Cat. No. W340-DE2-11



Speicherprogrammierbare Steuerungen

SYSMAC CS Series
SYSMAC CJ Series

REFERENZHANDBUCH DER PROGRAMMIERBEFEHLE



Kurzübersicht

1 Einführung
15 Übersicht über die Befehle
141 Befehle
1165 Ausführungszeit und Step-Anzahl der Befehle

Advanced Industrial Automation

SYSMAC CS-Serie

CS1G/H-CPU□□-EV1
CS1G/H-CPU□□H
CS1D-CPU□□H
CS1D-CPU□□S

SYSMAC CJ-Serie

CJ1G-CPU□□
CJ1G/H-CPU□□H
CJ1G-CPU□□P
CJ1M-CPU□□

Speicherprogrammierbare Steuerungen

Referenzhandbuch der Programmierbefehle

Überarbeitungsstand Juli 2004

Hinweis:

OMRON-Produkte sind zum Gebrauch durch einen qualifizierten Bediener gemäß angemessenen Verfahren und nur zu den in diesem Handbuch beschriebenen Zwecken gefertigt.

In diesem Handbuch werden Sicherheitshinweise nach folgenden Konventionen gekennzeichnet und eingeteilt. Beachten Sie stets die in diesen Hinweisen enthaltenen Informationen. Eine Nichtbeachtung der Sicherheitshinweise kann zu Verletzungen oder Sachschäden führen.

⚠ **GEFAHR** Kennzeichnet eine bevorstehende gefährliche Situation, die zum Tod oder zu ernsthaften Verletzungen führt, wenn sie nicht vermieden wird.

VORSICHT Kennzeichnet eine potenziell gefährliche Situation, die zum Tod oder zu schweren Verletzungen führen kann, wenn sie nicht vermieden wird.

Achtung Kennzeichnet eine potenziell gefährliche Situation, die zu kleineren oder mittelschweren Verletzungen oder Sachschäden führen kann, wenn sie nicht vermieden wird.

OMRON-Produktreferenzen

Alle OMRON-Produkte werden in diesem Handbuch groß geschrieben.

Die in manchen Anzeigen und auf manchen OMRON-Produkten verwendete Abkürzung „Ch“ bedeutet häufig „Wort“. Die übliche Abkürzung für „Wort“ (im Sinne von 16 Bits) in Anzeigen und auf OMRON-Produkten ist „Wd“.

Die Abkürzung „SPS“ steht für speicherprogrammierbare Steuerung. In manchen Anzeigen von Programmierkonsolen wird jedoch noch die Abkürzung „PC“ für "Programmable Controller" (Programmierbare Steuerung) verwendet. Dies ist nicht mit der üblichen Bedeutung von PC (Personal Computer) zu verwechseln.

Visuelle Hilfen

Die folgenden Überschriften tauchen in der linken Spalte des Handbuchs auf und helfen Ihnen, verschiedene Arten von Informationen zu finden.

Hinweis Kennzeichnet Informationen von besonderem Interesse für effizienten und zweckmäßigen Betrieb des Produkts.

1,2,3... 1. Kennzeichnet Auflistungen aller Art, z. B. Verfahren oder Checklisten.

© OMRON, 1999

Alle Rechte vorbehalten. Diese Publikation darf ohne vorherige schriftliche Genehmigung von OMRON weder als Ganzes noch in Auszügen in irgendeiner Form oder auf irgendeine Weise, sei es auf mechanischem oder elektronischem Wege oder durch Fotokopieren oder Aufzeichnen, reproduziert, auf einem Datensystem gespeichert oder übertragen werden.

In Bezug auf die hierin enthaltenen Informationen wird keine Patenthaftung übernommen. Da OMRON weiterhin an einer ständigen Verbesserung seiner Qualitätsprodukte arbeitet, sind Änderungen an den in diesem Handbuch enthaltenen Informationen ohne Ankündigung vorbehalten. Bei der Herstellung dieses Handbuchs wurden alle Vorsorgemaßnahmen ergriffen. Dennoch übernimmt OMRON keine Verantwortung für etwaige Fehler und Auslassungen. Es wird keine Haftung für Schäden übernommen, die aus der Nutzung von in diesem Dokument enthaltenen Informationen zurückzuführen sind.

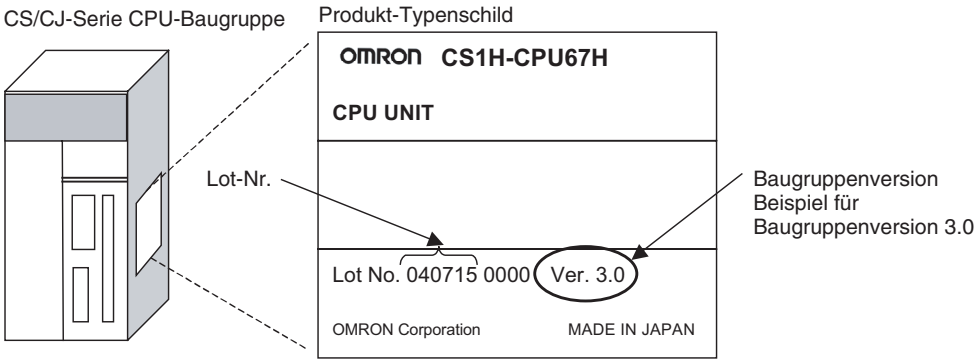
Baugruppenversionen von CPU-Baugruppen der CS/CJ-Serie

Baugruppen-versionen

Eine „Baugruppenversion“ wurde eingeführt, um CPU-Baugruppen der CS/CJ-Serie entsprechend der Funktionalitätsunterschiede, sich durch Weiterentwicklungen der Baugruppen ergeben, zu kennzeichnen. Dies gilt für die CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen.

Kennzeichnung der Baugruppenversionen auf den Produkten

Die Baugruppen-Version ist, wie unten gezeigt, rechts von der Lot-Nummer auf dem Typenschild der Produkte angegeben.



- CS1-H, CJ1-H und CJ1M CPU-Baugruppen (außer Low-End-Modelle), die am oder vor dem 4. November 2003 hergestellt wurden, besitzen keine Angabe der Baugruppenversion (d. h. die Stelle für die oben gezeigte Baugruppenversion ist unbeschriftet).
- Die Baugruppen-Versionsnummern der CS1-H, CJ1-H und CJ1M CPU-Baugruppen sowie der CS1D CPU-Baugruppe für Einzel-CPU-Systeme beginnen mit der Version 2.0.
- Die Baugruppen-Versionsnummern der CS1D CPU-Baugruppe für Duplex-CPU-Systeme beginnt mit Version 1.1.
- CPU-Baugruppen, für die keine Baugruppenversion angegeben wird, werden als CPU-Baugruppen *Vor Version □.□ bezeichnet, wie beispielsweise CPU-Baugruppen vor Version 2.0 oder CPU-Baugruppen vor Version 1.1*.

Überprüfung der Baugruppenversionen mit Hilfe von Programmiersoftware

CX-Programmer Version 4.0 kann zur Prüfung der Baugruppen-Version verwendet werden. Dazu gibt es zwei Möglichkeiten.

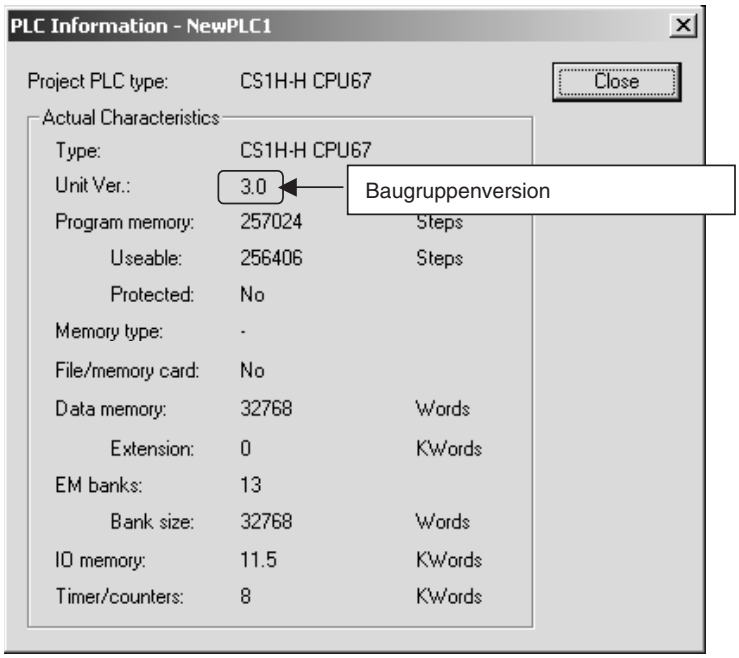
- Mithilfe der ***SPS-Informationen***
- Bei Verwendung der ***Baugruppen-Herstellungsinformationen*** (diese Methode kann auch für Spezial-E/A-Baugruppen und CPU-Bus-Baugruppen verwendet werden)

Hinweis CX-Programmer Version 3.3 oder frühere Versionen können nicht zur Überprüfung von Baugruppenversionen verwendet werden.

SPS-Information

- Wenn der Geräte- und CPU-Typ bekannt sind, wählen Sie ihn im Dialogfeld *Ändern der SPS* aus, gehen Sie online, und wählen Sie in den Menüs ***SPS - Bearbeiten - Information***.
- Wenn Geräte- und CPU-Typ nicht bekannt sind, diese aber direkt über ein serielles Kabel mit der CPU-Baugruppe verbunden sind, wählen Sie ***SPS - Auto Online***, um online zu gehen, und wählen Sie dann in den Menüs ***SPS - Bearbeiten - Information***.

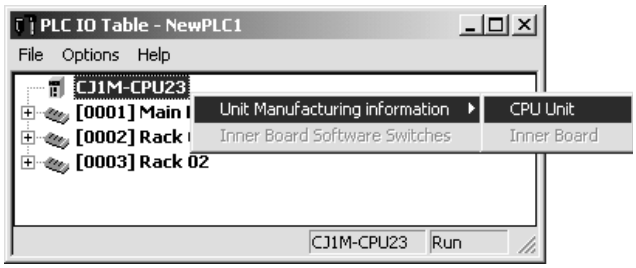
In beiden Fällen wird das nachfolgend abgebildete Dialogfeld *PLC Information* (SPS Information) angezeigt.



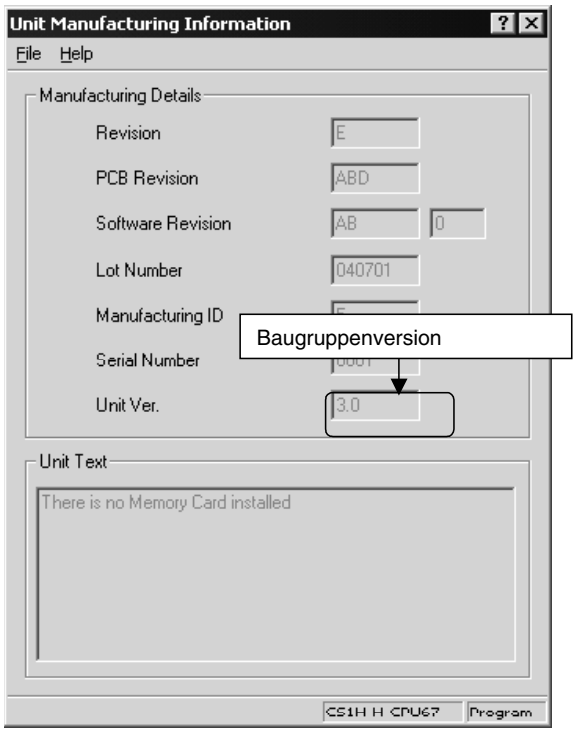
Verwenden Sie oben dargestellte Anzeige zur Prüfung der CPU-Baugruppenversion.

Baugruppen-Herstellungsinformationen

Klicken Sie mit der rechten Maustaste im Fenster E/A-Tabelle, und wählen Sie ***Fertigungsinformationen zur Baugruppe - CPU-Baugruppe*** *aus.*



Nun wird das nachfolgend abgebildete Dialogfeld *Unit Manufacturing Information* (Baugruppen-Herstellungsinformationen) angezeigt.



Verwenden Sie oben dargestellte Anzeige zur Prüfung der Baugruppen-Version der online verbundenen CPU-Baugruppe.

Verwendung der Baugruppen-Versionsetiketten

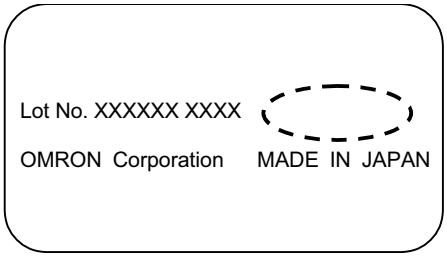
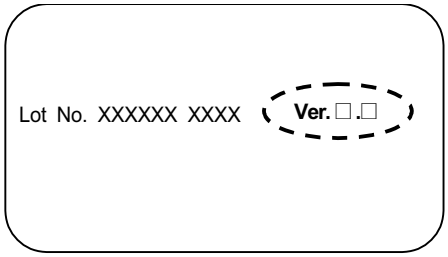
Die folgenden Baugruppen-Versionsetiketten werden mit der CPU-Baugruppe mitgeliefert.



Diese Etiketten können an der Vorderseite von CPU-Baugruppen angebracht werden, um sie von anderen CPU-Baugruppen abweichender Baugruppenversionen unterscheiden zu können.

Angabe der Baugruppenversion

In diesem Handbuch wird die Baugruppenversion einer CPU-Baugruppe wie in der folgenden Tabelle dargestellt angegeben.

Produkt-Typenschild **Bedeutung**	**CPU-Baugruppen ohne Angabe einer Baugruppenversion** 	**CPU-Baugruppen mit Angabe einer Version (Version □.□)** 
Bezeichnung einzelner CPU-Baugruppen (z. B. CS1H-CPU67H)	CS1-H CPU-Baugruppen vor Version 2.0	CS1H-CPU67H CPU-Baugruppe Ver. □.□
Bezeichnung von Gruppen von CPU-Baugruppen (z. B. CS1-H CPU-Baugruppen)	CS1-H CPU-Baugruppen vor Version 2.0	CS1-H CPU-Baugruppen Ver. □.□
Bezeichnung einer ganzen Serie von CPU-Baugruppen (z. B. CPU-Baugruppen der CS-Serie)	CS-Serie CPU-Baugruppen vor Version 2.0	CS-Serien-CPU-Baugruppen Ver. □.□

Baugruppenversionen und Chargennummern

Serie	Produktbezeichnung		Herstellungsdatum						
			Früher	Sep. 2003	Okt. 2003	Nov. 2003	Dez. 2003	Jun. 2004	Später
CS-Serie	CS1 CPU-Baugruppen	CS1□-CPU□□	Keine Baugruppenversion						
	CS1-V1 CPU-Baugruppen	CS1□-CPU□□-V1	Keine Baugruppenversion						
	CS1-H CPU-Baugruppen	CS1□-CPU□□H	CPU-Baugruppen vor Version 2.0			CPU Baugruppen Ver. 2.0 (Lot-Nr.: ab 031105)			CPU Baugruppen Ver. 3.0 (Lot-Nr.: ab 040622)
	CS1D CPU-Baugruppen: CPU-Baugruppen für Duplex-CPU-Systeme	CS1D-CPU□□H	CPU-Baugruppen vor Version 1.1				CPU-Baugruppen Ver. 1.1 (Lot-Nr.: ab 031120)		
	CS1D CPU-Baugruppen: CPU-Baugruppen für Einzel-CPU-Systeme	CS1D-CPU□□S					CPU Baugruppen Ver. 2.0 (Lot-Nr.: ab 031215)		
CJ-Serie	CJ1 CPU-Baugruppen	CJ1G-CPU□□	CPU-Baugruppen vor Version 2.0						
	CJ1-H CPU-Baugruppen	CJ1□-CPU□□H	CPU-Baugruppen vor Version 2.0			CPU-Baugruppen Ver. 2.0 (Lot-Nr.: ab 031105)			CPU-Baugruppen Ver. 3.0 (Lot-Nr.: ab 040623)
	CJ1M CPU-Baugruppen außer Low-End-Modelle	CJ1M-CPU□□	CPU-Baugruppen vor Version 2.0			CPU-Baugruppen Ver. 2.0 (Lot-Nr.: ab 031105)			CPU-Baugruppen Ver. 3.0 (Lot-Nr.: ab 040622)
	CJ1M CPU-Baugruppen, Low-End-Modelle	CJ1M-CPU11/21			Baugruppen-Version 2.0 (Lot-Nr.: ab 031002)				CPU-Baugruppen Version 3.0 (Lot-Nr.: ab 040629)
Programmiersoftware	CX-Programmer	WS02-CXPC1-EV□	Ver. 3.2		Ver. 3.3		Ver. 4.0		Ver. 5.0

Unterstützung von Funktionen nach Baugruppenversion

CS1-H CPU-Baugruppen (CS1□-CPU□□H)

Funktion		Baugruppenversion	
		CPU-Baugruppen vor Version 2.0	CPU-Baugruppen Ver. 2.0
Herunterladen und Heraufladen von einzelnen Tasks		---	OK
Verbesserter Leseschutz durch Verwendung von Passwörtern		---	OK
Schreibschutz vor FINS-Befehlen, die über Netzwerke an die CPU-Baugruppen gesendet werden		---	OK
Online-Netzwerkverbindungen ohne E/A-Tabellen		---	OK
Kommunikation über bis zu 8 Netzwerkebenen		---	OK
Online-Verbindung zu SPS-Systemen über NS-Terminals		OK ab Lot-Nummer 030201	OK
Einstellung der Worte für den ersten Steckplatz		OK bei bis zu 8 Gruppen	OK bei bis zu 64 Gruppen
Automatische Übertragung ohne Parameterdatei beim Einschalten der Versorgungsspannung		---	OK
Automatische Erkennung der E/A-Zuweisungsmethode für die automatische Übertragung beim Einschalten der Versorgungsspannung		---	---
Start-/Endzeiten des Betriebs		---	OK
Neue Anwendungsbefehle	MILH, MILR, MILC	---	OK
	=DT, <>DT, <DT, <=DT, >DT, >=DT	---	OK
	BCMP2	---	OK
	GRY	OK ab Lot-Nummer 030201	OK
	TPO	---	OK
	DSW, TKY, HKY, MTR, 7SEG	---	OK
	EXPLT, EGATR, ESATR, ECHRD, ECHWR	---	OK
	Lesen aus/Schreiben in CPU-Bus-Baugruppen mit IORD/IOWR	OK ab Lot-Nummer 030418	OK
	PRV2	---	---

CS1D CPU-Baugruppen

Funktion		CS1D CPU-Baugruppen für Duplex-CPU-Systeme (CS1D-CPU□□H)		CS1D CPU-Baugruppen für Einzel-CPU-Systeme (CS1D-CPU□□S)
		CPU-Baugruppen vor Version 1.1	CPU-Baugruppen Ver. 1.1	CPU-Baugruppen Ver. 2.0
Für CS1D CPU-Baugruppen spezifische Funktionen	Duplex-CPU-Baugruppen	OK	OK	---
	Online-Baugruppenaustausch	OK	OK	OK
	Duplex-Spannungsversorgungs-Baugruppen	OK	OK	OK
	Duplex-Controller-Link-Baugruppen	OK	OK	OK
	Duplex-Ethernet-Baugruppen	---	OK	OK
Herunterladen und Heraufladen von einzelnen Tasks		---	---	OK
Verbesserter Leseschutz durch Verwendung von Passwörtern		---	---	OK
Schreibschutz vor FINS-Befehlen, die über Netzwerke an die CPU-Baugruppen gesendet werden		---	---	OK
Online-Netzwerkverbindungen ohne E/A-Tabellen		---	---	OK
Kommunikation über bis zu 8 Netzwerkebenen		---	---	OK
Online-Verbindung zu SPS-Systemen über NS-Terminals		---	---	OK
Einstellung der Worte für den ersten Steckplatz		---	---	OK bei bis zu 64 Gruppen
Automatische Übertragung ohne Parameterdatei beim Einschalten der Versorgungsspannung		---	---	OK
Automatische Erkennung der E/A-Zuweisungsmethode für die automatische Übertragung beim Einschalten der Versorgungsspannung		---	---	---
Start-/Endzeiten des Betriebs		---	OK	OK
Neue Anwendungsbefehle	MILH, MILR, MILC	---	---	OK
	=DT, <>DT, <DT, <=DT, >DT, >=DT	---	---	OK
	BCMP2	---	---	OK
	GRY	---	---	OK
	TPO	---	---	OK
	DSW, TKY, HKY, MTR, 7SEG	---	---	OK
	EXPLT, EGATR, ESATR, ECHRD, ECHWR	---	---	OK
	Lesen aus/Schreiben in CPU-Bus-Baugruppen mit IORD/IOWR	---	---	OK
	PRV2	---	---	---

CJ1-H/CJ1M CPU-Baugruppen

Funktion		CJ1-H CPU-Baugruppen (CJ1□-CPU□□H)		CJ1M CPU-Baugruppen (außer Low-End-Modelle) (CJ1M-CPU□□)		CJ1M CPU-Baugruppen, Low-End-Modelle (CJ1M-CPU11/21)
		CPU-Baugruppen vor Version 2.0	CPU-Baugruppen Ver. 2.0	CPU-Baugruppen vor Version 2.0	CPU-Baugruppen Ver. 2.0	CPU-Baugruppen Ver. 2.0
Herunterladen und Heraufladen von einzelnen Tasks		---	OK	---	OK	OK
Verbesserter Leseschutz durch Verwendung von Passwörtern		---	OK	---	OK	OK
Schreibschutz vor FINS-Befehlen, die über Netzwerke an die CPU-Baugruppen gesendet werden		---	OK	---	OK	OK
Online-Netzwerkverbindungen ohne E/A-Tabellen		OK, aber nur, wenn die E/A-Tabelle erstellt wurde	OK	OK, aber nur, wenn die E/A-Tabelle erstellt wurde	OK	OK
Kommunikation über bis zu 8 Netzwerkebenen		OK bei bis zu 8 Gruppen	OK bei bis zu 64 Gruppen	OK bei bis zu 8 Gruppen	OK bei bis zu 64 Gruppen	OK bei bis zu 64 Gruppen
Online-Verbindung zu SPS-Systemen über NS-Terminals		OK ab Lot-Nummer 030201	OK	OK ab Lot-Nummer 030201	OK	OK
Einstellung der Worte für den ersten Steckplatz		---	OK	---	OK	OK
Automatische Übertragung ohne Parameterdatei beim Einschalten der Versorgungsspannung		---	OK	---	OK	OK
Automatische Erkennung der E/A-Zuweisungsmethode für die automatische Übertragung beim Einschalten der Versorgungsspannung		---	OK	---	OK	OK
Start-/Endzeiten des Betriebs		---	OK	---	OK	OK
Neue Anwendungsbefehle	MILH, MILR, MILC	---	OK	---	OK	OK
	=DT, <>DT, <DT, <=DT, >DT, >=DT	---	OK	---	OK	OK
	BCMP2	---	OK	OK	OK	OK
	GRY	OK ab Lot-Nummer 030201	OK	OK ab Lot-Nummer 030201	OK	OK
	TPO	---	OK	---	OK	OK
	DSW, TKY, HKY, MTR, 7SEG	---	OK	---	OK	OK
	EXPLT, EGATR, ESATR, ECHRD, ECHWR	---	OK	---	OK	OK
	Lesen aus/Schreiben in CPU-Bus-Baugruppen mit IORD/IOWR	---	OK	---	OK	OK
	PRV2	---	---	---	OK, jedoch nur für Modelle mit integrierten E/A	OK, jedoch nur für Modelle mit integrierten E/A

Von Baugruppen ab Ver. 3.0 unterstützte Funktionen

CS1-H CPU-Baugruppen (CS1□-CPU□□H)

Funktion		Baugruppenversion	
		Vor Version 2.0, Ver. 2.0	Ver. 3.0
Funktionsblöcke (CX-Programmer Ver. 5.0 oder neuer erforderlich)		---	OK
Serial Gateway (Konvertierung von FINS-Befehlen in CompoWay/F-Befehle an der integrierten seriellen Schnittstelle)		---	OK
Kommentarspeicher (im internen Flash-Speicher)		---	OK
Erweiterte Easy-Backup-Sicherung		---	OK
Neue Anwendungsbefehle	TXDU(256), RXDU(255) (ermöglichen die protokollfreie Kommunikation mittels serieller Kommunikationsbaugruppen ab Ver. 1.2)	---	OK
	Modell-Konvertierungsbefehle: XFERC(565), DISTC(566), COLLC(567), MOVBC(568), BCNTC(621)	---	OK
	Spezielle Funktionsblockbefehle: GETID(286)	---	OK
Weitere Befehle	TXD(235), RXD(236) (ermöglichen die protokollfreie Kommunikation mittels serieller Kommunikationsmodule ab Ver. 1.2)	---	OK

CS1D CPU-Baugruppen Baugruppenversion 3.0 wird nicht unterstützt.

CJ1-H/CJ1M CPU-Baugruppen (CJ1□-CPU□□H, CJ1G-CPU□□P, CJ1M-CPU□□)

Funktion		Baugruppenversion	
		Vor Version 2.0, Ver. 2.0	Ver. 3.0
Funktionsblöcke (CX-Programmer Ver. 5.0 oder höher erforderlich)		---	OK
Serial Gateway (Konvertierung von FINS-Befehlen in CompoWay/F-Befehle an der integrierten seriellen Schnittstelle)		---	OK
Kommentarspeicher (im internen Flash-Speicher)		---	OK
Erweiterte Easy-Backup-Sicherung		---	OK
Neue Anwendungsbefehle	TXDU(256), RXDU(255) (ermöglichen die protokollfreie Kommunikation mittels serieller Kommunikationsbaugruppen ab Ver. 1.2)	---	OK
	Modell-Konvertierungsbefehle: XFERC(565), DISTC(566), COLLC(567), MOVBC(568), BCNTC(621)	---	OK
	Spezielle Funktionsblockbefehle: GETID(286)	---	OK
Weitere Befehle	PRV(881) und PRV2(883): Einschließlich Hochfrequenz-Berechnungsmethoden zur Berechnung der Impulsfrequenz. (Nur CJ1M CPU-Baugruppen)	---	OK

Baugruppenversionen und Programmiergeräte

Um die bei den CPU-Baugruppen ab Ver. 2.0 neu hinzugekommenen Funktionen nutzen zu können, ist CX-Programmer Version 4.0 oder neuer erforderlich.

Um die bei den CPU-Baugruppen ab Ver. 3.0 neu hinzugekommenen Funktionsblöcke nutzen zu können, ist CX-Programmer Version 5.0 oder neuer erforderlich.

Die folgenden Tabellen zeigen die Beziehung zwischen Baugruppenversionen und Versionen von CX-Programmer.

Baugruppenversionen und Programmiergeräte

CPU-Baugruppe	Funktionen		CX-Programmer				Programmierkonsole
			Ver. 3.2 oder älter	Ver. 3.3	Ver. 4.0	Ver. 5.0 oder neuer	
CJ1M CPU-Baugruppen, Low-End-Modelle, Baugruppenver. 2.0	Neue Funktionen bei Baugruppenversion 2.0	Bei Verwendung neuer Funktionen	---	---	OK	OK	Keine Einschränkungen
		Ohne Verwendung neuer Funktionen	---	OK	OK	OK	
CS1-H, CJ1-H und CJ1M CPU-Baugruppen außer Low-End-Modelle, Baugruppen-Version 2.0	Neue Funktionen bei Baugruppenversion 2.0	Bei Verwendung neuer Funktionen	---	---	OK	OK	
		Ohne Verwendung neuer Funktionen	OK	OK	OK	OK	
CPU-Baugruppen CS1D für Einzel-CPU-Systeme, Baugruppenver. 2.0	Neue Funktionen bei Baugruppenversion 2.0	Bei Verwendung neuer Funktionen	---	---	OK	OK	
		Ohne Verwendung neuer Funktionen				OK	
CS1D CPU-Baugruppen für Duplex-CPU-Systeme, Baugruppenversion 1.	Neue Funktionen bei Baugruppenversion 1.1	Bei Verwendung neuer Funktionen	---	---	OK	OK	
		Ohne Verwendung neuer Funktionen	OK	OK	OK	OK	
CS/CJ-Serie, Baugruppenver. 3.0	Neue Funktionsblock-Funktionen bei Baugruppenversion 3.0	Bei Verwendung von Funktionsblöcken	---	---	---	OK	
		Ohne Verwendung von Funktionsblöcken	OK	OK	OK	OK	

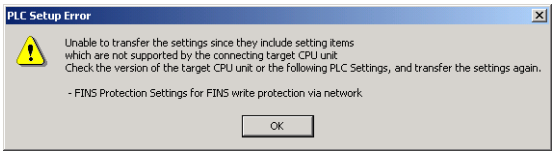
Hinweis Wie oben gezeigt, besteht keine Notwendigkeit zur Aktualisierung auf CX-Programmer Version 4.0, solange die in Baugruppenversion 2.0 bzw. 1.1 hinzugekommenen Funktionen nicht verwendet werden.

Gerätetyp-Einstellung

Die Baugruppenversion hat keinen Einfluss auf die mit CX-Programmer vorgenommene Einstellung für den Gerätetyp. Wählen Sie den Gerätetyp entsprechend den Angaben in der folgenden Tabelle unabhängig von der Baugruppenversion der CPU-Baugruppe.

Serie	CPU-Baugruppengruppe	CPU-Baugruppenmodell	Gerätetyp-Einstellung bei CX-Programmer Version 4.0 oder höher
CS-Serie	CS1-H CPU-Baugruppen	CS1G-CPU□□H	CS1G-H
		CS1H-CPU□□H	CS1H-H
	CS1D CPU-Baugruppen für Duplex-CPU-Systeme	CS1D-CPU□□H	CS1D-H (oder CS1H-H)
	CS1D CPU-Baugruppen für Einzel-CPU-Systeme	CS1D-CPU□□S	CS1D-S
CJ-Serie	CJ1-H CPU-Baugruppen	CJ1G-CPU□□H	CJ1G-H
		CJ1H-CPU□□H	CJ1H-H
	CJ1M CPU-Baugruppen	CJ1M-CPU□□	CJ1M

Behebung von Problemen mit Baugruppenversionen in CX-Programmer

Problem	Ursache	Lösung
 Nach Anzeige der oben abgebildeten Meldung wird im Meldungsfenster auf der Registerkarte *Kompilieren* ein Kompilierungsfehler angezeigt.	Es wurde der Versuch unternommen, mit Hilfe von CX-Programmer Version 4.0 oder höher ein Programm mit Befehlen, die nur von CPU-Baugruppen-Version 2.0 oder neuer unterstützt werden, auf ein CPU-Baugruppe vor Version 2.0 herunterzuladen.	Prüfen Sie das Programm oder wechseln Sie die CPU-Baugruppe, auf die das Programm heruntergeladen wird, zu einer CPU-Baugruppe der Version 2.0 oder neuer.
	Es wurde versucht, mit CX-Programmer Version 4.0 oder höher ein SPS-Setup, das Einstellungen enthält, die nur von CPU-Baugruppen der Version 2.0 oder neuer unterstützt werden (also nicht auf die Standardwerte gesetzt sind), auf eine CPU-Baugruppe vor Version 2.0 herunterzuladen.	Prüfen Sie die Einstellungen im SPS-Setup oder wechseln Sie die CPU-Baugruppe, auf die das Setup heruntergeladen wird, zu einer CPU-Baugruppe der Version 2.0 oder neuer.
"????" wird in einem Programm angezeigt, das von der SPS zu CX-Programmer übertragen wurde.	Es wurde mit CX-Programmer Version 3.3 oder niedriger versucht, ein Programm von einer CPU-Baugruppe Version 2.0 oder höher hochzuladen, das Befehle enthält, die nur von CPU-Baugruppenversion 2.0 oder höher unterstützt werden.	Die neuen Befehle können nicht unter Verwendung von CX-Programmer Version 3.3 oder früher heraufgeladen werden. Verwenden Sie CX-Programmer Version 4.0 oder höher.

INHALTSVERZEICHNIS

SICHERHEITSHINWEISE xxiii

1 Zielgruppe xxiv
2 Allgemeine Sicherheitshinweise xxiv
3 Sicherheitshinweise xxiv
4 Sicherheitshinweise für die Betriebsumgebung xxvi
5 Sicherheitshinweise zu Anwendungen xxvi
6 Konformität mit EU-Richtlinien xxx

KAPITEL 1
Einführung 1

1-1 Allgemeine Befehlseigenschaften 2
1-2 Befehlsausführungsprüfungen 13

KAPITEL 2
Übersicht über die Befehle 15

2-1 Klassifizierung der Befehle nach Funktion 16
2-2 Befehlsfunktionen 24
2-3 Alphabetische Liste der Befehle nach AWL-Kürzel 108
2-4 Liste der Befehl nach Funktionscode 125

KAPITEL 3
Befehle 141

3-1 Notation und Layout von Befehlsbeschreibungen 148
3-2 Befehlserweiterungen und neue Befehle 151
3-3 Sequenzeingabebefehle 153
3-4 Sequenzausgabebefehle 177
3-5 Sequenzsteuerbefehle 197
3-6 Zeitgeber- und Zählerbefehle 233
3-7 Vergleichsbefehle 275
3-8 Datenübertragungsbefehle 315
3-9 Datenverschiebungsbefehle 344
3-10 Inkrementier-/Dekrementierbefehle 393
3-11 Symbolarithmetische Befehle 409
3-12 Konvertierungsbefehle 465
3-13 Logikbefehle 517
3-14 Spezielle arithmetische Befehle 534
3-15 Gleitkomma-Arithmetikbefehle 558
3-16 Befehle für Gleitkommadaten doppelter Genauigkeit (nur CS1-H, CJ1-H, CJ1M und CS1D) 613
3-17 Tabellendatenverarbeitungsbefehle 660
3-18 Datensteuerungsbefehle 720
3-19 Unterprogramme 773
3-20 Interrupt-Steuerungsbefehle 798

3-21 Befehle für schnelle Zähler und Impulsausgaben 823
3-22 Schritt-Befehle 867
3-23 E/A-Baugruppenbefehle 885
3-24 Befehle für serielle Kommunikation 926
3-25 Netzwerkbefehle 973
3-26 Dateispeicherbefehle 1042
3-27 Anzeigebefehle: DISPLAY MESSAGE: MSG(046) 1058
3-28 Datum- und Uhrzeitbefehle 1061
3-29 Fehlersuchebefehle 1075
3-30 Fehlerdiagnosebefehle 1079
3-31 Sonstige Befehle 1104
3-32 Blockprogrammierbefehle 1124
3-33 Textverarbeitungsbefehle 1158
3-34 Task-Steuerbefehle 1192
3-35 Befehle zur Modellkonvertierung (nur CPU-Baugruppen ab Ver. 3.0) 1199

KAPITEL 4
Ausführungszeit und Step-Anzahl der Befehle 1219

4-1 Ausführungszeit und Step-Anzahl der Befehle für die CS-Serie 1221
4-2 Ausführungszeit und Step-Anzahl der Befehle für die CJ-Serie 1250

Appendices

A ASCII-Codetabelle 1285

Index 1287

Revisionshistorie 1295

Zu diesem Handbuch:

In diesem Handbuch wird die Kontaktplanprogrammierung für CPU-Baugruppen von SPS der CS/CJ-Serie beschrieben. Die SPS-Serien CS und CJ werden wie in der folgenden Tabelle aufgeführt unterteilt.

Baugruppe	CS-Serie	CJ-Serie
CPU-Baugruppen	CS1-H CPU-Baugruppen: CS1H-CPU□□H CS1G-CPU□□H	CJ1-H CPU-Baugruppen: CJ1H-CPU□□H CJ1G-CPU□□H CJ1G-CPU□□P
	CS1 CPU-Baugruppen: CS1H-CPU□□-EV1 CS1G-CPU□□-EV1	CJ1 CPU-Baugruppen: CJ1G-CPU□□-EV1 CJ1M CPU-Baugruppen: CJ1M-CPU□□
	CS1D CPU-Baugruppen: CS1D CPU-Baugruppen für Duplex-CPU-Systeme: CS1D-CPU□□H CS1D CPU-Baugruppen für Einzel-CPU-Systeme: CS1D-CPU□□S CS1D Prozess-CPU-Baugruppen: CS1D-CPU□□P	
E/A-Baugruppen	CS-Serie E/A-Baugruppen	CJ-Serie E/A-Baugruppen
Spezial-E/A-Baugruppen	CS-Serie Spezial-E/A-Baugruppen	CJ-Serie Spezial-E/A-Baugruppen
CPU-Bus-Baugruppen	CS-Serie CPU-Bus-Baugruppen	CJ-Serie CPU-Bus-Baugruppen
Spannungsversorgungs-Baugruppen	CS-Serie Spannungsversorgungs-Baugruppen	CJ-Serie Spannungsversorgungs-Baugruppen

Lesen Sie bitte dieses Handbuch und alle in der Tabelle auf der folgenden Seite aufgelisteten, dazugehörigen Handbücher und stellen Sie sicher, dass Sie alle Informationen verstanden haben, bevor Sie CPU-Baugruppen der CS/**CJ-Serie in einem SPS-System programmieren oder verwenden.**

Kapitel 1 Einführung zu den SPS der CS/CJ-Serie hinsichtlich des Befehlssatzes, den sie unterstützen.

Kapitel 2 Enthält verschiedene Listen von Befehlen, die als Referenz genutzt werden können.

Kapitel 3 Enthält Beschreibungen der einzelnen Befehle des Befehlssatzes der CS/CJ-Serie.

Kapitel 4 Enthält Befehlsausführungszeiten und die Anzahl der Steps für jeden Befehl der CS/CJ-Serie.

Zu diesem Handbuch, Fortsetzung

Name	Kat.-Nr.	Inhalt
SYSMAC CS/CJ-Serie CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H, CS1D-CPU□□H, CS1D-CPU□□S, CJ1G-CPU□□, CJ1M-CPU□□, CJ1G-CPU□□P, CJ1G/H-CPU□□H Speicherprogrammierbare Steuerungen Referenzhandbuch der Programmierbefehle	W340	Enthält Beschreibungen der von SPS-Systemen der CS/CJ-Serie unterstützten Befehle zur Kontaktplanprogrammierung. (Dieses Handbuch)
SYSMAC CS/CJ-Serie CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H, CS1D-CPU□□H, CS1D-CPU□□S, CJ1G-CPU□□, CJ1M-CPU□□, CJ1G-CPU□□P, CJ1G/H-CPU□□H Speicherprogrammierbare Steuerungen Programmierhandbuch	W394	In diesem Handbuch wird die Programmierung und Nutzung der Funktionen von SPS-Systemen der CS/CJ-Serie beschrieben.
SYSMAC CS-Serie CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H Speicherprogrammierbare Steuerungen Bedienerhandbuch	W339	Bietet eine detaillierte Beschreibung von Konstruktion, Installation, Wartung und grundlegender Funktionen von SPS-Systemen der CS-Serie.
SYSMAC CJ-Serie CJ1G-CPU□□, CJ1M-CPU□□, CJ1G-CPU□□P, CJ1G/H-CPU□□H Speicherprogrammierbare Steuerungen Bedienerhandbuch	W393	Bietet einen Überblick und eine Beschreibung von Konstruktion, Installation, Wartung und grundlegender Funktionen von SPS-Systemen der CJ-Serie.
SYSMAC CJ-Serie CJ1M-CPU21/22/23 Integrierte E/A-Funktionen Bedienerhandbuch	W395	Bietet eine Beschreibung der Funktionen der integrierten E/A bei CJ1M-CPU-Baugruppen.
SYSMAC CS-Serie CS1D-CPU□□H CPU-Baugruppen CS1D-CPU□□S CPU-Baugruppen CS1D-DPL1 Duplex-Baugruppe CS1D-PA207R Spannungsversorgungs-Baugruppe Duplex-System Bedienerhandbuch	W405	Bietet einen Überblick und eine Beschreibung von Konstruktion, Installation, Wartung und grundlegender Funktionen eines auf CS1D CPU-Baugruppen basierenden Duplex-Systems.
SYSMAC CS/CJ-Serie CQM1H-PRO01-E, C200H-PRO27-E, CQM1-PRO01-E Programmierkonsolen Bedienerhandbuch	W341	Enthält Informationen darüber, wie SPS-Systeme der CS/CJ-Serie über eine Programmierkonsole programmiert und gesteuert werden.
SYSMAC CS/CJ-Serie CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H, CS1D-CPU□□H, CS1D-CPU□□S, CJ1M-CPU□□, CJ1G-CPU□□, CJ1G-CPU□□P, CJ1G/H-CPU□□H, CS1W-SCB21-V1/41-V1, CS1W-SCU21-V1, CJ1W-SCU21-V1/41-V1 Kommunikationsbefehle Referenzhandbuch	W342	Enthält Beschreibungen der C-Serie- (Host-Link) und FINS-Kommunikationsbefehle, die mit SPS-Systemen der CS/CJ-Serie verwendet werden.
SYSMAC WS02-CXP□□-E CX-Programmer Version 3.□ Bedienerhandbuch	W414	Enthält Informationen zur Verwendung von CX-Programmer, einer Programmiersoftware, die SPS-Systeme der CS/CJ-Serie unterstützt, sowie der in CX-Programmer enthaltenen Software CX-Net.
SYSMAC WS02-CXP□□-E CX-Programmer Version 4.□ Bedienerhandbuch	W425	
SYSMAC WS02-CXP□□-E CX-Programmer Version 5.□ Bedienerhandbuch	W437	

Name	Kat.-Nr.	Inhalt
SYSMAC WS02-CXP□□-E CX-Programmer Bedienerhandbuch (Funktionsblöcke)	W438	Bietet Beschreibungen der Funktionsblock-spezifischen Spezifikationen und Verfahren. Diese Informationen sind nur bei Verwendung von Funktionsblöcken in Kombination mit CX-Programmer der Version 5.0 und der CS1-H/CJ1-H/CJ1M CPU-Baugruppe der Version 3.0 erforderlich. Detaillierte Informationen zur sonstigen Verwendung von CX-Programmer Ver. 5.0 finden Sie im *CX-Programmer Handbuch Version 5.□ Bedienerhandbuch* (W437).
SYSMAC CS/CJ-Serie CS1W-SCB21-V1/41-V1, CS1W-SCU21-V1, CJ1W-SCU21-V1/41-V1 Serielle Kommunikationsmodule/-baugruppen Bedienerhandbuch	W336	Beschreibung der Nutzung von seriellen Kommunikations-Modulen und -Baugruppen zur Kommunikation mit externen Geräten, insbesondere auch der Verwendung von Standardsystemprotokollen für OMRON-Produkte.
SYSMAC WS02-PSTC1-E CX-Protocol Bedienerhandbuch	W344	Beschreibung der Nutzung von CX-Protocol zum Erstellen von Protokollmakros für die Kommunikation mit externen Geräten.

VORSICHT Fehler die dadurch verursacht werden, weil Sie die in diesem Handbuch enthaltenen Informationen nicht lesen oder nicht verstehen, können zur Verletzung oder zum Tod von Personen, zur Beschädigung des Produkts bzw. zu Fehlfunktionen führen. Lesen Sie jedes Kapitel vollständig durch, und führen Sie die vorgestellten Maßnahmen und Bedienverfah-

SICHERHEITSHINWEISE

In diesem Abschnitt finden Sie allgemeine Sicherheitshinweise für die Nutzung der speicherprogrammierbaren Steuerungen (SPS) der CS/CJ-Serie sowie zugehöriger Geräte.

Die in diesem Abschnitt enthaltenen Informationen sind wichtig für die sichere und zuverlässige Anwendung von speicherprogrammierbaren Steuerungen. Bevor Sie versuchen, ein SPS-System einzurichten oder zu betreiben, müssen Sie diesen Abschnitt lesen und die hierin enthaltenen Informationen verstehen.

1 Zielgruppe xxiv
2 Allgemeine Sicherheitshinweise xxiv
3 Sicherheitshinweise xxiv
4 Sicherheitshinweise für die Betriebsumgebung xxvi
5 Sicherheitshinweise zu Anwendungen xxvi
6 Konformität mit EU-Richtlinien xxx
6-1 Anwendbare Richtlinien xxx
6-2 Konzepte xxx
6-3 Konformität mit EU-Richtlinien xxxi
6-4 Methoden zur Entstörung von Relaisausgängen xxxi

1 Zielgruppe

Dieses Handbuch ist für die Anwendung von den nachfolgend aufgeführten Personengruppen bestimmt, die über entsprechende Kenntnisse auf dem Gebiet elektrischer Steuerungssysteme verfügen sollten:

- Personen, die mit der Installation von Automatisierungs-Systemen befasst sind.
- Personen, die mit der Konstruktion von Automatisierungs-Systemen befasst sind.
- Personen, die Automatisierungs-Systeme und -Anlagen verwalten.

2 Allgemeine Sicherheitshinweise

Der Anwender darf das Produkt nur entsprechend den in diesem Handbuch niedergelegten Vorgaben einsetzen.

Bevor Sie dieses Produkt unter Bedingungen anwenden, die nicht in diesem Handbuch beschrieben sind oder das Produkt in nuklearen Steuerungssystemen, Bahnnetzen, Luftfahrtsystemen, Fahrzeugen, Verbrennungssystemen, medizinischen Geräten, Fahrgeschäften, Sicherheitsgeräten und anderen Systemen, Maschinen und Geräten anwenden, die bei unsachgemäßer Anwendung ernsthaften Einfluss auf Leben und Eigentum haben, konsultieren Sie bitte Ihre OMRON-Vertretung.

Stellen Sie sicher, dass die Nennleistungen und Betriebsmerkmale des Produktes den Anforderungen der Systeme, Maschinen und Anlagen genügen. Die Systeme, Maschinen und Anlagen ihrerseits sollten mit redundanten Sicherheitsmechanismen ausgestattet sein.

Dieses Handbuch enthält Informationen zu Programmierung und Betrieb des Produkts. Lesen Sie dieses Handbuch vor Verwendung des Produkts durch, und halten Sie dieses Handbuch während des Betriebs zu Referenzzwecken immer griffbereit.

⚠ VORSICHT Es ist außerordentlich wichtig, dass SPS und alle SPS-Baugruppen nur für den vorgegebenen Einsatzzweck und unter den angegebenen Bedingungen verwendet werden. Dies gilt besonders für Anwendungen, bei denen direkt oder indirekt die Gefahr von Personenschäden besteht. Wenden Sie sich an Ihre OMRON-Vertretung, bevor Sie ein SPS-System für die oben aufgeführten Anwendungen einsetzen.

3 Sicherheitshinweise

⚠ VORSICHT Die CPU-Baugruppe führt E/A-Aktualisierungen auch dann durch, wenn das Programm angehalten ist (d. h. auch in der PROGRAM-Betriebsart). Achten Sie auf Einhaltung der Sicherheitsvorschriften, bevor Sie den Status eines beliebigen Speichers ändern, der einer E/A-Baugruppe, einer Spezial-E/A-Baugruppe oder einem CPU-Bus zugewiesen ist. Jede Änderung der Daten, die einer Baugruppe zugewiesen sind, kann Auswirkungen auf die angeschlossenen Lasten haben. Folgende Vorgänge können zu einer Änderung des Speicherstatus führen.

- Übertragen von Speicherdaten von einem Programmiergerät in die CPU-Baugruppe.
- Ändern der aktuellen Werte im Speicher von einem Programmiergerät aus.
- Zwangsweises Setzen oder Rücksetzen von Bits über ein Programmiergerät.
- Übertragen von Speicherdateien von einer Speicherkarte oder dem EM-Dateispeicher in die CPU-Baugruppe.
- Übertragen von Speicherdaten von einem Host-Computer oder von einer anderen SPS im Netzwerk.

⚠ VORSICHT Entnehmen Sie keine Baugruppe bei eingeschalteter Spannungsversorgung aus dem System. Es besteht die Gefahr eines elektrischen Schlags.

⚠ VORSICHT Berühren Sie Klemmen oder Klemmenblöcke nicht bei eingeschalteter Versorgungsspannung. Es besteht die Gefahr eines elektrischen Schlags.

⚠ **VORSICHT** Versuchen Sie nicht, Baugruppen zu zerlegen, instand zu setzen oder zu modifizieren. Bei jedem Versuch besteht das Risiko von Fehlfunktionen sowie die Gefahr eines elektrischen Schlags bzw. Brandgefahr.

⚠ **VORSICHT** Statten Sie externe Schaltkreise (d. h., nicht die in der SPS) mit Sicherheitsmaßnahmen aus, damit die Sicherheit im System gewährleistet bleibt, wenn aufgrund einer Fehlfunktion der SPS oder eines anderen externen Faktors mit Auswirkungen auf den SPS-Betrieb eine Störung auftritt. Dabei kann es sich z. B. um die im Folgenden genannten Maßnahmen handeln. Andernfalls besteht die Gefahr von schweren Unfällen.

- Externe Schaltungen müssen mit Not-Aus-, Verriegelungs-, Begrenzungsschaltungen oder ähnlichen Sicherheitsmaßnahmen ausgestattet werden.
- Die SPS schaltet alle Ausgänge auf AUS, wenn durch die Selbstdiagnosefunktion ein Fehler (schwerwiegender Systemfehler) erkannt wird oder wenn ein FALS-Befehl (schwerwiegender anwenderdefinierter Fehler) ausgeführt wird. Zur Gewährleistung der Sicherheit im System müssen als Gegenmaßnahme für solche Fehler externe Sicherheitsmaßnahmen getroffen werden.
- Die SPS-Ausgänge können aufgrund von Ablagerungen oder Verbrennungen an den Ausgangsrelaiskontakten bzw. Zerstörung von Ausgangstransistoren irrtümlicherweise auf EIN oder AUS geschaltet bleiben. Zur Gewährleistung der Sicherheit im System müssen als Gegenmaßnahme für solche Probleme externe Sicherheitsmaßnahmen getroffen werden.
- Wenn der 24-V-DC-Ausgang (Hilfsspannungsversorgung der SPS) überlastet oder kurzgeschlossen wird, fällt möglicherweise die Spannung ab, was dazu führt, dass die Ausgänge auf AUS geschaltet werden. Zur Gewährleistung der Sicherheit im System müssen als Gegenmaßnahme für solche Probleme externe Sicherheitsmaßnahmen getroffen werden.

⚠ **Achtung** Überprüfen Sie die Sicherheit, bevor Sie unter Verwendung einer Programmiersoftware im Dateispeicher (Speicherkarte oder EM-Dateispeicher) gespeicherte Datendateien an den E/A-Bereich (CIO) der CPU-Baugruppe übertragen. Andernfalls können unabhängig von der Betriebsart der CPU-Baugruppe Fehlfunktionen an den mit der Ausgangsbaugruppe verbundenen Geräten auftreten.

⚠ **Achtung** Der Kunde muss durch entsprechende Vorkehrungen die Systemsicherheit für den Fall sicherstellen, dass bedingt durch unterbrochene Signalleitungen, vorübergehende Spannungsausfälle oder beliebige andere Ursachen falsche, fehlende oder ungewöhnliche Signale auftreten. Wenn der Betrieb aufgrund fehlender Sicherheitsvorkehrungen nicht ordnungsgemäß erfolgt, kann dies zu schweren Unfällen führen.

⚠ **Achtung** Führen Sie die Online-Bearbeitung der Programme oder Parameter nur durch, nachdem Sie überprüft haben, dass eine Verlängerung der Zykluszeit keine negativen Auswirkungen mit sich bringt. Andernfalls sind die Eingangssignale möglicherweise nicht lesbar.

⚠ **Achtung** Das Anwenderprogramm und die Parameterdaten werden von den CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen beim Schreiben in die CPU-Baugruppe automatisch im Flash-Speicher gesichert. Der E/A-Speicher (einschließlich der DM-, EM- und HR-Bereiche) wird jedoch nicht in den Flash-Speicher geschrieben. Die Inhalte der DM-, EM- und HR-Bereiche können während eines Ausfalls der Versorgungsspannung durch eine Batterie erhalten werden. Wenn die Batterie nicht ordnungsgemäß funktioniert, sind die Inhalte dieser Bereiche nach einem Ausfall der Versorgungsspannung evtl. nicht mehr korrekt. Wenn der Inhalt der DM-, EM- und HR-Bereiche zur Steuerung externer Ausgänge verwendet wird, müssen geeignete Maßnahmen zur Vermeidung von falschen Ausgaben ergriffen werden, wenn das Batterie-Fehlerbit (A40204) auf EIN gesetzt ist.

⚠ **Achtung** Stellen Sie die Sicherheit am Zielknotenpunkt sicher, bevor Sie ein Programm an einen anderen Knotenpunkt übertragen oder den Inhalt des E/A-Speicherbereichs verändern. Wenn Sie eine dieser Aktionen durchführen, ohne die Sicherheit zu überprüfen, kann dies zu Verletzungen führen.

⚠ Achtung Ziehen Sie die Schrauben am Klemmenblock, speziell die der Spannungsversorgungsbaugruppe, mit dem im Bedienerhandbuch angegebenen Drehmoment fest. Lose Schrauben können zu Schmorbränden oder Fehlfunktionen führen.

⚠ Achtung Berühren Sie die Spannungsversorgungsbaugruppe nicht bei eingeschalteter Versorgungsspannung bzw. direkt nach Ausschalten der Versorgungsspannung. Die Spannungsversorgungs-Baugruppe kann in diesen Fällen heiß sein, und Sie können sich Verbrennungen zuziehen.

⚠ Achtung Lassen Sie beim Anschließen eines PCs oder anderer Peripheriegeräte an eine SPS besondere Vorsicht walten, wenn an diese eine nicht isolierte, an eine externe Spannungsversorgung angeschlossene Baugruppe (CS1W-CLK12/52(-V1) oder CS1W-ETN01) angeschlossen ist. Wenn die 24-V-Seite der externen Spannungsversorgung und die 0-V-Seite des Peripheriegerätes gleichzeitig geerdet sind, tritt ein Kurzschluss auf. Erden Sie beim Anschluss eines Peripheriegerätes an diese Art von SPS entweder die 0-V-Seite der externen Spannungsversorgung, oder erden Sie die externe Spannungsversorgung überhaupt nicht.

4 Sicherheitshinweise für die Betriebsumgebung

⚠ Achtung Betreiben Sie die SPS nicht an folgenden Orten:

- Orte, die direkter Sonneneinstrahlung ausgesetzt sind.
- Orte, an denen Temperaturen oder Luftfeuchtigkeit außerhalb der in den technischen Daten angegebenen Bereiche herrschen.
- Orte, die starken Temperaturschwankungen und damit Kondensatbildung ausgesetzt sind.
- Orte, an denen korrosive oder entzündliche Gase vorhanden sind.
- Orte, die dem Einfluss von Stäuben (besonders Eisenstaub) oder Salzen ausgesetzt sind.
- Orte, die dem Einfluss von Feuchtigkeit, Öl oder Chemikalien ausgesetzt sind.
- Orte, die Stößen oder Schwingungen ausgesetzt sind.

⚠ Achtung Ergreifen Sie beim Installieren von Systemen an folgenden Orten angemessene und geeignete Gegenmaßnahmen:

- Orte mit statischer Aufladung und anderen Störungen.
- Orte mit starken elektromagnetischen Feldern.
- Orte, die möglicherweise Radioaktivität ausgesetzt sind.
- Orte in der Nähe von Spannungsversorgungseinrichtungen.

⚠ Achtung Die Betriebsumgebung des SPS-Systems kann große Auswirkungen auf die Lebensdauer und Zuverlässigkeit des Systems haben. Ungeeignete Betriebsumgebungen können Fehlfunktionen, Ausfälle und andere unvorhergesehene Probleme in Bezug auf das SPS-System zur Folge haben. Achten Sie darauf, dass die Betriebsumgebung zum Zeitpunkt der Installation die angegebenen Bedingungen erfüllt und dass diese Bedingungen während der Lebensdauer des Systems erfüllt bleiben.

5 Sicherheitshinweise zu Anwendungen

Beachten Sie bei der Verwendung des SPS-Systems die folgenden Sicherheitshinweise.

- Verwenden Sie CX-Programmer (Programmier-Software, die unter Windows läuft), wenn Sie mehr als eine Task programmieren möchten. Eine Programmierkonsole kann verwendet werden, um nur eine zyklische Task plus Interrupt-Tasks zu programmieren. Eine Programmierkonsole kann jedoch auch zum Bearbeiten von Multitask-Programmen verwendet werden, die ursprünglich mit CX-Programmer erstellt wurden.

⚠ VORSICHT Beachten Sie stets diese Sicherheitshinweise. Das Nichtbeachten der folgenden Sicherheitshinweise kann schwere und möglicherweise tödliche Verletzungen zur Folge haben.

- Achten Sie bei der Installation der SPS auf eine ordnungsgemäße Erdung, die den geltenden Vorschriften entspricht. Bei fehlender oder nicht vorschriftsmäßiger Erdung besteht die Gefahr eines elektrischen Schlags.
- Eine den Vorschriften entsprechende Erdung ist auch dann erforderlich, wenn die Schutzerdungsklemme (GR) und die Signalerdungsklemme (LG) der Spannungsversorgungs-Baugruppe miteinander verbunden werden.
- Schalten Sie die Spannungsversorgung zur SPS immer AUS, bevor Sie versuchen, eine der folgenden Arbeiten durchzuführen. Bei Nichtabschalten der Versorgungsspannung besteht das Risiko von Fehlfunktionen und die Gefahr eines elektrischen Schlags.
  - Anbringen oder Abbauen von Spannungsversorgungsbaugruppen, E/A-Baugruppen, CPU-Baugruppen, internen Modulen oder sonstigen Baugruppen.
  - Zusammensetzen der Baugruppen.
  - Einstellen der DIP-Schalter oder Drehschalter.
  - Anschließen von Kabeln oder Verdrahten des Systems.
  - Anschließen oder Lösen von Steckverbindungen.

⚠ Achtung Das Nichtbeachten der folgenden Sicherheitshinweise kann einen fehlerhaften Betrieb der SPS oder des Systems zur Folge haben, oder es könnten die SPS oder die SPS-Baugruppen beschädigt werden. Beachten Sie stets diese Sicherheitshinweise.

- Die Anwenderprogramm- und Parameterbereichsdaten in CS1-H, CS1D, CJ1-H und CJ1M CPU-Baugruppen werden im integrierten Flash-Speicher gesichert. Die BKUP-Anzeige auf der Vorderseite der CPU-Baugruppe leuchtet, während die Datensicherung ausgeführt wird. Schalten Sie die Versorgungsspannung der CPU-Baugruppe nicht aus, solange die BKUP-Anzeige leuchtet. Die Daten werden nicht gesichert, wenn die Versorgungsspannung ausgeschaltet wird.
- Wenn Sie eine CPU-Baugruppe der CS-Serie erstmalig verwenden, setzen Sie die mit der Baugruppe mitgelieferte Batterie CS1W-BAT1 ein, und löschen Sie vor dem Programmieren sämtliche Speicherbereiche mittels eines Programmiergeräts. Wenn Sie die interne Uhr verwenden, schalten Sie nach dem Einsetzen der Batterie die Versorgungsspannung ein, und stellen Sie die Uhr mit Hilfe eines Programmiergeräts oder des Befehls DATE(735) ein. Die Uhr läuft erst, wenn die Zeit eingestellt wurde.
- Bei Auslieferung der CPU-Baugruppe ist das SPS-Setup so eingestellt, dass die CPU-Baugruppe in der Betriebsart startet, der mit dem Betriebsartenschalter der Programmierkonsole eingestellt ist. Wenn keine Programmierkonsole angeschlossen ist, startet eine CS1-CPU-Baugruppe der CS-Serie in der PROGRAM-Betriebsart, die CS1-H, CS1D, CJ1, CJ1-H oder CJ1M CPU-Baugruppen starten in der RUN-Betriebsart, so dass der Betrieb sofort aufgenommen wird. Starten Sie niemals absichtlich oder unabsichtlich den Betrieb, ohne sich zu vergewissern, dass dies sicher ist.
- Setzen Sie beim Erstellen einer AUTOEXEC.IOM-Datei über ein Programmiergerät (eine Programmierkonsole oder CX-Programmer) zum automatischen Übertragen von Daten beim Start die erste Schreibadresse auf D20000, und achten Sie darauf, dass die Größe der geschriebenen Daten nicht die Größe des DM-Bereichs übersteigt. Wenn die Datendatei beim Start von der Speicherkarte gelesen wird, werden beginnend bei D20000 Daten in die CPU-Baugruppe geschrieben, selbst wenn beim Erstellen der Datei AUTOEXEC.IOM eine andere Adresse eingestellt war. Wird ferner der DM-Bereich überschritten (dies ist beim Verwenden von CX-Programmer möglich), werden die übrigen Daten in den EM-Bereich geschrieben.
- Schalten Sie die Versorgungsspannung der SPS stets vor der der gesteuerten Anlage ein. Wird die Versorgungsspannung der SPS erst nach der der gesteuerten Anlage eingeschaltet, können zeitweilig Fehler in Steuerungssystemsignalen auftreten, da die Ausgangsklemmen an DC-Ausgangs-Bau-

gruppen und anderen Baugruppen vorübergehend auf EIN geschaltet werden, wenn die Versorgungsspannung der SPS eingeschaltet wird.

- Der Anwender muss durch entsprechende Vorkehrungen die Systemsicherheit für den Fall sicherstellen, dass bedingt durch Schäden an internen Schaltkreisen (Relais, Transistoren oder andere Komponenten) von Ausgangs-Baugruppen intern auf AUS gesetzte Ausgänge weiterhin auf EIN geschaltet bleiben.
- Der Anwender muss durch entsprechende Vorkehrungen die Systemsicherheit für den Fall sicherstellen, dass bedingt durch unterbrochene Signalleitungen, vorübergehende Spannungsausfälle oder beliebige andere Ursachen falsche, fehlende oder ungewöhnliche Signale auftreten.
- Der Anwender muss Verriegelungs- und Begrenzungsschaltungen sowie ähnliche Sicherheitsmaßnahmen in externen Schaltungen (d. h. nicht die in der speicherprogrammierbaren Steuerung) vorsehen.
- Schalten Sie die Spannungsversorgung zur SPS nicht AUS, während Daten übertragen werden. Schalten Sie die Spannungsversorgung insbesondere dann nicht AUS, wenn eine Speicherkarte gelesen oder beschrieben wird. Die Speicherkarte darf auch nicht entfernt werden, wenn die BUSY-Anzeige leuchtet. Drücken Sie zum Entnehmen einer Speicherkarte zunächst die Speicherkartentaste, und warten Sie dann, bis die BUSY-Anzeige erlischt, bevor Sie die Speicherkarte entnehmen.
- Wenn das E/A-Haltebit auf EIN gesetzt ist, werden die Ausgänge der SPS nicht auf AUS geschaltet und behalten ihren vorherigen Status bei, wenn die SPS von der RUN- oder MONITOR- in die PROGRAM-Betriebsart umgeschaltet wird. Stellen Sie sicher, dass in diesem Fall keine Gefahren von den angeschlossenen Geräten ausgehen. (Wenn der Betrieb aufgrund eines schwerwiegenden Fehlers, einschließlich jenen mit dem Befehl FALS(007) erzeugten, angehalten wird, werden alle Ausgänge von Ausgangsbaugruppen auf AUS geschaltet, und nur der interne Ausgangsstatus wird beibehalten.)
- Der Speicherinhalt der DM-, EM- und HR-Bereiche in der CPU-Baugruppe wird durch eine Batterie gesichert. Wenn die Spannung der Batterie abfällt, können diese Daten verloren gehen. Ergreifen Sie mit Hilfe des Batterie-Fehlerbits (A40204) Vorkehrungen für die Reinitialisierung von Daten oder andere Maßnahmen für den Fall, dass die Spannung der Batterie abfällt.
- Wenn eine SPS der CS-Serie mit 200 bis 240 V Wechselspannung versorgt wird, müssen Sie stets die Drahtbrücken von den Spannungsauswahlklemmen der Spannungsversorgungs-Baugruppe entfernen (mit Ausnahme von Spannungsversorgungs-Baugruppen mit weitem Eingangsspannungsbereich). Das Produkt wird zerstört, wenn bei angebrachten Drahtbrücken 200 bis 240 V AC angelegt werden.
- Verwenden Sie stets die in den Bedienerhandbüchern angegebenen Versorgungsspannungen. Eine falsche Spannung kann zu Fehlfunktionen oder Bränden führen.
- Ergreifen Sie geeignete Maßnahmen, um sicherzustellen, dass das Gerät mit der angegebenen Nennspannung und -frequenz versorgt wird. Lassen Sie an Installationsorten, an denen die Spannungsversorgung nicht stabil ist, besondere Vorsicht walten. Eine falsche Spannungsversorgung kann zu Fehlfunktionen führen.
- Installieren Sie externe Trennschalter, und ergreifen Sie weitere Sicherheitsmaßnahmen gegen Kurzschlüsse in externer Verdrahtung. Bei unzureichenden Sicherheitsmaßnahmen gegen Kurzschlüsse besteht Brandgefahr.
- Legen Sie keine Spannungen an die Eingangsbaugruppen an, die die Nenneingangsspannung übersteigen. Bei zu hohen Spannungen besteht Brandgefahr.
- Legen Sie keine Spannungen an die Ausgangsbaugruppen an und schließen Sie keine Lasten an die Ausgangsbaugruppen an, die die maximale Schaltleistung übersteigen. Bei übermäßigen Spannungen oder Lasten besteht Brandgefahr.

- Trennen Sie vor der Durchführung von Isolationswiderstandsmessungen oder Spannungsbeständigkeitstests die Signalmasse (LG) von der Gehäuseerde (GR). Andernfalls kann es zu Verbrennungen kommen.
- Installieren Sie die Baugruppen ordnungsgemäß, wie in den Bedienerhandbüchern angegeben. Bei unsachgemäßer Installation der Baugruppen besteht die Gefahr von Fehlfunktionen.
- Bei SPS-Systemen der CS-Serie muss sichergestellt werden, dass sämtliche Befestigungsschrauben der Baugruppen und des Baugruppenträgers mit dem in den entsprechenden Handbüchern angegebenen Drehmoment festgezogen werden. Ein falsches Anzugsdrehmoment kann zu Fehlfunktionen führen.
- Stellen Sie sicher, dass alle Klemmenschrauben und Schrauben von Steckverbindern mit dem in den entsprechenden Handbüchern angegebenen Drehmoment festgezogen werden. Ein falsches Anzugsdrehmoment kann zu Fehlfunktionen führen.
- Belassen Sie beim Verdrahten die Schutzfolie an der Baugruppe. Beim Entfernen der Schutzfolie besteht die Gefahr einer Fehlfunktion, wenn Fremdkörper in die Baugruppe eindringen.
- Entfernen Sie die Schutzfolie nach Abschluss der Verdrahtungsarbeiten, um eine ordnungsgemäße Wärmeableitung zu gewährleisten. Bei nicht entfernter Schutzfolie besteht die Gefahr von Fehlfunktionen.
- Verwenden Sie für die Verdrahtung Crimp-Kabelschuhe. Schließen Sie keine abisolierten Litzendrähte direkt an Klemmen an. Beim Anschließen von abisolierten Litzendrähten besteht die Gefahr von Schmorbränden.
- Verdrahten Sie alle Anschlüsse ordnungsgemäß.
- Überprüfen Sie die gesamte Verdrahtung und alle Schaltereinstellungen sorgfältig, bevor Sie die Spannungsversorgung einschalten. Bei unsachgemäßer Verdrahtung besteht Brandgefahr.
- Installieren Sie Baugruppen erst, nachdem Sie die Klemmenblöcke und Steckverbinder eingehend überprüft haben.
- Stellen Sie sicher, dass die Klemmenblöcke, Speicherbaugruppen, Verlängerungskabel und andere Teile mit Arretierungen ordnungsgemäß eingerastet sind. Bei fehlender Arretierung besteht die Gefahr von Fehlfunktionen.
- Überprüfen Sie vor Beginn des Betriebs die Schaltereinstellungen, den Inhalt des DM-Bereichs sowie andere Voreinstellungen. Bei Aufnahme des Betriebs ohne sachgemäße Einstellungen bzw. Daten besteht die Gefahr von unerwartetem Verhalten.
- Überprüfen Sie das Anwenderprogramm auf ordnungsgemäße Ausführung, bevor Sie es tatsächlich für die SPS einsetzen. Wird das Programm nicht überprüft, besteht die Gefahr von unerwartetem Verhalten.
- Vergewissern Sie sich, dass keine negativen Auswirkungen im System auftreten können, bevor Sie eine der folgenden Aktionen durchführen. Andernfalls besteht die Gefahr von unerwartetem Verhalten.
  - Ändern der Betriebsart der SPS (einschließlich der Einstellung der Betriebsart beim Start).
  - Zwangsweises Setzen/Rücksetzen eines Bits im Speicher.
  - Ändern des aktuellen Wertes eines Wortes oder eines voreingestellten Wertes im Speicher.
- Ziehen Sie nicht an den Kabeln, und biegen Sie die Kabel nicht über das natürliche Maß hinaus. Andernfalls können die Kabel dabei brechen.
- Stellen Sie keine Gegenstände auf die Kabel oder andere Leitungen. Die Kabel können dabei brechen.
- Verwenden Sie keine handelsüblichen RS-232C-Computerkabel, sondern ausschließlich die hier aufgeführten Spezialkabel. Müssen Kabel angefertigt werden, so beachten Sie unbedingt die Angaben im Bedienerhandbuch. Bei Verwendung von handelsüblichen Kabeln besteht die Gefahr, dass externe Geräte oder die CPU-Baugruppe beschädigt werden.

- Verbinden Sie niemals Stift 6 (5-V-Spannungsversorgung) des RS-232C-Anschlusses der CPU-Baugruppe mit einem anderen Gerät als den Adaptern NT-AL001 oder CJ1W-CIF11. Das externe Gerät oder die CPU-Baugruppe können andernfalls beschädigt werden.
- Achten Sie beim Austauschen von Komponenten darauf, dass die neue Komponente die richtigen Nenndaten besitzt. Andernfalls besteht die Gefahr von Fehlfunktionen oder Brandgefahr.
- Achten Sie vor Berühren der SPS darauf, dass Sie zunächst einen geerdeten Metallgegenstand berühren, um statische Aufladung abzuleiten. Andernfalls besteht die Gefahr von Fehlfunktionen oder Beschädigungen.
- Bewahren Sie Leiterplatten bei Transport oder Lagerung in antistatischem Material auf, um sie vor statischer Elektrizität zu schützen. Achten Sie auf die Einhaltung der zulässigen Lagerungstemperatur.
- Berühren Sie Leiterplatten oder die darauf befindlichen Komponenten nicht mit bloßen Händen. Auf den Leiterplatten befinden sich scharfe Anschlussdrähte und andere Teile, die bei unsachgemäßer Handhabung zu Verletzungen führen können.
- Schließen Sie die Batterieanschlüsse niemals kurz. Die Batterie darf nicht aufgeladen, zerlegt, erhitzt oder verbrannt werden. Setzen Sie die Batterie keinen starken Erschütterungen aus. Andernfalls besteht die Gefahr, dass die Batterie undicht wird, bricht, Hitze erzeugt oder sich entzündet. Entsorgen Sie Batterien, die heruntergefallen sind oder auf sonstige Weise heftigen Erschütterungen ausgesetzt waren. Batterien, die Erschütterungen ausgesetzt waren, können beim Gebrauch möglicherweise undicht sein.
- Nach UL-Standards dürfen Batterien nur von erfahrenen Technikern ausgewechselt werden. Lassen Sie die Batterien nicht von Personen austauschen, die nicht dafür qualifiziert sind.
- Bei einer SPS der CJ-Serie müssen die Schieber an den Ober- und Unterseiten der Spannungsversorgungs-Baugruppen, CPU-Baugruppen, E/A-Baugruppen, Spezial-E/A-Baugruppen und CPU-Bus-Baugruppen vollständig verriegelt werden (sie müssen hörbar einrasten). Die SPS funktioniert möglicherweise nicht ordnungsgemäß, wenn die Schieber nicht eingerastet sind.
- Bringen Sie bei einer SPS der CJ-Serie stets die Endabdeckung an der letzten Baugruppe rechts an. Die SPS funktioniert ohne die Endabdekkung nicht ordnungsgemäß.
- Fehlerhaft eingerichtete Data-Link-Tabellen oder -Parameter können zu unerwartetem Systemverhalten führen. Stellen Sie auch bei korrekt eingerichteten Data-Link-Tabellen und -Parametern vor dem Starten und Stoppen von Data-Links sicher, dass dies keine unerwünschten Auswirkungen auf das System hat.
- CPU-Bus-Baugruppen werden neu gestartet, wenn Routing-Tabellen von einem Programmiergerät auf die CPU-Baugruppe übertragen werden. Das Neustarten der Baugruppen ist zum Lesen und Aktivieren der neuen Routing-Tabellen erforderlich. Stellen Sie vor dem Neustart der CPU-Bus-Baugruppen sicher, dass dies keine unerwünschten Auswirkungen auf das System hat.

6 Konformität mit EU-Richtlinien

6-1 Anwendbare Richtlinien

- EMV-Richtlinien
- Niederspannungsrichtlinie

6-2 Konzepte

EMV-Richtlinien

OMRON-Automationsgeräte, die den EU-Richtlinien entsprechen, entsprechen ebenfalls den damit zusammenhängenden EMV-Richtlinien, so dass sie problem-

los in andere Gerätekombinationen oder in eine Gesamtanlage integriert werden können. Die tatsächlichen Produkte wurden auf Konformität mit EMV-Normen geprüft (siehe folgender Hinweis). Ob die Produkte jedoch in dem vom Kunden verwendeten System den Normen entsprechen, muss der Kunde selbst prüfen.

Die Abstrahlungseigenschaften von OMRON-Produkten, die den EU-Richtlinien entsprechen, variieren je nach Konfiguration, Verdrahtung und anderen Bedingungen der Installation oder der Schaltschränke, in denen die OMRON-Geräte installiert sind. Der Kunde muss daher eine abschließende Prüfung durchführen, um die Bestätigung zu erhalten, dass die Geräte und die Gesamtanlage den EMV-Normen entsprechen.

Hinweis Die folgenden EMV-Normen (elektromagnetische Verträglichkeit) sind anwendbar:

EMS (Elektromagnetische Störfestigkeit): EN61131-2 (CS-Serie)/ EN61000-6-2 (CJ-Serie)

EMI (Elektromagnetische Störaussendung): EN61000-6-4 (Abstrahlung: 10-m-Bestimmungen)

Niederspannungsrichtlinie

Stellen Sie immer sicher, dass Geräte, die mit Spannungen von 50 bis 1.000 V AC und 75 bis 1.500 V DC arbeiten, die erforderlichen Sicherheitsnormen für die SPS erfüllen (EN61131-2).

6-3 Konformität mit EU-Richtlinien

Die SPS der CS/CJ-Serie entsprechen den EU-Richtlinien. Um zu gewährleisten, dass die Anlage oder das Gerät, in dem die SPS der CS/CJ-Serie verwendet wird, den EU-Richtlinien entspricht, muss die SPS wie folgt installiert sein:

1,2,3...

1. Die SPS der CS/CJ-Serie muss in einem Schaltschrank installiert sein.
2. Sie müssen eine Schutzisolierung oder eine doppelte Isolierung für die DC-Spannungsversorgungen der Kommunikations- und E/A-Baugruppen anbringen.
3. SPS der CS/CJ-Serie, die den EU-Richtlinien entsprechen, entsprechen ebenfalls der allgemeinen Emissionsnorm (EN61000-6-4). Die Abstrahlungseigenschaften (10-m-Bestimmungen) variieren je nach Konfiguration des verwendeten Schaltschranks, anderen an den Schaltschrank angeschlossenen Geräten, Verdrahtung und weiteren Bedingungen. Daher müssen Sie überprüfen, ob die Gesamtanlage oder -ausrüstung den EU-Richtlinien entspricht.

6-4 Methoden zur Entstörung von Relaisausgängen

Die SPS der CS/CJ-Serie entsprechen den allgemeinen Emissionsnormen (EN61000-6-4) der EMV-Richtlinien. Eine durch Schalten des Relaisausgangs erzeugte Störung erfüllt diese Norm jedoch möglicherweise nicht. In diesem Fall muss ein Entstörfilter an der Lastseite angeschlossen werden, oder für die SPS müssen andere geeignete externe Gegenmaßnahmen eingerichtet werden.

Die zur Erfüllung der Normen vorzunehmenden Gegenmaßnahmen variieren je nach Gerät auf der Lastseite, Verdrahtung, Konfiguration der Anlagen usw. Es folgen Beispiele für Gegenmaßnahmen zur Verringerung der erzeugten Störungen.

Gegenmaßnahmen

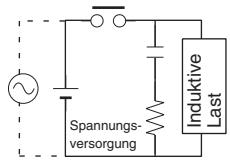
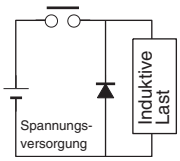
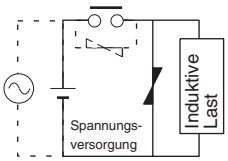
(Nähere Einzelheiten finden Sie in EN61000-6-4.)

Es sind keine Gegenmaßnahmen erforderlich, wenn die Frequenz der Lastschaltung im Gesamtsystem, in das die SPS eingebunden ist, unter fünf Schaltungen pro Minute liegt.

Es sind Gegenmaßnahmen erforderlich, wenn die Frequenz der Lastschaltung im Gesamtsystem, in das die SPS eingebunden ist, über fünf Schaltungen pro Minute liegt.

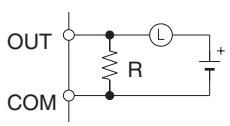
Beispiele für Gegenmaßnahmen

Wird eine induktive Last geschaltet, schließen Sie einen Überspannungsschutz, Dioden usw. parallel zu der Last oder dem Kontakt an, wie unten gezeigt.

Schaltung	Strom		Eigenschaften	Erforderliche Komponenten
	AC	DC		
RC-Glied 	Ja	Ja	Wenn es sich bei der Last um ein Relais oder eine Magnetspule handelt, gibt es eine Verzögerung zwischen dem Öffnen des Schaltkreises und dem Rücksetzen der Last. Wenn die Versorgungsspannung 24 oder 48 V beträgt, setzen Sie den Überspannungsschutz parallel zur Last ein. Wenn die Versorgungsspannung 100 bis 200 V beträgt, setzen Sie den Überspannungsschutz zwischen den Kontakten ein.	Die Kapazität des Kondensators muss zwischen 1 und 0,5 μF pro Kontaktstrom von 1 A, und der Widerstand muss zwischen 0,5 und 1 Ω pro Kontaktspannung von 1 V betragen. Diese Werte variieren jedoch entsprechend der Last und den Eigenschaften des Relais. Ermitteln Sie experimentell die optimalen Werte. Berücksichtigen Sie dabei, dass der Kondensator die Funkenentladung beim Öffnen der Kontakte unterdrücken und der Widerstand den beim Schließen der Kontakte fließenden Ladestrom des Kondensators begrenzen soll. Die Durchschlagfestigkeit des Kondensators muss zwischen 200 und 300 V liegen. Verwenden Sie bei Wechselstrom-Schaltkreisen einen Kondensator ohne Polarität.
Freilaufdiode 	Nein	Ja	Die parallel zur Last geschaltete Diode (Freilaufdiode) leitet den beim Abbau des Magnetfelds entstehenden Strom wieder in die Last, wo er aufgrund des Ohmschen Widerstands der Last in Wärme umgewandelt wird. Diese durch diese Methode verursachte zeitliche Verzögerung zwischen dem Öffnen des Schaltkreises und dem Rücksetzen der Last ist länger als die durch die Kondensator/ Widerstand-Methode verursachte.	Der Wert für die Durchschlagfestigkeit der Diode in Sperrrichtung muss mindestens zehnmal so hoch sein wie der Spannungswert in der Schaltung. Der Durchlassstrom der Diode muss gleich oder größer als der Laststrom sein. Bei Niederspannungsschaltkreisen genügt es, wenn die Spitzensperrspannung der Freilaufdiode dem Zwei- bis Dreifachen der Lastspannung entspricht.
Varistor 	Ja	Ja	Der Varistor wird bei Überschreitung einer bestimmten Spannung, die größer als die Lastspannung gewählt werden muss, niederohmig. Auf diese Weise wird der beim Abbau des Magnetfelds der induktiven Last entstehende Strom kurzgeschlossen und die Entstehung eines Schaltfunkens an den Kontakten verhindert. Es tritt eine zeitliche Verzögerung zwischen dem Öffnen des Schaltkreises und dem Rücksetzen der Last auf. Beträgt die Versorgungsspannung 24 oder 48 V, so schalten Sie den Varistor parallel zur Last. Beträgt die Versorgungsspannung 100 oder 200 V, so schalten Sie den Varistor parallel zu den Schaltkontakten.	---

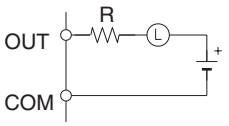
Beim Schalten einer Last mit einem hohen Einschaltstrom, etwa einer Glühlampe, reduzieren Sie den Einschaltstrom wie unten gezeigt.

Abhilfemaßnahme 1



Durch einen Vorschaltwiderstand fließt bereits ca. ein Drittel des Nennstroms der Glühlampe.

Abhilfemaßnahme 2



Verwendung eines Begrenzungswiderstands

KAPITEL 1
Einführung

Dieses Kapitel beinhaltet Informationen über allgemeine Befehlseigenschaften sowie über die Fehler, die während der Befehlsausführung auftreten können.

1-1 Allgemeine Befehlseigenschaften 2
1-1-1 Programmkapazität 2
1-1-2 Befehle mit Flankenausführung 3
1-1-3 Befehlsvariationen 4
1-1-4 Befehlsposition und Ausführungsbedingungen 5
1-1-5 Eingabe von Daten in Operanden 5
1-1-6 Datenformate 11
1-2 Befehlsausführungsprüfungen 13
1-2-1 Fehler bei der Befehlsausführung 13
1-2-2 Schwerwiegende Fehler (Programmfehler) 13

1-1 Allgemeine Befehlseigenschaften

1-1-1 Programmkapazität

Die Programmkapazität ist ein Maß für die Größe des Anwenderprogrammbereichs in der CPU-Baugruppe und wird als Anzahl an Programm-Steps ausgedrückt. Je nach Befehl und Operanden belegt ein Befehl der CS/CJ-Serie zwischen ein und sieben Steps im Anwenderprogramm.

CS-Serie

In den folgenden Tabellen sind die maximalen Anzahlen von Steps aufgeführt, die in jeder CPU-Baugruppe der CS-Serie programmiert werden können.

- CS1-H CPU-Baugruppen

Produktbezeichnung	Programmkapazität	E/A-Punkte
CS1H-CPU67H	250 kSteps	5.120
CS1H-CPU66H	120 kSteps	
CS1H-CPU65H	60 kSteps	
CS1H-CPU64H	30 kSteps	
CS1H-CPU63H	20 kSteps	
CS1G-CPU45H	60 kSteps	
CS1G-CPU44H	30 kSteps	1.280
CS1G-CPU43H	20 kSteps	960
CS1G-CPU42H	10 kSteps	

- CS1 CPU-Baugruppen

Produktbezeichnung	Programmkapazität	E/A-Punkte
CS1H-CPU67-E	250 kSteps	5.120
CS1H-CPU66-E	120 kSteps	
CS1H-CPU65-E	60 kSteps	
CS1H-CPU64-E	30 kSteps	
CS1H-CPU63-E	20 kSteps	
CS1G-CPU45-E	60 kSteps	
CS1G-CPU44-E	30 kSteps	1.280
CS1G-CPU43-E	20 kSteps	960
CS1G-CPU42-E	10 kSteps	

- CS1D CPU-Baugruppen für Einzel-CPU-Systems

Produktbezeichnung	Programmkapazität	E/A-Punkte
CS1D-CPU67H	250 kSteps	5.120
CS1D-CPU65H	60 kSteps	

CS1D CPU-Baugruppen für Duplex-CPU-Systeme

Produktbezeichnung	Programmkapazität	E/A-Punkte
CS1D-CPU42S	10 kSteps	960
CS1D-CPU44S	30 kSteps	1.280
CS1D-CPU65S	60 kSteps	5.120
CS1D-CPU67S	250 kSteps	

CJ-Serie

In den folgenden Tabellen sind die maximalen Anzahlen von Steps aufgeführt, die in jeder CPU-Baugruppe der CJ-Serie programmiert werden können.

- CJ1-H CPU-Baugruppen

Produktbezeichnung	Programmkapazität	E/A-Punkte
CJ1H-CPU67H	250 kSteps	2.560
CJ1H-CPU66H	120 kSteps	
CJ1H-CPU65H	60 kSteps	
CJ1G-CPU45H	60 kSteps	1.280
CJ1G-CPU44H	30 kSteps	
CJ1G-CPU43H	20 kSteps	960
CJ1G-CPU42H	10 kSteps	

- CJ1 CPU-Baugruppen

Produktbezeichnung	Programmkapazität	E/A-Punkte
CJ1G-CPU45	60 kSteps	1.280
CJ1G-CPU44	30 kSteps	

- CJ1M CPU-Baugruppen

Produktbezeichnung	Programmkapazität	E/A-Punkte
CJ1M-CPU23	20 kSteps	640
CJ1M-CPU22	10 kSteps	320
CJ1M-CPU21	5 kSteps	160
CJ1M-CPU13	20 kSteps	640
CJ1M-CPU12	10 kSteps	320
CJ1M-CPU11	5 kSteps	160

Hinweis Anders als bei früheren SPS-Systemen von OMRON, bei denen die Programmkapazität in Worten angegeben wurde, erfolgt die Angabe der Programmkapazität bei SPS-Systemen der CJ-Serie in Steps. Einfach gesagt heißt das, 1 Step entspricht 1 Wort. Der für jeden Befehl erforderliche Speicherplatz ist bei einigen Befehlen der CS/CJ-Serie dennoch abweichend und es treten Fehler auf, wenn die Kapazität eines Anwenderprogramms eines anderen SPS-Systems für eine SPS der CS/CJ-Serie auf der Grundlage, dass 1 Wort gleich 1 Step ist, konvertiert wird. Am Ende dieses Kapitels finden Sie im Abschnitt *KAPITEL 4 Ausführungszeit und Step-Anzahl der Befehle* Richtlinien zur Bestimmung der Kapazität eines solchermaßen konvertierten Programms.

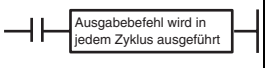
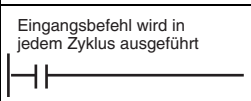
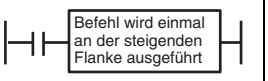
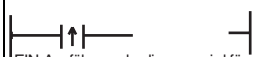
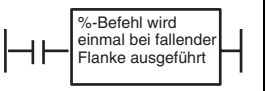
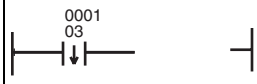
Die Anzahl der Steps in einem Programm entspricht nicht der Anzahl der Befehle. So benötigen beispielsweise die Befehle LD und OUT jeweils einen Step, MOV(021) hingegen benötigt drei Steps. Andere Befehle benötigen bis zu jeweils 7 Steps. Die Anzahl der für einen Befehl erforderlichen Steps erhöht sich für jeden im Befehl verwendeten Operanden doppelter Länge um einen Step. Z.B. benötigt MOVL(498) normalerweise 3 Steps, es sind aber 4 Steps erforderlich, wenn für den Quellwortoperanden eine Konstante spezifiziert wird. In *KAPITEL 4 Ausführungszeit und Step-Anzahl der Befehle* finden Sie Informationen zur Anzahl der für jeden Befehl erforderlichen Steps.

1-1-2 Befehle mit Flankenausführung

Die meisten Befehle für SPS der CS/CJ-Serie stehen in Varianten ohne Flankenüberwachung und Ausführung an der steigenden Flanke zur Verfügung, einige zudem in einer Variante mit Ausführung an der fallenden Flanke.

- Ein Befehl ohne Flankenüberwachung wird in jedem Zyklus ausgeführt.

- Ein Befehl mit Ausführung an der steigenden Flanke wird nur einmal ausgeführt, nachdem seine Ausführungsbedingung von AUS zu EIN wechselt.
- Ein Befehl mit Ausführung an der fallenden Flanke wird nur einmal ausgeführt, nachdem seine Ausführungsbedingung von EIN zu AUS wechselt.

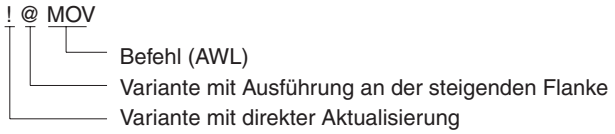
Variante	Befehlstyp	Funktion	Format	Beispiel
Keine Flankenüberwachung	Ausgabebefehle (Befehle, die eine Ausführungsbedingung erfordern)	Der Befehl wird in jedem Zyklus ausgeführt, solange die Ausführungsbedingung wahr (EIN) ist.		
	Eingangsbefehle (Befehle, die als Ausführungsbedingungen verwendet werden)	Die Bit-Verarbeitung (wie z.B. Lesen, Vergleichen oder Testen) wird in jedem Zyklus ausgeführt. Die Ausführungsbedingung ist wahr, wenn das Ergebnis den Zustand EIN hat.		
Ausführung an steigender Flanke (mit @-Präfix)	Ausgabebefehle	Der Befehl wird nur einmal ausgeführt, wenn die Ausführungsbedingung von AUS zu EIN wechselt.		 MOV(021) wird bei jedem AUS-zu-EIN-Wechsel in CIO 000102 einmal ausgeführt.
	Eingangsbefehle (Befehle, die als Ausführungsbedingungen verwendet werden)	Die Bit-Verarbeitung (wie z.B. Lesen, Vergleichen oder Testen) wird in jedem Zyklus ausgeführt. Die Ausführungsbedingung ist für einen Zyklus wahr, wenn das Ergebnis von AUS zu EIN wechselt.		 EIN-Ausführungsbedingung wird für nur einen Zyklus pro AUS-zu-EIN-Wechsel in CIO 000103 erzeugt.
Ausführung an der fallenden Flanke (mit %-Präfix)	Ausgabebefehle	Der Befehl wird nur einmal ausgeführt, wenn die Ausführungsbedingung von EIN zu AUS wechselt.		 SET wird einmal bei jedem EIN-zu-AUS-Wechsel in CIO 000102 ausgeführt.
	Eingangsbefehle (Befehle, die als Ausführungsbedingungen verwendet werden)	Die Bit-Verarbeitung (wie z.B. Lesen, Vergleichen oder Testen) wird in jedem Zyklus ausgeführt. Die Ausführungsbedingung ist für einen Zyklus wahr, wenn das Ergebnis von EIN zu AUS wechselt.		 EIN-Ausführungsbedingung wird für nur einen Zyklus pro EIN-zu-AUS-Wechsel in CIO 000103 erzeugt.

Hinweis Die Option für Ausführung an der fallenden Flanke (%) ist nur bei den Befehlen LD, AND, OR und RSET verfügbar. Um von anderen Befehlen Variationen mit Ausführung an der fallenden Flanke zu erzeugen, muss die Ausführung des Befehls durch Arbeits-Bits gesteuert werden, die mit DIFD(014) oder DOWN(522) gesetzt werden.

1-1-3 Befehlsvariationen

Die Präfixe (@, % und !) können einem Befehl hinzugefügt werden, um einen Befehl mit Flankenüberwachung oder eine direkte Aktualisierung zu erstellen.

Variante		Präfix	Funktion
Flankenüberwachung	Ausführung an steigender Flanke	@	Erzeugt einen Befehl mit Ausführung an der steigenden Flanke.
	Ausführung an fallender Flanke	%	Erzeugt einen Befehl mit Ausführung an der fallenden Flanke.
Direkte Aktualisierung		!	Die Operandendaten des Befehls im E/A-Bereich werden bei Ausführung des Befehls aktualisiert.



1-1-4 Befehlsposition und Ausführungsbedingungen

Die nachfolgende Tabelle zeigt Orte, an denen Befehle programmiert werden können. Die Tabelle zeigt außerdem, wann ein Befehl eine Ausführungsbedingung benötigt und wann nicht. Detaillierte Informationen zu spezifischen Befehlen finden Sie in *KAPITEL 2 Übersicht über die Befehle*.

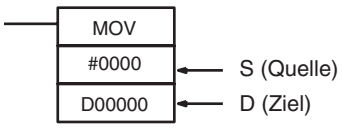
Befehlstyp		Ort	Ausführungsbedingung	Format	Beispiele
Eingang	Befehle, die logische Bedingungen starten	An der linken Stromschiene oder am Beginn eines Befehlsblocks	Nicht erforderlich		LD, LD TST und Symbolvergleichsbefehle, wie z.B. LD >
	Verbindungsbefehle	Zwischen einem Startbefehl und einem Ausgangsbefehl	Erforderlich		AND, OR, AND TST und Symbolvergleichsbefehle, wie z.B. AND >, UP, DOWN, NOT
Ausgang		An der rechten Stromschiene	Erforderlich		Die Mehrheit der Befehle (wie z.B. OUT und MOV)
			Nicht erforderlich		Befehle wie END, JME, FOR und ILC

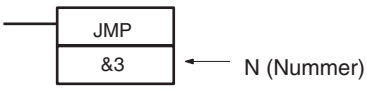
Zusätzlich zu diesen Befehlen sind die SPS der Serie CS/CJ mit Blockprogrammierungsbefehlen ausgestattet. Detaillierte Informationen hierzu finden in den Erläuterungen zur Blockprogrammierung.

Hinweis Wenn einem Befehl, der eine solche erfordert, keine Ausführungsbedingung vorangeht, tritt ein Programmfehler auf, wenn das Programm durch ein Peripheriegerät geprüft wird.

1-1-5 Eingabe von Daten in Operanden

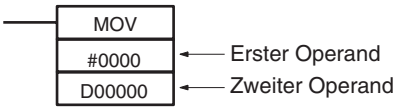
Operanden sind Parameter, die im Voraus als E/A-Speicheradressen oder Konstanten festgelegt werden, welche zur Ausführung des Befehls erforderlich sind. Grundsätzlich gibt es drei Arten von Operanden: Quellenoperanden, Zieloperanden und Nummern.





Operand		Üblicher Code	Inhalt	
Quelle	Daten enthaltende Adresse oder die Daten selbst	S	Quellenoperand	Andere Quelldaten als Steuerdaten
		C	Steuerdaten	Steuerdaten mit einem Bit oder Bits, die die Befehlsausführung steuern
Ziel	Adresse, an der die Daten gespeichert werden	D	---	
Nummer	Enthält eine Nummer, wie z.B. eine Sprungnummer oder Unterprogrammnummer.	N	---	

Hinweis Ein Befehlsoperand kann auch anhand seiner Position im Befehl (erster Operand, zweiter Operand,...) bezeichnet werden. Die für den Operanden verwendeten Codes ändern sich mit der spezifischen Funktion des Operanden.



Spezifizieren von Bit-Adressen

Beschreibung	Beispiel	Befehlsbeispiel
Zur Spezifizierung einer Bitadresse müssen Wortadresse und Bitadresse direkt spezifiziert werden. □□□□ □□ Bitnummer Wortadresse **Hinweis** Das Wortadresse + Bitnummer-Format wird nicht für Zeitgeber-/Zähler-Ablaufbits oder Task-Flags verwendet.	0001 02 Bit 02 Wort CIO 0001	0001 02

Spezifizieren von Wort-Adressen

Beschreibung	Beispiel	Befehlsbeispiel
Zur Spezifizierung einer Wortadresse muss die Wortadresse direkt spezifiziert werden. □□□□ Wortadresse	0003 Wort CIO 0003 D00200 Wort D00200	MOV 0003 D00200

Spezifizieren indirekter DM/EM-Adressen im Binärmodus

Beschreibung	Beispiel	Befehlsbeispiel
Wenn das @-Präfix vor der DM- oder EM-Adresse eingegeben wird, spezifiziert der Inhalt dieses Worts ein anderes Wort, welches als Operand verwendet wird. Der Inhalt kann entsprechend der gewünschten Wortadresse im DM- oder EM-Bereich zwischen 0000 und 7FFF (0 bis 32.767) liegen. @D□□□□□ Inhalt → 00000 bis 32767 (0000 bis 7FFF) D	---	---
Wenn der Inhalt von @D□□□□□ zwischen 0000 und 7FFF (00000 bis 32.767) liegt, wird das entsprechende Wort zwischen D00000 und D32767 spezifiziert.	@D00300 0 1 0 0 Dezimal: 256 Spezifiziert D00256. @-Präfix hinzufügen.	MOV #0001 @D00300
Wenn der Inhalt von @D□□□□□ zwischen 8000 und FFFF (32.768 bis 65.535) liegt, wird das entsprechende Wort zwischen E0_00000 und E0_32767 in EM-Bank 0 spezifiziert.	@D00300 8 0 0 1 Dezimal: 32.769 Spezifiziert E0_00001.	---

Beschreibung	Beispiel	Befehlsbeispiel
Wenn der Inhalt von @En□_□□□□□ zwischen 0000 und 7FFF (00000 bis 32.767) liegt, wird das entsprechende Wort zwischen En□_00000 und En□_32767 spezifiziert.	@E1_00200 0 1 0 1 Dezimal: 257 ↓ Spezifiziert E1_00257.	MOV #0001 @E1_00200
Wenn der Inhalt von @En□_□□□□□ zwischen 8000 und FFFF (32.768 bis 65.535) liegt, wird das entsprechende Wort zwischen E (□+1) _00000 und E (□+1) _32767 (in der nächsten EM-Bank) spezifiziert.	@E1_00200 8 0 0 2 Dezimal: 32770 ↓ Spezifiziert E2_00002.	

Hinweis Wenn im SPS-Setup der Binärmodus gewählt ist, werden DM-Bereich und aktuelle EM-Bankadresse (Bank 0 bis C) als aufeinander folgende Speicheradressen behandelt. Ein Wort in EM-Bank 0 wird spezifiziert, wenn ein indirekt adressiertes DM-Wort einen Wert enthält, der größer als 32.767 ist. Beispielsweise wird E00000 in Bank 0 spezifiziert, wenn das indirekt adressierende DM-Wort einen hexadezimalen Wert von 8000 (32.768) enthält.

Ein Wort in der nächsten EM-Bank wird spezifiziert, wenn ein indirekt adressiertes EM-Wort einen Wert enthält, der größer als 32.767 ist. Beispielsweise wird E3_00000 spezifiziert, wenn das indirekt adressierende EM-Wort in Bank 2 einen hexadezimalen Wert von 8000 (32.768) enthält.

Spezifizieren indirekter DM/EM-Adressen im BCD-Modus

Verfahren	Beschreibung	Beispiel	Befehlsbeispiel
Indirekte DM/EM-Adressierung (BCD-Modus)	Wenn das *-Präfix vor der DM- oder EM-Adresse eingegeben wird, spezifiziert der BCD-Inhalt dieses Worts ein anderes Wort, welches als Operand verwendet wird. Der Inhalt kann entsprechend der gewünschten Wortadresse im DM- oder EM-Bereich zwischen 0000 und 9999 liegen. *D□□□□□ ↓ Inhalt 0000 bis 9999 (BCD) ↓ D	*D00200 0 1 0 0 ↓ Spezifiziert D00100. Präfix * hinzufügen.	MOV #0001 *D00200

Adressierung von Indexregistern

Verfahren	Beschreibung		Beispiel	Befehlsbeispiel
Direktes Adressieren von Indexregistern	MOVR(560) überträgt die SPS-Speicheradresse eines Worts oder Bits in ein Indexregister (IR0 bis IR15). (MOVRW(561) überträgt die SPS-Speicheradresse eines Zeitgeber- oder Zähleristwerts in ein Indexregister.)		IR0 IR2	MOVR 0010 IR0 Speichert die SPS-Speicheradresse von CIO 0010 in IR0. MOVR 000102 IR2 Speichert die SPS-Speicheradresse von CIO 000102 in IR2.
Indirektes Adressieren mittels Indexregistern	Grundfunktion (kein Offset)	Das an der E/A-Speicheradresse in IR□ enthaltene Wort oder Bit wird als Operand verwendet. Geben Sie vor dem Indexregister ein Komma ein, um eine indirekte Adressierung anzugeben. (Die Bit-/Wortbezeichnung kann durch den Befehl oder Operanden bestimmt werden.)	,IR0 ,IR1	LD ,IR0 Lädt den Bit-Status an der in IR0 enthaltenen E/A-Speicheradresse. MOV #0001, IR1 Überträgt #0001 in das Wort an der in IR1 enthaltenen E/A-Speicheradresse.
	Konstanter Offset	Der Offset-Wert (–2.048 bis +2.047) wird zu der in IR□ enthaltenen E/A-Speicheradresse hinzuaddiert, und die sich daraus ergebende Adresse wird als Operand verwendet. (Der Offset wird bei Ausführung des Befehls in Binärdaten umgewandelt.)	+5 ,IR0 +31 ,IR1	LD +5 ,IR0 Addiert 5 zu der in IR0 enthaltenen E/A-Speicheradresse und lädt den Status des Bits an dieser Adresse. MOV #0001 +31 ,IR1 Addiert 31 zu der in IR1 enthaltenen E/A-Speicheradresse und überträgt #0001 in das Wort an dieser Adresse.
	Datenregister-Offset	Der vorzeichenbehaftete Binärinhalt des Datenregisters wird zu der in IR□ enthaltenen E/A-Speicheradresse hinzuaddiert, und die sich daraus ergebende Adresse wird als Operand verwendet.	DR0 ,IR0 DR0 ,IR1	LD DR0 ,IR0 Addiert den Inhalt von DR0 zu der in IR0 enthaltenen E/A-Speicheradresse und lädt den Status des Bits an dieser Adresse. MOV #0001 DR0 ,IR1 Addiert den Inhalt von DR0 zu der in IR1 enthaltenen E/A-Speicheradresse und überträgt #0001 in das Wort an dieser Adresse.
	Auto-Inkrement	Nachdem die E/A-Speicheradresse aus IR□ gelesen wurde, wird der Inhalt des Indexregisters um eins oder zwei inkrementiert. Inkrementierung um 1:,R□+ Inkrementierung um 2:,IR□++ **Hinweis** Indexregister werden bei Ausführung des Befehls inkrementiert, auch wenn ein Fehler auftritt und das Fehler-Falg auf EIN gesetzt wird.	,IR0 + + ,IR1 +	LD ,IR0 + + Lädt den Bit-Status der in IR0 enthaltenen E/A-Speicheradresse und inkrementiert das Register dann um zwei. MOV #0001 ,IR1 + Überträgt #0001 in das Wort an der in IR1 enthaltenen E/A-Speicheradresse und inkrementiert das Register dann um eins.
	Auto-Dekrement	Der Inhalt von IR□ wird um ein oder zwei dekrementiert und anschließend wird die E/A-Speicheradresse im Register als Operand verwendet. Dekrementierung um 1:,–IR□ Dekrementierung um 2:,– –IR□ **Hinweis** Indexregister werden bei Ausführung des Befehls dekrementiert, auch wenn ein Fehler auftritt und das Fehler-Flag auf EIN gesetzt wird.	, – – IR0 , – IR1	LD , – – IR0 Dekrementiert den Inhalt von IR0 um 2 und lädt anschließend den Status des Bits an der E/A-Speicheradresse. MOV #0001 , – IR1 Dekrementiert den Inhalt von IR0 um eins und überträgt anschließend #0001 in das Wort an dieser E/A-Speicheradresse.

Hinweis Stellen Sie sicher, dass die Inhalte von Indexregistern eine gültige E/A-Speicheradresse spezifizieren.

Spezifizieren von Konstanten

Verfahren	Anwendbarkeit auf Operanden	Datenformat	Code	Bereich	Beispiel
Konstante (16-Bit-Daten)	Alle Binärdaten und Binärdaten innerhalb eines Bereichs	Binär ohne Vorzeichen	#	#0000 bis #FFFF	---
		Dezimal mit Vorzeichen	±	-32.768 bis +32.767	---
		Dezimal ohne Vorzeichen	&	&0 bis &66.535	---
	Alle BCD-Daten und BCD-Daten innerhalb eines Bereichs	BCD	#	#0000 bis #9999	---
Konstante (32-Bit-Daten)	Alle Binärdaten und Binärdaten innerhalb eines Bereichs	Binär ohne Vorzeichen	#	#0000 0000 bis #FFFF FFFF	---
		Dezimal mit Vorzeichen	+ –	-2.147.483.648 bis +2.147.483.647	---
		Dezimal ohne Vorzeichen	&	&0 bis &4.294.967.295	---
	Alle BCD-Daten und BCD-Daten innerhalb eines Bereichs	BCD	#	#0000 0000 bis #9999 9999	---

Spezifizieren von Textzeichenketten

Verfahren	Beschreibung	Code	Beispiele	Befehlsbeispiel
Textzeichenketten	Text wird im ASCII-Format (1 Byte/Zeichen außer Sonderzeichen) gespeichert, beginnend mit dem niedrigsten Byte des niedrigsten Worts im Bereich. Bei einer ungeraden Zeichenanzahl wird 00 (NULL) im höheren Byte des letzten Worts im Bereich gespeichert. Bei einer geraden Zeichenanzahl wird 0000 (zwei NULLEN) im Wort nach dem letzten Wort im Bereich gespeichert.		"ABCDE" "A" "B" "C" "D" "E" NULL ‖ 41 42 43 44 45 00 "ABCD" "A" "B" "C" "D" NULL NULL ‖ 41 42 43 44 00 00	MOV$ D00100 D00200 D00100 41 42 D00101 43 44 D00102 45 00 ↓ D00200 41 42 D00201 43 44 D00202 45 00

In der folgenden Tabelle sind die Zeichen aufgeführt, die im ASCII-Format ausgedrückt werden können.

Niederwertigstes Bit \ Höchstwertiges Bit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	‘	p				―	タ	ミ		
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			”	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			$	4	D	T	d	t			、	エ	ト	ヤ		
5			%	5	E	U	e	u			・	オ	ナ	ユ		
6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7			’	7	G	W	g	w			ァ	キ	ヌ	ラ		
8			(	8	H	X	h	x			ィ	ク	ネ	リ		
9			)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ェ	コ	ハ	レ		
B			+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C			,	<	L	¥	l	|			ャ	シ	フ	ワ		
D			―	=	M	]	m	}			ュ	ス	ヘ	ン		
E			.	>	N	^	n	~			ョ	セ	ホ	゛		
F			/	?	O	_	o				ッ	ソ	マ	°		

Hinweis Folgende Befehle werden auch ausgeführt, wenn die Eingangsbedingungen auf AUS gesetzt sind. Wenn bei der indirekten Spezifizierung von Speicheradressen mit Hilfe von Auto-Inkrement oder Auto-Dekrement (IR+ oder IR-) in einem Operanden einer dieser Befehle verwendet wird, wird der Wert im Indexregister (IR) unabhängig von der Eingangsbedingung bei jedem Zyklus aktualisiert (Inkrementierung oder Dekrementierung um eins in jedem Zyklus). Dies muss beim Schreiben eines Programms berücksichtigt werden.

Klassifizierung	Befehle
Eingabebefehle	LD, LD NOT, AND, AND NOT, OR, OR NOT, LD TST(350), LD TSTN(351), AND TST(350), AND TSTN(351), OR TST(350), OR TSTN(351)
Ausgabebefehle	OUT, OUT NOT, DIFU(013), DIFD(014)
Sequenz-Steuerbefehle	JMP(004), FOR(512)
Zeitgeber- und Zählerbefehle	TIM/TIMX(550), TIMH(015)/TIMHX(551), TMHH(540)/ TMHHX(552), TTIM(087)/TTIMX(555), TIML(542)/ TIMLX(553), MTIM(533)/MTIMX(554), CNT/CNTX(546), CNTR(012)/CNTRX(548)
Vergleichsbefehle	Symbolvergleichsbefehle (LD, AND, OR =, usw. (Funktionscodes: 300, 305, 310, 320 und 325))
Gleitkomma-Arithmetikbefehle einfacher Genauigkeit	Gleitkommadatenvergleich einfacher Genauigkeit (LD, AND, OR = F, usw.(Funktionscodes: 329 bis 334))
Gleitkommaarithmetikbefehle doppelter Genauigkeit	Gleitkommadatenvergleich doppelter Genauigkeit (LD, AND, OR = D, usw.(Funktionscodes: 335 bis 340))
Blockprogrammierungsbefehle	BPPS(811), BPRS(812), EXIT(806), EXIT(806) NOT, IF(802), IF(802) NOT, WAIT(805), WAIT(805) NOT, TIMW(813)/TIMWX(816), CNTW(814)/CNTWX(818), TMHW(815)/TMHWX(817), LEND(810), LEND(810) NOT
Textzeichenketten-Verarbeitungsbefehle	Zeichenkettenvergleich (LD, AND, OR =$, usw. (Funktionscodes: 670 bis 675))

Die folgenden Beispiele für die Kontaktplanprogrammierung zeigen, wie Indexregister behandelt werden.

Beispiel 1

Kontaktplan:

```
LD P_Off
OUT, IR0+
```

Funktionsweise: Wenn die SPS-Speicheradresse 000013 in IR0 gespeichert ist.

Die Eingangsbedingung ist auf AUS gesetzt (P_Off ist das Immer-AUS-Flag), also setzt der OUT-Befehl die Adresse 000013, die indirekt durch IR0 adressiert ist, auf AUS. Der OUT-Befehl wird ausgeführt, sodass IR0 inkrementiert wird. Dadurch wird die SPS-Speicheradresse 000014, die durch Inkrementieren von IR0 um +1 erhalten wird, gespeichert. Demzufolge setzt der OUT-Befehl im folgenden Zyklus die Adresse 000014 auf AUS.

Beispiel 2

Kontaktplan:

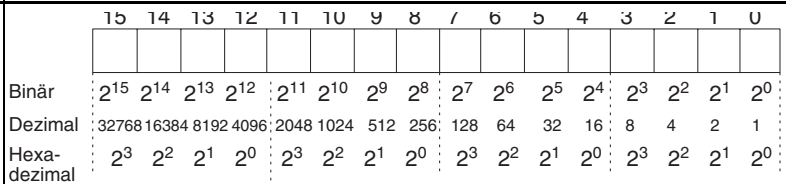
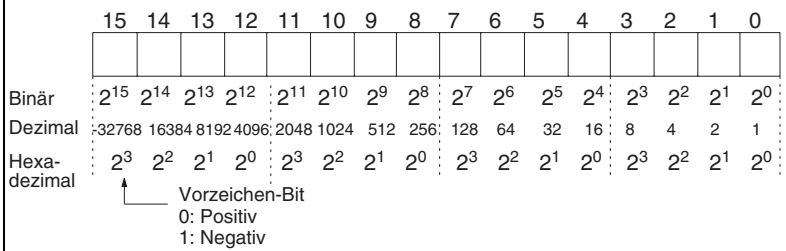
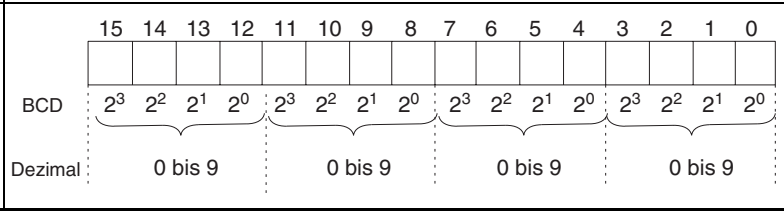
```
LD P_Off
SET, IR0+
```

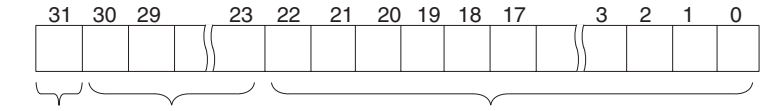
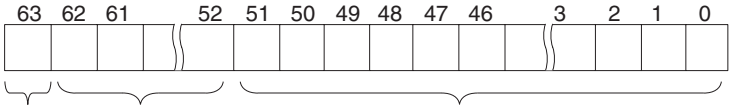
Funktionsweise: Wenn die SPS-Speicheradresse 000013 in IR0 gespeichert ist.

Die Eingangsbedingung ist auf AUS gesetzt (P_Off ist das Immer-AUS-Flag), so dass der SET-Befehl nicht ausgeführt wird. Daher wird IR0 nicht inkrementiert und der in IR0 gespeicherte Wert bleibt die SPS-Speicheradresse 000013.

1-1-6 Datenformate

In der folgenden Tabelle sind die Datenformate aufgeführt, die bei den SPS der CS/CJ-Serie verwendet werden können.

Name	Format	Dezimal-bereich	Hexadezi-malbereich
Binärdaten ohne Vorzeichen		0 bis 65.535	0000 bis FFFF
Binärdaten mit Vorzeichen		-32.768 bis +32.767	8000 bis 7FFF
BCD-Daten		0 bis 9.999	0000 bis 9999

Name	Format	Dezimal-bereich	Hexadezi-malbereich
Gleit-komma-Dezimal-daten	 Vorzeichen der Mantisse: Exponent: Mantisse Binär Wert = $(-1)^{\text{Vorzeichen}} \times 1,[\text{Mantisse}] \times 2^{\text{Exponent}}$ Vorzeichen (Bit 31): negativ oder 0: positiv Mantisse: Die Mantisse besteht aus 23 Bits von Bit 00 bis Bit 22 und enthält den Dezimalbruchteil in 1,□□□..... im Binärformat. Exponent: Der Exponent besteht aus 8 Bits von Bit 23 bis Bit 30 und enthält n plus 127 in 2^n als Binärzahl. **Hinweis** Dieses Format entspricht den IEEE754-Standards für Gleitkommadaten einfacher Genauigkeit und wird nur bei Befehlen verwendet, mit denen Gleitkommadaten konvertiert oder berechnet werden. Es kann zur Einstellung oder Überwachung im SPS-Speicher- und Datenansichts-Fenster im CX-Programmer verwendet werden (wird von den Programmierkonsolen nicht unterstützt). In dieser Eigenschaft müssen Anwender dieses Format nicht kennen, doch sie müssen wissen, dass die Formatierung zwei Worte belegt.	---	---
Gleit-komma-Dezimal-daten dop-pelter Genauig-keit	 Vorzeichen der Mantisse: Exponent: Mantisse Binär Wert = $(-1)^{\text{Vorzeichen}} \times 1,[\text{Mantisse}] \times 2^{\text{Exponent}}$ Vorzeichen (Bit 63) 1: negativ oder 0: positiv Mantisse: Die 52 Bits von Bit 00 bis Bit 51 enthalten die Mantisse, d. h., den Dezimalbruchteil in 1,□□□.... im Binärformat. Exponent: Die 11 Bits von Bit 52 bis 62 enthalten den Exponenten. Der Exponent wird im Binärformat als 1023 plus n in 2^n ausgedrückt. **Hinweis** Dieses Format entspricht den IEEE754-Standards für Gleitkommadaten doppelter Genauigkeit und wird nur bei Befehlen verwendet, mit denen Gleitkommadaten konvertiert oder berechnet werden. Es kann zur Einstellung oder Überwachung im SPS-Speicher- und Datenansichts-Fenster im CX-Programmer verwendet werden (wird von den Programmierkonsolen nicht unterstützt). In dieser Eigenschaft müssen Anwender dieses Format nicht kennen, doch sie müssen wissen, dass die Formatierung vier Worte belegt.	---	---

Binärzahlen mit Vorzeichen

Binärzahlen mit negativem Vorzeichen werden als Zweierkomplement des Hexadezimal-Absolutwerts ausgedrückt. Bei einem Dezimalwert von –12.345 entspricht der Absolutwert 3039 hexadezimal. Das Zweierkomplement ist 10000 – 3039 (beides hexadezimal) oder CFC7.

Um eine Binärzahl mit negativem Vorzeichen (CFC7) in eine Dezimalzahl zu konvertieren, wird das Zweierkomplement dieser Zahl (10000 – CFC7 = 3039) gebildet, dieses in eine Dezimalzahl (3039 hexadezimal = 12.345 dezimal) konvertiert und ein Minuszeichen (–12.345) hinzugefügt.

1-2 Befehlsausführungsprüfungen

1-2-1 Fehler bei der Befehlsausführung

Befehlsoperanden und Anordnung werden geprüft, wenn ein Befehl von einem Peripheriegerät eingegeben oder eine Programmprüfung von einem Peripheriegerät (keine Programmierkonsole) ausgeführt wird. Dies sind aber keine abschließende Prüfungen. Folgende vier Fehler können bei Ausführung eines Befehls auftreten.

Befehlsverarbeitungsfehler (ER-Flag EIN)

Normalerweise sind Befehlsverarbeitungsfehler keine schwerwiegenden Fehler, aber das SPS-Setup kann so eingestellt werden, dass Befehlsverarbeitungsfehler als schwerwiegende Fehler behandelt werden. Wenn diese Einstellung erfolgt ist, wird das Befehlsverarbeitungs-Fehlerflag (A29508) bei Auftreten eines Befehlsverarbeitungsfehlers auf EIN gesetzt und die Programmausführung wird gestoppt.

Zugriffsfehler (AER-Flag EIN)

Normalerweise sind Zugriffsfehler keine schwerwiegenden Fehler, aber das SPS-Setup kann so eingestellt werden, dass diese Fehler als schwerwiegende Fehler gelten. Wenn diese Einstellung erfolgt ist, werden Zugriffsfehler-Bit (A29510) und BCD-Fehler-bei-indirekter-DM/EM-Adressierung-Bit bei Auftreten eines Zugriffsfehlers auf EIN gesetzt und die Programmausführung wird gestoppt.

Unzulässiger-Befehl-Fehler

Das Unzulässiger-Befehl-Fehlerflag (A29514) wird bei Auftreten dieses Fehlers auf EIN gesetzt und die Programmausführung wird gestoppt.

UM-Überlauffehler (Anwenderprogrammspeicher)

Das Anwenderprogrammspeicherüberlauf-Fehlerflag (A29514) wird bei Auftreten dieses Fehlers auf EIN gesetzt und die Programmausführung wird gestoppt.

1-2-2 Schwerwiegende Fehler (Programmfehler)

Die Programmausführung wird bei Auftreten eines der folgenden Programmfehler gestoppt. Wenn ein Programmfehler auftritt, wird die Tasknummer des beim Stoppen der Programmausführung ausgeführten Tasks in A294 geschrieben und die Programmadresse in A298 und A299 geschrieben.

Verwenden Sie den Inhalt dieser Worte zum Lokalisieren des Programmfehlers und korrigieren Sie ihn ggf.

Adresse	Beschreibung
A294	Die Task-Nummer der aktuellen Tasks wird in dieses Wort geschrieben, wenn die Programmausführung aufgrund eines Programmfehlers gestoppt wird. Zyklische Tasks tragen die Nummern 0000 bis 001F (Zyklische Tasks 0 bis 31). Interrupt-Tasks tragen die Task-Nummern 8000 bis 80FF (Interrupt-Tasks 0 bis 255).
A298 und A299	Die aktuelle Programmadresse wird in diese Worte geschrieben, wenn die Programmausführung aufgrund eines Programmfehlers gestoppt wird. A299 enthält die linken (höchstwertigen) Stellen der Programmadresse und A298 enthält die rechten (niederwertigsten) Stellen der Programmadresse.

Alle Fehler, für die das Fehlerflag oder Zugriffsfehlerflag auf EIN gesetzt wird, werden als Programmfehler behandelt. In der folgenden Tabelle sind die Programmfehler aufgelistet. Das SPS-Setup kann so eingestellt werden, dass die Programmausführung bei Auftreten dieser Fehler gestoppt wird.

Fehlertyp	Beschreibung	Entsprechende Flags
Kein END-Befehl	Es gibt keinen END(001) Befehl im Programm.	END-fehlt-Fehlerflag (A29511)
Task-Fehler	Es gibt drei mögliche Ursachen für einen Task-Fehler: 1) Es gibt keinen ausführbereiten zyklischen Task. 2) Dem Task ist kein Programm zugeordnet. 3) Es wurde ein Interrupt generiert, aber der entsprechende Interrupt-Task ist nicht vorhanden.	Task-Fehlerflag (A29512)
Befehlsverarbeitungs-fehler*	Die CPU hat versucht, einen Befehl auszuführen, aber die im Befehlsoperanden enthaltenen Daten waren ungültig. *Wenn das SPS-Setup so eingestellt wurde, dass Befehlsfehler als schwerwiegende Fehler (Programmfehler) betrachtet werden, wird das Befehlsausführungs-Fehlerflag (A29508) auf EIN gesetzt und die Programmausführung gestoppt.	Fehler-Flag (ER), Befehlsausführungs-Fehlerflag (A29508)
Zugriffsfehler*	Es gibt fünf mögliche Ursachen für einen Zugriffsfehler: 1) Lesen/Schreiben im Parameterbereich. 2) Schreiben in nicht vorhandenen Speicher. 3) Lesen/Beschreiben in einer in EM-Dateispeicher umgewandelten EM-Bank. 4) Schreiben in einem schreibgeschützten Bereich. 5) Der Inhalt von einem DM/EM-Wort war nicht in BCD, obwohl die SPS für indirekte BCD-Adressierung eingerichtet ist. *Wenn das SPS-Setup so eingestellt wurde, dass Befehlsfehler als schwerwiegende Fehler (Programmfehler) betrachtet werden, wird das Unzulässiger-Zugriff-Fehlerflag (A29510) auf EIN gesetzt und die Programmausführung gestoppt.	Zugriffsfehler-Flag (AER) Unzulässiger-Zugriff-Fehlerflag (A29510)
BCD-Fehler-bei-indirekter-DM/EM-Adressierung-Bit*	Der Inhalt von einem DM/EM-Wort war nicht in BCD, obwohl die SPS für indirekte BCD-Adressierung eingerichtet ist. *Wenn das SPS-Setup so eingestellt wurde, dass Befehlsfehler als schwerwiegende Fehler (Programmfehler) betrachtet werden, wird das BCD-Fehler-bei-indirekter-DM/EM-Adressierung-Bit (A29509) auf EIN gesetzt und die Programmausführung gestoppt.	Zugriffsfehler-Flag (AER) BCD-Fehler-bei-indirekter-DM/EM-Adressierung-Bit (A29509)
Flankenauswertungs-Überlauffehler	Befehle mit Flankenauswertung wurden während der Online-Bearbeitung (über 31.072 mal) wiederholt eingefügt und gelöscht.	Flankenauswertungs-Überlauffehlerflag (A29513)
Anwenderprogramm-speicher-Überlauffehler	Die letzte Adresse im Anwenderprogrammspeicher (UM) wurde überschritten.	Anwenderprogrammspeicher-Überlauffehlerflag (A29515)
Unzulässiger-Befehl-Fehler	Das Programm enthält einen nicht ausführbaren Befehl.	Unzulässiger-Befehl-Fehlerflag (A29514)

KAPITEL 2
Übersicht über die Befehle

Dieser Abschnitt enthält eine Übersicht der für SPS der CS/CJ-Serie verwendeten Befehle.

2-1 Klassifizierung der Befehle nach Funktion 16
2-2 Befehlsfunktionen 24
2-2-1 Eingabebefehle 24
2-2-2 Ausgabebefehle 26
2-2-3 Sequenz-Steuerbefehle 29
2-2-4 Zeitgeber- und Zählerbefehle 33
2-2-5 Vergleichsbefehle 37
2-2-6 Datenübertragungsbefehle 41
2-2-7 Datenverschiebungsbefehle 44
2-2-8 Inkrementier-/Dekrementierbefehle 48
2-2-9 Arithmetische Befehle 49
2-2-10 Konvertierungsbefehle 54
2-2-11 Logikbefehle 60
2-2-12 Spezielle arithmetische Befehle 62
2-2-13 Gleitkomma-Arithmetikbefehle 63
2-2-14 Gleitkomma-Arithmetikbefehle doppelter Genauigkeit 67
2-2-15 Tabellendatenverarbeitungsbefehle 71
2-2-16 Regelungsbefehle 75
2-2-17 Unterprogrammbefehle 79
2-2-18 Interrupt-Steuerbefehle 80
2-2-19 Befehle für schnelle Zähler und Impulsausgabe (nur CJ1M-CPU21/22/23) 82
2-2-20 Schrittbefehle 84
2-2-21 E/A-Baugruppenbefehle 84
2-2-22 Befehle für die serielle Kommunikation 87
2-2-23 Netzwerkbefehle 88
2-2-24 Dateispeicherbefehle 91
2-2-25 Anzeigebefehle 92
2-2-26 Uhrfunktionsbefehle 92
2-2-27 Fehlersuch-Befehle 93
2-2-28 Fehlerdiagnosebefehle 94
2-2-29 Sonstige Befehle 95
2-2-30 Blockprogrammierbefehle 96
2-2-31 Textverarbeitungsbefehle 102
2-2-32 Task-Steuerbefehle 105
2-2-33 Befehle zur Modellkonvertierung (nur CPU-Baugruppen ab Ver. 3.0) 106
2-2-34 Spezielle Funktionsblockbefehle 107
2-3 Alphabetische Liste der Befehle nach AWL-Kürzel 108
2-4 Liste der Befehl nach Funktionscode 125

2-1 Klassifizierung der Befehle nach Funktion

In der folgenden Tabelle sind die Befehle der CS/CJ-Serie nach ihrer Funktion aufgelistet. (In *Kapitel 3, Befehle,* sind die Befehle nach ihrer Funktion klassifiziert aufgeführt.)

*Befehle oder Befehlsgruppen, die mit einem einzelnen Sternchen markiert sind, werden nur von den CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

**Befehle oder Befehlsgruppen, die mit zwei Sternchen markiert sind, werden nur von CJ1M CPU-Baugruppen unterstützt.

***Befehle oder Befehlsgruppen, die mit drei Sternchen markiert sind, werden nicht von CS1D CPU-Baugruppen für Duplex-CPU-Systeme unterstützt.

Hinweis

1. Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0
2. Nur CJ1M-CPU21/22/23 CPU-Baugruppen ab Ver. 2.0
3. Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0
   CJ1M CPU-Baugruppe (vor Version 2.0 oder Baugruppenversion ab 2.0)

Klassifizierung	Unterklasse	AWL	Befehl	AWL	Befehl	AWL	Befehl
Basisbefehle	**Eingang**	LD	LOAD	LD NOT	LOAD NOT	AND	AND
		AND NOT	AND NOT	OR	OR	OR NOT	OR NOT
		AND LD	AND LOAD	OR LD	OR LOAD	---	---
	Ausgang	OUT	OUTPUT	OUT NOT	OUTPUT NOT	---	---
Sequenz-Eingabebefehle	**---**	NOT	NOT	UP	CONDITION ON	DOWN	CONDITION OFF
	Bit-Test	LD TST	LD BIT TEST	LD TSTN	LD BIT TEST NOT	AND TST	AND BIT TEST NOT
		AND TSTN	AND BIT TEST NOT	OR TST	OR BIT TEST	OR TSTN	OR BIT TEST NOT
Sequenz-Ausgabebefehle	**---**	KEEP	KEEP	DIFU	DIFFERENTIATE UP	DIFD	DIFFERENTIATE DOWN
		OUTB*	SINGLE BIT OUTPUT	---	---	---	---
	Setzen/Zurücksetzen	SET	SET	RSET	RESET	SETA	MULTIPLE BIT SET
		RSTA	MULTIPLE BIT RESET	SETB*	SINGLE BIT SET	RSTB*	SINGLE BIT RESET
Sequenz-Steuerbefehle	**---**	END	END	NOP	NO OPERATION	---	---
	Verriegelung	IL	INTERLOCK	ILC	INTERLOCK CLEAR	MILH	MULTI-INTERLOCK DIFFERENTIATION HOLD
		MILR (siehe Hinweis 1)	MULTI-INTERLOCK DIFFERENTIATION RELEASE	MILC (siehe Hinweis 1)	MULTI-INTERLOCK CLEAR	---	---
	Sprung	JMP	JUMP	JME	JUMP END	CJP	CONDITIONAL JUMP
		CJPN	CONDITIONAL JUMP	JMP0	MULTIPLE JUMP	JME0	MULTIPLE JUMP END
	Repeat	FOR	FOR-NEXT LOOPS	BREAK	BREAK LOOP	NEXT	FOR-NEXT LOOPS

Klassifizierung	Unterklasse		AWL	Befehl	AWL	Befehl	AWL	Befehl
Zeitgeber- und Zählerbefehle	**BCD**	**Zeitgeber (mit Zeitgebernummern)**	TIM	TIMER	TIMH	HIGH-SPEED TIMER	TMHH	ONE-MS TIMER
			TTIM	ACCUMULATIVE TIMER	---	---	---	---
		Zeitgeber (ohne Zeitgebernummern)	TIML	LONG TIMER	MTIM	MULTI-OUTPUT TIMER	---	---
		Zähler (mit Zählernummern)	CNT	COUNTER	CNTR	REVERSIBLE TIMER	CNR	RESET TIMER/ COUNTER
	Binär*	**Zeitgeber (mit Zeitgebernummern)**	TIMX	TIMER	TIMHX	HIGH-SPEED TIMER	TMHHX	ONE-MS TIMER
			TTIMX	ACCUMULATIVE TIMER	---	---	---	---
		Zeitgeber (ohne Zeitgebernummern)	TIMLX	LONG TIMER	MTIMX	MULTI-OUTPUT TIMER	---	---
		Zähler (mit Zählernummern)	CNTX	COUNTER	CNTRX	REVERSIBLE TIMER	CNRX	RESET TIMER/ COUNTER
Vergleichsbefehle	**Symbolvergleich**		LD, AND, OR + =, <>, <, <=, >, >=	Symbolvergleich (ohne Vorzeichen)	LD, AND, OR + =, <>, <, <=, >, >= + L	Symbolvergleich (Doppelwort, ohne Vorzeichen)	LD, AND, OR + =, <>, <, <=, >, >= +S	Symbolvergleich (mit Vorzeichen)
			LD, AND, OR + =, <>, <, <=, >, >= + SL	Symbolvergleich (Doppelwort, mit Vorzeichen)	LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT (siehe Hinweis 1)	Zeitvergleich	---	---
	Datenvergleich (Bedingungs-Flags)		CMP	UNSIGNED COMPARE	CMPL	DOUBLE UNSIGNED COMPARE	CPS	SIGNED BINARY COMPARE
			CPSL	DOUBLE SIGNED BINARY COMPARE	ZCP*	AREA RANGE COMPARE	ZCPL*	DOUBLE AREA RANGE COMPARE
	Tabellenvergleich		MCMP	MULTIPLE COMPARE	TCMP	TABLE COMPARE	BCMP	UNSIGNED BLOCK COMPARE
			BCMP2 (siehe Hinweis 3)	EXPANDED BLOCK COMPARE	---	---	---	---
Kopierbefehle	**Einzel-/ Doppelwort**		MOV	MOVE	MOVL	DOUBLE MOVE	MVN	MOVE NOT
			MVNL	DOUBLE MOVE NOT	---	---	---	---
	Bit/Stelle		MOVB	MOVE BIT	MOVD	MOVE DIGIT	---	---
	Austausch		XCHG	DATA EXCHANGE	XCGL	DOUBLE DATA EXCHANGE	---	---
	Block-/Bitübertragung		XFRB	MULTIPLE BIT TRANSFER	XFER	BLOCK TRANSFER	BSET	BLOCK SET
	Verteilen/ Sammeln		DIST	SINGLE WORD DISTRIBUTE	COLL	DATA COLLECT	---	---
	Indexregister		MOVR	MOVE TO REGISTER	MOVRW	MOVE TIMER/ COUNTER PV TO REGISTER	---	---

Klassifizierung	Unterklasse	AWL	Befehl	AWL	Befehl	AWL	Befehl
Datenverschiebebefehle	**1-Bit-Verschiebung**	SFT	SHIFT REGISTER	SFTR	REVERSIBLE SHIFT REGISTER	ASLL	DOUBLE SHIFT LEFT
		ASL	ARITHMETIC SHIFT LEFT	ASR	ARITHMETIC SHIFT RIGHT	ASRL	DOUBLE SHIFT RIGHT
	0000 hex asynchron	ASFT	ASYNCHRONOUS SHIFT REGISTER	---	---	---	---
	Wortweises Verschieben	WSFT	WORD SHIFT	---	---	---	---
	1-Bit-Rotation	ROL	ROTATE LEFT	ROLL	DOUBLE ROTATE LEFT	RLNC	ROTATE LEFT WITHOUT CARRY
		RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	ROR	ROTATE RIGHT	RORL	DOUBLE ROTATE RIGHT
		RRNC	ROTATE RIGHT WITHOUT CARRY	RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	---	---
	1-Stelle-Verschiebung	SLD	ONE DIGIT SHIFT LEFT	SRD	ONE DIGIT SHIFT RIGHT	---	---
	n-Bit-Daten verschieben	NSFL	SHIFT N-BIT DATA LEFT	NSFR	SHIFT N-BIT DATA RIGHT	---	---
	n-Bit-Verschiebung	NASL	SHIFT N-BITS LEFT	NSLL	DOUBLE SHIFT N-BITS LEFT	NASR	SHIFT N-BITS RIGHT
		NSRL	DOUBLE SHIFT N-BITS RIGHT	---	---	---	---
Inkrementier-/Dekrementierbefehle	**BCD**	++B	INCREMENT BCD	++BL	DOUBLE INCREMENT BCD	– –B	DECREMENT BCD
		– –BL	DOUBLE DECREMENT BCD	---	---	---	---
	Binär	++	INCREMENT BINARY	++L	DOUBLE INCREMENT BINARY	– –	DECREMENT BINARY
		– –L	DOUBLE DECREMENT BINARY	---	---	---	---

Klassifizierung	Unterklasse	AWL	Befehl	AWL	Befehl	AWL	Befehl
Symbolarithmetische Befehle	**Binäre Addition**	+	SIGNED BINARY ADD WITHOUT CARRY	+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+C	SIGNED BINARY ADD WITH CARRY
		+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	---	---	---	---
	BCD-Addition	+B	BCD ADD WITHOUT CARRY	+BL	DOUBLE BCD ADD WITHOUT CARRY	+BC	BCD ADD WITH CARRY
		+BCL	DOUBLE BCD ADD WITH CARRY	---	---	---	---
	Binär-Subtraktion	–	SIGNED BINARY SUBTRACT WITHOUT CARRY	–L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	–C	SIGNED BINARY SUBTRACT WITH CARRY
		–CL	DOUBLE SIGNED BINARY WITH CARRY	---	---	---	---
	BCD-Subtraktion	–B	BCD SUBTRACT WITHOUT CARRY	–BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	–BC	BCD SUBTRACT WITH CARRY
		–BCL	DOUBLE BCD SUBTRACT WITH CARRY	---	---	---	---
	Binär-Multiplikation	*	SIGNED BINARY MULTIPLY	*L	DOUBLE SIGNED BINARY MULTIPLY	*U	UNSIGNED BINARY MULTIPLY
		*UL	DOUBLE UNSIGNED BINARY MULTIPLY	---	---	---	---
	BCD-Multiplikation	*B	BCD MULTIPLY	*BL	DOUBLE BCD MULTIPLY	---	---
	Binär-Division	/	SIGNED BINARY DIVIDE	/L	DOUBLE SIGNED BINARY DIVIDE	/U	UNSIGNED BINARY DIVIDE
		/UL	DOUBLE UNSIGNED BINARY DIVIDE	---	---	---	---
	BCD-Division	/B	BCD DIVIDE	/BL	DOUBLE BCD DIVIDE	---	---

Klassifizierung	Unterklasse	AWL	Befehl	AWL	Befehl	AWL	Befehl
Konvertierungsbefehle	**BCD/Binärkonvertierung**	BIN	BCD-TO-BINARY	BINL	DOUBLE BCD-TO-DOUBLE BINARY	BCD	BINARY-TO-BCD
		BCDL	DOUBLE BINARY-TO-DOUBLE BCD	NEG	2'S COMPLEMENT	NEGL	DOUBLE 2'S COMPLEMENT
		SIGN	16-BIT TO 32-BIT SIGNED BINARY	---	---	---	---
	Dekoder/ Kodierer	MLPX	DATA DECODER	DMPX	DATA ENCODER	---	---
	ASCII/HEX-Konvertierung	ASC	ASCII CONVERT	HEX	ASCII TO HEX	---	---
	Zeilen-/Spalten-Konvertierung	LINE	COLUMN TO LINE	COLM	LINE TO COLUMN	---	---
	Binär mit Vorzeichen/BCD-Konvertierung	BINS	SIGNED BCD-TO-BINARY	BISL	DOUBLE SIGNED BCD-TO-BINARY	BCDS	SIGNED BINARY-TO-BCD
		BDSL	DOUBLE SIGNED BINARY-TO-BCD	GRY (siehe Hinweis 1)	GRAY CODE CONVERSION	---	---
Logikbefehle	**Logisches AND/OR**	ANDW	LOGICAL AND	ANDL	DOUBLE LOGICAL AND	ORW	LOGICAL OR
		ORWL	DOUBLE LOGICAL OR	XORW	EXCLUSIVE OR	XORL	DOUBLE EXCLUSIVE OR
		XNRW	EXCLUSIVE NOR	XNRL	DOUBLE EXCLUSIVE NOR	---	---
	Komplement	COM	COMPLEMENT	COML	DOUBLE COMPLEMENT	---	---
Spezielle arithmetische Befehle	**---**	ROTB	BINARY ROOT	ROOT	BCD SQUARE ROOT	APR	ARITHMETIC PROCESS
		FDIV	FLOATING POINT DIVIDE	BCNT	BIT COUNTER	---	---
Gleitkomma-Arithmetikbefehle	**Gleitkomma/Binär-Konvertierung**	FIX	FLOATING TO 16-BIT	FIXL	FLOATING TO 32-BIT	FLT	16-BIT TO FLOATING
		FLTL	32-BIT TO FLOATING	---	---	---	---
	Gleitkomma-Grundarithmetik	+F	FLOATING-POINT ADD	–F	FLOATING-POINT SUBTRACT	/F	FLOATING-POINT DIVIDE
		*F	FLOATING-POINT MULTIPLY	---	---	---	---
	Gleitkomma-Trigonometrie	RAD	DEGREES TO RADIANS	DEG	RADIANS TO DEGREES	SIN	SINE
		COS	COSINE	TAN	TANGENT	ASIN	ARC SINE
		ACOS	ARC COSINE	ATAN	ARC TANGENT	---	---
	Gleitkommaarithmetik	SQRT	SQUARE ROOT	EXP	EXPONENT	LOG	LOGARITHM
		PWR	EXPONENTIAL POWER	---	---	---	---
	Symbolvergleich und Konvertierung*	LD, AND, OR + =, <>, <, <=, >, >= + F	Symbolvergleich (Gleitkommadaten einfacher Genauigkeit)	FSTR*	FLOATING-POINT TO ASCII	FVAL*	ASCII TO FLOATING-POINT

Klassifizierung	Unterklasse	AWL	Befehl	AWL	Befehl	AWL	Befehl
Gleitkommaarithmetikbefehle doppelter Genauigkeit*	**Gleitkomma/Binär-Konvertierung**	FIXD	DOUBLE FLOATING TO 16-BIT	FIXLD	DOUBLE FLOATING TO 32-BIT	DBL	16-BIT TO DOUBLE FLOATING
		DBLL	32-BIT TO DOUBLE FLOATING	---	---	---	---
	Gleitkomma-Grundarithmetik	+D	DOUBLE FLOATING-POINT ADD	−D	DOUBLE FLOATING-POINT SUBTRACT	/D	DOUBLE FLOATING-POINT DIVIDE
		⁎D	DOUBLE FLOATING-POINT MULTIPLY	---	---	---	---
	Gleitkomma-Trigonometrie	RADD	DOUBLE DEGREES TO RADIANS	DEGD	DOUBLE RADIANS TO DEGREES	SIND	DOUBLE SINE
		COSD	DOUBLE COSINE	TAND	DOUBLE TANGENT	ASIND	DOUBLE ARC SINE
		ACOSD	DOUBLE ARC COSINE	ATAND	DOUBLE ARC TANGENT	---	---
	Gleitkommaarithmetik	SQRTD	DOUBLE SQUARE ROOT	EXPD	DOUBLE EXPONENT	LOGD	DOUBLE LOGARITHM
		PWRD	DOUBLE EXPONENTIAL POWER	---	---	---	---
	Symbolvergleich	LD, AND, OR + =, <>, <, <=, >, >= + D	Symbolvergleich (Gleitkomma doppelter Genauigkeit)	---	---	---	---
Tabellendaten-Verarbeitungsbefehle	**Stapelverarbeitung**	SSET	SET STACK	PUSH	PUSH ONTO STACK	LIFO	LAST IN FIRST OUT
		FIFO	FIRST IN FIRST OUT	SNUM*	STACK SIZE READ	SREAD*	STACK DATA READ
		SWRIT*	STACK DATA OVERWRITE	SINS*	STACK DATA INSERT	SDEL*	STACK DATA DELETE
	Verarbeitung von 1 Datensatz/ mehrere Worten	DIM	DIMENSION RECORD TABLE	SETR	SET RECORD LOCATION	GETR	GET RECORD NUMBER
	Datensatz-zu-Wort-Verarbeitung	SRCH	DATA SEARCH	MAX	FIND MAXIMUM	MIN	FIND MINIMUM
		SUM	SUM	FCS	FRAME CHECKSUM	---	---
	Byte-Verarbeitung	SWAP	SWAP BYTES	---	---	---	---
Datensteuerungsbefehle	**---**	PID	PID CONTROL	PIDAT*	PID CONTROL WITH AUTOTUNING	LMT	LIMIT CONTROL
		BAND	DEAD BAND CONTROL	ZONE	DEAD ZONE CONTROL	TPO (siehe Hinweis 1)	TIME-PROPORTIONAL OUTPUT
		SCL	SCALING	SCL2	SCALING 2	SCL3	SCALING 3
		AVG	AVERAGE	---	---	---	---
Unterprogrammbefehle	**---**	SBS	SUBROUTINE CALL	MCRO	MACRO	SBN	SUBROUTINE ENTRY
		RET	SUBROUTINE RETURN	GSBS*	GLOBAL SUBROUTINE CALL	GSBN*	GLOBAL SUBROUTINE ENTRY
		GRET*	GLOBAL SUBROUTINE RETURN	---	---	---	---

Klassifizierung	Unterklasse	AWL	Befehl	AWL	Befehl	AWL	Befehl
Interrupt-Steuerungsbefehle	---	MSKS***	SET INTERRUPT MASK	MSKR***	READ INTERRUPT MASK	CLI***	CLEAR INTERRUPT
		DI	DISABLE INTERRUPTS	EI	ENABLE INTERRUPTS	---	---
Befehle für schnelle Zähler/ Impulsausgabe**	---	INI	MODE CONTROL	PRV	HIGH-SPEED COUNTER PV READ	PRV2 (siehe Hinweis 2)	COUNTER FREQUENCY CONVERT
		CTBL	COMPARISON TABLE LOAD	SPED	SPEED OUTPUT	PULS	SET PULSES
		PLS2	PULSE OUTPUT	ACC	ACCELERATION CONTROL	ORG	ORIGIN SEARCH
Step-Befehle	---	PWM	PULSE WITH VARIABLE DUTY FACTOR	STEP	STEP DEFINE	SNXT	STEP START
Befehle für E/A-Baugruppen	---	IORF	I/O REFRESH	SDEC	7-SEGMENT DECODER	DSW (siehe Hinweis 1)	DIGITAL SWITCH INPUT
		TKY (siehe Hinweis 1)	TEN KEY INPUT	HKY (siehe Hinweis 1)	HEXADECIMAL KEY INPUT	MTR (siehe Hinweis 1)	MATRIX INPUT
		7SEG (siehe Hinweis 1)	7-SEGMENT DISPLAY OUTPUT	IORD	INTELLIGENT I/O READ	IOWR	INTELLIGENT I/O WRITE
		DLNK*	CPU BUS UNIT I/O REFRESH	---	---	---	---
Befehle für serielle Kommunikation	---	PMCR	PROTOCOL MACRO	TXD	TRANSMIT	RXD	RECEIVE
		STUP	CHANGE SERIAL PORT SETUP	---	---	---	---
Netzwerkbefehle	---	SEND	NETWORK SEND	RECV	NETWORK RECEIVE	CMND	DELIVER COMMAND
		EXPLT (siehe Hinweis 1)	SEND GENERAL EXPICIT	EGATR (siehe Hinweis 1)	EXPLICIT GET ATTRIBUTE	ESATR (siehe Hinweis 1)	EXPLICIT SET ATTRIBUTE
		ECHRD (siehe Hinweis 1)	EXPLICIT WORD READ	ECHWR (siehe Hinweis 1)	EXPLICIT WORD WRITE	---	---
Anzeigebefehle	---	MSG	DISPLAY MESSAGE	---	---	---	---
Dateispeicherbefehle	---	FREAD	READ DATA FILE	FWRIT	WRITE DATA FILE	---	---
Uhrfunktionsbefehle	---	CADD	CALENDAR ADD	CSUB	CALENDAR SUBTRACT	SEC	HOURS TO SECONDS
		HMS	SECONDS TO HOURS	DATE	CLOCK ADJUSTMENT	---	---
Fehlerbeseitigungs-Befehle	---	TRSM	TRACE MEMORY SAMPLING	---	---	---	---
Fehlerdiagnosebefehle	---	FAL	FAILURE ALARM	FALS	SEVERE FAILURE ALARM	FPD	FAILURE POINT DETECTION

Klassifizierung	Unterklasse		AWL	Befehl	AWL	Befehl	AWL	Befehl
Sonstige Befehle	**---**		STC	SET CARRY	CLC	CLEAR CARRY	EMBC	SELECT EM BANK
			WDT	EXTEND MAXIMUM CYCLE TIME	CCS*	SAVE CONDITION FLAGS	CCL*	LOAD CONDITION FLAGS
			FRMCV*	CONVERT ADDRESS FROM CV	TOCV*	CONVERT ADDRESS TO CV	IOSP***	DISABLE PERIPHERAL SERVICING
			IORS***	ENABLE PERIPHERAL SERVICING	---	---	---	---
Blockprogrammierungsbefehle	**Blockprogrammierbereich definieren**		BPRG	BLOCK PROGRAM BEGIN	BEND	BLOCK PROGRAM END	---	---
	Blockprogramm starten/stoppen		BPPS	BLOCK PROGRAM PAUSE	BPRS	BLOCK PROGRAM RESTART	---	---
	EXIT		EXIT *Bit-Adresse*	Bedingter END-Befehl	EXIT NOT *Bit-Adresse*	Bedingter END NOT-Befehl	Eingangsbedingung EXIT	Bedingter END-Befehl
	Verarbeitung einer IF-Verzweigung		IF *Bit-Adresse*	CONDITIONAL BLOCK BRANCHING	IF NOT *Bit-Adresse*	CONDITIONAL BLOCK BRANCHING (NOT)	ELSE	CONDITIONAL BLOCK BRANCHING (ELSE)
			IEND	CONDITIONAL BLOCK BRANCHING END	---	---	---	---
	WAIT		WAIT *Bit-Adresse*	ONE CYCLE AND WAIT	WAIT NOT *Bit-Adresse*	ONE CYCLE AND WAIT NOT	*Eingangsbedingung* WAIT	ONE CYCLE AND WAIT
	Zeitgeber/Zähler	**BCD**	TIMW	TIMER WAIT	CNTW	COUNTER WAIT	TMHW	HIGH-SPEED TIMER WAIT
		Binär*	TIMWX	TIMER WAIT	CNTWX	COUNTER WAIT	TMHWX	HIGH-SPEED TIMER WAIT
	Wiederholung		LOOP	LOOP BLOCK	LEND *Bit-Adresse*	LOOP BLOCK END	LEND NOT *Bit-Adresse*	LOOP BLOCK END NOT
			Eingangsbedingung LEND	LOOP BLOCK END	---	---	---	---
Textzeichenketten-Verarbeitungsbefehle	**---**		MOV$	MOV STRING	+$	CONCATENATE STRING	LEFT$	GET STRING LEFT
			RIGHT$	GET STRING RIGHT	MID$	GET STRING MIDDLE	FIND$	FIND IN STRING
			LEN$	STRING LENGTH	RPLC$	REPLACE IN STRING	DEL$	DELETE STRING
			XCHG$	EXCHANGE STRING	CLR$	CLEAR STRING	INS$	INSERT INTO STRING
			LD, AND, OR + =$, <>$, <$, <=$, >$, >=$	STRING COMPARISON	---	---	---	---
Task-Steuerbefehle	**---**		TKON	TASK ON	TKOF	TASK OFF	---	---

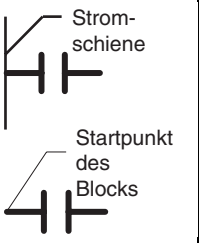
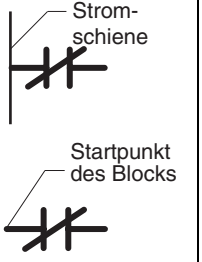
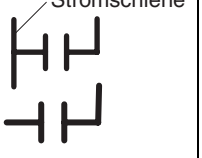
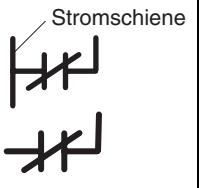
2-2 Befehlsfunktionen

2-2-1 Eingabebefehle

*1: Nicht unterstützt von CS1D CPU-Baugruppen für Duplex-CPU-Systeme.

*2: Nur unterstützt durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D.

*3: Nur unterstützt durch CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M.

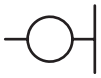
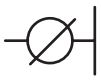
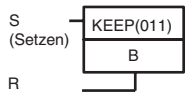
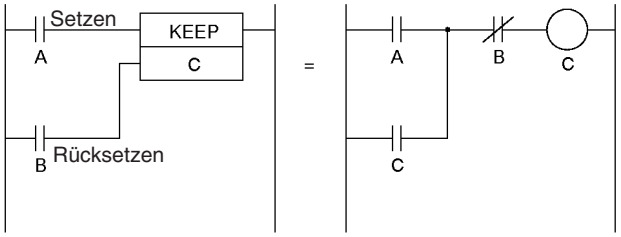
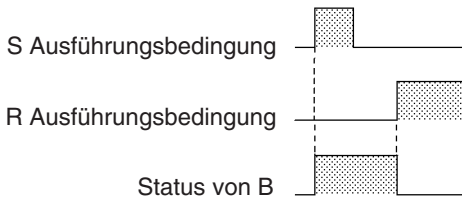
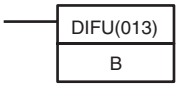
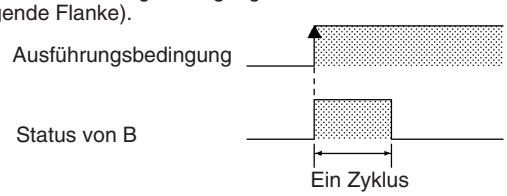
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungsbedingung	Seite
LOAD (Laden) LD @LD %LD !LD*1 !@LD*1 !%LD*1		Bildet den Anfang der logischen Verknüpfung und erzeugt eine EIN/AUS-Ausführungsbedingung auf der Grundlage des EIN/AUS-Status des spezifizierten Operanden-Bits.	Anfang der logischen Verknüpfung Nicht erforderlich	153
LOAD NOT (Laden invertiert) LD NOT @LD NOT*2 %LD NOT*2 !LD NOT*1 !@LD NOT*3 !%LD NOT*3		Bildet den Anfang der logischen Verknüpfung und erzeugt eine EIN/AUS-Ausführungsbedingung auf der Grundlage des invertierten EIN/AUS-Status des spezifizierten Operanden-Bits.	Anfang der logischen Verknüpfung Nicht erforderlich	155
AND (UND) AND @AND %AND !AND*1 !@AND*1 !%AND*1		Bildet eine logische UND-Verknüpfung zwischen dem Status des spezifizierten Operanden-Bits und der aktuellen Ausführungsbedingung.	Fortsetzung der logischen Verknüpfung Erforderlich	157
AND NOT (UND invertiert) AND NOT @AND NOT*2 %AND NOT*2 !AND NOT*1 !@AND NOT*3 !%AND NOT*3		Invertiert den Status des spezifizierten Operanden-Bits und bildet ein logisches UND mit der aktuellen Ausführungsbedingung.	Fortsetzung der logischen Verknüpfung Erforderlich	159
OR (ODER) OR @OR %OR !OR*1 !@OR*1 !%OR*1		Bildet eine logische ODER-Verknüpfung zwischen dem EIN/AUS-Status des spezifizierten Operanden-Bits und der aktuellen Ausführungsbedingung.	Fortsetzung der logischen Verknüpfung Erforderlich	161
OR NOT (ODER invertiert) OR NOT @OR NOT*2 %OR NOT*2 !OR NOT*1 !@OR NOT*3 !%OR NOT*3		Invertiert den Status des spezifizierten Operanden-Bits und bildet ein logisches ODER mit der aktuellen Ausführungsbedingung.	Fortsetzung der logischen Verknüpfung Erforderlich	163

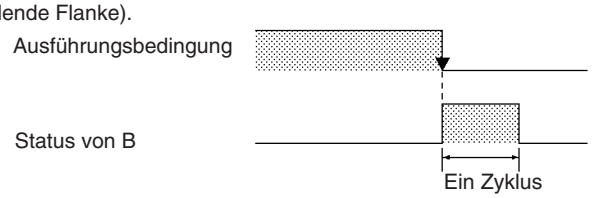
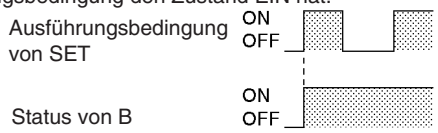
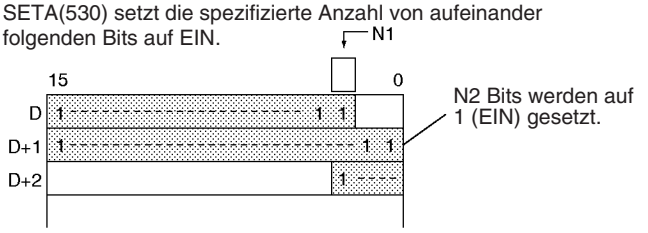
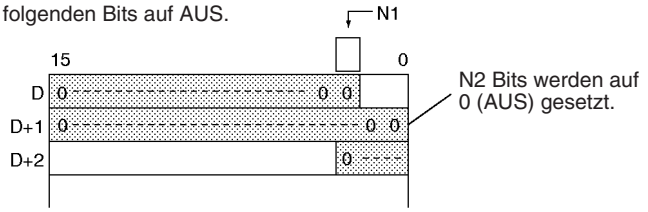
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
AND LOAD (UND Laden) AND LD	Logischer Block Logischer Block	Führt eine logische AND-Bedingung zwischen logischen Blöcken ein. LD bis — Logischer Block A LD bis — Logischer Block B AND LD ····· Reihenschaltung von logischem Block A und logischem Block B	Fortsetzung der logischen Verknüpfung Erforderlich	164
OR LOAD (ODER Laden) OR LD	Logischer Block Logischer Block	Führt eine logische ODER-Bedingung zwischen logischen Blöcken ein. LD bis — Logischer Block A LD bis — Logischer Block B OR LD ····· Parallelschaltung von logischem Block A und logischem Block B.	Fortsetzung der logischen Verknüpfung Erforderlich	166
NOT (Invertierung) NOT 520	---	Invertiert die Ausführungsbedingung.	Fortsetzung der logischen Verknüpfung Erforderlich	172
CONDITION ON (steigende Flanke) UP 521	UP(521)	UP(521) setzt die Ausführungsbedingung für einen Zyklus auf EIN, wenn die Ausführungsbedingung von AUS zu EIN wechselt.	Fortsetzung der logischen Verknüpfung Erforderlich	173
CONDITION OFF (fallende Flanke) DOWN 522	DOWN(522)	DOWN(522) setzt die Ausführungsbedingung für einen Zyklus auf EIN, wenn die Ausführungsbedingung von EIN auf AUS wechselt.	Fortsetzung der logischen Verknüpfung Erforderlich	173
BIT TEST LD TST 350	TST(350) S N **S**: Quellwort **N**: Bitnummer	LD TST(350), AND TST(350) sowie OR TST(350) werden im Programm wie LD, AND und OR verwendet. Die Ausführungsbedingung ist auf EIN gesetzt, wenn das spezifizierte Bit im spezifizierten Wort den Status EIN hat. Entsprechend ist sie auf AUS gesetzt, wenn das Bit den Status AUS hat.	Fortsetzung der logischen Verknüpfung Nicht erforderlich	174
BIT TEST LD TSTN 351	TSTN(351) S N **S**: Quellwort **N**: Bitnummer	LD TSTN(351), AND TSTN(351) sowie OR TSTN(351) werden im Programm wie LD NOT, AND NOT und OR NOT verwendet. Die Ausführungsbedingung ist auf AUS gesetzt, wenn das spezifizierte Bit im spezifizierten Wort den Zustand EIN hat. Entsprechend ist sie auf EIN gesetzt, wenn das Bit den Zustand AUS hat.	Fortsetzung der logischen Verknüpfung Nicht erforderlich	174
BIT TEST AND TST 350	AND TST(350) S N **S**: Quellwort **N**: Bitnummer	LD TST(350), AND TST(350) sowie OR TST(350) werden im Programm wie LD, AND und OR verwendet. Die Ausführungsbedingung ist auf EIN gesetzt, wenn das spezifizierte Bit im spezifizierten Wort den Status EIN hat. Entsprechend ist sie auf AUS gesetzt, wenn das Bit den Status AUS hat.	Fortsetzung der logischen Verknüpfung Erforderlich	174
BIT TEST AND TSTN 351	AND TSTN(351) S N **S**: Quellwort **N**: Bitnummer	LD TSTN(351), AND TSTN(351) sowie OR TSTN(351) werden im Programm wie LD NOT, AND NOT und OR NOT verwendet. Die Ausführungsbedingung ist auf AUS gesetzt, wenn das spezifizierte Bit im spezifizierten Wort den Zustand EIN hat. Entsprechend ist sie auf EIN gesetzt, wenn das Bit den Zustand AUS hat.	Fortsetzung der logischen Verknüpfung Erforderlich	174

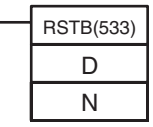
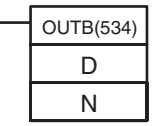
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
BIT TEST OR TST 350	TST(350) S N **S**: Quellwort **N**: Bitnummer	LD TST(350), AND TST(350) sowie OR TST(350) werden im Programm wie LD, AND und OR verwendet. Die Ausführungsbedingung ist auf EIN gesetzt, wenn das spezifizierte Bit im spezifizierten Wort den Status EIN hat. Entsprechend ist sie auf AUS gesetzt, wenn das Bit den Status AUS hat.	Fortsetzung der logischen Verknüpfung Erforderlich	174
BIT TEST OR TSTN 351	TSTN(351) S N **S**: Quellwort **N**: Bitnummer	LD TSTN(351), AND TSTN(351) sowie OR TSTN(351) werden im Programm wie LD NOT, AND NOT und OR NOT verwendet. Die Ausführungsbedingung ist auf AUS gesetzt, wenn das spezifizierte Bit im spezifizierten Wort den Zustand EIN hat. Entsprechend ist sie auf EIN gesetzt, wenn das Bit den Zustand AUS hat.	Fortsetzung der logischen Verknüpfung Erforderlich	174

2-2-2 Ausgabebefehle

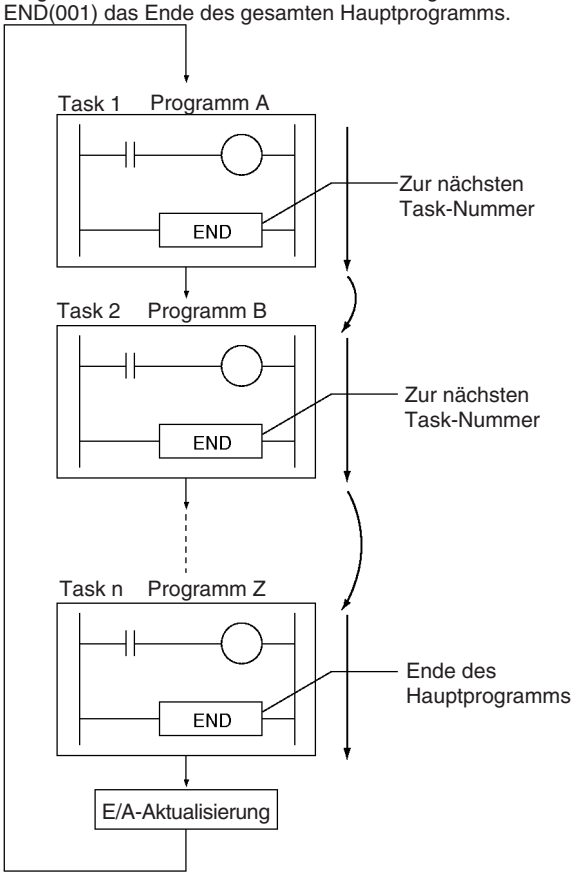
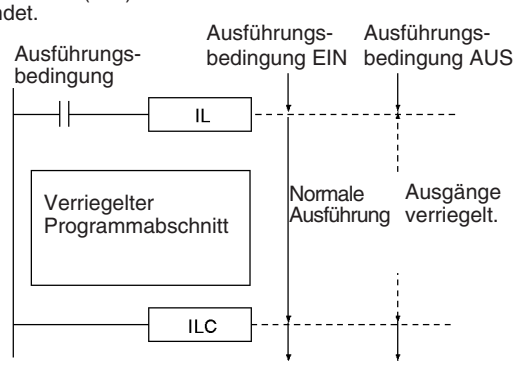
*1: Nicht unterstützt von CS1D CPU-Baugruppen für Duplex-CPU-Systeme.

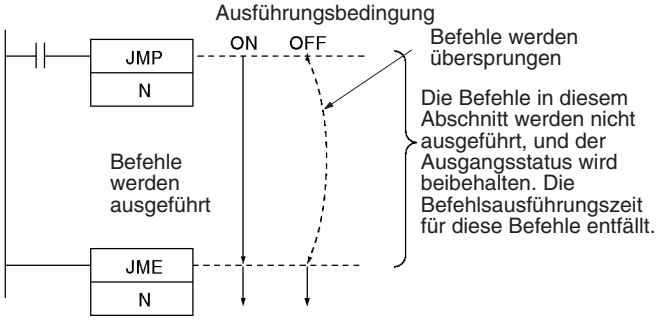
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
OUTPUT (Ausgabe) OUT !OUT*1		Gibt das Ergebnis (Ausführungsbedingung) der logischen Verarbeitung an das spezifizierte Bit aus.	Ausgang Erforderlich	177
OUTPUT NOT (Ausgabe invertiert) OUT NOT !OUT NOT*1		Invertiert das Ergebnis (Ausführungsbedingung) der logischen Verarbeitung und gibt es an das spezifizierte Bit aus.	Ausgang Erforderlich	178
KEEP (Halten) KEEP !KEEP*1 011	 **B**: Bit	Arbeitet als Halterelais.  	Ausgang Erforderlich	180
DIFFERENTIATE UP (steigende Flanke) DIFU !DIFU*1 013	 **B**: Bit	DIFU(013) setzt das angegebene Bit für einen Zyklus auf EIN, wenn die Ausführungsbedingung von AUS zu EIN wechselt (steigende Flanke). 	Ausgang Erforderlich	184

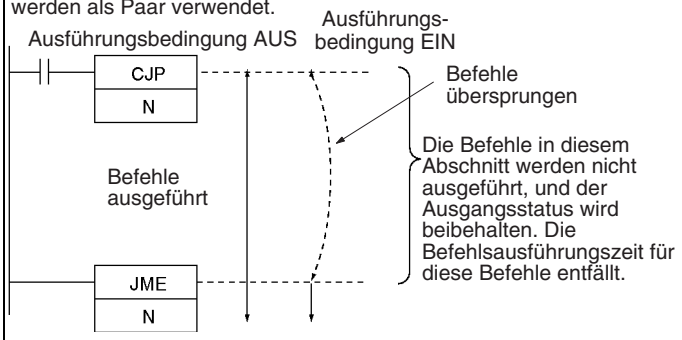
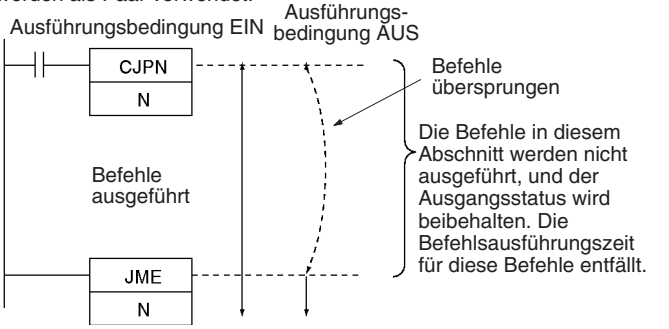
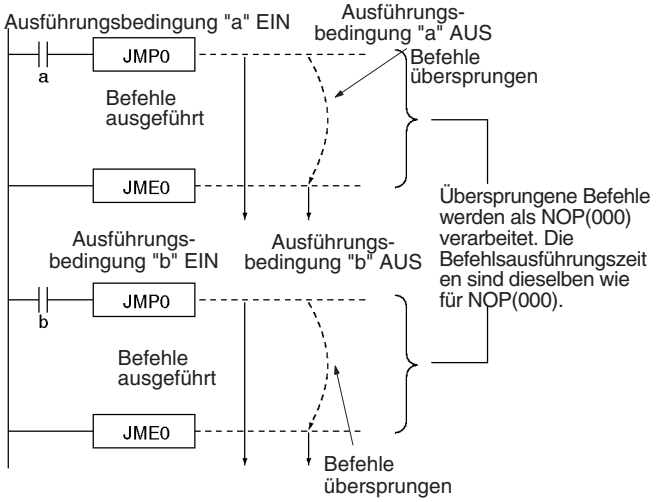
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
DIFFERENTIATE DOWN (fallende Flanke) DIFD !DIFD*1 014	DIFD(014) B **B**: Bit	DIFD(014) setzt das angegebene Bit für einen Zyklus auf EIN, wenn die Ausführungsbedingung von EIN zu AUS wechselt (fallende Flanke). 	Ausgang Erforderlich	184
SET (Setzen) SET @SET %SET !SET*1 !@SET*1 !%SET*1	SET B **B**: Bit	SET setzt das Operanden-Bit auf EIN, wenn die Ausführungsbedingung den Zustand EIN hat. 	Ausgang Erforderlich	187
RESET (Rücksetzen) RSET @RSET %RSET !RSET*1 !@RSET*1 !%RSET*1	RSET B **B**: Bit	RSET setzt das Operanden-Bit auf AUS, wenn die Ausführungsbedingung den Zustand EIN hat. 	Ausgang Erforderlich	187
MULTIPLE BIT SET (Mehrere Bits setzen) SETA @SETA 530	SETA(530) D N1 N2 **D**: Anfangswort **N1**: Anfangsbit **N2**: Anzahl der Bits	SETA(530) setzt die spezifizierte Anzahl von aufeinander folgenden Bits auf EIN. 	Ausgang Erforderlich	189
MULTIPLE BIT RESET (Mehrere Bits rücksetzen) RSTA @RSTA 531	RSTA(531) D N1 N2 **D**: Anfangswort **N1**: Anfangsbit **N2**: Anzahl der Bits	RSTA(531) setzt die spezifizierte Anzahl von aufeinander folgenden Bits auf AUS. 	Ausgang Erforderlich	189
SINGLE BIT SET (Einzelbit setzen) (nur CS1-H, CJ1-H, CJ1M und CS1D) SETB @SETB !SETB*1 !@SETB*1	SETB(532) D N **D**: Wortadresse **N**: Bitnummer	SETB(532) setzt das spezifizierte Bit im spezifizierten Wort auf EIN, wenn die Ausführungsbedingung den Zustand EIN hat. Anders als der SET-Befehl kann SETB(532) zum Setzen eines Bits in einem DM- oder EM-Wort verwendet werden.	Ausgang Erforderlich	192

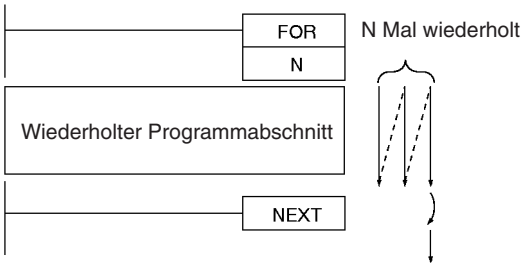
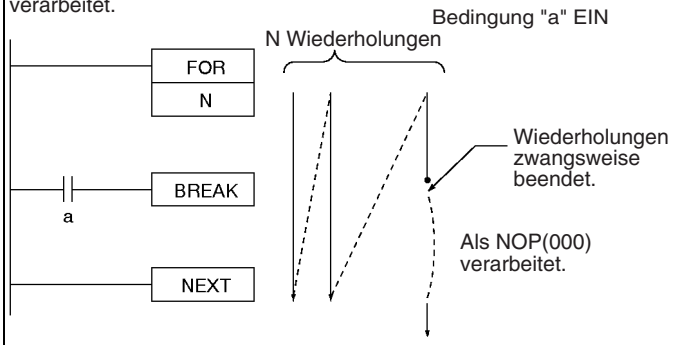
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
SINGLE BIT RESET (Einzelbit rücksetzen) (nur CS1-H, CJ1-H, CJ1M oder CS1D) RSTB @RSTB !RSTB*1 !@RSTB*1	 **D**: Wortadresse **N**: Bitnummer	RSTB(533) setzt das spezifizierte Bit im spezifizierten Wort auf AUS, wenn die Ausführungsbedingung den Zustand EIN hat. Anders als der RSET-Befehl kann RSTB(533) zum Rücksetzen eines Bits in einem DM- oder EM-Wort verwendet werden.	Ausgang Erforderlich	192
SINGLE BIT OUTPUT (Einzelbit-Ausgabe) (nur CS1-H, CJ1-H, CJ1M oder CS1D) OUTB @OUTB !OUTB*1	 **D**: Wortadresse **N**: Bitnummer	OUTB(534) gibt das Ergebnis (Ausführungsbedingung) der logischen Verarbeitung an das spezifizierte Bit aus. Anders als der OUT-Befehl kann OUTB(534) zur Steuerung eines Bits in einem DM- oder EM-Wort verwendet werden.	Ausgang Erforderlich	195

2-2-3 Sequenz-Steuerbefehle

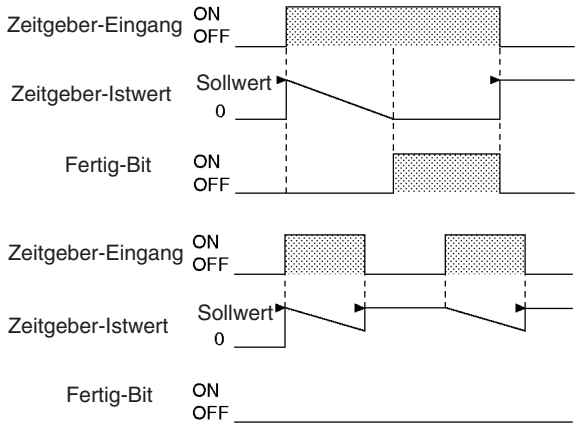
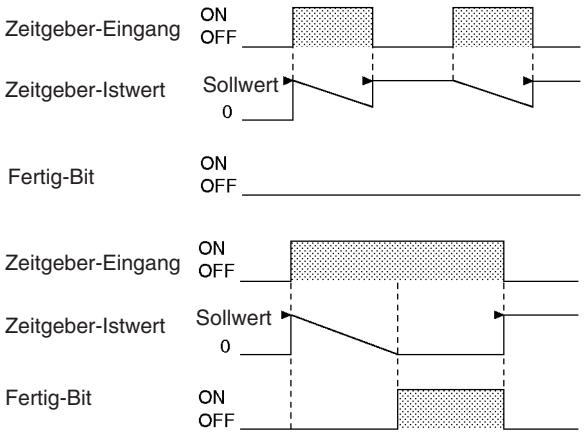
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
END (Ende) END 001	END(001)	Kennzeichnet das Ende eines Programms. END(001) beendet die Ausführung eines Programms für diesen Zyklus. Nach END(001) geschriebene Befehle werden nicht ausgeführt. Die Ausführung fährt mit dem Programm mit der nächsten Task-Nummer fort. Wenn das aktuell ausgeführte Programm die höchste Task-Nummer im Programm hat, markiert END(001) das Ende des gesamten Hauptprogramms.  	Ausgang Nicht erforderlich	197
NO OPERATION (Befehl ohne Funktion) NOP 000		Dieser Befehl hat keine Funktion. (Durch NOP(000) erfolgt keine Verarbeitung.)	Ausgang Nicht erforderlich	198
INTERLOCK (Verriegelung) IL 002	IL(002)	Verriegelt alle Ausgänge zwischen IL(002) und ILC(003), wenn die Ausführungsbedingung für IL(002) den Zustand AUS hat. IL(002) und ILC(003) werden normalerweise als Paar verwendet.  	Ausgang Erforderlich	201

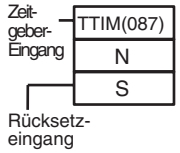
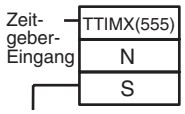
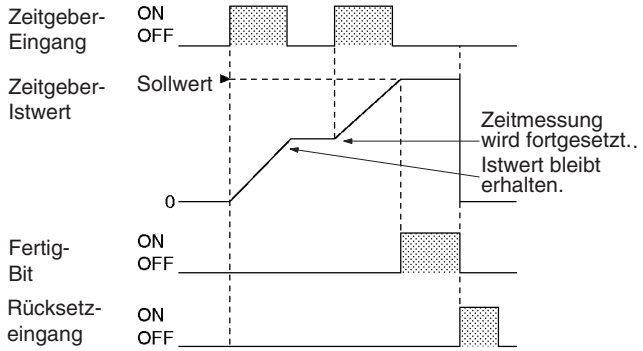
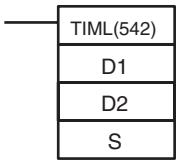
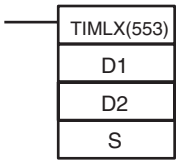
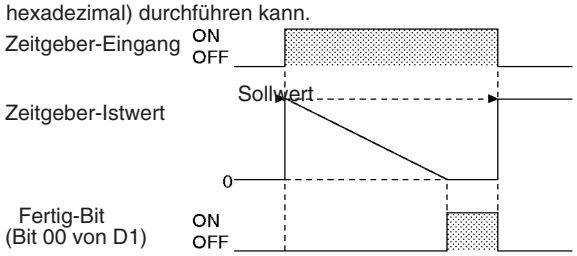
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
INTERLOCK CLEAR (Verriegelung aufheben) ILC 003	ILC(003)	Alle Ausgänge zwischen IL(002) und ILC(003) sind verriegelt, wenn die Ausführungsbedingung für IL(002) den Zustand AUS hat. IL(002) und ILC(003) werden normalerweise als Paar verwendet.	Ausgang Nicht erforderlich	201
MULTI-INTERLOCK DIFFERENTIATION HOLD (Mehrfachverriegelung, Flanke halten) MILH 517 Nur für CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0	MILH(517) N D **N**: Verriegelungsnummer **D**: Verriegelungsstatus-Bit	Wenn die Ausführungsbedingung für MILH(517) den Zustand AUS hat, werden die Ausgänge für alle Befehle zwischen diesem MILH(517)-Befehl und dem nächsten MILC(519)-Befehl verriegelt. MILH(517) und MILC(519) werden als Paar verwendet. MILH(517)/MILC(519)-Befehle können verschachtelt werden (z. B. MILH(517) – MILH(517) – MILC(519) – MILC(519)). Befindet sich ein Befehl mit Flankenauswertung (DIFU, DIFD bzw. ein Befehl mit einem Präfix @ oder %) zwischen MILH(517) und dem entsprechenden MILC(519)-Befehl, wird dieser Befehl nach Aufhebung der Verriegelung ausgeführt, wenn die Flankenauswertungsbedingung des Befehls eingetreten ist.	Ausgang Erforderlich	205
MULTI-INTERLOCK DIFFERENTIATION RELEASE (Mehrfachverriegelung, Flanke nicht halten) MILR 518 Nur für CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0	MILR(518) N D **N**: Verriegelungsnummer **D**: Verriegelungsstatus-Bit	Wenn die Ausführungsbedingung für MILR(518) den Zustand AUS hat, werden die Ausgänge für alle Befehle zwischen diesem MILR(518)-Befehl und dem nächsten MILC(519)-Befehl verriegelt. MILR(518) und MILC(519) werden als Paar verwendet. Die MILR(518)/MILC(519)-Verriegelungen können verschachtelt werden (z. B. MILR(518) – MILR(518) – MILC(519) – MILC(519)). Bei einem Befehl mit Flankenauswertung (DIFU, DIFD bzw. ein Befehl mit einem Präfix @ oder %) zwischen MILR(518) und dem entsprechenden MILC(519)-Befehl wird dieser Befehl nach Aufhebung der Verriegelung nicht ausgeführt, auch wenn die Flankenauswertungsbedingung des Befehls eingetreten ist.	Ausgang Erforderlich	205
MULTI-INTERLOCK CLEAR (Mehrfachverriegelung aufheben) MILC 519 Nur für CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0	MILC(519) N **N**: Verriegelungsnummer	Hebt eine Verriegelung auf, die durch einen MILH(517)- oder MILR(518)-Befehl mit der gleichen Verriegelungsnummer gestartet wurde. Alle Ausgaben zwischen MILH(517)/MILR(518) und dem entsprechenden MILC(519)-Befehl mit der gleichen Verriegelungsnummer werden verriegelt, wenn die Ausführungsbedingung für MILH(517)/MILR(518) den Zustand AUS hat.	Ausgang Nicht erforderlich	205
JUMP (Sprung) JMP 004	JMP(004) N **N**: Sprungnummer	Wenn die Ausführungsbedingung für JMP(004) den Zustand AUS hat, springt die Programmausführung direkt zum ersten JME(005) mit derselben Sprungnummer im Programm. JMP(004) und JME(005) werden als Paar verwendet. 	Ausgang Erforderlich	219
JUMP END (Sprungende) JME 005	JME(005) N **N**: Sprungnummer	Zeigt das Ende eines durch JMP(004) oder CJP(510) eingeleiteten Sprungs an.	Ausgang Nicht erforderlich	219

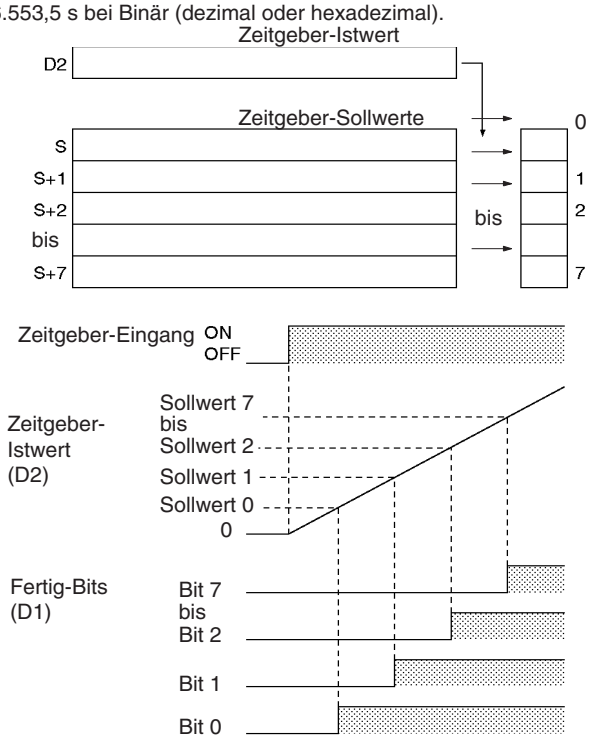
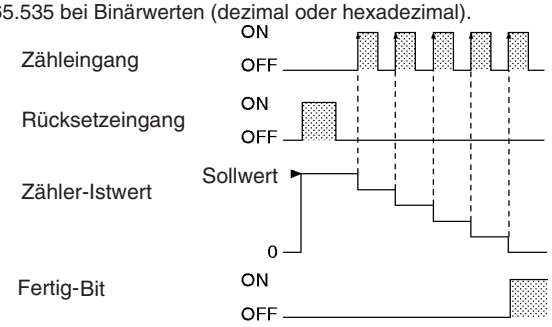
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
CONDITIONAL JUMP (Bedingter Sprung) CJP 510	CJP(510) N **N**: Sprungnummer	Die Funktion von CJP(510) ist im Prinzip die Umkehrung der Funktion von JMP(004). Wenn die Ausführungsbedingung für CJP(510) den Zustand EIN hat, springt die Programmausführung direkt zum ersten JME(005) mit derselben Sprungnummer im Programm. CJP(510) und JME(005) werden als Paar verwendet. 	Ausgang Erforderlich	223
CONDITIONAL JUMP NOT (Bedingter Sprung invertiert) CJPN 511	CJPN(511) N **N**: Sprungnummer	Die Funktion von CJPN(511) ist annähernd dieselbe wie von JMP(004). Wenn die Ausführungsbedingung für CJPN(511) den Zustand AUS hat, springt die Programmausführung direkt zum ersten JME(005) mit derselben Sprungnummer im Programm. CJPN(511) und JME(005) werden als Paar verwendet. 	Ausgang Nicht erforderlich	223
MULTIPLE JUMP (Mehrfachsprung) JMP0 515	JMP0(515)	Wenn die Ausführungsbedingung für JMP0(515) den Zustand AUS hat, werden alle Befehle zwischen JMP0(515) und dem nächsten JME0(516) im Programm als NOP(000) verarbeitet, also nicht verarbeitet. JMP0(515) und JME0(516) sind als Paare zu verwenden. Es gibt keine Begrenzung für die Anzahl der JMP0/JME0-Paare, die im Programm verwendet werden können. 	Ausgang Erforderlich	227
MULTIPLE JUMP END (Mehrfachsprung-Ende) JME0 516	JME0(516)	Wenn die Ausführungsbedingung für JMP0(515) den Zustand AUS hat, werden alle Befehle zwischen JMP0(515) und dem nächsten JME0(516) im Programm als NOP(000) verarbeitet, also nicht verarbeitet. JMP0(515) und JME0(516) sind als Paare zu verwenden. Es gibt keine Begrenzung der Anzahl der JMP0/JME0-Paare, die im Programm verwendet werden können.	Ausgang Nicht erforderlich	227

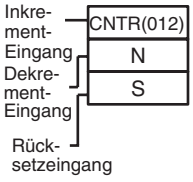
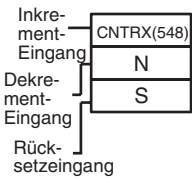
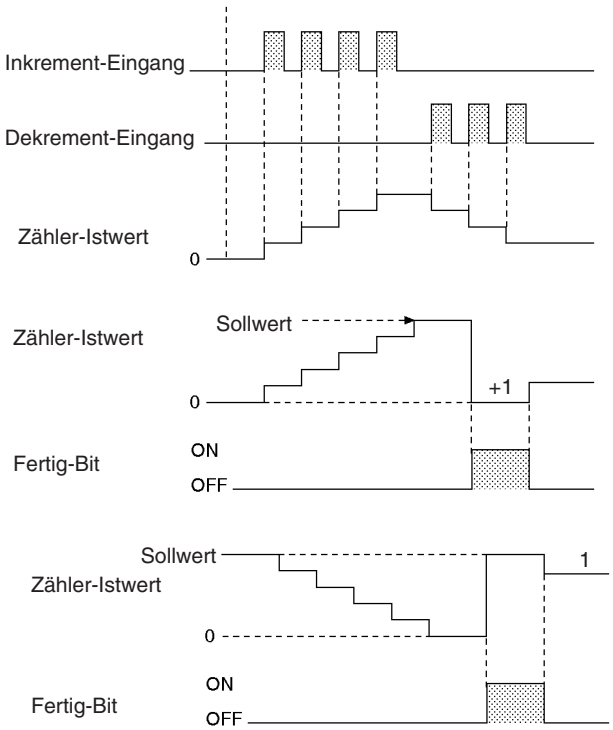
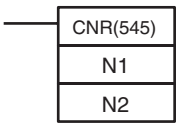
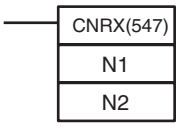
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
FOR-NEXT-Programmschleife FOR 512	FOR(512) N **N**: Anzahl der Schleifen	Die Befehle zwischen FOR(512) und NEXT(513) werden N-mal wiederholt. FOR(512) und NEXT(513) werden als Paar verwendet. 	Ausgang Nicht erforderlich	229
BREAK LOOP (FOR-NEXT-Schleife verlassen) BREAK 514	BREAK(514)	Wird in einer FOR-NEXT-Schleife programmiert, um die Schleife bei einer bestimmten Ausführungsbedingung zu unterbrechen. Die verbliebenen Befehle in der Schleife werden als NOP(000)-Befehle verarbeitet. 	Ausgang Erforderlich	232
FOR-NEXT-Programmschleife NEXT 513	NEXT(513)	Die Befehle zwischen FOR(512) und NEXT(513) werden N-mal wiederholt. FOR(512) und NEXT(513) werden als Paar verwendet.	Ausgang Nicht erforderlich	229

2-2-4 Zeitgeber- und Zählerbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
TIMER (Zeitgeber) TIM (BCD) TIMX (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	TIM N S **N**: Zeitgebernummer **S**: Sollwert TIMX(550) N S **N**: Zeitgebernummer **S**: Sollwert	TIM/TIMX(550) ist ein dekrementierender Zeitgeber mit Schritten von 0,1 s. Der Einstellbereich für den Sollwert reicht von 0 bis 999,9 s bei BCD-Werten und von 0 bis 6.553,5 s bei Binärwerten (dezimal oder hexadezimal). 	Ausgang Erforderlich	235
HIGH-SPEED TIMER (Schneller Zeitgeber) TIMH 015 (BCD) TIMHX 551 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	TIMH(015) N S **N**: Zeitgebernummer **S**: Sollwert TIMHX(551) N S **N**: Zeitgebernummer **S**: Sollwert	TIMH(015)/TIMHX(551) ist ein dekrementierender Zeitgeber mit 10-ms-Schritten. Der Einstellbereich für den Sollwert reicht von 0 bis 99,99 s bei BCD-Werten und von 0 bis 655,35 s bei Binärwerten (dezimal oder hexadezimal). 	Ausgang Erforderlich	240
ONE-MS TIMER (1-ms-Zeitgeber) TMHH 540 (BCD) TMHHX 552 (BCD) (nur CS1-H, CJ1-H, CJ1M und CS1D)	TMHH(540) N S **N**: Zeitgebernummer **S**: Sollwert TMHHX(552) N S **N**: Zeitgebernummer **S**: Sollwert	TMHH(540)/TMHHX(552) sind dekrementierende Zeitgeber mit 1-ms-Schritten. Der Einstellbereich für den Sollwert reicht von 0 bis 9,999 s bei BCD-Werten und von 0 bis 65,535 s bei Binärwerten (dezimal oder hexadezimal). Die Zeitablaufdiagramme für TMHH(540) und TMHHX(552) entsprechen denen, wie sie oben für TIMH(015) dargestellt sind.	Ausgang Erforderlich	244

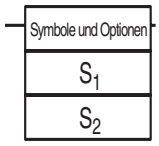
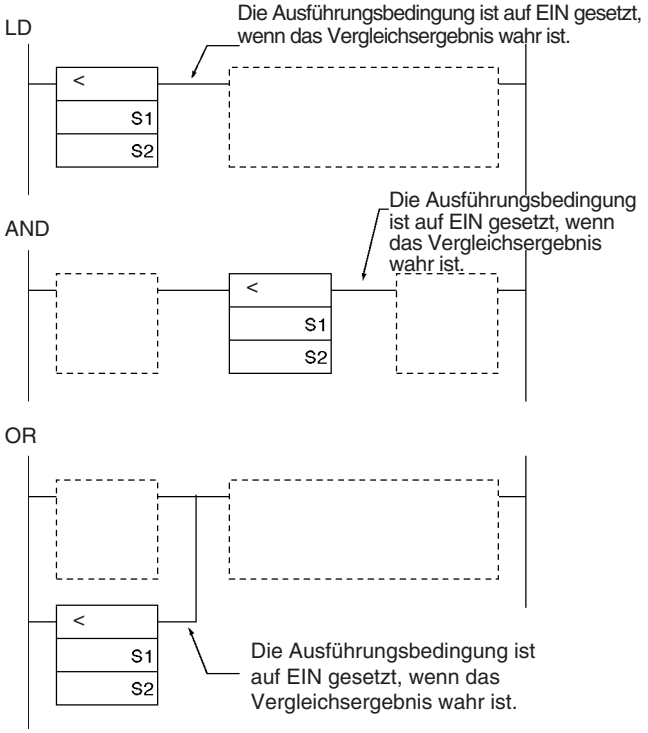
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
ACCUMULATIVE TIMER (Akkumulierender Zeitgeber) TTIM 087 (BCD) TTIMX 555 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	 **N**: Zeitgeber-nummer **S**: Sollwert  **N**: Zeitgeber-nummer **S**: Sollwert	TTIM(087)/TTIMX(555) ist ein inkrementierender Zeitgeber mit Schritten von 0,1 s. Der Einstellbereich für den Sollwert reicht von 0 bis 999,9 s bei BCD-Werten und von 0 bis 6.553,5 s bei Binärwerten (dezimal oder hexadezimal). 	Ausgang Erforderlich	247
LONG TIMER (Langzeitgeber) TIML 542 (BCD) TIMLX 553 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	 **D1**: Fertig-Bit **D2**: Istwertwort **S**: Sollwertwort  **D1**: Fertig-Bit **D2**: Istwertwort **S**: Sollwertwort	TIML(542)/TIMLX(553) ist ein dekrementierender Zeitgeber mit Schritten von 0,1 s, der eine Zeitmessung bis zu ca. 115 Tagen bei BCD-Werten und bis zu 49.710 Tagen bei Binärwerten (dezimal oder hexadezimal) durchführen kann. 	Ausgang Erforderlich	251

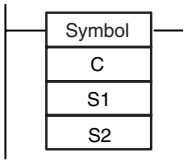
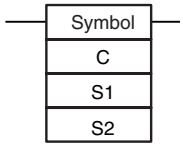
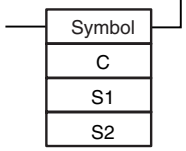
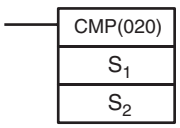
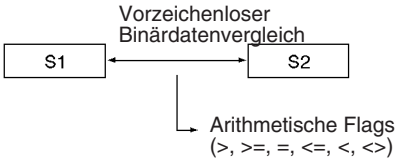
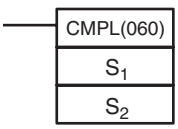
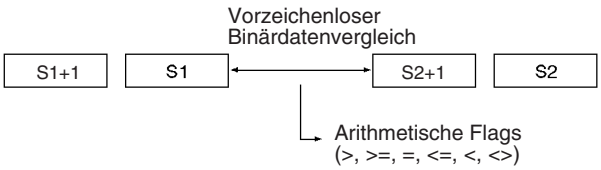
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
MULTI-OUTPUT TIMER (Mehrfachausgabe-Zeitgeber) MTIM 543 (BCD) MTIMX 554 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	MTIM(543) D1 D2 S **D1**: Fertig-Bits **D2**: Istwertwort **S**: Erstes Sollwertwort MTIMX(554) D1 D2 S **D1**: Fertig-Bits **D2**: Istwertwort **S**: Erstes Sollwertwort	MTIM(543)/MTIMX(554) ist ein in 0,1-s-Schritten inkrementierender Zeitgeber mit 8 unabhängigen Sollwerten und Fertig-Bits. Der Einstellbereich für den Sollwert ist 0 bis 999,9 s bei BCD und 0 bis 6.553,5 s bei Binär (dezimal oder hexadezimal). 	Ausgang Erforderlich	254
COUNTER (Zähler) CNT (BCD) CNTX 546 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	Zähleingang CNT N S Rücksetzeingang **N**: Zählernummer **S**: Sollwert Zähleingang CNTX(546) N S Rücksetzeingang **N**: Zählernummer **S**: Sollwert	CNT/CNTX(546) ist ein dekrementierender Zähler. Der Einstellbereich für den Sollwert reicht von 0 bis 9.999 bei BCD-Werten und von 0 bis 65.535 bei Binärwerten (dezimal oder hexadezimal). 	Ausgang Erforderlich	260

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
REVERSIBLE COUNTER (Umkehrbarer Zähler) CNTR 012 (BCD) CNTRX 548 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	 **N**: Zählernummer **S**: Sollwert  **N**: Zählernummer **S**: Sollwert	CNTR(012)/CNTRX(548) ist ein reversierbarer Zähler. 	Ausgang Erforderlich	263
RESET TIMER/ COUNTER (Zeitgeber/Zähler zurücksetzen) CNR @CNR 545 (BCD) CNRX @CNRX 547 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	 N_1: Erste Zeitgeber-/Zähler-Nummer N_2: Letzte Zeitgeber-/Zähler-Nummer  N_1: Erste Zeitgeber-/Zähler-Nummer N_2: Letzte Zeitgeber-/Zähler-Nummer	CNR(545)/CNRX(547) setzt die Zeitgeber oder Zähler innerhalb des spezifizierten Bereichs der Zeitgeber- bzw. Zählernummern zurück. Setzt den Istwert auf das Maximum: 9999 (CNR) bzw. 65535 (CNRX).	Ausgang Erforderlich	267

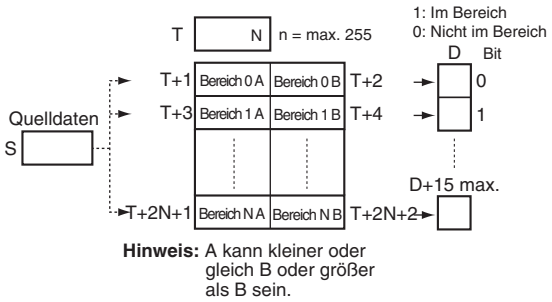
2-2-5 Vergleichsbefehle

*1: Nicht unterstützt von CS1D CPU-Baugruppen für Duplex-CPU-Systeme.

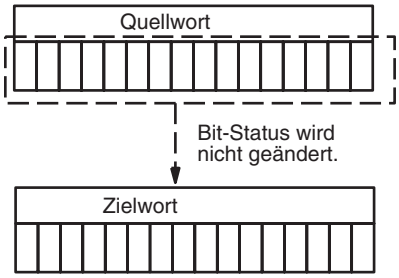
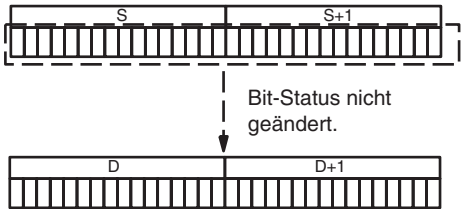
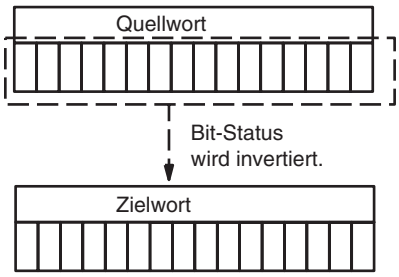
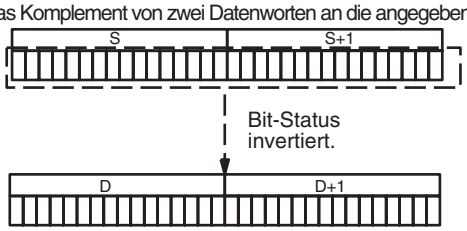
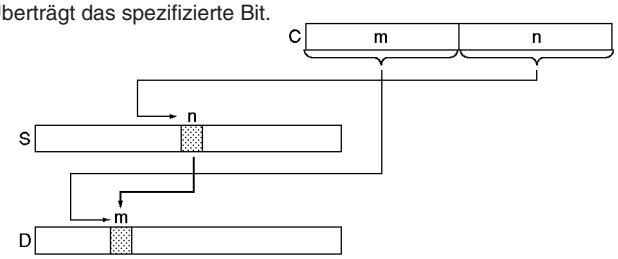
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
Symbolvergleich (ohne Vorzeichen) LD, AND, OR + =, <>, <, <=, >, >= 300 (=) 305 (<>) 310 (<) 315 (<=) 320 (>) 325(>=)	 **S_1**: Vergleichsdaten 1 **S_2**: Vergleichsdaten 2	Symbolvergleichsbefehle (vorzeichenlos) vergleichen zwei Werte (Konstanten und/oder den Inhalt spezifizierter Worte) und erzeugen eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Die Symbolvergleichsbefehle können auf 3 Arten angewendet werden: als LD (LOAD), AND und OR. 	LD: Nicht erforderlich AND, OR: Erforderlich	275
Symbolvergleich (Doppelwort, ohne Vorzeichen) LD, AND, OR + =, <>, <, <=, >, >= + L 301 (=) 306 (<>) 311 (<) 316 (<=) 321 (>) 326 (>=)	**S_1**: Vergleichsdaten 1 **S_2**: Vergleichsdaten 2	Symbolvergleichsbefehle (Doppelwort, ohne Vorzeichen) vergleichen zwei Werte (Konstanten und/oder den Inhalt spezifizierter Doppelwortdaten) ohne Vorzeichen und erzeugen eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Die Symbolvergleichsbefehle können auf drei Arten angewendet werden: als LD (LOAD), AND und OR.	LD: Nicht erforderlich AND, OR: Erforderlich	275
Symbolvergleich (mit Vorzeichen) LD, AND, OR + =, <>, <, <=, >, >= +S 302 (=) 307 (<>) 312 (<) 317 (<=) 322 (>) 327 (>=)	**S_1**: Vergleichsdaten 1 **S_2**: Vergleichsdaten 2	Symbolvergleichsbefehle (mit Vorzeichen) vergleichen zwei Werte (Konstanten oder den Inhalt spezifizierter Worte) mit Vorzeichen und erzeugen eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Die Symbolvergleichsbefehle können auf 3 Arten angewendet werden: als LD (LOAD), AND und OR.	LD: Nicht erforderlich AND, OR: Erforderlich	275

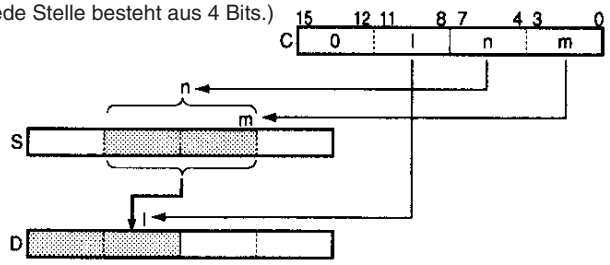
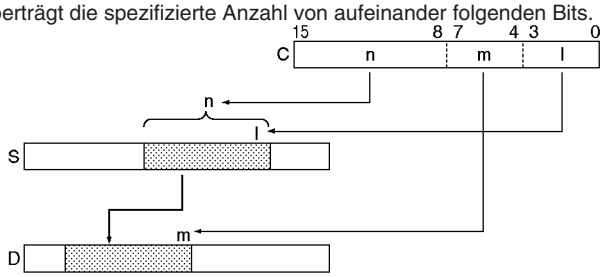
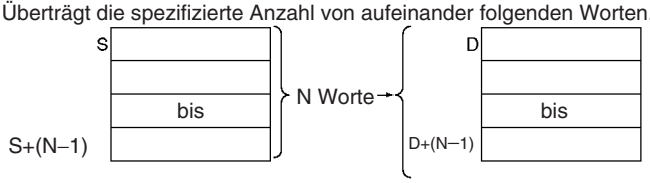
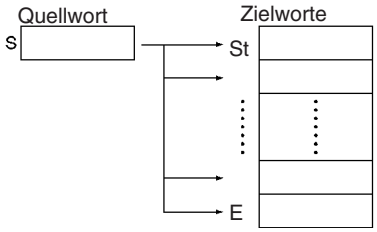
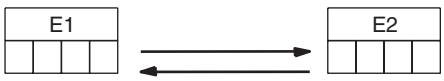
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
Symbolvergleich (Doppelwort, mit Vorzeichen) LD, AND, OR + =, <>, <, <=, >, >= +SL 303 (=) 308 (<>) 313 (<) 318 (<=) 323 (>) 328 (>=)	**S_1**: Vergleichsdaten 1 **S_2**: Vergleichsdaten 2	Symbolvergleichsbefehle (Doppelwort, mit Vorzeichen) vergleichen zwei Werte (Konstanten oder den Inhalt spezifizierter Doppelwortdaten) mit Vorzeichen und erzeugen eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Die Symbolvergleichsbefehle können auf 3 Arten angewendet werden: als LD (LOAD), AND und OR.	LD: Nicht erforderlich AND, OR: Erforderlich	275
Zeitvergleich LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT 341 (= DT) 342 (<> DT) 343 (< DT) 344 (<= DT) 345 (> DT) 346 (>= DT) (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	LD (LOAD):  AND:  OR:  **C:** Steuerwort **S1:** Erstes Wort der Ist-Zeit **S2:** Erstes Wort der Vergleichszeit	Zeitvergleichsbefehle vergleichen zwei BCD-Zeitwerte und erzeugen eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Die Zeitvergleichsbefehle können auf 3 Arten angewendet werden: als LD (LOAD), AND und OR. Zeitwerte (Jahr, Monat, Tag, Stunde, Minute und Sekunde) können beim Vergleich maskiert/demaskiert werden, so dass die Erstellung von Kalender-Zeitfunktionen einfach ist.	LD: Nicht erforderlich AND, OR: Erforderlich	281
UNSIGNED COMPARE (Vergleich ohne Vorzeichen) CMP !CMP*1 020	 **S_1**: Vergleichsdaten 1 **S_2**: Vergleichsdaten 2	Vergleicht zwei vorzeichenlose Binärwerte (Konstanten und/oder den Inhalt spezifizierter Worte) und gibt das Ergebnis über die arithmetischen System-Flags aus. 	Ausgang Erforderlich	287
DOUBLE UNSIGNED COMPARE (Doppelwort-Vergleich ohne Vorzeichen) CMPL 060	 **S_1**: Vergleichsdaten 1 **S_2**: Vergleichsdaten 2	Vergleicht zwei vorzeichenlose 2-Wort-Binärwerte (Konstanten und/oder den Inhalt spezifizierter Worte) und gibt das Ergebnis über die arithmetischen System-Flags aus. 	Ausgang Erforderlich	290

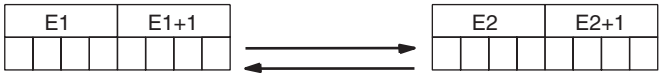
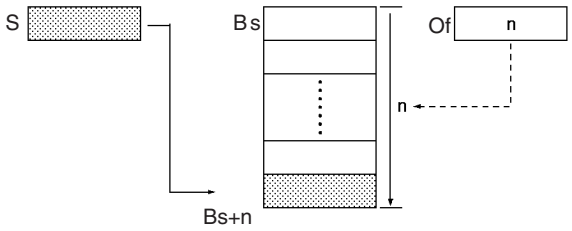
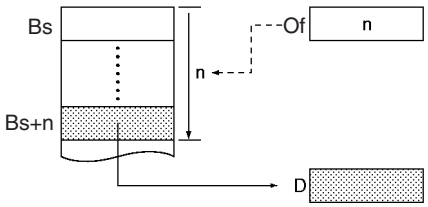
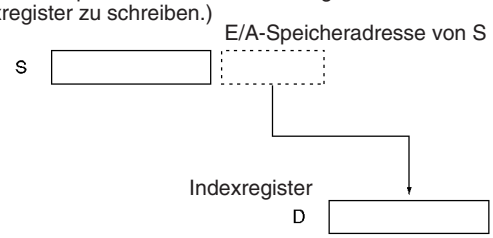
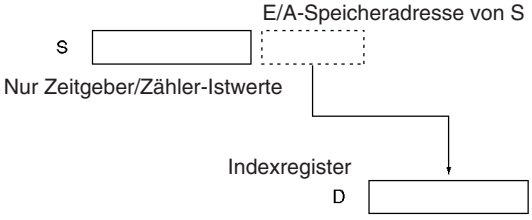
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
SIGNED BINARY COMPARE (Binärdatenvergleich mit Vorzeichen) CPS !CPS*1 114	CPS(114) S_1 S_2 **S1**: Vergleichsdaten 1 **S2**: Vergleichsdaten 2	Vergleicht zwei Binärwerte mit Vorzeichen (Konstanten und/oder den Inhalt spezifizierter Worte) und gibt das Ergebnis über die arithmetischen System-Flags aus. Binärwertvergleich mit Vorzeichen S1 ⟷ S2 Arithmetische Flags (>, >=, =, <=, <, <>)	Ausgang Erforderlich	293
DOUBLE SIGNED BINARY COMPARE (Doppelwort-Binärdatenvergleich mit Vorzeichen) CPSL 115	CPSL(115) S_1 S_2 **S1**: Vergleichsdaten 1 **S2**: Vergleichsdaten 2	Vergleicht zwei Doppelwort-Binärwerte mit Vorzeichen (Konstanten und/oder den Inhalt spezifizierter Worte) und gibt das Ergebnis über die arithmetischen System-Flags aus. Binärwertvergleich mit Vorzeichen S1+1 S1 ⟷ S2+1 S2 Arithmetische Flags (>, >=, =, <=, <, <>)	Ausgang Erforderlich	296
MULTIPLE COMPARE (Mehrfach-Vergleich) MCMP @MCMP 019	MCMP(019) S_1 S_2 R **S1**: Erstes Wort von Satz 1 **S2**: Erstes Wort von Satz 2 **R**: Ergebniswort	Vergleicht 16 aufeinander folgende Worte mit 16 anderen aufeinander folgenden Worten und setzt das entsprechende Bit im Ergebniswort auf EIN, wenn die Inhalte der Worte **nicht** gleich sind. Vergleich S1 ⟷ S2 → R S1+1 ⟷ S2+1 → 1 S1+14 ⟷ S2+14 → 14 S1+15 ⟷ S2+15 → 15 0: Die Worte sind gleich. 1: Die Worte sind nicht gleich.	Ausgang Erforderlich	299
TABLE COMPARE (Tabellenvergleich) TCMP @TCMP 085	TCMP(085) S T R **S**: Quelldaten **T**: Erstes Wort der Tabelle **R**: Ergebniswort	Vergleicht die Quelldaten mit dem Inhalt von 16 Worten und setzt das entsprechende Bit im Ergebniswort auf EIN, wenn die Inhalte identisch sind. Vergleich S ⟷ T → R T+1 → 1 T+14 → 14 T+15 → 15 1: Die Daten sind gleich. 0: Die Daten sind nicht gleich.	Ausgang Erforderlich	301
UNSIGNED BLOCK COMPARE (Vorzeichenloser Blockvergleich) BCMP @BCMP 068	BCMP(068) S T R **S**: Quelldaten **T**: Erstes Wort der Tabelle **R**: Ergebniswort	Vergleicht die Quelldaten mit 16 Bereichen (durch 16 untere und 16 obere Grenzwerte definiert), und setzt das entsprechende Bit im Ergebniswort auf EIN, wenn sich die Quelldaten innerhalb des Bereichs befinden. Bereiche 1: Im Bereich 0: Nicht im Bereich Unterer Grenzwert Oberer Grenzwert R Quelldaten S T bis T+1 → 0 T+2 bis T+3 → 1 T+28 bis T+29 → 14 T+30 bis T+31 → 15	Ausgang Erforderlich	304

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
EXPANDED BLOCK COMPARE (Erweiterter Blockvergleich) BCMP2 @BCMP2 502 (Nur für CS1-H, CJ1-H und CS1D CPU-Baugruppen ab Ver. 2.0) CJ1M CPU-Baugruppe (vor Version 2.0 oder Baugruppenversion ab 2.0)	BCMP2(502) S T R **S**: Quelldaten **T**: Erstes Wort des Blocks **R**: Ergebniswort	Vergleicht die Quelldaten mit bis zu 256 Bereichen (durch obere und untere Grenzwerte definiert) und setzt das entsprechende Bit im Ergebniswort auf EIN, wenn die Quelldaten innerhalb eines Bereichs liegen.  **Hinweis:** A kann kleiner oder gleich B oder größer als B sein.	Ausgang Erforderlich	306
AREA RANGE COMPARE (Bereichsvergleich) ZCP @ZCP 088 (nur CS1-H, CJ1-H, CJ1M und CS1D)	ZCP(088) CD LL UL **CD**: Vergleichsdaten (1 Wort) **LL**: Unterer Grenzwert des Bereichs **UL**: Oberer Grenzwert des Bereichs	Vergleicht den 16-Bit-Binärwert ohne Vorzeichen in CD (Wortinhalt oder Konstante) mit dem durch LL (unterer Grenzwert) und UL (oberer Grenzwert) definierten Bereich und gibt die Ergebnisse über System-Arithmetik-Flags aus.	Ausgang Erforderlich	310
DOUBLE AREA RANGE COMPARE (Doppelwort-Bereichsvergleich) ZCPL @ZCPL 116 (nur CS1-H, CJ1-H, CJ1M und CS1D)	ZCPL(116) CD LL UL **CD**: Vergleichsdaten (2 Worte) **LL**: Unterer Grenzwert des Bereichs **UL**: Oberer Grenzwert des Bereichs	Vergleicht den 32-Bit-Binärwert ohne Vorzeichen in CD und CD+1 (Wortinhalt oder Konstante) mit dem durch LL (unterer Grenzwert) und UL (oberer Grenzwert) definierten Bereich und gibt die Ergebnisse über System-Arithmetik-Flags aus.	Ausgang Erforderlich	313

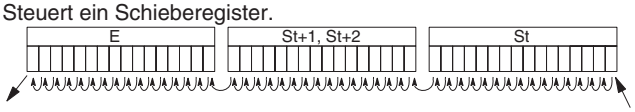
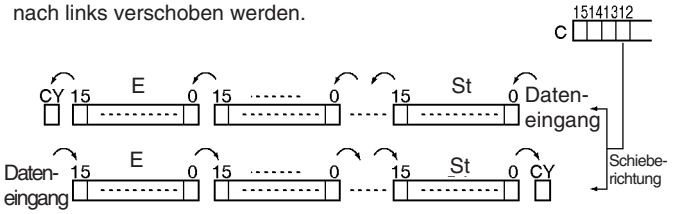
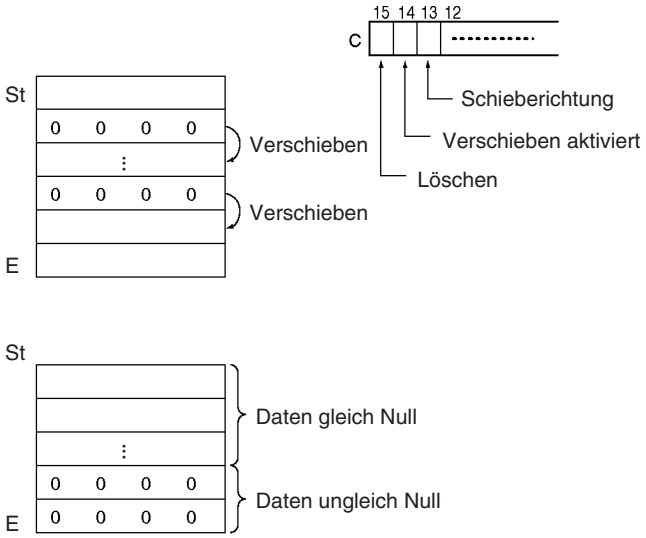
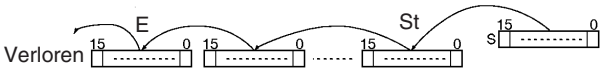
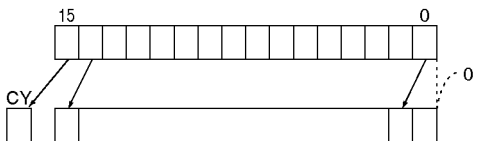
2-2-6 Datenübertragungsbefehle

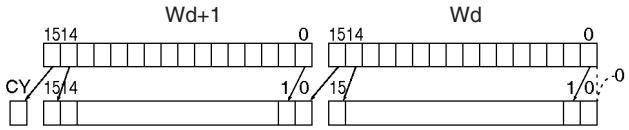
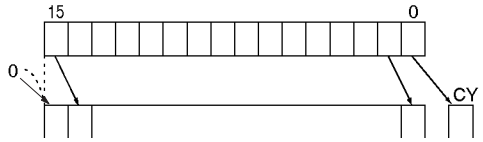
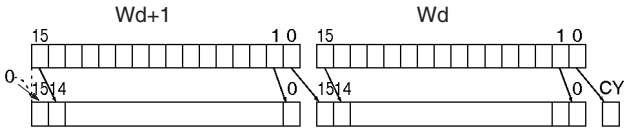
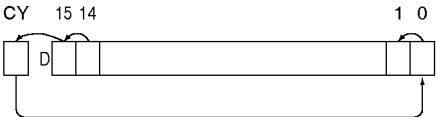
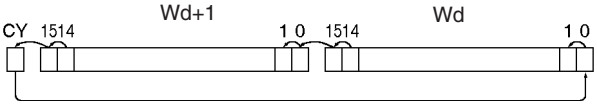
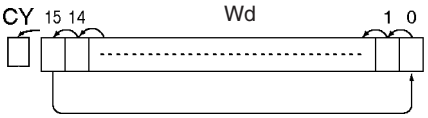
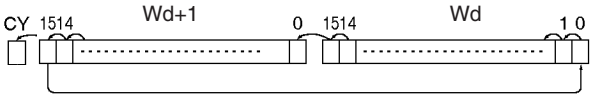
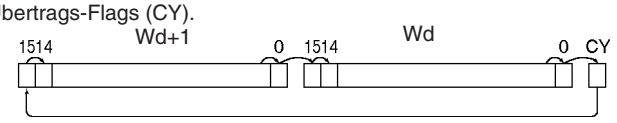
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
MOVE (Übertragen) MOV @MOV !MOV !@MOV 021	MOV(021) S D **S**: Quelle **D**: Ziel	Überträgt ein Datenwort an das angegebene Wort. 	Ausgang Erforderlich	315
DOUBLE MOVE (Doppelwort übertragen) MOVL @MOVL 498	MOVL(498) S D **S**: 1. Quellwort **D**: 1. Zielwort	Überträgt zwei Datenworte an die angegebenen Worte. 	Ausgang Erforderlich	318
MOVE NOT (Übertragen invertiert) MVN @MVN 022	MVN(022) S D **S**: Quelle **D**: Ziel	Überträgt das Komplement eines Datenworts an das angegebene Wort. 	Ausgang Erforderlich	317
DOUBLE MOVE NOT (Doppelwort übertragen invertiert) MVNL @MVNL 499	MVNL(499) S D **S**: 1. Quellwort **D**: 1. Zielwort	Überträgt das Komplement von zwei Datenworten an die angegebenen Worte. 	Ausgang Erforderlich	320
MOVE BIT (Bit übertragen) MOVB @MOVB 082	MOVB(082) S C D **S**: Quellwort oder -Daten **C**: Steuerwort **D**: Zielwort	Überträgt das spezifizierte Bit. 	Ausgang Erforderlich	321

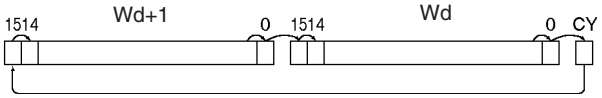
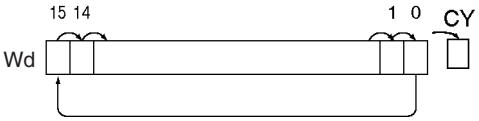
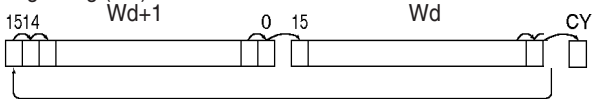
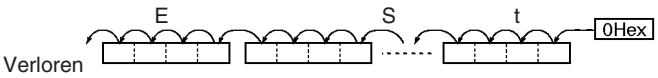
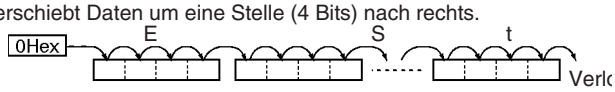
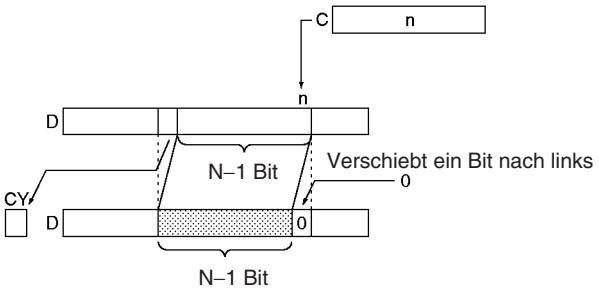
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
MOVE DIGIT (Stelle übertragen) MOVD @MOVD 083	MOVD(083) S C D **S**: Quellwort oder Daten **C**: Steuerwort **D**: Zielwort	Überträgt die spezifizierte(n) Stelle oder Stellen. (Jede Stelle besteht aus 4 Bits.) 	Ausgang Erforderlich	323
MULTIPLE BIT TRANSFER (Mehrere Bits übertragen) XFRB @XFRB 062	XFRB(062) C S D **C**: Steuerwort **S**: 1. Quellwort **D**: 1. Zielwort	Überträgt die spezifizierte Anzahl von aufeinander folgenden Bits. 	Ausgang Erforderlich	326
BLOCK TRANSFER (Blockübertragung) XFER @XFER 070	XFER(070) N S D **N**: Anzahl der Worte **S**: 1. Quellwort **D**: 1. Zielwort	Überträgt die spezifizierte Anzahl von aufeinander folgenden Worten. 	Ausgang Erforderlich	328
BLOCK SET (Block setzen) BSET @BSET 071	BSET(071) S St E **S**: Quellwort **St**: Anfangswort **E**: Endwort	Kopiert dasselbe Wort in einen Bereich aufeinander folgender Worte. 	Ausgang Erforderlich	331
DATA EXCHANGE (Datenaustausch) XCHG @XCHG 073	XCHG(073) E1 E2 **E1**: Erstes Austauschwort **E2**: Zweites Austauschwort	Tauscht den Inhalt der zwei spezifizierten Worte aus. 	Ausgang Erforderlich	333

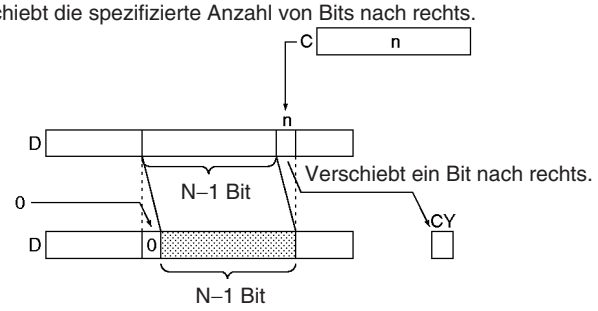
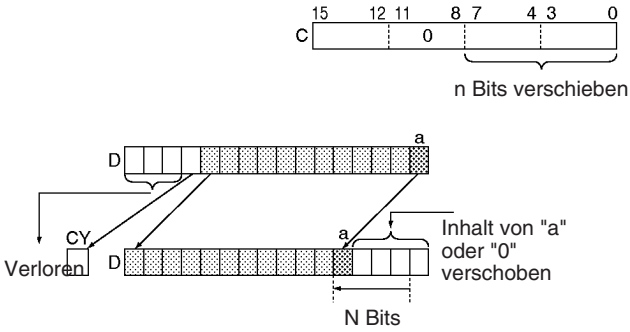
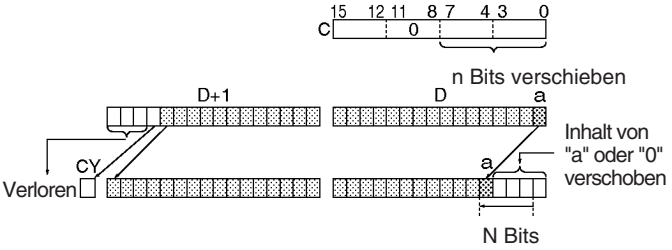
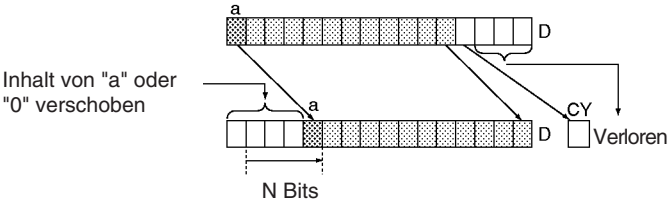
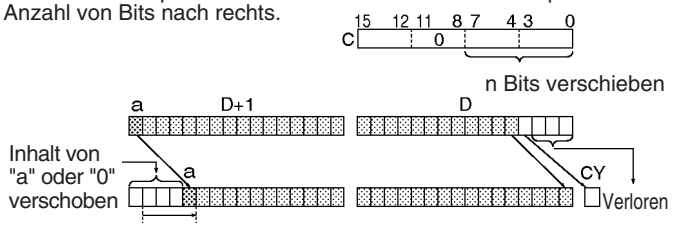
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DOUBLE DATA EXCHANGE (Doppelwort-Datenaustausch) XCGL @XCGL 562	XCGL(562) E1 E2 **E1**: Erstes Austauschwort **E2**: Zweites Austauschwort	Tauscht den Inhalt von zwei aufeinander folgenden Worten mit dem von zwei anderen aufeinander folgenden Worten aus. 	Ausgang Erforderlich	334
SINGLE WORD DISTRIBUTE (Einzelwortverteilung) DIST @DIST 080	DIST(080) S Bs Of **S**: Quellwort **Bs**: Ziel-Basisadresse **Of**: Offset	Überträgt das Quellwort an das durch Addition eines Offsetwerts zur Basisadresse errechnete Zielwort. 	Ausgang Erforderlich	336
DATA COLLECT (Daten sammeln) COLL @COLL 081	COLL(081) Bs Of D **Bs**: Quellen-Basisadresse **Of**: Offset **D**: Zielwort	Überträgt das Quellwort (errechnet durch Addition eines Offsetwerts zur Basisadresse) an das Zielwort. 	Ausgang Erforderlich	338
MOVE TO REGISTER (In Register übertragen) MOVR @MOVR 560	MOVR(560) S D **S**: Quelle (gewünschtes Wort oder Bit) **D**: Ziel (Indexregister)	Schreibt die interne E/A-Speicheradresse des spezifizierten Worts, Bits oder Zeitgeber/Zähler-Fertig-Bits in das spezifizierte Indexregister. (Verwenden Sie den Befehl MOVRW(561), um die interne E/A-Speicheradresse eines Zeitgeber/Zähler-Istwerts in ein Indexregister zu schreiben.) 	Ausgang Erforderlich	340
MOVE TIMER/ COUNTER PV TO REGISTER (Zeitgeber-/Zähler-Istwert in Register übertragen) MOVRW @MOVRW 561	MOVRW(561) S D **S**: Quelle (gewünschte Zeitgeber/Zähler-Nummer) **D**: Ziel (Indexregister)	Schreibt die interne E/A-Speicheradresse des spezifizierten Zeitgeber- oder Zähler-Istwerts in das spezifizierte Indexregister. (Verwenden Sie den Befehl MOVR(560), um die interne E/A-Speicheradresse eines Worts, Bits oder Zeitgeber/Zähler-Fertig-Bits in ein Indexregister zu schreiben.) 	Ausgang Erforderlich	342

2-2-7 Datenverschiebungsbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
SHIFT REGISTER (Schieberegister) SFT 010	Daten-eingang Schiebe-eingang Rück-setzeing SFT(010) St E **St**: Anfangswort **E**: Endwort	Steuert ein Schieberegister. 	Ausgang Erforderlich	345
REVERSIBLE SHIFT REGISTER (Umkehrbares Schieberegister) SFTR @SFTR 084	SFTR(084) C St E **C**: Steuerwort **St**: Anfangswort **E**: Endwort	Erstellt ein Schieberegister, in dem Daten nach rechts oder nach links verschoben werden. 	Ausgang Erforderlich	346
ASYNCHRONOUS SHIFT REGISTER (Asynchrones Schieberegister) ASFT @ASFT 017	ASFT(017) C St E **C**: Steuerwort **St**: Anfangswort **E**: Endwort	Verschiebt alle Wortdaten ungleich Null innerhalb des spezifizierten Wortbereichs entweder in Richtung St oder in Richtung E, wobei die 0000-hex-Wortdaten ersetzt werden. 	Ausgang Erforderlich	349
WORD SHIFT (Wortverschiebung) WSFT @WSFT 016	WSFT(016) S St E **S**: Quellwort **St**: Anfangswort **E**: Endwort	Verschiebt Daten zwischen St und E in Worteinheiten. 	Ausgang Erforderlich	352
ARITHMETIC SHIFT LEFT (Arithmetisches Verschieben nach links) ASL @ASL 025	ASL(025) Wd **Wd**: Wort	Verschiebt den Inhalt von Wd ein Bit nach links. 	Ausgang Erforderlich	354

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DOUBLE SHIFT LEFT (Doppelwort verschieben nach links) ASLL @ASLL 570	ASLL(570) Wd **Wd**: Wort	Verschiebt den Inhalt von Wd und Wd +1 ein Bit nach links. 	Ausgang Erforderlich	355
ARITHMETIC SHIFT RIGHT (Arithmetisches Verschieben nach rechts) ASR @ASR 026	ASR(026) Wd **Wd**: Wort	Verschiebt den Inhalt von Wd ein Bit nach rechts. 	Ausgang Erforderlich	357
DOUBLE SHIFT RIGHT (Doppelwort verschieben nach rechts) ASRL @ASRL 571	ASRL(571) Wd **Wd**: Wort	Verschiebt den Inhalt von Wd und Wd +1 ein Bit nach rechts. 	Ausgang Erforderlich	358
ROTATE LEFT (Rotation nach links) ROL @ROL 027	ROL(027) Wd **Wd**: Wort	Verschiebt alle Wd-Bits ein Bit nach links einschließlich des Übertrags-Flags (CY 	Ausgang Erforderlich	360
DOUBLE ROTATE LEFT (Doppelwort rotieren nach links) ROLL @ROLL 572	ROLL(572) Wd **Wd**: Wort	Verschiebt alle Bits von Wd und Wd +1 um ein Bit nach links einschließlich des Übertrags-Flags (CY). 	Ausgang Erforderlich	362
ROTATE LEFT WITHOUT CARRY (Rotation nach links ohne Übertrag) RLNC @RLNC 574	RLNC(574) Wd **Wd**: Wort	Verschiebt alle Bits von Wd ein Bit nach links ohne Übertrags-Flag (CY). 	Ausgang Erforderlich	367
DOUBLE ROTATE LEFT WITHOUT CARRY (Doppelwort rotieren nach links ohne Übertrag) RLNL @RLNL 576	RLNL(576) Wd **Wd**: Wort	Verschiebt alle Bits von Wd und Wd +1 um ein Bit nach links ohne das Übertrags-Flag (CY). 	Ausgang Erforderlich	369
ROTATE RIGHT (Rotation nach rechts) ROR @ROR 028	ROR(028) Wd **Wd**: Wort	Verschiebt alle Wd-Bits ein Bit nach rechts einschließlich des Übertrags-Flags (CY). 	Ausgang Erforderlich	364

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DOUBLE ROTATE RIGHT (Doppelwort rotieren nach rechts) RORL @RORL 573	RORL(573) Wd **Wd**: Wort	Verschiebt alle Bits von Wd und Wd+1 um ein Bit nach rechts einschließlich des Übertrags-Flags (CY). 	Ausgang Erforderlich	365
ROTATE RIGHT WITHOUT CARRY (Rotation nach rechts ohne Übertrag) RRNC @RRNC 575	RRNC(575) Wd **Wd**: Wort	Verschiebt alle Bits von Wd ein Bit nach rechts ohne das Übertrags-Flag (CY). Der Inhalt des niederwertigsten Bits (rechts) von Wd wird zum höchstwertigen Bit (links) und dem Übertrags-Flag (CY) verschoben. 	Ausgang Erforderlich	371
DOUBLE ROTATE RIGHT WITHOUT CARRY (Doppelwort rotieren nach rechts ohne Übertrag) RRNL @RRNL 577	RRNL(577) Wd **Wd**: Wort	Verschiebt alle Bits von Wd und Wd+1 um ein Bit nach rechts ohne das Übertrags-Flag (CY). Der Inhalt des niederwertigsten Bits (rechts) von Wd+1 wird zum höchstwertigen Bit (links) von Wd und dem Übertrags-Flag (CY) verschoben. 	Ausgang Erforderlich	372
ONE DIGIT SHIFT LEFT (Eine Stelle nach links verschieben) SLD @SLD 074	SLD(074) St E **St**: Anfangswort **E**: Endwort	Verschiebt Daten um eine Stelle (4 Bits) nach links. 	Ausgang Erforderlich	374
ONE DIGIT SHIFT RIGHT (Eine Stelle nach rechts verschieben) SRD @SRD 075	SRD(075) St E **St**: Anfangswort **E**: Endwort	Verschiebt Daten um eine Stelle (4 Bits) nach rechts. 	Ausgang Erforderlich	376
SHIFT N-BIT DATA LEFT (N Bits Daten nach links verschieben) NSFL @NSFL 578	NSFL(578) D C N **D**: Anfangswort für das Verschieben **C**: Anfangsbit **N**: Verschiebedatenlänge	Verschiebt die spezifizierte Anzahl von Bits nach links. 	Ausgang Erforderlich	377

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
SHIFT N-BIT DATA RIGHT (N Bits Daten nach rechts verschieben) NSFR @NSFR 579	NSFR(579) D C N **D**: Anfangswort für das Verschieben **C**: Anfangsbit **N**: Verschiebe-datenlänge	Verschiebt die spezifizierte Anzahl von Bits nach rechts. 	Ausgang Erforderlich	379
SHIFT N-BITS LEFT (N Bits nach links verschieben) NASL @NASL 580	NASL(580) D C **D**: Verschiebewort **C**: Steuerwort	Verschiebt die spezifizierten 16 Bits Wortdaten um die spezifizierte Anzahl von Bits nach links. 	Ausgang Erforderlich	381
DOUBLE SHIFT N-BITS LEFT (Doppelwort N Bits nach links verschieben) NSLL @NSLL 582	NSLL(582) D C **D**: Verschiebewort **C**: Steuerwort	Verschiebt die spezifizierten 32 Bits Wortdaten um die spezifizierte Anzahl von Bits nach links. 	Ausgang Erforderlich	384
SHIFT N-BITS RIGHT (N Bits nach rechts verschieben) NASR @NASR 581	NASR(581) D C **D**: Verschiebewort **C**: Steuerwort	Verschiebt die spezifizierten 16 Bits Wortdaten um die spezifizierte Anzahl von Bits nach rechts. 	Ausgang Erforderlich	387
DOUBLE SHIFT N-BITS RIGHT (Doppelwort N Bits nach rechts verschieben) NSRL @NSRL 583	NSRL(583) D C **D**: Verschiebewort **C**: Steuerwort	Verschiebt die spezifizierten 32 Bits Wortdaten um die spezifizierte Anzahl von Bits nach rechts. 	Ausgang Erforderlich	389

2-2-8 Inkrementier-/Dekrementierbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
INCREMENT BINARY (Binär-Doppelwort inkrementieren) ++ @++ 590	++(590) Wd **Wd**: Wort	Inkrementiert den 4-stelligen Hexadezimal-Inhalt des spezifizierten Worts um 1. Wd +1 → Wd	Ausgang Erforderlich	393
DOUBLE INCREMENT BINARY (Binär-Doppelwort inkrementieren) ++L @++L 591	++L(591) Wd **Wd**: Wort	Inkrementiert den 8-stelligen Hexadezimal-Inhalt der spezifizierten Worte um 1. Wd+1 Wd +1 → Wd+1 Wd	Ausgang Erforderlich	395
DECREMENT BINARY (Binärwert dekrementieren) – – @– – 592	– –(592) Wd **Wd**: Wort	Dekrementiert den 4-stelligen Hexadezimal-Inhalt des spezifizierten Worts um 1. Wd –1 → Wd	Ausgang Erforderlich	397
DOUBLE DECREMENT BINARY (Binär-Doppelwort dekrementieren) – –L @– –L 593	– – L(593) Wd **Wd**: Erstes Wort	Dekrementiert den 8-stelligen Hexadezimal-Inhalt der spezifizierten Worte um 1. Wd+1 Wd –1 → Wd+1 Wd	Ausgang Erforderlich	399
INCREMENT BCD (BCD-Wert inkrementieren) ++B @++B 594	++B(594) Wd **Wd**: Wort	Inkrementiert den 4-stelligen BCD-Inhalt des spezifizierten Worts um 1. Wd +1 → Wd	Ausgang Erforderlich	401
DOUBLE INCREMENT BCD (BCD-Doppelwort inkrementieren) ++BL @++BL 595	++BL(595) Wd **Wd**: Erstes Wort	Inkrementiert den 8-stelligen BCD-Inhalt der spezifizierten Worte um 1. Wd+1 Wd +1 → Wd+1 Wd	Ausgang Erforderlich	403
DECREMENT BCD (BCD-Wert dekrementieren) – –B @– –B 596	– –B(596) Wd **Wd**: Wort	Dekrementiert den 4-stelligen BCD-Inhalt des spezifizierten Worts um 1. Wd –1 → Wd	Ausgang Erforderlich	405
DOUBLE DECREMENT BCD (BCD-Doppelwort dekrementieren) – –BL @– –BL 597	– –BL(597) Wd **Wd**: Erstes Wort	Dekrementiert den 8-stelligen BCD-Inhalt der spezifizierten Worte um 1. Wd+1 Wd –1 → Wd+1 Wd	Ausgang Erforderlich	407

2-2-9 Arithmetische Befehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
SIGNED BINARY ADD WITHOUT CARRY (Vorzeichenbehaftete Binär-Addition ohne Übertrag) + @+ 400	+(400) Au Ad R **Au**: 1. Summand **Ad**: 2. Summand **R**: Ergebniswort	Addiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten. Au (Binär mit Vorzeichen) + Ad (Binär mit Vorzeichen) CY wird auf EIN gesetzt, wenn ein Übertrag vorhanden ist. CY R (Binär mit Vorzeichen)	Ausgang Erforderlich	410
DOUBLE SIGNED BINARY ADD WITHOUT CARRY (Vorzeichenbehaftete Doppelwort-Binär-Addition ohne Übertrag) +L @+L 401	+L(401) Au Ad R **Au**: 1. Summand, erstes Wort **Ad**: 2. Summand, erstes Wort **R**: Erstes Ergebniswort	Addiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten. Au+1 Au (Binär mit Vorzeichen) + Ad+1 Ad (Binär mit Vorzeichen) CY wird auf EIN gesetzt, wenn ein Übertrag vorhanden ist. CY R+1 R (Binär mit Vorzeichen)	Ausgang Erforderlich	412
SIGNED BINARY ADD WITH CARRY (Vorzeichenbehaftete Binär-Addition mit Übertrag) +C @+C 402	+C(402) Au Ad R **Au**: 1. Summand **Ad**: 2. Summand **R**: Ergebniswort	Addiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten mit Übertrags-Flag (CY). Au (Binär mit Vorzeichen) Ad (Binär mit Vorzeichen) + CY CY wird auf EIN gesetzt, wenn ein Übertrag vorhanden ist. CY R (Binär mit Vorzeichen)	Ausgang Erforderlich	414
DOUBLE SIGNED BINARY ADD WITH CARRY (Vorzeichenbehaftete Doppelwort-Binär-Addition mit Übertrag) +CL @+CL 403	+CL(403) Au Ad R **Au**: 1. Summand, erstes Wort **Ad**: 2. Summand, erstes Wort **R**: Erstes Ergebniswort	Addiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten mit Übertrags-Flag (CY). Au+1 Au (Binär mit Vorzeichen) Ad+1 Ad (Binär mit Vorzeichen) CY wird auf EIN gesetzt, wenn ein Übertrag vorhanden ist. + CY CY R+1 R (Binär mit Vorzeichen)	Ausgang Erforderlich	416
BCD ADD WITHOUT CARRY (BCD-Addition ohne Übertrag) +B @+B 404	+B(404) Au Ad R **Au**: 1. Summand **Ad**: 2. Summand **R**: Ergebniswort	Addiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten. Au (BCD) + Ad (BCD) CY wird auf EIN gesetzt, wenn ein Übertrag vorhanden ist. CY R (BCD)	Ausgang Erforderlich	418

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
DOUBLE BCD ADD WITHOUT CARRY (Doppelwort-BCD-Addition ohne Übertrag) +BL @+BL 405	+BL(405) Au Ad R **Au**: 1. Summand, erstes Wort **Ad**: 2. Summand, erstes Wort **R**: Erstes Ergebniswort	Addiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten. Au+1 Au (BCD) + Ad+1 Ad (BCD) CY R+1 R (BCD) CY wird auf EIN gesetzt, wenn ein Übertrag vorhanden ist.	Ausgang Erforderlich	419
BCD ADD WITH CARRY (BCD-Addition mit Übertrag) +BC @+BC 406	+BC(406) Au Ad R **Au**: 1. Summand **Ad**: 2. Summand **R**: Ergebniswort	Addiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten mit Übertrags-Flag (CY). Au (BCD) Ad (BCD) + CY CY wird auf EIN gesetzt, wenn ein Übertrag vorhanden ist. CY R (BCD)	Ausgang Erforderlich	421
DOUBLE BCD ADD WITH CARRY (Doppelwort-BCD-Addition mit Übertrag) +BCL @+BCL 407	+BCL(407) Au Ad R **Au**: 1. Summand, erstes Wort **Ad**: 2. Summand, erstes Wort **R**: Erstes Ergebniswort	Addiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten mit Übertrags-Flag (CY). Au+1 Au (BCD) Ad+1 Ad (BCD) + CY CY wird auf EIN gesetzt, wenn ein Übertrag vorhanden ist. CY R+1 R (BCD)	Ausgang Erforderlich	423
SIGNED BINARY SUBTRACT WITHOUT CARRY (Vorzeichenbehaftete Binär-Subtraktion ohne Übertrag) – @– 410	–(410) Mi Su R **Mi**: Minuend-Wort **Su**: Subtrahend-Wort **R**: Ergebniswort	Subtrahiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten. Mi (Binär mit Vorzeichen) – Su (Binär mit Vorzeichen) CY wird auf EIN gesetzt, wenn ein negativer Übertrag vorhanden ist. CY R (Binär mit Vorzeichen)	Ausgang Erforderlich	424
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY (Vorzeichenbehaftete Doppelwort-Binär-Subtraktion ohne Übertrag) –L @–L 411	– L(411) Mi Su R **Mi**: Minuend-Wort **Su**: Subtrahend-Wort **R**: Ergebniswort	Subtrahiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten. Mi+1 Mi (Binär mit Vorzeichen) – Su+1 Su (Binär mit Vorzeichen) CY wird auf EIN gesetzt, wenn ein negativer Übertrag vorhanden ist. CY R+1 R (Binär mit Vorzeichen)	Ausgang Erforderlich	426

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
SIGNED BINARY SUBTRACT WITH CARRY (Vorzeichenbehaftete Binär-Subtraktion mit Übertrag) −C @−C 412	−C(412) Mi Su R **Mi**: Minuend-Wort **Su**: Subtrahend-Wort **R**: Ergebniswort	Subtrahiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten mit Übertrags-Flag (CY). Mi (Binär mit Vorzeichen) Su (Binär mit Vorzeichen) − CY CY wird auf EIN gesetzt, wenn ein negativer Übertrag vorhanden ist. CY R (Binär mit Vorzeichen)	Ausgang Erforderlich	430
DOUBLE SIGNED BINARY WITH CARRY (Vorzeichenbehaftete Doppelwort-Binär-Subtraktion mit Übertrag) −CL @−CL 413	−CL(413) Mi Su R **Mi**: Minuend-Wort **Su**: Subtrahend-Wort **R**: Ergebniswort	Subtrahiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten mit Übertrags-Flag (CY). Mi+1 Mi (Binär mit Vorzeichen) Su+1 Su (Binär mit Vorzeichen) CY wird auf EIN gesetzt, wenn ein negativer Übertrag vorhanden ist. − CY CY R+1 R (Binär mit Vorzeichen)	Ausgang Erforderlich	432
BCD SUBTRACT WITHOUT CARRY (BCD-Subtraktion ohne Übertrag) −B @−B 414	−B(414) Mi Su R **Mi**: Minuend-Wort **Su**: Subtrahend-Wort **R**: Ergebniswort	Subtrahiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten. Mi (BCD) − Su (BCD) CY wird auf EIN gesetzt, wenn ein Übertrag vorhanden ist. CY R (BCD)	Ausgang Erforderlich	435
DOUBLE BCD SUBTRACT WITHOUT CARRY (Doppelwort-BCD-Subtraktion ohne Übertrag) −BL @−BL 415	−BL(415) Mi Su R **Mi**: Erstes Minuend-Wort **Su**: Erstes Subtrahend-Wort **R**: Erstes Ergebniswort	Subtrahiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten. Mi+1 Mi (BCD) − Su+1 Su (BCD) CY wird auf EIN gesetzt, wenn ein negativer Übertrag vorhanden ist. CY R+1 R (BCD)	Ausgang Erforderlich	436
BCD SUBTRACT WITH CARRY (BCD-Subtraktion mit Übertrag) −BC @−BC 416	−BC(416) Mi Su R **Mi**: Minuend-Wort **Su**: Subtrahend-Wort **R**: Ergebniswort	Subtrahiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten mit Übertrags-Flag (CY). Mi (BCD) Su (BCD) − CY CY wird auf EIN gesetzt, wenn ein negativer Übertrag vorhanden ist. CY R (BCD)	Ausgang Erforderlich	440

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
DOUBLE BCD SUBTRACT WITH CARRY (Doppelwort-BCD-Subtraktion mit Übertrag) −BCL @−BCL 417	−BCL(417) Mi Su R **Mi**: Erstes Minuend-Wort **Su**: Erstes Subtrahend-Wort **R**: Erstes Ergebniswort	Subtrahiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten mit Übertrags-Flag (CY). Mi+1 Mi (BCD) Su+1 Su (BCD) − CY CY wird auf EIN gesetzt, wenn ein negativer Übertrag vorhanden ist. CY R+1 R (BCD)	Ausgang Erforderlich	441
SIGNED BINARY MULTIPLY (Vorzeichenbehaftete Binär-Multiplikation) * @* 420	*(420) Md Mr R **Md**: Multiplikand-Wort **Mr**: Multiplikator-Wort **R**: Ergebniswort	Multipliziert 4-stellige vorzeichenbehaftete Hexadezimaldaten und/oder -konstanten. Md (Binär mit Vorzeichen) × Mr (Binär mit Vorzeichen) R+1 R (Binär mit Vorzeichen)	Ausgang Erforderlich	443
DOUBLE SIGNED BINARY MULTIPLY (Vorzeichenbehaftete Doppelwort-Binär-Multiplikation) *L @*L 421	*L(421) Md Mr R **Md**: Erstes Multiplikand-Wort **Mr**: Erstes Multiplikator-Wort **R**: Erstes Ergebniswort	Multipliziert 8-stellige vorzeichenbehaftete Hexadezimaldaten und/oder -konstanten. Md+1 Md (Binär mit Vorzeichen) × Mr+1 Mr (Binär mit Vorzeichen) R+3 R+2 R+1 R (Binär mit Vorzeichen)	Ausgang Erforderlich	445
UNSIGNED BINARY MULTIPLY (Vorzeichenlose Binär-Multiplikation) *U @*U 422	*U(422) Md Mr R **Md**: Multiplikand-Wort **Mr**: Multiplikator-Wort **R**: Ergebniswort	Multipliziert 4-stellige vorzeichenlose Hexadezimaldaten und/oder -konstanten. Md (Binär ohne Vorzeichen) × Mr (Binär ohne Vorzeichen) R+1 R (Binär ohne Vorzeichen)	Ausgang Erforderlich	447
DOUBLE UNSIGNED BINARY MULTIPLY (Vorzeichenlose Doppelwort-Binär-Multiplikation) *UL @*UL 423	*UL(423) Md Mr R **Md**: Erstes Multiplikand-Wort **Mr**: Erstes Multiplikator-Wort **R**: Erstes Ergebniswort	Multipliziert 8-stellige vorzeichenlose Hexadezimaldaten und/oder -konstanten. Md+1 Md (Binär ohne Vorzeichen) × Mr+1 Mr (Binär ohne Vorzeichen) R+3 R+2 R+1 R (Binär ohne Vorzeichen)	Ausgang Erforderlich	449

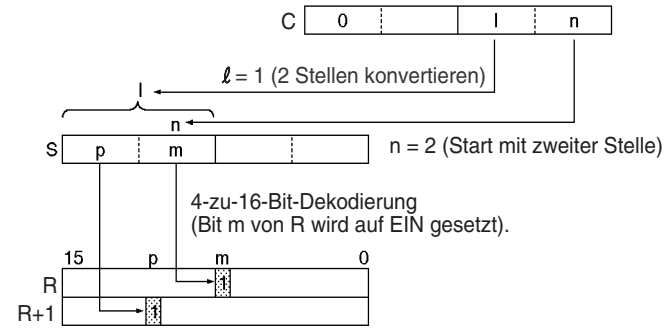
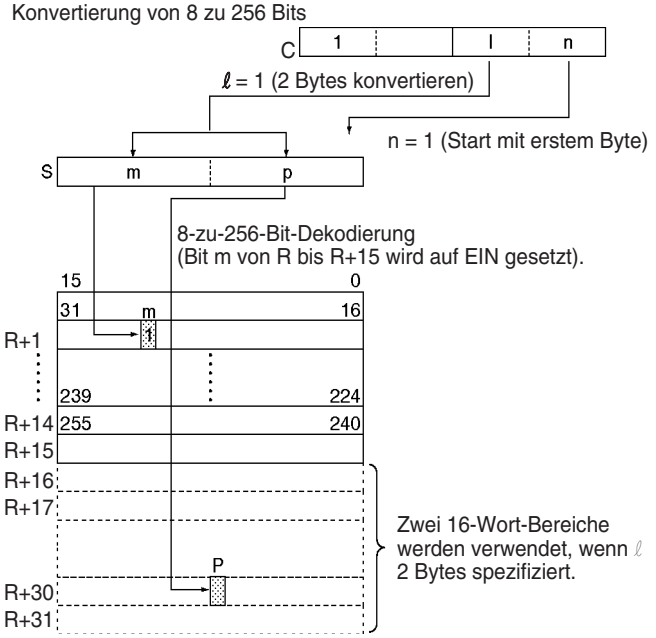
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
BCD MULTIPLY (BCD-Multiplikation) *B @*B 424	*B(424) Md Mr R **Md**: Multiplikand-Wort **Mr**: Multiplikator-Wort **R**: Ergebniswort	Multipliziert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten. Md (BCD) × Mr (BCD) R+1 | R (BCD)	Ausgang Erforderlich	450
DOUBLE BCD MULTIPLY (BCD-Doppelwort-Multiplikation) *BL @*BL 425	*BL(425) Md Mr R **Md**: Erstes Multiplikand-Wort **Mr**: Erstes Multiplikator-Wort **R**: Erstes Ergebniswort	Multipliziert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten. Md+1 | Md (BCD) × Mr+1 | Mr (BCD) R+3 | R+2 | R+1 | R (BCD)	Ausgang Erforderlich	452
SIGNED BINARY DIVIDE (Vorzeichenbehaftete Binär-Division) / @/ 430	/(430) Dd Dr R **Dd**: Dividend-Wort **Dr**: Divisor-Wort **R**: Ergebniswort	Dividiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten mit Vorzeichen. Dd (Binär mit Vorzeichen) ÷ Dr (Binär mit Vorzeichen) R+1 | R (Binär mit Vorzeichen) Rest Quotient	Ausgang Erforderlich	454
DOUBLE SIGNED BINARY DIVIDE (Vorzeichenbehaftete Doppelwort-Binär-Division) /L @/L 431	/L(431) Dd Dr R **Dd**: Erstes Dividend-Wort **Dr**: Erstes Divisor-Wort **R**: Erstes Ergebniswort	Dividiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten mit Vorzeichen. Dd+1 | Dd (Binär mit Vorzeic ÷ Dr+1 | Dr (Binär mit Vorzeic R+3 | R+2 | R+1 | R (Binär mit Vorzeic Rest Quotient	Ausgang Erforderlich	456
UNSIGNED BINARY DIVIDE (Vorzeichenlose Binär-Division) /U @/U 432	/U(432) Dd Dr R **Dd**: Dividend-Wort **Dr**: Divisor-Wort **R**: Ergebniswort	Dividiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten ohne Vorzeichen. Dd (Binär ohne Vorzeichen) ÷ Dr (Binär ohne Vorzeichen) R+1 | R (Binär ohne Vorzeichen) Rest Quotient	Ausgang Erforderlich	458

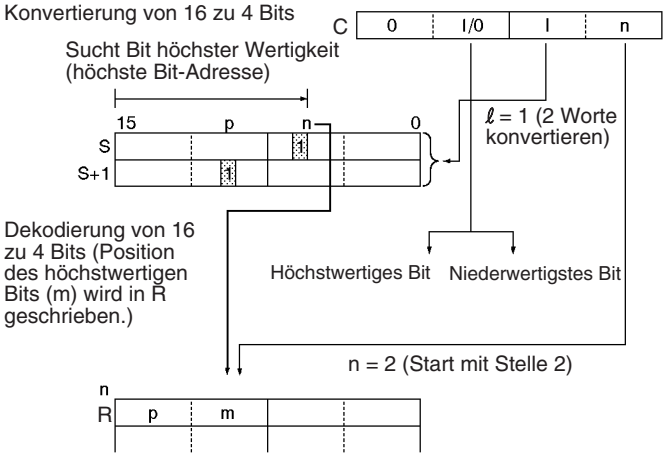
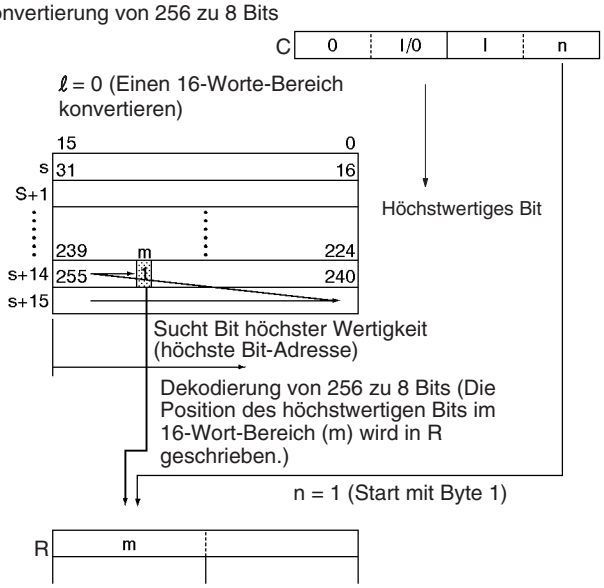
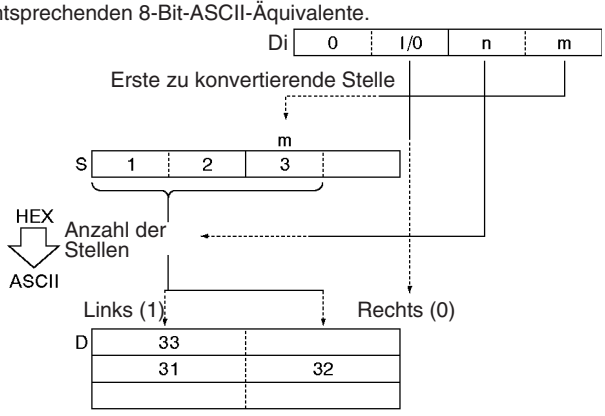
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DOUBLE UNSIGNED BINARY DIVIDE (Vorzeichenlose Doppelwort-Binär-Division) /UL @/UL 433	/UL(433) Dd Dr R **Dd**: Erstes Dividend-Wort **Dr**: Erstes Divisor-Wort **R**: Erstes Ergebniswort	Dividiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten ohne Vorzeichen. Dd+1 Dd (Binär ohne Vorzeichen) ÷ Dr+1 Dr (Binär ohne Vorzeichen) R+3 R+2 R+1 R (Binär ohne Vorzeichen) Rest Quotient	Ausgang Erforderlich	460
BCD DIVIDE (BCD-Wert-Division) /B @/B 434	/B(434) Dd Dr R **Dd**: Dividend-Wort **Dr**: Divisor-Wort **R**: Ergebniswort	Dividiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten. Dd (BCD) ÷ Dr (BCD) R+1 R (BCD) Rest Quotient	Ausgang Erforderlich	462
DOUBLE BCD DIVIDE (Doppelwort-BCD-Division) /BL @/BL 435	/BL(435) Dd Dr R **Dd**: Erstes Dividend-Wort **Dr**: Erstes Divisor-Wort **R**: Erstes Ergebniswort	Dividiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten. Dd+1 Dd (BCD) ÷ Dr+1 Dr (BCD) R+3 R+2 R+1 R (BCD) Rest Quotient	Ausgang Erforderlich	464

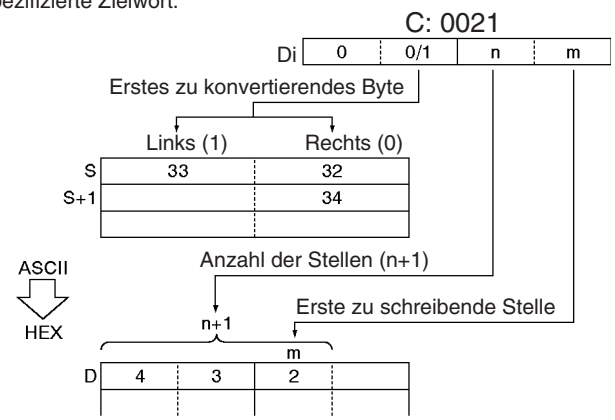
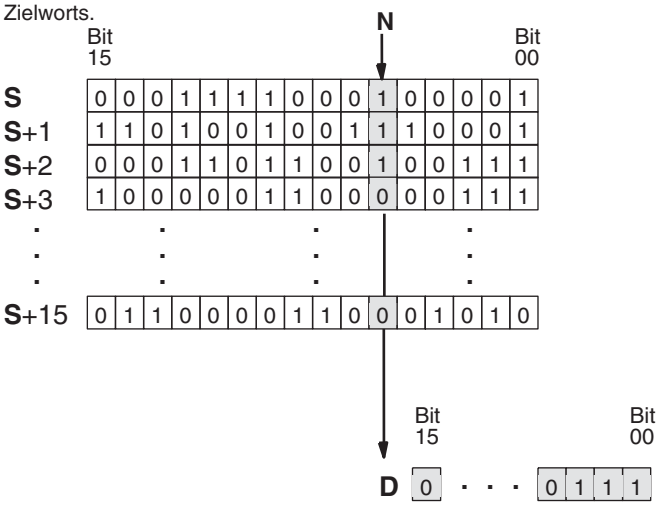
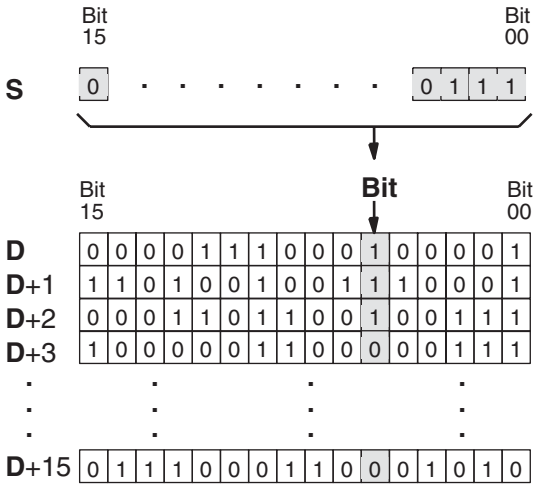
2-2-10 Konvertierungsbefehle

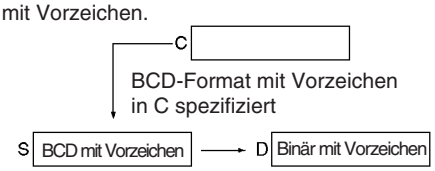
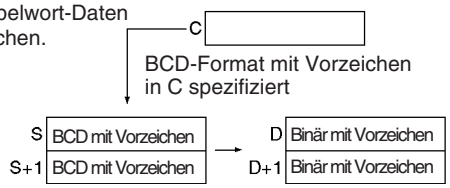
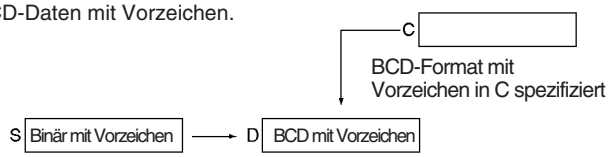
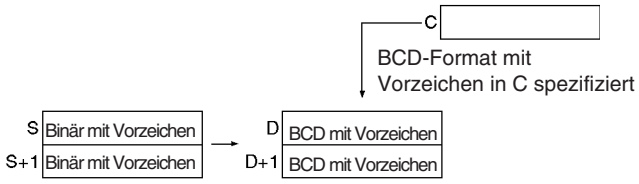
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
BCD-TO-BINARY (BCD zu Binär) BIN @BIN 023	BIN(023) S R **S**: Quellwort **R**: Ergebniswort	Konvertiert BCD-Daten in Binärdaten. S (BCD) → R (BIN)	Ausgang Erforderlich	466
DOUBLE BCD-TO-DOUBLE BINARY (BCD-Doppelwort zu Binär-Doppelwort) BINL @BINL 058	BINL(058) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Konvertiert 8-stellige BCD-Daten in 8-stellige Hexadezimaldaten (32 Bit binär). S (BCD) → R (BIN) S+1 (BCD) → R+1 (BIN)	Ausgang Erforderlich	467

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
BINARY-TO-BCD (Binär zu BCD) BCD @BCD 024	BCD(024) S R **S**: Quellwort **R**: Ergebniswort	Konvertiert ein Wort Binärdaten in ein Wort BCD-Daten. S (BIN) → R (BCD)	Ausgang Erforderlich	469
DOUBLE BINARY-TO-DOUBLE BCD (Binär-Doppelwort zu BCD-Doppelwort) BCDL @BCDL 059	BCDL(059) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Konvertiert 8-stellige Hexadezimaldaten (32 Bit binär) in 8-stellige BCD-Daten. S (BIN), S+1 (BIN) → R (BCD), R+1 (BCD)	Ausgang Erforderlich	470
2'S COMPLEMENT (2er-Komplement) NEG @NEG 160	NEG(160) S R **S**: Quellwort **R**: Ergebniswort	Berechnet das Zweierkomplement eines Worts von Hexadezimaldaten. Zweierkomplement (Komplement + 1) $\overline{(S)}$ → (R)	Ausgang Erforderlich	472
DOUBLE 2'S COMPLEMENT (Doppelwort-2er-Komplement) NEGL @NEGL 161	NEGL(161) S R **S**: Erstes Quellwor **R**: Erstes Ergebniswort	Berechnet das Zweierkomplement von Doppelwort-Hexadezimaldaten. Zweierkomplement (Komplement + 1) $\overline{(S+1, S)}$ → (R+1, R)	Ausgang Erforderlich	474
16-BIT TO 32-BIT SIGNED BINARY (16-Bit- in 32-Bit-Binärwerte mit Vorzeichen) SIGN @SIGN 600	SIGN(600) S R **S**: Quellwort **R**: Erstes Ergebniswort	Erweitert einen 16-Bit-Binärwert mit Vorzeichen in sein 32-Bit-Äquivalent. MSB S MSB = 1: FFFF hex MSB = 0: 0000 Hex D+1 D D = Inhalt von S	Ausgang Erforderlich	476

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DATA DECODER (Daten-Dekoder) MLPX @MLPX 076	MLPX(076) S C R **S**: Quellwort **C**: Steuerwort **R**: Erstes Ergebniswort	Liest den numerischen Wert der spezifizierten Stelle (bzw. des Bytes) im Quellwort, setzt das entsprechende Bit im Ergebniswort (bzw. dem 16-Wort-Bereich) auf EIN und setzt alle anderen Bits im Ergebniswort (bzw. dem 16-Wort-Bereich) auf AUS. Konvertierung von 4 zu 16 Bits  Konvertierung von 8 zu 256 Bits 	Ausgang Erforderlich	477

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
DATA ENCODER (Daten-Kodierer) DMPX @DMPX 077	DMPX(077) S R C **S**: Erstes Quellwort **R**: Ergebniswort **C**: Steuerwort	Sucht den Ort des ersten oder letzten auf EIN gesetzten Bits des Quellworts (bzw. des 16-Worte-Bereichs) und schreibt diesen Wert in die spezifizierte Stelle (bzw. das Byte) im Ergebniswort. Konvertierung von 16 zu 4 Bits  Konvertierung von 256 zu 8 Bits 	Ausgang Erforderlich	482
ASCII CONVERT (ASCII-Konvertierung) ASC @ASC 086	ASC(086) S Di D **S**: Quellwort **Di**: Stellen-spezifikator **D**: Erstes Zielwort	Konvertiert 4-Bit-Hexadezimalstellen im Quellwort in die entsprechenden 8-Bit-ASCII-Äquivalente. 	Ausgang Erforderlich	486

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
ASCII TO HEX (ASCII zu hex) HEX @HEX 162	HEX(162) S Di D **S**: Erstes Quellwort **Di**: Stellen-spezifikator **D**: Zielwort	Konvertiert bis zu 4 Bytes ASCII-Daten im Quellwort in die Hexadezimal-Äquivalente und schreibt diese Stellen in das spezifizierte Zielwort. 	Ausgang Erforderlich	490
COLUMN TO LINE (Spalte zu Zeile) LINE @LINE 063	LINE(063) S N D **S**: Erstes Quellwort **N**: Bitnummer **D**: Zielwort	Konvertiert eine Spalte Bits eines 16-Worte-Bereichs (die gleiche Bit-Nummer in 16 aufeinander folgenden Worten) in die 16 Bits des Zielworts. 	Ausgang Erforderlich	494
LINE TO COLUMN (Zeile zu Spalte) COLM @COLM 064	COLM(064) S D N **S**: Quellwort **D**: Erstes Zielwort **N**: Bitnummer	Konvertiert die 16 Bits eines Quellworts in eine Spalte mit Bits in einem 16-Worte-Bereich von Zielworten (die gleiche Bitnummer in 16 aufeinander folgenden Worten). 	Ausgang Erforderlich	496

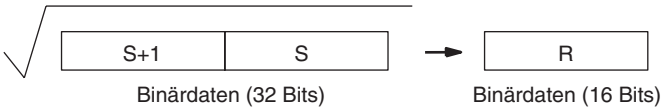
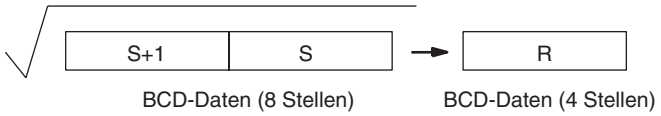
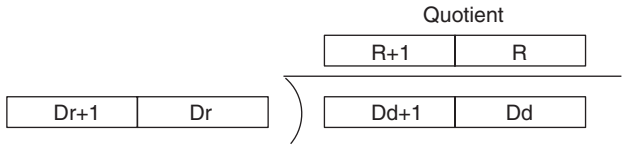
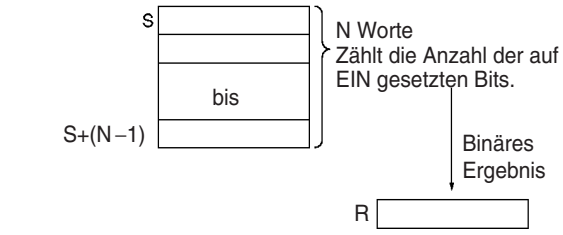
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
SIGNED BCD-TO-BINARY (BCD mit Vorzeichen zu Binär) BINS @BINS 470	BINS(470) C S D **C**: Steuerwort **S**: Quellwort **D**: Zielwort	Konvertiert ein Wort BCD-Daten mit Vorzeichen in ein Wort Binärdaten mit Vorzeichen. 	Ausgang Erforderlich	499
DOUBLE SIGNED BCD-TO-BINARY (BCD-Doppelwort mit Vorzeichen zu Binär) BISL @BISL 472	BISL(472) C S D **C**: Steuerwort **S**: Erstes Quellwort **D**: Erstes Zielwort	Konvertiert BCD-Doppelwort-Daten mit Vorzeichen zu Binär-Doppelwort-Daten mit Vorzeichen. 	Ausgang Erforderlich	502
SIGNED BINARY-TO-BCD (Vorzeichenbehafteter Binärwert zu BCD) BCDS @BCDS 471	BCDS(471) C S D **C**: Steuerwort **S**: Quellwort **D**: Zielwort	Konvertiert ein Wort Binärdaten mit Vorzeichen in ein Wort BCD-Daten mit Vorzeichen. 	Ausgang Erforderlich	505
DOUBLE SIGNED BINARY-TO-BCD (Vorzeichenbehaftetes Binär-Doppelwort zu BCD) BDSL @BDSL 473	BDSL(473) C S D **C**: Steuerwort **S**: Erstes Quellwort **D**: Erstes Zielwort	Konvertiert Binär-Doppelwort-Daten mit Vorzeichen zu BCD-Doppelwort-Daten mit Vorzeichen. 	Ausgang Erforderlich	507
GRAY CODE CONVERSION (Gray-Code konvertieren) GRY 474 (Nur CS/CJ-Serie ab Baugruppenversion 2.0, einschließlich CS1-H, CJ1-H und CJ1M CPU-Baugruppen ab Lot-Nummer 030201)	GRY(474) C S D **C**: Steuerwort **S**: Quellwort **D**: Erstes Zielwort	Konvertiert Gray-Code-Daten im spezifizierten Wort in Binär-, BCD- oder Winkeldaten (°) mit der spezifizierten Auflösung.	Ausgang Erforderlich	511

2-2-11 Logikbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
LOGICAL AND (Logisches UND) ANDW @ANDW 034	ANDW(034) I_1 I_2 R I_1: Eingang 1 I_2: Eingang 2 **R**: Ergebniswort	Bildet eine logische AND-Verknüpfung zwischen entsprechenden Bits in Einzelworten von Wortdaten und/oder Konstanten. **$I_1 . I_2 \rightarrow$ R** I_1 I_2 R: 1 1 1; 1 0 0; 0 1 0; 0 0 0	Ausgang Erforderlich	517
DOUBLE LOGICAL AND (Logisches Doppelwort-UND) ANDL @ANDL 610	ANDL(610) I_1 I_2 R I_1: Eingang 1 I_2: Eingang 2 **R**: Ergebniswort	Bildet eine logische AND-Verknüpfung zwischen entsprechenden Bits in Doppelworten von Wortdaten und/oder Konstanten. **$(I_1, I_1+1) . (I_2, I_2+1) \rightarrow (R, R+1)$** I_1, I_1+1 I_2, I_2+1 R, R+1: 1 1 1; 1 0 0; 0 1 0; 0 0 0	Ausgang Erforderlich	519
LOGICAL OR (Logisches ODER) ORW @ORW 035	ORW(035) I_1 I_2 R **I1:** Eingang 1 **I2:** Eingang 2 **R:** Ergebniswort	Bildet eine logische OR-Verknüpfung zwischen entsprechenden Bits in Einzelworten von Wortdaten und/oder Konstanten. **$I_1 + I_2 \rightarrow$ R** I_1 I_2 R: 1 1 1; 1 0 1; 0 1 1; 0 0 0	Ausgang Erforderlich	520
DOUBLE LOGICAL OR (Logisches Doppelwort-ODER) ORWL @ORWL 611	ORWL(611) I_1 I_2 R I_1: Eingang 1 I_2: Eingang 2 **R**: Ergebniswort	Bildet eine logische OR-Verknüpfung zwischen entsprechenden Bits in Doppelworten von Wortdaten und/oder Konstanten. **$(I_1, I_1+1) + (I_2, I_2+1) \rightarrow (R, R+1)$** I_1, I_1+1 I_2, I_2+1 R, R+1: 1 1 1; 1 0 1; 0 1 1; 0 0 0	Ausgang Erforderlich	522
EXCLUSIVE OR (Exklusives ODER) XORW @XORW 036	XORW(036) I_1 I_2 R I_1: Eingang 1 I_2: Eingang 2 **R**: Ergebniswort	Bildet eine logische Exklusiv-OR-Verknüpfung zwischen entsprechenden Bits in Einzelworten von Wortdaten und/oder Konstanten. **$I_1 . \overline{I_2} + \overline{I_1} . I_2 \rightarrow$ R** I_1 I_2 R: 1 1 0; 1 0 1; 0 1 1; 0 0 0	Ausgang Erforderlich	524

<table>
<tr><th>Befehl
AWL
Code</th><th>Symbol/Operand</th><th>Funktion</th><th>Position
Ausführungs-
bedingung</th><th>Seite</th></tr>
<tr><td>DOUBLE EXCLUSIVE OR (Exklusives Doppelwort-ODER)
XORL
@XORL
612</td><td>XORL(612)
I_1
I_2
R
I_1: Eingang 1
I_2: Eingang 2
R: Ergebniswort</td><td>Bildet eine logische exklusive OR-Verknüpfung zwischen entsprechenden Bits in Doppelworten von Wortdaten und/oder Konstanten.
$(I_1,I_1+1).\ (\overline{I_2,I_2+1}) + (\overline{I_1,I_1+1}).\ (I_2,I_2+1) \rightarrow (R,R+1)$
<table><tr><th>I_1,I_1+1</th><th>I_2,I_2+1</th><th>R, R+1</th></tr><tr><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table></td><td>Ausgang Erforderlich</td><td>526</td></tr>
<tr><td>EXCLUSIVE NOR (Exklusives ODER invertiert)
XNRW
@XNRW
037</td><td>XNRW(037)
I_1
I_2
R
I_1: Eingang 1
I_2: Eingang 2
R: Ergebniswort</td><td>Bildet eine logische exklusive NOR-Verknüpfung zwischen entsprechenden Einzelworten von Wortdaten und/oder Konstanten.
$I_1.I_2+\overline{I_1}.\overline{I_2} \rightarrow R$
<table><tr><th>I_1</th><th>I_2</th><th>R</th></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td></tr></table></td><td>Ausgang Erforderlich</td><td>528</td></tr>
<tr><td>DOUBLE EXCLUSIVE NOR (Exklusives Doppelwort-ODER invertiert)
XNRL
@XNRL
613</td><td>XNRL(613)
I_1
I_2
R
I_1: Eingang 1
I_2: Eingang 2
R: Erstes Ergebniswort</td><td>Bildet eine logische exklusive NOR-Verknüpfung zwischen entsprechenden Bits in Doppelworten von Wortdaten und/oder Konstanten.
$(I_1,I_1+1).\ (I_2,I_2+1) + (\overline{I_1,I_1+1}).\ (\overline{I_2,I_2+1}) \rightarrow (R,R+1)$
<table><tr><th>I_1,I_1+1</th><th>I_2,I_2+1</th><th>R, R+1</th></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td></tr></table></td><td>Ausgang Erforderlich</td><td>529</td></tr>
<tr><td>COMPLEMENT (Komplement)
COM
@COM
029</td><td>COM(029)
Wd
Wd: Wort</td><td>Setzt in alle auf EIN gesetzten Bits in Wd auf AUS und alle auf AUS gesetzten Bits auf EIN.
$\overline{Wd} \rightarrow Wd$: 1 → 0 und 0 → 1</td><td>Ausgang Erforderlich</td><td>531</td></tr>
<tr><td>DOUBLE COMPLEMENT (Doppelwort-Komplement)
COML
@COML
614</td><td>COML(614)
Wd
Wd: Wort</td><td>Setzt in Wd und Wd+1 alle auf EIN gesetzten Bits auf AUS und alle auf AUS gesetzten Bits auf EIN.
$\overline{(Wd+1, Wd)} \rightarrow (Wd+1, Wd)$</td><td>Ausgang Erforderlich</td><td>533</td></tr>
</table>

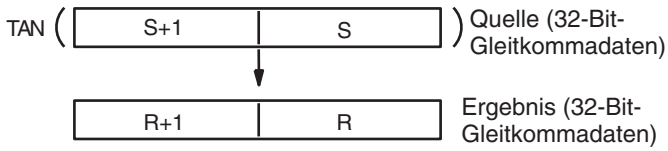
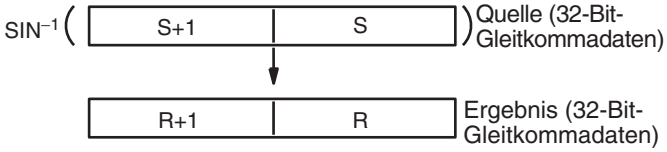
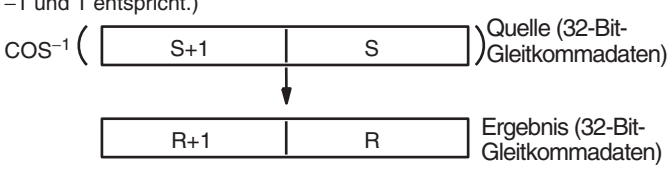
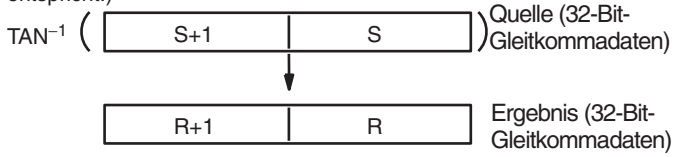
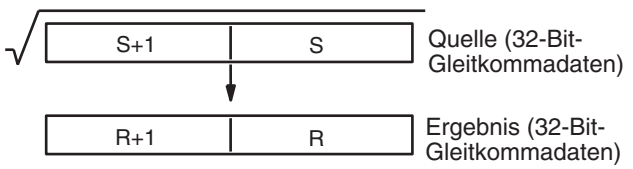
2-2-12 Spezielle arithmetische Befehle

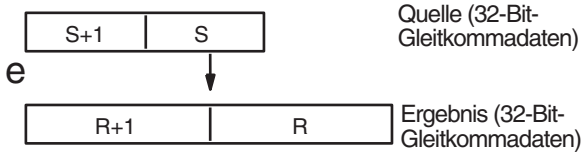
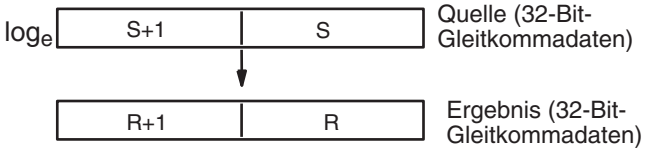
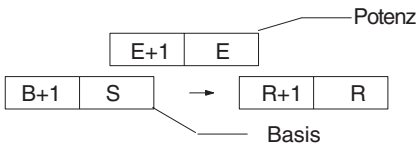
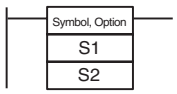
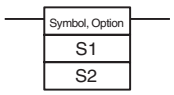
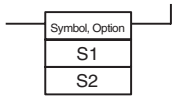
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
BINARY ROOT (Binär-Quadratwurzel) ROTB @ROTB 620	ROTB(620) S R **S**: Erstes Quellwort **R**: Ergebniswort	Berechnet die Quadratwurzel aus dem 32-Bit-Binärinhalt der spezifizierten Worte und gibt den ganzzahligen Teil des Ergebnisses in das spezifizierte Ergebniswort aus. 	Ausgang Erforderlich	534
BCD SQUARE ROOT (BCD-Quadratwurzel) ROOT @ROOT 072	ROOT(072) S R **S**: Erstes Quellwort **R**: Ergebniswort	Berechnet die Quadratwurzel aus einer 8-stelligen BCD-Zahl und gibt den ganzzahligen Teil des Ergebnisses in das spezifizierte Ergebniswort aus. 	Ausgang Erforderlich	536
ARITHMETIC PROCESS (Arithmetische Verarbeitung) APR @APR 069	APR(069) C S R **C**: Steuerwort **S**: Quelldaten **R**: Ergebniswort	Berechnet den Sinus, Kosinus oder eine lineare Extrapolation der Quelldaten. Mit der linearen Extrapolation können alle Beziehungen zwischen X und Y durch Geradensegmente angenähert werden.	Ausgang Erforderlich	540
FLOATING POINT DIVIDE (Gleitkomma-Division) FDIV @FDIV 079	FDIV(079) Dd Dr R **Dd**: Erstes Dividend-Wort **Dr**: Erstes Divisor-Wort **R**: Erstes Ergebniswort	Dividiert eine 7-stellige Gleitkommazahl durch eine andere. Die Gleitkommazahlen werden in wissenschaftlicher Schreibweise ausgedrückt (7-stellige Mantisse und 1-stelliger Exponent). 	Ausgang Erforderlich	552
BIT COUNTER (Bitzähler) BCNT @BCNT 067	BCNT(067) N S R **N**: Anzahl der Worte **S**: Erstes Quellwort **R**: Ergebniswort	Zählt die Gesamtanzahl der auf EIN gesetzten Bits in den spezifizierten Worten. 	Ausgang Erforderlich	556

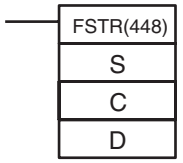
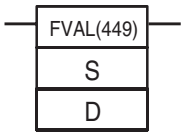
2-2-13 Gleitkomma-Arithmetikbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
FLOATING TO 16-BIT (Gleitkomma zu 16 Bit Binär) FIX @FIX 450	FIX(450) S R **S**: Erstes Quellwort **R**: Ergebniswort	Konvertiert einen 32-Bit-Gleitkommawert in 16-Bit-Binärdaten mit Vorzeichen und gibt das Ergebnis in das spezifizierte Ergebniswort aus. S+1 | S — Gleitkommadaten (32 Bits) ↓ R — Binärdaten mit Vorzeichen (16 Bits	Ausgang Erforderlich	563
FLOATING TO 32-BIT (Gleitkomma zu 32 Bit Binär) FIXL @FIXL 451	FIXL(451) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Konvertiert einen 32-Bit-Gleitkommawert in 32-Bit-Binärdaten mit Vorzeichen und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. S+1 | S — Gleitkommadaten (32 Bits) ↓ R+1 | R — Binärdaten mit Vorzeichen (32 Bits)	Ausgang Erforderlich	565
16-BIT TO FLOATING (16 Bit Binär zu Gleitkomma) FLT @FLT 452	FLT(452) S R **S**: Quellwort **R**: Erstes Ergebniswort	Konvertiert einen 16-Bit-Binärwert mit Vorzeichen in 32-Bit-Gleitkommadaten und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. S — Binärdaten mit Vorzeichen (16 Bits) ↓ R+1 | R — Gleitkommadaten (32 Bits)	Ausgang Erforderlich	566
32-BIT TO FLOATING (32 Bit Binär zu Gleitkomma) FLTL @FLTL 453	FLTL(453) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Konvertiert einen 32-Bit-Binärwert mit Vorzeichen in 32-Bit-Gleitkommadaten und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. S+1 | S — Binärdaten mit Vorzeichen (32 Bits) ↓ R+1 | R — Gleitkommadaten (32 Bits)	Ausgang Erforderlich	568
FLOATING-POINT ADD (Gleitkomma-Addition) +F @+F 454	+F(454) Au Ad R **Au**: 1. Summand, erstes Wort **AD**: 2. Summand, erstes Wort **R**: Erstes Ergebniswort	Addiert zwei 32-Bit-Gleitkommazahlen und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. Au+1 | Au — 1. Summand (Gleitkommadaten, 32 Bits) + Ad+1 | Ad — 2. Summand (Gleitkommadaten, 32 Bits) R+1 | R — Ergebnis (Gleitkommadaten, 32 Bits)	Ausgang Erforderlich	570
FLOATING-POINT SUBTRACT (Gleitkomma-Subtraktion) –F @–F 455	F(455) Mi Su R **Mi**: Erstes Minuend-Wort **Su**: Erstes Subtrahend-Wort **R**: Erstes Ergebniswort	Subtrahiert eine 32-Bit-Gleitkommazahl von einer anderen und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. Mi+1 | Mi — Minuend (Gleitkommadaten, 32 Bits) – Su+1 | Su — Subtrahend (Gleitkommadaten, 32 Bits) R+1 | R — Ergebnis (Gleitkommadaten, 32 Bits)	Ausgang Erforderlich	572

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
FLOATING-POINT MULTIPLY (Gleitkomma-Multiplikation) *F @*F 456	*F(456) Md Mr R **Md**: Erstes Multiplikand-Wort **Mr**: Erstes Multiplikator-Wort **R**: Erstes Ergebniswort	Multipliziert zwei 32-Bit-Gleitkommazahlen miteinander und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. Md+1 | Md — Multiplikand (Gleitkommadaten, 32 Bits) × Mr+1 | Mr — Multiplikator (Gleitkommadaten, 32 Bits) R+1 | R — Ergebnis (Gleitkommadaten, 32 Bits)	Ausgang Erforderlich	574
FLOATING-POINT DIVIDE (Gleitkomma-Division) /F @/F 457	/F(457) Dd Dr R **Dd**: Erstes Dividend-Wort **Dr**: Erstes Divisor-Wort **R**: Erstes Ergebniswort	Dividiert eine 32-Bit-Gleitkommazahl durch eine andere und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. Dd+1 | Dd — Dividend (Gleitkommadaten, 32 Bits) ÷ Dr+1 | Dr — Divisor (Gleitkommadaten, 32 Bits) R+1 | R — Ergebnis (Gleitkommadaten, 32 Bits)	Ausgang Erforderlich	576
DEGREES TO RADIANS (Grad in Radianten) RAD @RAD 458	RAD(458) S R **S**: Erstes Quellwor **R**: Erstes Ergebniswort	Konvertiert eine 32-Bit-Gleitkommazahl von Grad zu Radianten und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. S+1 | S — Quelle (Grad, 32-Bit-Gleitkommadaten) R+1 | R — Ergebnis (Radianten, 32-Bit-Gleitkommadaten)	Ausgang Erforderlich	578
RADIANS TO DEGREES (Radianten in Grad) DEG @DEG 459	DEG(459) S R **S**: Erstes Quellwor **R**: Erstes Ergebniswort	Konvertiert eine 32-Bit-Gleitkommazahl von Radianten zu Grad und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. S+1 | S — Quelle (Radianten, 32-Bit-Gleitkommadaten) R+1 | R — Ergebnis (Grad, 32-Bit-Gleitkommadaten)	Ausgang Erforderlich	579
SINE (Sinus) SIN @SIN 460	SIN(460) S R **S**: Erstes Quellwo **R**: Erstes Ergebniswort	Berechnet den Sinus einer 32-Bit-Gleitkommazahl (in Radianten) und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. SIN (S+1 | S) Quelle (32-Bit-Gleitkommadaten) R+1 | R — Ergebnis (32-Bit-Gleitkommadaten)	Ausgang Erforderlich	581
COSINE (Kosinus) COS @COS 461	COS(461) S R **S**: Erstes Quellwor **R**: Erstes Ergebniswort	Berechnet den Kosinus einer 32-Bit-Gleitkommazahl (in Radianten) und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. COS(S+1 | S) Quelle (32-Bit-Gleitkommadaten) R+1 | R — Ergebnis (32-Bit-Gleitkommadaten)	Ausgang Erforderlich	583

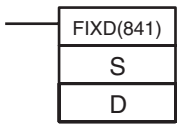
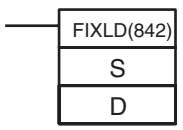
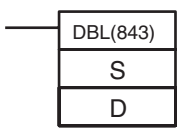
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
TANGENT (Tangens) TAN @TAN 462	TAN(462) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Tangens einer 32-Bit-Gleitkommazahl (in Radianten) und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. 	Ausgang Erforderlich	585
ARC SINE (Arkussinus) ASIN @ASIN 463	ASIN(463) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Arkussinus einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. (Die Arkussinusfunktion ist die Umkehrung der Sinusfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Sinuswert zwischen –1 und 1 entspricht.) 	Ausgang Erforderlich	587
ARC COSINE (Arkuskosinus) ACOS @ACOS 464	ACOS(464) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Arkuskosinus einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. (Die Arkuskosinusfunktion ist die Umkehrung der Kosinusfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Kosinuswert zwischen –1 und 1 entspricht.) 	Ausgang Erforderlich	589
ARC TANGENT (Arkustangens) ATAN @ATAN 465	ATAN(465) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Arkustangens einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. (Die Arkustangensfunktion ist die Umkehrung der Tangensfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Tangenswert entspricht.) 	Ausgang Erforderlich	591
SQUARE ROOT (Quadratwurzel) SQRT @SQRT 466	SQRT(466) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet die Quadratwurzel einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. 	Ausgang Erforderlich	593

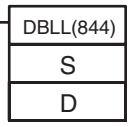
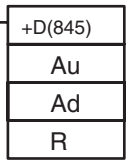
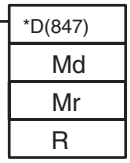
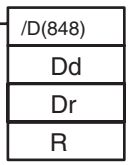
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
EXPONENT (Exponent) EXP @EXP 467	EXP(467) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet die natürliche Exponentialgröße (Basis e) einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten 	Ausgang Erforderlich	595
LOGARITHM (Logarithmus) LOG @LOG 468	LOG(468) S R **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den natürlichen Logarithmus (Basis e) einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. 	Ausgang Erforderlich	597
EXPONENTIAL POWER (Potenz-funktion) PWR @PWR 840	PWR(840) B E R **B**: Erstes Basiswort **E**: Erstes Exponenten-Wort	Berechnet die Potenz einer 32-Bit-Gleitkommazahl mit einer anderen 32-Bit-Gleitkommazahl. 	Ausgang Erforderlich	599
FLOATING SYMBOL COMPARISON (Gleitkomma-wert-Vergleich) (nur CS1-H, CJ1-H, CJ1M und CS1D) LD, AND oder OR + =F (329), <>F (330), <F (331), <=F (332), >F (333), oder >=F (334)	Verwenden bei LD:  Verwenden bei AND:  Verwenden bei OR:  **S1**: Vergleichsdaten 1 **S2**: Vergleichsdaten 2	Vergleicht die spezifizierten Daten oder Konstanten einfacher Genauigkeit (32 Bits) und erzeugt eine EIN-Ausführungsbedingung, wenn das Vergleichsergebnis wahr ist. Gleitkomma-Symbolvergleichsbefehle können auf drei Arten verwendet werden: LD (Load), AND und OR.	LD: Nicht erforderlich AND oder OR: Erforderlich	600

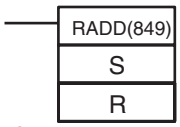
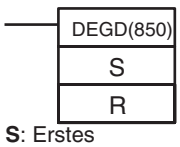
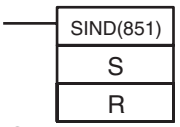
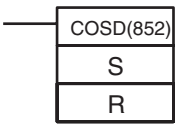
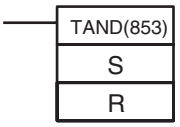
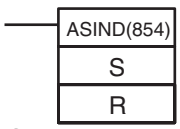
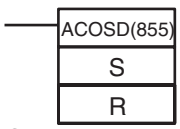
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
FLOATING-POINT TO ASCII (Gleitkommawert zu Zeichenkette) (nur CS1-H, CJ1-H, CJ1M und CS1D) FSTR @FSTR 448	 **S**: Erstes Quellwo **C**: Steuerwort **D**: Zielwort	Konvertiert die spezifizierten Gleitkommadaten einfacher Genauigkeit (32-Bit-Dezimalkomma- oder Exponentialformat) in eine Zeichenkette (ASCII) und gibt das Ergebnis in das Zielwort aus.	Ausgang Erforderlich	604
ASCII TO FLOATING-POINT (Zeichenkette zu Gleitkomma) (nur CS1-H, CJ1-H, CJ1M und CS1D) FVAL @FVAL 449	 **S:** Quellwort **D**: Erstes Zielwort	Konvertiert die spezifizierte Zeichenkette (ASCII) der Gleitkommadaten einfacher Genauigkeit (Dezimalkomma- oder Exponentialformat) zu 32-Bit-Gleitkommadaten einfacher Genauigkeit und gibt das Ergebnis in die Zielworte aus.	Ausgang Erforderlich	609

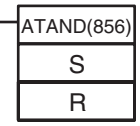
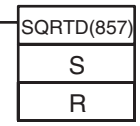
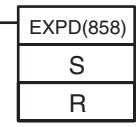
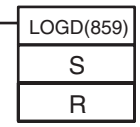
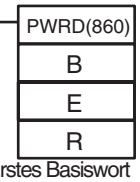
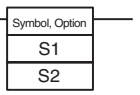
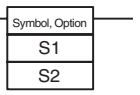
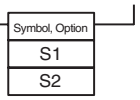
2-2-14 Gleitkomma-Arithmetikbefehle doppelter Genauigkeit

Die Gleitkomma-Arithmetikbefehle doppelter Genauigkeit werden nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

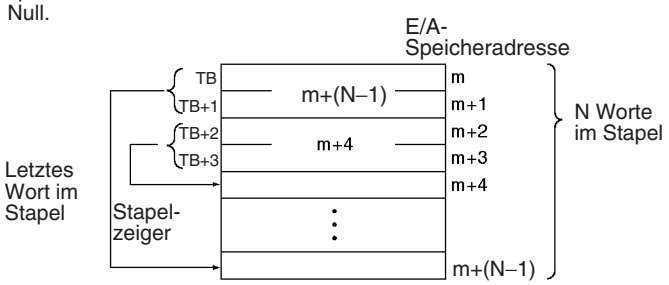
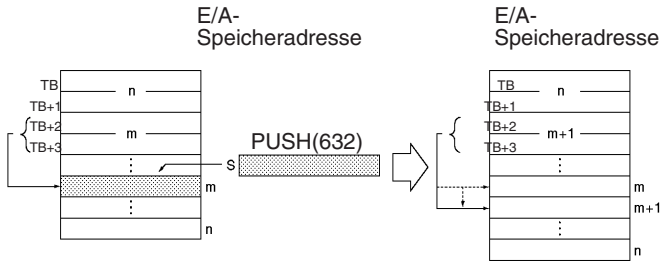
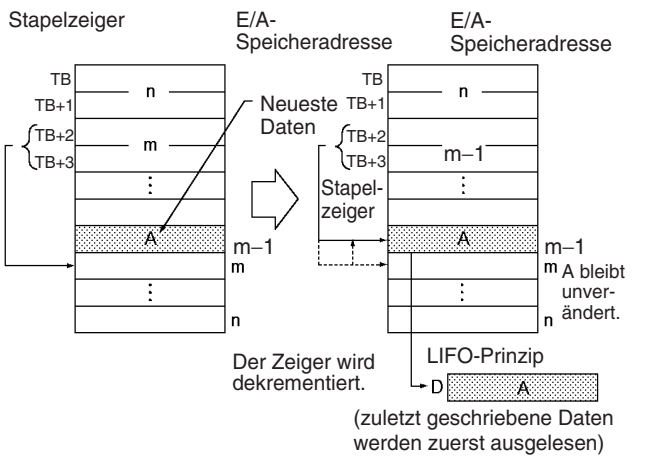
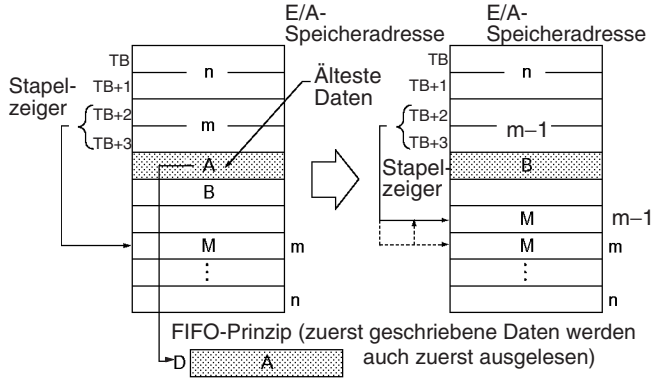
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DOUBLE FLOATING TO 16-BIT BINARY (Gleitkommadaten doppelter Genauigkeit zu 16 Bit binär) FIXD @FIXD 841	 **S**: Erstes Quellwort **D**: Zielwort	Konvertiert die spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) zu 16-Bit-Daten mit Vorzeichen und gibt das Ergebnis in das Zielwort aus.	Ausgang Erforderlich	620
DOUBLE FLOATING TO 32-BIT BINARY (Gleitkommadaten doppelter Genauigkeit zu 32 Bit binär) FIXLD @FIXLD 842	 **S**: Erstes Quellwort **D**: Erstes Zielwort	Konvertiert die spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) zu 32-Bit-Daten mit Vorzeichen und gibt das Ergebnis in die Zielworte aus.	Ausgang Erforderlich	621
16-BIT BINARY TO DOUBLE FLOATING (16 Bit binär zu Gleitkommadaten doppelter Genauigkeit) DBL @DBL 843	 **S**: Quellwort **D**: Erstes Zielwort	Konvertiert die spezifizierten 16-Bit-Binärdaten mit Vorzeichen zu Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Zielworte aus.	Ausgang Erforderlich	623

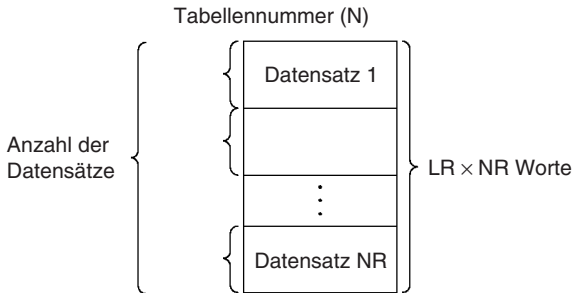
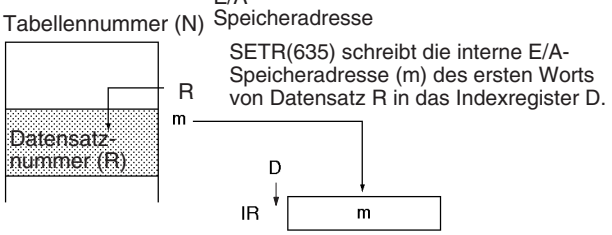
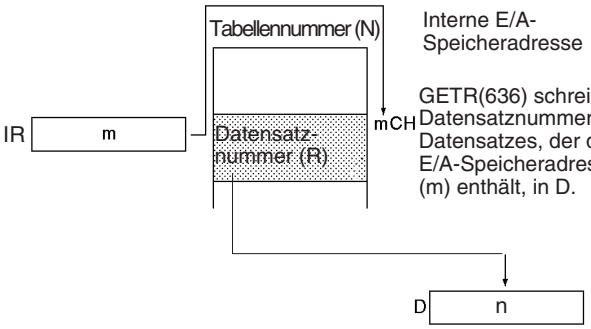
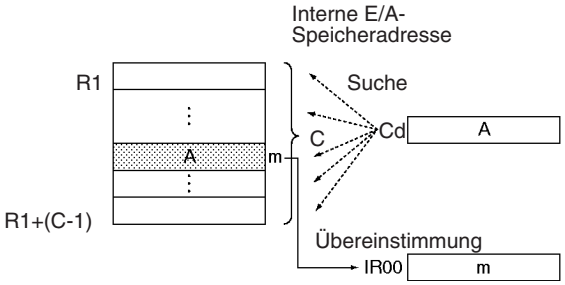
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
32-BIT BINARY TO DOUBLE FLOATING (32 Bit binär zu Gleitkommadaten doppelter Genauigkeit) DBLL @DBLL 844	 **S**: Erstes Quellwort **D**: Erstes Zielwort	Konvertiert die spezifizierten 32-Bit-Binärdaten mit Vorzeichen zu Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Zielworte aus.	Ausgang Erforderlich	624
DOUBLE FLOATING-POINT ADD (Gleitkomma-Addition doppelter Genauigkeit) +D @+D 845	 **Au**: 1. Summand, erstes Wort **Ad**: 2. Summand, erstes Wort **R**: Erstes Ergebniswort	Addiert die spezifizierten Gleitkommawerte doppelter Genauigkeit (je 64 Bit) und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	626
DOUBLE FLOATING-POINT SUBTRACT (Gleitkomma-Subtraktion doppelter Genauigkeit) −D @−D 846	 **Mi**: Erstes Minuend-Wort **Su**: Erstes Subtrahend-Wort **R**: Erstes Ergebniswort	Subtrahiert die spezifizierten Gleitkommawerte doppelter Genauigkeit (je 64 Bits) voneinander und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	628
DOUBLE FLOATING-POINT MULTIPLY (Gleitkomma-Multiplikation doppelter Genauigkeit) *D @*D 847	 **Md**: Erstes Multiplikand-Wort **Mr**: Erstes Multiplikator-Wort **R**: Erstes Ergebniswort	Multipliziert die spezifizierten Gleitkommawerte doppelter Genauigkeit (je 64 Bits) miteinander und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	630
DOUBLE FLOATING-POINT DIVIDE (Gleitkomma-Division doppelter Genauigkeit) /D @/D 848	 **Dd**: Erstes Dividend-Wort **Dr**: Erstes Divisor-Wort **R**: Erstes Ergebniswort	Dividiert die spezifizierten Gleitkommawerte doppelter Genauigkeit (je 64 Bits) durch einander und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	632

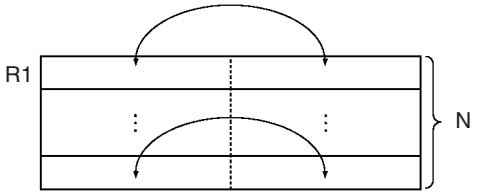
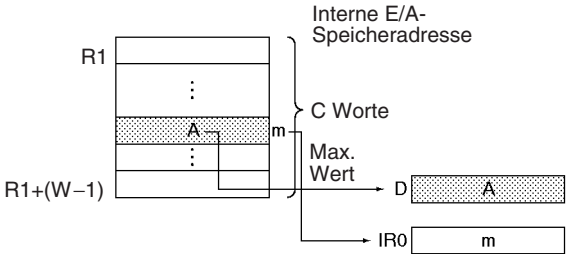
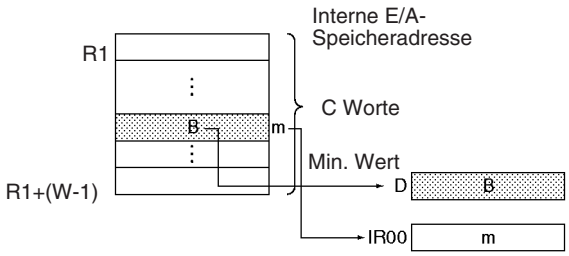
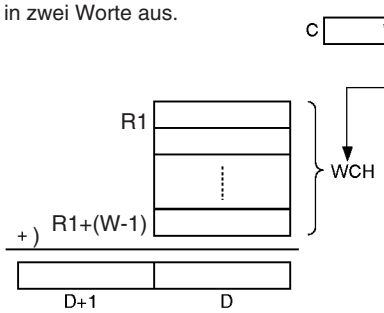
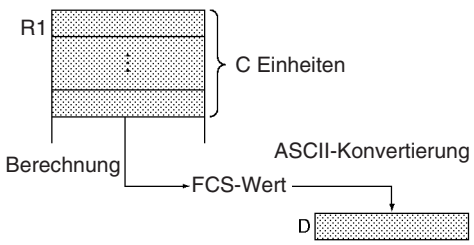
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
DOUBLE DEGREES TO RADIANS (Grad nach Radianten, Gleitkommadaten doppelter Genauigkeit) RADD @RADD 849	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Konvertiert die spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) von Grad in Radianten und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	634
DOUBLE RADIANS TO DEGREES (Radianten zu Grad, Gleitkommadaten doppelter Genauigkeit) DEGD @DEGD 850	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Konvertiert die spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) von Radianten in Grad und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	636
DOUBLE SINE (Sinus, Gleitkommadaten doppelter Genauigkeit) SIND @SIND 851	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Sinus des Winkels (in Radianten) aus den spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	637
DOUBLE COSINE (Kosinus, Gleitkommadaten doppelter Genauigkeit) COSD @COSD 852	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Kosinus des Winkels (in Radianten) aus den spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	639
DOUBLE TANGENT (Tangens, Gleitkommadaten doppelter Genauigkeit) TAND @TAND 853	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Tangens des Winkels (in Radianten) aus den spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	641
DOUBLE ARC SINE (Arkussinus, Gleitkommadaten doppelter Genauigkeit) ASIND @ASIND 854	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Winkel (in Radianten) aus dem Sinuswert der spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Ergebnisworte aus. (Die Arkussinusfunktion ist die Umkehrung der Sinusfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Sinuswert zwischen -1 und 1 entspricht.)	Ausgang Erforderlich	643
DOUBLE ARC COSINE (Arkuskosinus, Gleitkommadaten doppelter Genauigkeit) ACOSD @ACOSD 855	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Winkel (in Radianten) aus dem Kosinuswert der spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Ergebnisworte aus. (Die Arkuskosinusfunktion ist die Umkehrung der Kosinusfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Kosinuswert zwischen -1 und 1 entspricht.)	Ausgang Erforderlich	645

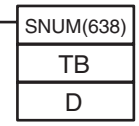
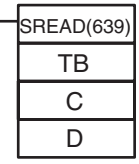
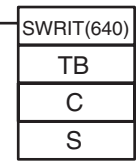
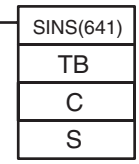
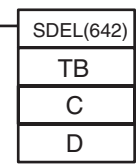
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungsbedingung	Seite
DOUBLE ARC TANGENT (Arkustangens, Gleitkommadaten doppelter Genauigkeit) ATAND @ATAND 856	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den Winkel (in Radianten) aus dem Tangenswert der spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Ergebnisworte aus. (Die Arkustangensfunktion ist die Umkehrung der Tangensfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Tangenswert entspricht.)	Ausgang Erforderlich	647
DOUBLE SQUARE ROOT (Quadratwurzel, Gleitkommadaten doppelter Genauigkeit) SQRTD @SQRTD 857	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet die Quadratwurzel aus den spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	649
DOUBLE EXPONENT (Exponent, Gleitkommadaten doppelter Genauigkeit) EXPD @EXPD 858	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den natürlichen Exponenten (zur Basis e) der spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	651
DOUBLE LOGARITHM (Logarithmus, Gleitkommadaten doppelter Genauigkeit) LOGD @LOGD 859	 **S**: Erstes Quellwort **R**: Erstes Ergebniswort	Berechnet den natürlichen Logarithmus (Basis e) aus den spezifizierten Gleitkommadaten doppelter Genauigkeit (64 Bits) und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	653
DOUBLE EXPONENTIAL POWER (Potenz, Gleitkommadaten doppelter Genauigkeit) PWRD @PWRD 860	 **B**: Erstes Basiswort **E**: Erstes Exponenten-Wort **R**: Erstes Ergebniswort	Berechnet die Potenz einer Gleitkommazahl doppelter Genauigkeit (64 Bits) mit einer anderen Gleitkommazahl doppelter Genauigkeit und gibt das Ergebnis in die Ergebnisworte aus.	Ausgang Erforderlich	655
DOUBLE SYMBOL COMPARISON (Symbolvergleich, Gleitkommadaten doppelter Genauigkeit) LD, AND oder OR + =D (335), <>=D (336), <=D (337), <=D (338), >=D (339), oder >=D (340)	Verwenden als LD:  Verwenden als AND:  Verwenden als OR:  **S1**: Vergleichsdaten 1 **S2**: Vergleichsdaten 2	Vergleicht die spezifizierten Daten doppelter Genauigkeit (64 Bits) und erzeugt eine EIN-Ausführungsbedingung, wenn das Vergleichsergebnis wahr ist. Gleitkomma-Symbolvergleichsbefehle können auf drei Arten verwendet werden: LD (Load), AND und OR.	LD: Nicht erforderlich AND oder OR: Erforderlich	657

2-2-15 Tabellendatenverarbeitungsbefehle

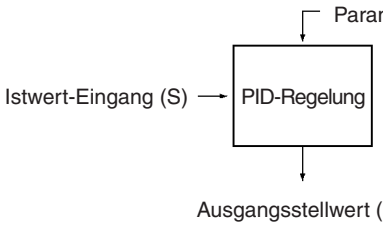
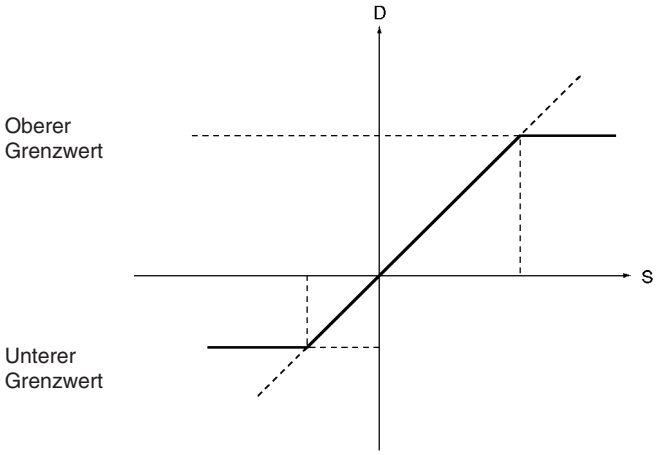
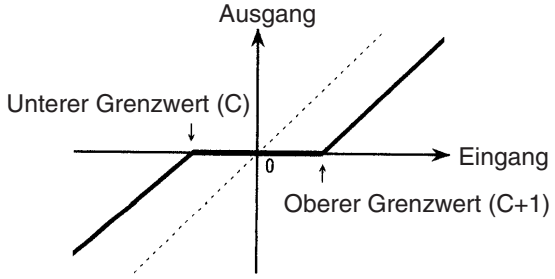
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
SET STACK (Stapel definieren) SSET @SSET 630	SSET(630) TB N **TB**: Erste Stapeladresse **N**: Anzahl der Worte	Definiert einen Stapel der spezifizierten Länge beginnend mit dem spezifizierten Wort und initialisiert die Worte im Datenbereich alle auf Null. 	Ausgang Erforderlich	666
PUSH ONTO STACK (In Stapel schreiben) PUSH @PUSH 632	PUSH(632) TB S **TB**: Erste Stapeladresse **S**: Quellwort	Schreibt ein Datenwort in den spezifizierten Stapel. 	Ausgang Erforderlich	669
LAST IN FIRST OUT (LIFO-Speicher) LIFO @LIFO 634	LIFO(634) TB D **TB**: Erste Stapeladresse **D**: Zielwort		Ausgang Erforderlich	675
FIRST IN FIRST OUT (FIFO-Speicher) FIFO @FIFO 633	FIFO(633) TB D **TB**: Erste Stapeladresse **D**: Zielwort	Liest das zuerst in den spezifizierten Stapel geschriebene Datenwort (die ältesten Daten im Stapel). 	Ausgang Erforderlich	672

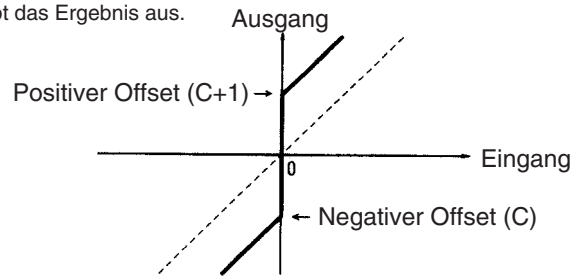
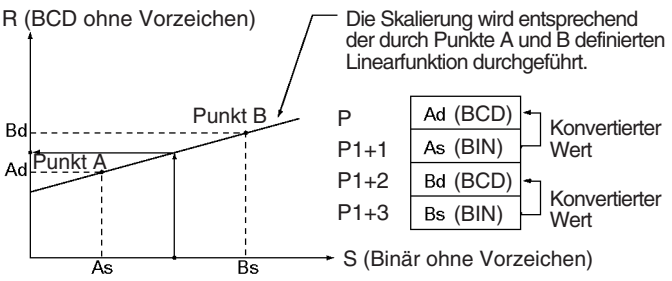
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
DIMENSION RECORD TABLE (Datensatztabellen-Definition) DIM @DIM 631	DIM(631) N LR NR TB **N**: Tabellennummer **LR**: Länge jedes Datensatzes **NR**: Anzahl der Datensätze **TB**: Erstes Wort der Tabelle	Definiert eine Datensatztabelle durch Angabe der Länge der einzelnen Datensätze und der Anzahl der Datensätze. Es können bis zu 16 Datensatztabellen definiert werden. 	Ausgang Erforderlich	678
SET RECORD LOCATION (Datensatzposition setzen) SETR @SETR 635	SETR(635) N R D **N**: Tabellennummer **R**: Datensatznummer **D**: Ziel-Indexregister	Schreibt die Position des spezifizierten Datensatzes (die interne E/A-Speicheradresse, an der der Datensatz beginnt) in das spezifizierte Indexregister.  SETR(635) schreibt die interne E/A-Speicheradresse (m) des ersten Worts von Datensatz R in das Indexregister D.	Ausgang Erforderlich	681
GET RECORD NUMBER (Datensatznummer lesen) GETR @GETR 636	GETR(636) N IR D **N**: Tabellennummer **IR**: Indexregister **D**: Zielwort	Gibt die Datensatznummer des Datensatzes an der E/A-Speicheradresse, die im spezifizierten Indexregister enthalten ist, zurück.  GETR(636) schreibt die Datensatznummer des Datensatzes, der die E/A-Speicheradresse (m) enthält, in D.	Ausgang Erforderlich	683
DATA SEARCH (Daten suchen) SRCH @SRCH 181	SRCH(181) C R1 Cd **C**: Erstes Steuerwort **R1**: Erstes Wort im Bereich **Cd**: Vergleichsdaten	Sucht nach einem Datenwort innerhalb eines Bereichs. 	Ausgang Erforderlich	685

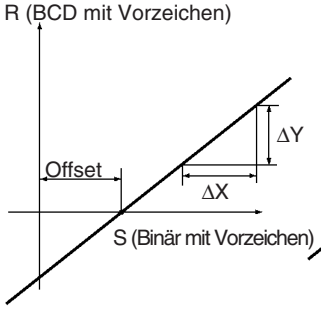
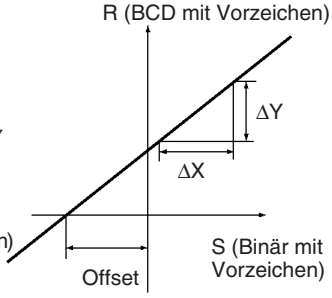
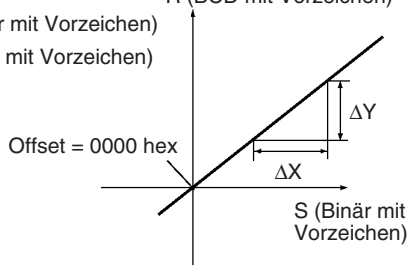
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
SWAP BYTES (Bytes austauschen) SWAP @SWAP 637	SWAP(637) N R1 **N**: Anzahl der Worte **R1**: Erstes Wort im Bereich	Vertauscht die Bytes eines Wortes in allen Worten innerhalb des Bereichs. Byte-Position wird ausgetauscht. 	Ausgang Erforderlich	687
FIND MAXIMUM (Maximalwert suchen) MAX @MAX 182	MAX(182) C R1 D **C**: Erstes Steuerwort **R1**: Erstes Wort im Bereich **D**: Zielwort	Sucht den maximalen Wert innerhalb des Bereichs. 	Ausgang Erforderlich	689
FIND MINIMUM (Minimalwert suchen) MIN @MIN 183	MIN(183) C R1 D **C**: Erstes Steuerwort **R1**: Erstes Wort im Bereich **D**: Zielwort	Sucht den Minimalwert innerhalb des Bereichs. 	Ausgang Erforderlich	693
SUM (Summe) SUM @SUM 184	SUM(184) C R1 D **C**: Erstes Steuerwort **R1**: Erstes Wort im Bereich **D**: Erstes Zielwort	Addiert die Bytes oder Worte innerhalb des Bereichs und gibt das Ergebnis in zwei Worte aus. 	Ausgang Erforderlich	697
FRAME CHECKSUM (Frame-Prüfsumme) FCS @FCS 180	FCS(180) C R1 D **C**: Erstes Steuerwort **R1**: Erstes Wort im Bereich **D**: Erstes Zielwort	Berechnet den ASCII-Frameprüfsummenwert für den spezifizierten Bereich. 	Ausgang Erforderlich	700

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
STACK SIZE READ (Stapelgröße lesen) (nur CS1-H, CJ1-H, CJ1M und CS1D) SNUM @SNUM 638	 **TB**: Erste Stapel-adresse **D**: Zielwort	Zählt die Menge der Stapeldaten (Anzahl der Worte) im spezifizierten Stapel.	Ausgang Erforderlich	704
STACK DATA READ (Stapeldaten lesen) (nur CS1-H, CJ1-H, CJ1M und CS1D) SREAD @SREAD 639	 **TB**: Erste Stapel-adresse **C**: Offsetwert **D**: Zielwort	Liest die Daten des spezifizierten Datenelements des Stapels. Der Offset-Wert bestimmt die Stelle des gewünschten Datenelements (Anzahl der Datenelemente vor der aktuellen Zeigerposition).	Ausgang Erforderlich	707
STACK DATA OVERWRITE (Stapeldaten überschreiben) (nur CS1-H, CJ1-H, CJ1M und CS1D) SWRIT @SWRIT 640	 **TB**: Erste Stapeladresse **C**: Offsetwert **S**: Quelldaten	Schreibt die Eingangsdaten für das spezifizierte Datenelement in den Stapel (überschreibt vorhandene Daten). Der Offset-Wert bestimmt die Stelle des gewünschten Datenelements (Anzahl der Datenelemente vor der aktuellen Zeigerposition).	Ausgang Erforderlich	710
STACK DATA INSERT (Stapeldaten einfügen) (nur CS1-H, CJ1-H, CJ1M und CS1D) SINS @SINS 641	 **TB**: Erste Stapeladresse **C**: Offsetwert **S**: Quelldaten	Fügt die Quelldaten an der spezifizierten Stelle in den Stapel ein und verschiebt den Rest der Daten im Stapel nach unten. Der Offset-Wert bestimmt die Stelle des Einfügepunktes (Anzahl der Datenelemente vor der aktuellen Zeigerposition).	Ausgang Erforderlich	713
STACK DATA DELETE (Stapeldaten löschen) (nur CS1-H, CJ1-H, CJ1M und CS1D) SDEL @SDEL 642	 **TB**: Erste Stapeladresse **C**: Offsetwert **D**: Zielwort	Löscht das Datenelement an der spezifizierten Stelle im Stapel und verschiebt den Rest der Daten im Stapel nach oben. Der Offset-Wert bestimmt die Stelle des Löschpunktes (Anzahl der Datenelemente vor der aktuellen Zeigerposition).	Ausgang Erforderlich	716

2-2-16 Regelungsbefehle

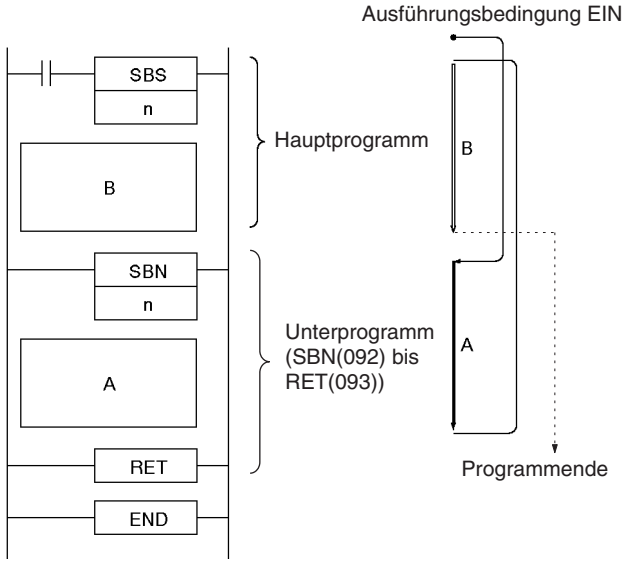
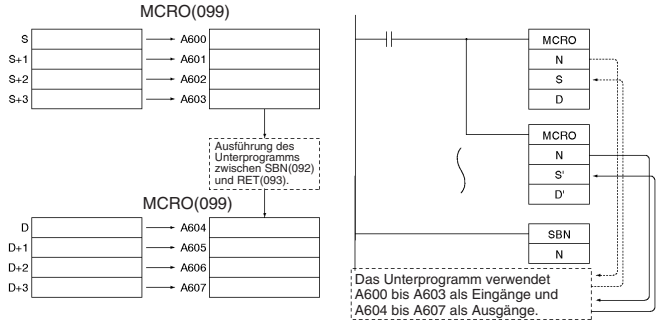
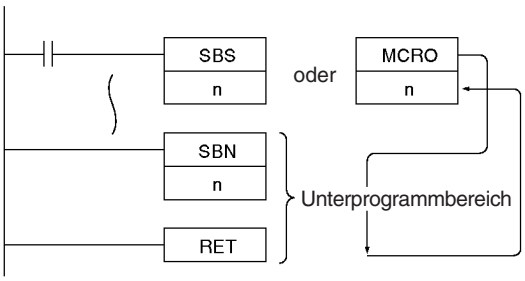
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
PID CONTROL (PID-Regelung) PID 190	PID(190) S C D **S**: Eingangswort **C**: Erstes Parameterwort **D**: Ausgangswort	Führt eine PID-Regelung gemäß den spezifizierten Parametern aus. 	Ausgang Erforderlich	720
PID CONTROL WITH AUTOTUNING (PID-Regelung mit Autotuning) PIDAT 191 (nur CS1-H, CJ1-H und CJ1M)	PIDAT(191) S C D **S**: Eingangswort **C**: Erstes Parameterwort **D**: Ausgangswort	Führt eine PID-Regelung gemäß der spezifizierten Parameter aus. Die PID-Konstanten können mit PIDAT(191) durch Autotuning eingestellt werden.	Ausgang Erforderlich	731
LIMIT CONTROL (Begrenzungs-funktion) LMT @LMT 680	LMT(680) S C D **S**: Eingangswort **C**: Erstes Grenzwert-Wort **D**: Ausgangswort	Begrenzt die Ausgabedaten durch den oberen und unteren Grenzwert. 	Ausgang Erforderlich	741
DEAD BAND CONTROL (Totband-Steuerung) BAND @BAND 681	BAND(681) S C D **S**: Eingangswort **C**: Erstes Grenzwert-Wort **D**: Ausgangswort	Steuert Ausgabedaten in Abhängigkeit von den eingestellten Parametern für die Totzone. 	Ausgang Erforderlich	743

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DEAD ZONE CONTROL (Totzonen-Steuerung) ZONE @ZONE 682	ZONE(682) S C D **S**: Eingangswort **C**: Erstes Grenzwert-Wort **D**: Ausgangswort	Addiert den spezifizierten Offset zu den Eingangsdaten und gibt das Ergebnis aus. 	Ausgang Erforderlich	746
TIME-PROPORTIONAL OUTPUT (Zeitproportionale Ausgabe) TPO 685 (Nur CS/CJ-Serie ab Baugruppenversion 2.0)	TPO(685) S C R **S**: Eingangswort **C**: Erstes Parameterwort **R**: Impulsausgabe-Bit	Verwendet das Tastverhältnis oder den Ausgangsstellwert aus dem spezifizierten Wort als Eingabe, konvertiert das Tastverhältnis in eine zeitproportionale Ausgabe auf der Basis der spezifizierten Parameter und gibt das Ergebnis über den spezifizierten Ausgang aus.	Ausgang Erforderlich	749
SCALING (Skalierung) SCL @SCL 194	SCL(194) S P1 R **S**: Quellwort **P1**: Erstes Parameterwort **R**: Ergebniswort	Konvertiert Binärdaten ohne Vorzeichen in BCD-Daten ohne Vorzeichen entsprechend der spezifizierten Linearfunktion. 	Ausgang Erforderlich	757

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
SCALING 2 (Skalierung 2) SCL2 @SCL2 486	SCL2(486) S P1 R **S**: Quellwort **P1**: Erstes Parameterwort **R**: Ergebniswort	Konvertiert Binärdaten mit Vorzeichen in BCD-Daten mit Vorzeichen entsprechend der spezifizierten Linearfunktion. Ein Offset kann bei der Definition der Linearfunktion eingegeben werden. **Positiver Offset**  **Negativer Offset**  P1 Offset (Binär mit Vorzeichen) P1+1 ΔY (Binär mit Vorzeichen) P1+2 ΔX (BCD mit Vorzeichen) **Offset von 0000** 	Ausgang Erforderlich	762

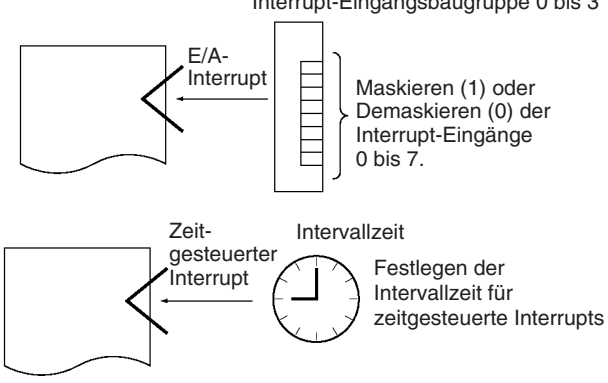
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
SCALING 3 (Skalierung 3) SCL3 @SCL3 487	SCL3(487) S P1 R **S**: Quellwort **P1**: Erstes Parameterwort **R**: Ergebniswort	Konvertiert BCD-Daten mit Vorzeichen in Binärdaten mit Vorzeichen entsprechend der spezifizierten Linearfunktion. Ein Offset kann bei der Definition der Linearfunktion eingegeben werden. **Positiver Offset** R (Binär mit Vorzeichen) Max. Konvertierung Max. Konvertierung ΔY ΔX Offset Min. Konvertierung S (BCD mit Vorzeichen) **Negativer Offset** R (Binär mit Vorzeichen) ΔY ΔX Offset S (BCD mit Vorzeichen) Min. Konvertierung **Offset von 0000** R (Binär mit Vorzeichen) Max. Konvertierung ΔY ΔX S (BCD mit Vorzeichen) Min. Konvertierung	Ausgang Erforderlich	766
AVERAGE (Mittelwert) AVG 195	AVG(195) S N R **S**: Quellwort **N**: Anzahl der Zyklen **R**: Ergebniswort	Berechnet den Mittelwert eines Eingangsworts über die spezifizierte Anzahl von Zyklen. S: Quellwort N: Anzahl der Zyklen R R+1 Zeiger Mittelwert-gültig-Flag Mittelwert R+2 R+3 N Werte R+N+1	Ausgang Erforderlich	769

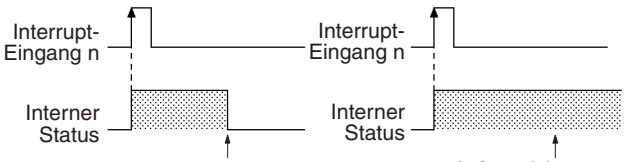
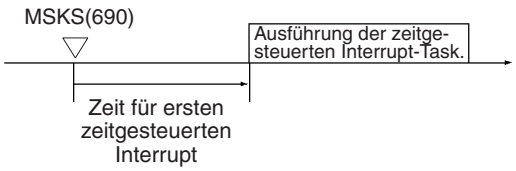
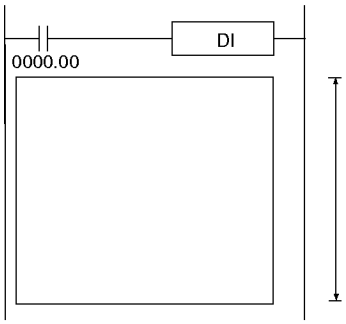
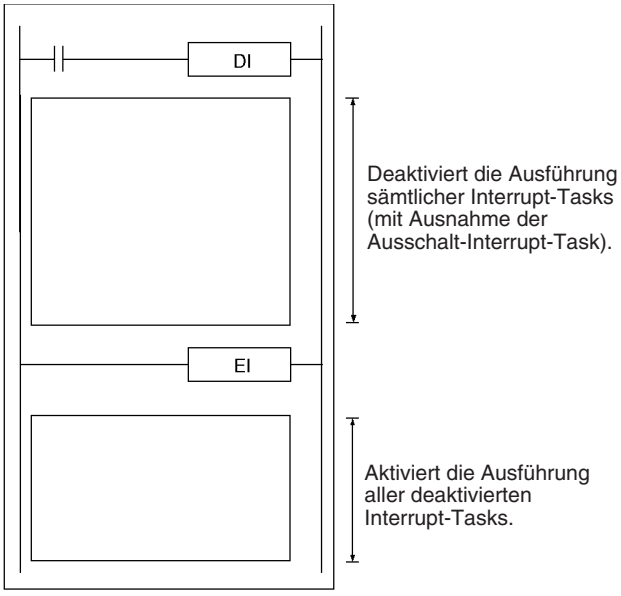
2-2-17 Unterprogrammbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
SUBROUTINE CALL (Unterprogrammaufruf) SBS @SBS 091	SBS(091) N **N**: Unterprogramm-nummer	Ruft das Unterprogramm mit der spezifizierten Unterprogrammnummer auf und führt es aus. 	Ausgang Erforderlich	773
MACRO (Makro) MCRO @MCRO 099	MCRO(099) N S D **N**: Unterprogramm-Nummer **S**: Erstes Eingangs-Parameterwort **D**: Erstes Ausgangs-Parameterwort	Ruft das Unterprogramm mit der spezifizierten Unterprogramm-Nummer auf und führt das Programm unter Verwendung der Eingangsparameter in S bis S+3 und der Ausgangsparameter in D bis D+3 aus. 	Ausgang Erforderlich	779
SUBROUTINE ENTRY (Unterprogramm-Anfang) SBN 092	SBN(092) N **N**: Unterprogramm-Nummer	Kennzeichnet den Beginn eines Unterprogramms mit der spezifizierten Unterprogramm-Nummer. 	Ausgang Nicht erforderlich	783
SUBROUTINE RETURN (Unterprogramm verlassen) RET 093	RET(093)	Kennzeichnet das Ende eines Unterprogramms.	Ausgang Nicht erforderlich	786

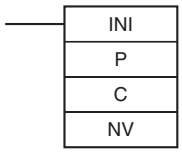
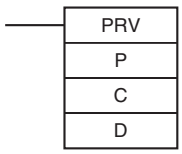
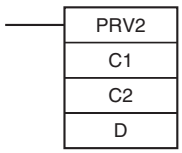
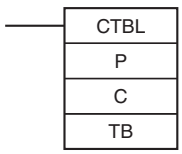
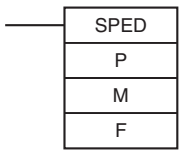
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
GLOBAL SUB-ROUTINE CALL (Globales Unterprogramm aufrufen) (nur CS1-H, CJ1-H, CJ1M und CS1D) GSBS 750	GSBS(750) N **N**: Unterprogramm-Nummer	Ruft das Unterprogramm mit der spezifizierten Unterprogrammnummer auf und führt es aus.	Ausgang Nicht erforderlich	786
GLOBAL SUB-ROUTINE ENTRY (Globales Unterprogramm, Anfang) (nur CS1-H, CJ1-H, CJ1M und CS1D) GSBN 751	GSBN(751) N **N**: Unterprogramm-Nummer	Kennzeichnet den Beginn eines Unterprogramms mit der spezifizierten Unterprogramm-Nummer.	Ausgang Nicht erforderlich	794
GLOBAL SUB-ROUTINE RETURN (Globales Unterprogramm verlassen) (nur CS1-H, CJ1-H, CJ1M und CS1D) GRET 752	GRET(752)	Kennzeichnet das Ende eines Unterprogramms.	Ausgang Nicht erforderlich	797

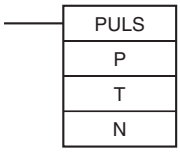
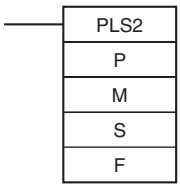
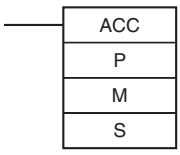
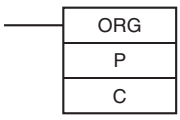
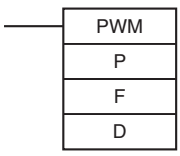
2-2-18 Interrupt-Steuerbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
SET INTERRUPT MASK (Interrupt-Maske setzen) (Nicht unterstützt von CS1D CPU-Baugruppen für Duplex-CPU-Systeme.) MSKS @MSKS 690	MSKS(690) N S **N**: Interrupt-Kennzeichner **S**: Interruptdaten	Zum Einstellen der Interrupt-Verarbeitung für E/A-Interrupts oder zeitgesteuerte Interrupts. Beide Interrupt-Tasks (E/A und zeitgesteuert) werden maskiert (deaktiviert), wenn die SPS erstmalig eingeschaltet wird. MSKS(690) kann zum Maskieren oder Demaskieren von E/A-Interrupts verwendet werden sowie zum Einstellen der Intervallzeit für zeitgesteuerte Interrupts. 	Ausgang Erforderlich	798
READ INTERRUPT MASK (Interrupt-Maske lesen) (Nicht unterstützt von CS1D CPU-Baugruppen für Duplex-CPU-Systeme.) MSKR @MSKR 692	MSKR(692) N D **N**: Interrupt-Kennzeichner **D**: Zielwort	Liest die aktuellen Interrupt-Verarbeitungseinstellungen die mit MSKS(690) gesetzt wurden.	Ausgang Erforderlich	804

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
CLEAR INTERRUPT (Interrupt zurücksetzen) (Nicht unterstützt von CS1D CPU-Baugruppen für Duplex-CPU-Systeme.) CLI @CLI 691	CLI(691) N S **N**: Interrupt-Kennzeichner **S**: Interruptdaten	Löscht oder erhält aufgezeichnete Interrupt-Eingänge für E/A-Interrupts oder legt die Zeit bis zum ersten Interrupt bei zeitgesteuerten Interrupts fest. N = 0 bis 3  Aufgezeichneter Interrupt gelöscht Aufgezeichneter Interrupt erhalten N = 4 bis 5 	Ausgang Erforderlich	809
DISABLE INTERRUPTS (Interrupts deaktivieren) DI @DI 693	DI(693)	Deaktiviert die Ausführung sämtlicher Interrupt-Tasks mit Ausnahme der Ausschalt-Interrupt-Task. 	Ausgang Erforderlich	814
ENABLE INTERRUPTS (Interrupts aktivieren) EI 694	EI(694)	Aktiviert die Ausführung aller mit DI(693) deaktivierten Interrupt-Tasks. 	Ausgang Nicht erforderlich	816

2-2-19 Befehle für schnelle Zähler und Impulsausgabe (nur CJ1M-CPU21/22/23)

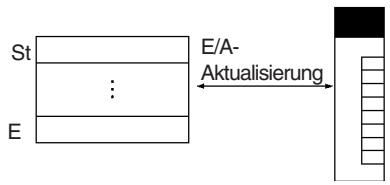
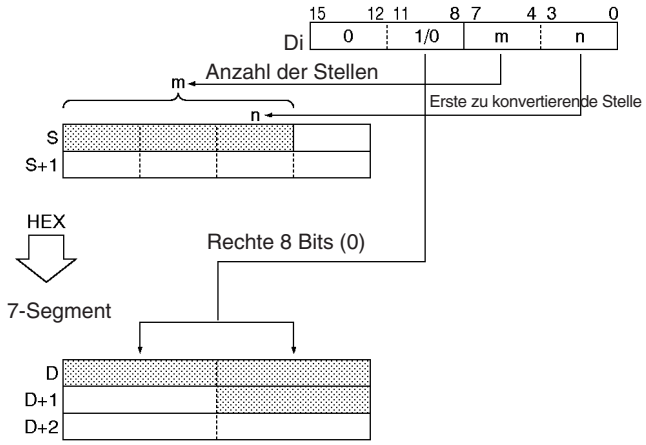
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
MODE CONTROL (Betriebsartensteuerung) INI @INI 880	 **P:**Schnittstellenbezeichner **C:** Steuerdaten **NV:** Erstes Wort mit neuem Istwert	INI(880) wird zum Starten und Anhalten des Zielwertvergleichs, zur Änderung des Istwerts eines schnellen Zählers, zur Änderung des Istwerts eines Interrupt-Eingangs (Zählermodus), zur Änderung des Istwerts einer Impulsausgabe oder zum Anhalten einer Impulsausgabe verwendet.	Ausgang Erforderlich	823
HIGH-SPEED COUNTER PV READ (Istwert von schnellem Zähler lesen) PRV @PRV 881	 **P:**Schnittstellenbezeichner **C:** Steuerdaten **D:** Erstes Zielwort	PRV(881) wird zum Lesen des Istwerts eines schnellen Zählers, einer Impulsausgabe oder eines Interrupt-Eingangs (Zählermodus) verwendet.	Ausgang Erforderlich	827
COUNTER FREQUENCY CONVERT (Impulsfrequenzkonvertierung) PRV2 883 (Nur CJ1M CPU-Baugruppen ab Ver. 2.0)	 **C1:** Steuerdaten **C2:** Impulse/Umdrehung **D:** Erstes Zielwort	Liest den Impulsfrequenzeingang eines schnellen Zählers und wandelt die Frequenz in eine Drehzahl (Umdrehungen/Minute) um oder den Istwert des Zählers in eine Gesamtanzahl von Umdrehungen. Das Ergebnis wird als 8-stellige Hexadezimalzahl in die Zielworte ausgegeben. Impulsfrequenzen können nur vom schnellen Zähler 0 gelesen werden.	Ausgang Erforderlich	833
COMPARISON TABLE LOAD (Zähler-Vergleichstabelle laden) CTBL @CTBL 882	 **P:**Schnittstellenbezeichner **C:** Steuerdaten **TB:** Erstes Vergleichstabellenwort	CTBL(882) wird zur Registrierung einer Vergleichstabelle und zur Ausführung von Zielwert- oder Bereichsvergleichen mit dem Istwert eines schnellen Zählers verwendet.	Ausgang Erforderlich	837
SPEED OUTPUT (Drehzahlausgabe) SPED @SPED 885	 **P:**Schnittstellenbezeichner **M:** Ausgangsbetriebsart **F:** Erstes Impulsfrequenzwort	SPED(885) wird zur Festlegung der Frequenz und zur Ausführung der Impulsausgabe ohne Beschleunigung oder Verzögerung verwendet.	Ausgang Erforderlich	841

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
SET PULSES (Impulsausgabe) PULS @PULS 886	 **P:**Schnittstellen-bezeichner **T:** Impulstyp **N:** Anzahl der Impulse	PULS(886) wird verwendet, um die Anzahl der Impulse für die Impulsausgabe festzulegen.	Ausgang Erforderlich	846
PULSE OUTPUT (Impulsausgabe 2) PLS2 @PLS2 887	 **P:**Schnittstellen-bezeichner **M:** Ausgangsbetriebsart **S:** Erstes Wort der Einstellungentabelle **F:** Erstes Wort der Startfrequenz	PLS2(887) wird verwendet, um die Impulsfrequenz und die Beschleunigungs-/Verzögerungsrampen einzustellen und die Impulsausgabe zu starten. Nur Positionierung ist möglich.	Ausgang Erforderlich	849
ACCELERATION CONTROL (Beschleunigungseinstellung) ACC @ACC 888	 **P:**Schnittstellen-bezeichner **M:** Ausgangsbetriebsart **S:** Erstes Wort der Einstellungentabelle	ACC(888) wird verwendet, um die Impulsfrequenz und die Beschleunigungs-/Verzögerungsrampen einzustellen und die Impulsausgabe zu starten. Sowohl Positionierung als auch Geschwindigkeitssteuerung ist möglich.	Ausgang Erforderlich	855
ORIGIN SEARCH (Nullpunktsuche) ORG @ORG 889	 **P:**Schnittstellen-bezeichner **C:** Steuerdaten	ORG(889) wird zur Durchführung von Nullpunktsuche und Nullpunktrückkehr verwendet.	Ausgang Erforderlich	862
PULSE WITH VARIABLE DUTY FACTOR (Impuls mit variablem Tastverhältnis) PWM @ 891	 **P:** Schnittstellen-bezeichner **F:** Frequenz **D:** Tastverhältnis	PWM(891) wird zur Ausgabe von Impulsen mit variablem Tastverhältnis verwendet.	Ausgang Erforderlich	865

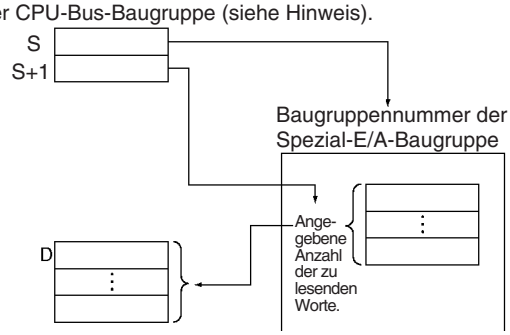
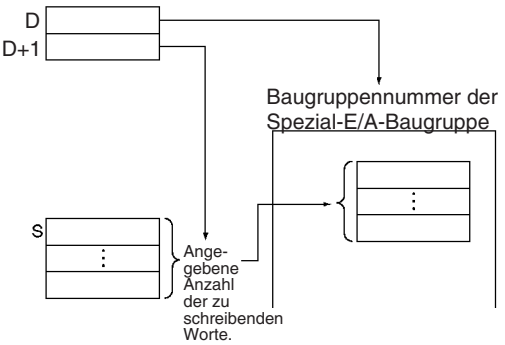
2-2-20 Schrittbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
STEP DEFINE (Schrittdefinition) STEP 008	STEP(008) B **B**: Bit	STEP(008) funktioniert je nach seiner Position und ob ein Steuerbit spezifiziert wurde oder nicht, auf die folgenden 2 Arten: (1) Startet einen bestimmten Schritt. (2) Beendet die Schrittkette (d. h. die Schrittausführung).	Ausgang Erforderlich	868
STEP START (nächster Schritt) SNXT 009	SNXT(009) B **B**: Bit	SNXT(009) wird auf die folgenden drei Arten verwendet: (1) Starten des Schrittprogramms, (2) zum Fortfahren mit dem nächsten Schritt, (3) zum Beenden der Schrittkette.	Ausgang Erforderlich	868

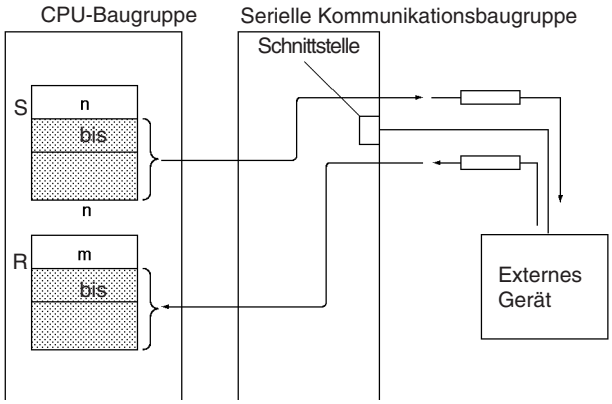
2-2-21 E/A-Baugruppenbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
I/O REFRESH (E/A-Aktualisierung) IORF @IORF 097	IORF(097) St E **St**: Anfangswort **E**: Endwort	Aktualisiert die spezifizierten E/A-Worte. E/A-Bitbereich oder Spezial-E/A-Baugruppen-Bitbereich E/A-Baugruppe oder Spezial-E/A-Baugruppe 	Ausgang Erforderlich	885
7-SEGMENT DECODER (7-Segment-Dekoder) SDEC @SDEC 078	SDEC(078) S Di D **S**: Quellwort **Di**: Stellen-spezifikator **D**: Erstes Zielwort	Konvertiert den hexadezimalen Inhalt der angegebenen Stelle(n) in 8-Bit-7-Segment- Anzeigecode und gibt ihn in die höher- bzw. niederwertigen 8 Bits der spezifizierten Zielworte aus. 	Ausgang Erforderlich	888

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DIGITAL SWITCH INPUT (Digital-schaltereingabe) DSW 210 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	DSW(210) I O D C1 C2 **I:** Dateneingangswort (D0 bis D3) **O:** Ausgangswort **D:**Erstes Ergebniswort **C1:** Anzahl der Stellen **C2:** Systemwort	Liest den Wert, der auf einem an eine Eingangs- oder Ausgangsbaugruppe angeschlossenen externen Daumenradschalter eingestellt ist, und speichert die 4- bzw. 8-stelligen BCD-Daten in den spezifizierten Worten.	Ausgang Erforderlich	890
TEN KEY INPUT (Zehnertastatur-eingabe) TKY 211 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	TKY(211) I D1 D2 **I**: Dateneingangswort **D1**: Erstes Registerwort **D2**: Tasteneingabewort	Liest numerische Daten von einem an eine Eingangs- und Ausgangsbaugruppe angeschlossenen numerischen Tastenfeld und speichert bis zu 8 Stellen BCD-Daten in den spezifizierten Worten.	Ausgang Erforderlich	896
HEXADECIMAL KEY INPUT (Hexadezimale Tastatureingabe) HKY 212 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	HKY(212) I O D C **I**: Dateneingangswort **O**: Ausgangswort **D**: Erstes Registerwort **C**: Systemwort	Liest numerische Daten von einem an eine Eingangs- und Ausgangsbaugruppe angeschlossenen Hexadezimal-Tastenblock und speichert bis zu 8 Stellen Hexadezimaldaten in den spezifizierten Worten.	Ausgang Erforderlich	899
MATRIX INPUT (Matrixeingabe) MTR 213 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	MTR(213) I O D C **I**: Dateneingangswort **O**: Ausgangswort **D**: Erstes Zielwort **C**: Systemwort	Liest bis zu 64 Signale von einer an eine Eingangs- und Ausgangsbaugruppe (mit 8 Eingängen und 8 Ausgängen) angeschlossenen 8 × 8-Matrix ein und speichert diese 64-Bit-Daten in 4 Zielworten.	Ausgang Erforderlich	904

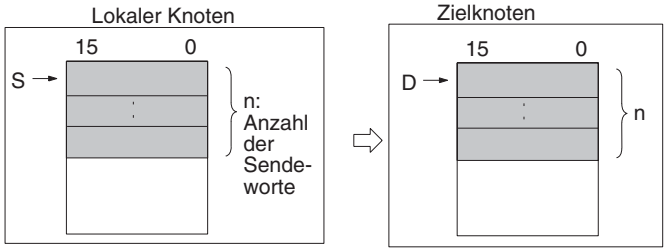
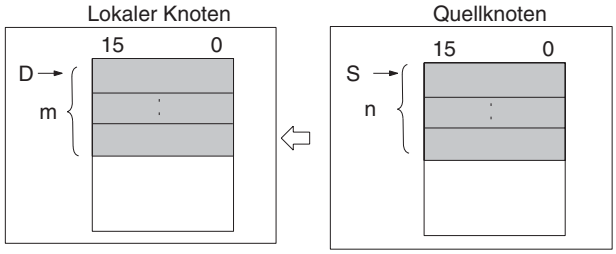
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
7-SEGMENT DISPLAY OUTPUT (7-Segment-Anzeigen-Ausgabe) 7SEG 214 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	7SEG(214) S O C D **S**: Erstes Quellwort **O**: Ausgangswort **C**: Steuerdaten **D**: Systemwort	Konvertiert die Eingangsdaten (4-stellige oder 8-stellige BCD-Daten) in Daten für eine 7-Segment-Anzeige und gibt diese Daten in das spezifizierte Ausgangswort aus.	Ausgang Erforderlich	908
INTELLIGENT I/O READ (Lesen von Speicher aus intelligenten Spezial-E/A-Baugruppen) IORD @IORD 222	IORD(222) C S D **C**: Steuerdaten **S**: Übertragungsquelle und Anzahl der Worte **D**: Übertragungsziel und Anzahl der Worte	Liest den Inhalt des Speicherbereichs der Spezial-E/A-Baugruppe oder CPU-Bus-Baugruppe (siehe Hinweis).  **Hinweis:** Nur CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 (einschließlich CS1-H, CJ1-H und CJ1M CPU-Baugruppen ab Lot-Nummer 030418) können von CPU-Bus-Baugruppen lesen.	Ausgang Erforderlich	913
INTELLIGENT I/O WRITE (Schreiben von Speicher in intelligenten Spezial-E/A-Baugruppen) IOWR @IOWR 223	IOWR(223) C S D **C**: Steuerdaten **S**: Übertragungsquelle und Anzahl der Worte **D**: Übertragungsziel und Anzahl der Worte	Schreibt in den internen Speicher der Spezial-E/A-Baugruppe oder der CPU-Bus-Baugruppe (siehe Hinweis).  **Hinweis:** Nur CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 (einschließlich CS1-H, CJ1-H und CJ1M CPU-Baugruppen ab Lot-Nummer 030418) können in CPU-Bus-Baugruppen schreiben.	Ausgang Erforderlich	917
CPU BUS UNIT I/O REFRESH (CPU-Bus-Baugruppe E/A-Aktualisierung) (nur CS1-H, CJ1-H, CJ1M und CS1D) DLNK @DLNK 226	DLNK(226) N **N**: Baugruppennummer	Aktualisiert die E/A in der CPU-Bus-Baugruppe mit der spezifizierten Baugruppennummer.	Ausgang Erforderlich	921

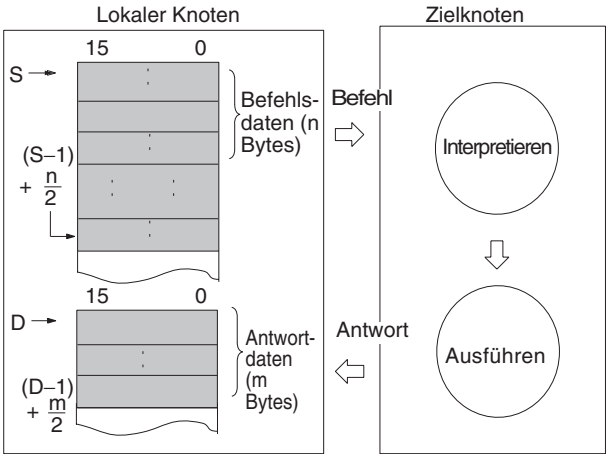
2-2-22 Befehle für die serielle Kommunikation

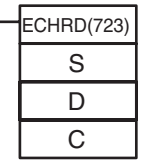
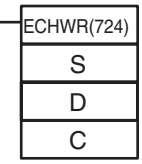
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
PROTOCOL MACRO (Protokoll-Makro) PMCR @PMCR 260	PMCR(260). C1 C2 S R **C1**: Steuerwort 1 **C2**: Steuerwort 2 **S**: Erstes Sendewort **R**: Erstes Empfangswort	Ruft eine bei einem seriellen Kommunikationsmodul (nur CS-Serie) oder einer seriellen Kommunikationsbaugruppe registrierte Kommunikationssequenz auf und führt diese aus. 	Ausgang Erforderlich	928
TRANSMIT (Senden) TXD @TXD 236	TXD(236) S C N **S**: Erstes Quellwort **C**: Steuerwort **N**: Anzahl der Bytes 0000 bis 0100 hex (0 bis 256 dezimal)	Gibt die festgelegte Anzahl an Bytes Daten über die in der CPU-Baugruppe integrierte RS-232C-Schnittstelle oder die serielle Schnittstelle eines seriellen Kommunikationsmoduls (Version 1.2 oder höher) aus.	Ausgang Erforderlich	937
RECEIVE (Empfangen) RXD @RXD 235	RXD(235) D C N **D**: Erstes Zielwort **C**: Steuerwort **N**: Anzahl der Bytes zum Speichern von 0000 bis 0100 hex (0 bis 256 dezimal)	Liest die festgelegte Anzahl an Bytes Daten über die in der CPU-Baugruppe integrierte RS-232C-Schnittstelle oder die serielle Schnittstelle eines seriellen Kommunikationsmoduls (Version 1.2 oder höher).	Ausgang Erforderlich	944
TRANSMIT VIA SERIAL COMMUNICATIONS UNIT (Senden über serielle Kommunikationsbaugruppe) TXDU @TXDU 256	TXDU(256) S C N **S**: Erstes Quellwort **C**: Erstes Steuerwort **N**: Anzahl Bytes 0000 bis 0100 hex (0 bis 256 dezimal)	Gibt die angegebene Anzahl Datenbytes über die serielle Schnittstelle einer seriellen Kommunikationsbaugruppe (ab Version 1.2) aus. Die Datenausgabe erfolgt protokollfrei. Sind im zugeordneten Setup-Bereich Start- und/oder Endcode festgelegt, werden diese verwendet.	Ausgang Erforderlich	952

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
RECEIVE VIA SERIAL COMMUNICATIONS UNIT (Empfangen über serielle Kommunikationsbaugruppe) RXDU @RXDU 255	RXDU(255) D C N **D**: Erstes Zielwort **C**: Erstes Steuerwort **N**: Anzahl zu speichernder Bytes, 0000 bis 0100 hex (1 bis 256 dezimal)	Liest die angegebene Anzahl Datenbytes über die serielle Schnittstelle einer seriellen Kommunikationsbaugruppe (ab Version 1.2). Der Datenempfang erfolgt protokollfrei. Sind im zugeordneten Setup-Bereich Start- und/oder Endcode festgelegt, werden diese beachtet.	Ausgang Erforderlich	960
CHANGE SERIAL PORT SETUP (Änderung der seriellen Schnittstellen-Konfiguration) STUP @STUP 237	STUP(237) C S **C**: Steuerwort (Schnittstelle) **S**: Erstes Quellwort	Ändert die Kommunikationsparameter einer seriellen Schnittstelle der CPU-Baugruppe, einer seriellen Kommunikationsbaugruppe (CPU-Bus-Baugruppe) oder eines seriellen Kommunikationsmoduls. Damit ermöglicht STUP(237) die Änderung des Protokollmodus während des SPS-Betriebs.	Ausgang Erforderlich	968

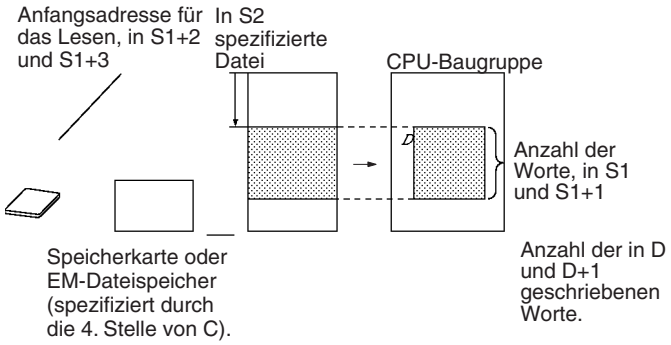
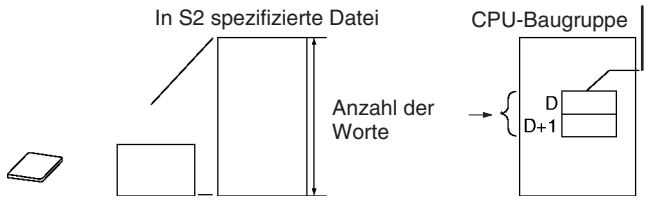
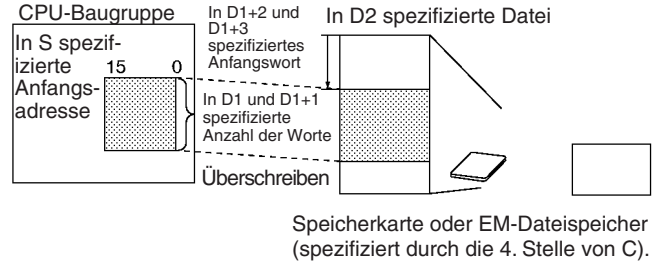
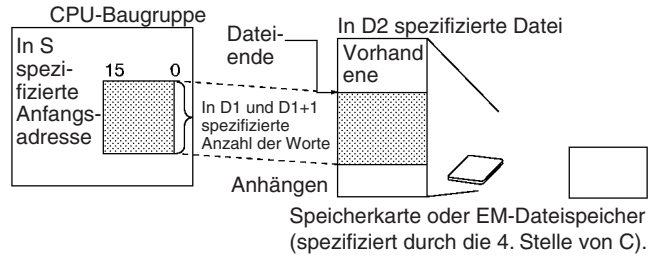
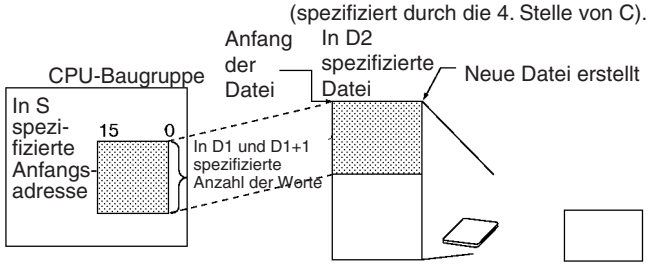
2-2-23 Netzwerkbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
NETWORK SEND (Senden über Netzwerk) SEND @SEND 090	SEND(090) S D C **S**: Erstes Quellwort **D**: Erstes Zielwort **C**: Erstes Steuerwort	Überträgt Daten zu einem Knoten im Netzwerk. 	Ausgang Erforderlich	991
NETWORK RECEIVE (Empfang über Netzwerk) RECV @RECV 098	RECV(098) S D C **S**: Erstes Quellwort **D**: Erstes Zielwort **C**: Erstes Steuerwort	Fordert Daten von einem Knoten im Netzwerk an und empfängt die Daten. 	Ausgang Erforderlich	997

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DELIVER COMMAND (FINS-Befehl senden) CMND @CMND 490	CMND(490) S D C **S**: Erstes Befehlswort **D**: Erstes Antwort-Wort **C**: Erstes Steuerwort	Sendet FINS-Befehle und empfängt die Antwort.  	Ausgang Erforderlich	1003
EXPLICIT MESSAGE SEND (Explizite Meldung senden) EXPLT 720 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	EXPLT(720) S D C **S**: Erstes Wort der zu sendenden Meldung **D**: Erstes Wort der empfangenen Meldung **C**: Erstes Steuerwort	Sendet eine explizite Meldung mit einem beliebigen Service-Code.	Ausgang Erforderlich	1013
EXPLICIT GET ATTRIBUTE (Explizite Meldung, Attribut abrufen) EGATR 721 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	EGATR(721) S D C **S**: Erstes Wort der zu sendenden Meldung **D**: Erstes Wort der empfangenen Meldung **C**: Erstes Steuerwort der Meldung	Liest Statusinformationen mit einer expliziten Meldung (Get Attribute Single, Service-Code: 0E hex).	Ausgang Erforderlich	1021
EXPLICIT SET ATTRIBUTE (Explizite Meldung, Attribut setzen) ESATR 722 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	ESATR(722) S C **S**: Erstes Wort der gesendeten Meldung **C**: Erstes Steuerwort	Schreibt Statusinformationen mit einer expliziten Meldung (Set Attribute Single, Service-Code: 0E hex)	Ausgang Erforderlich	1028

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
EXPLICIT WORD READ (Explizite Meldung, SPS-Wort lesen) ECHRD 723 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	 **S**: Erstes Quellwort in der dezentralen CPU-Baugruppe **D**: Erstes Zielwort in der lokalen CPU-Baugruppe **C**: Erstes Steuerwort	Liest Daten für die lokale CPU-Baugruppe von einer dezentralen CPU-Baugruppe im Netzwerk. (Die dezentrale CPU-Baugruppe muss explizite Meldungen unterstützten.)	Ausgang Erforderlich	1034
EXPLICIT WORD WRITE (Explizite Meldung, SPS-Wort schreiben) ECHWR 724 (Nur CPU-Baugruppen der CS/CJ-Serie ab Ver. 2.0)	 **S**: Erstes Quellenwort in der lokalen CPU-Baugruppe **D**: Erstes Zielwort in der dezentralen CPU-Baugruppe **C**: Erstes Steuerwort	Schreibt Daten von der lokalen CPU-Baugruppe in eine dezentrale CPU-Baugruppe im Netzwerk. (Die dezentrale CPU-Baugruppe muss explizite Meldungen unterstützten.)	Ausgang Erforderlich	1038

2-2-24 Dateispeicherbefehle

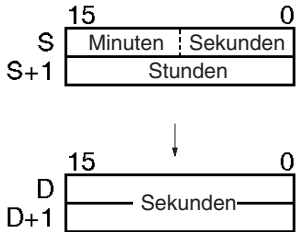
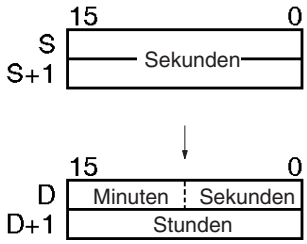
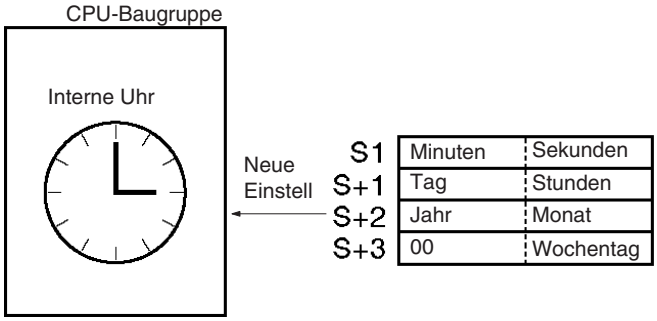
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
READ DATA FILE (Datei lesen) FREAD @FREAD 700	FREAD(700) C S1 S2 D **C**: Steuerwort **S1**: Erstes Quellwort **S2**: Dateiname **D**: Erstes Zielwort	Liest Daten bzw. die spezifizierte Datenmenge aus der Datendatei im Dateispeicher in den Datenbereich der CPU-Baugruppe.  	Ausgang Erforderlich	1045
WRITE DATA FILE (Datei schreiben) FWRIT @FWRIT 701	FWRIT(701) C D1 D2 S **C**: Steuerwort **D1**: Erstes Zielwort **D2**: Dateiname **S**: Erstes Quellwort	Überschreibt Daten in der Datendatei im Dateispeicher mit den Daten aus dem Datenbereich der CPU-Baugruppe, oder hängt diese Daten an. Wenn die spezifizierte Datei nicht vorhanden ist, wird eine neue Datei mit diesem Dateinamen erstellt.   	Ausgang Erforderlich	1052

2-2-25 Anzeigebefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
DISPLAY MESSAGE (Meldungsanzeige) MSG @MSG 046	MSG(046) N M **N**: Meldungsnummer **M**: Erstes Meldungswort	Liest die spezifizierten sechzehn Worte der erweiterten ASCII-Daten und zeigt die Meldung auf einem Peripheriegerät, wie zum Beispiel einer Programmierkonsole, an.	Ausgang Erforderlich	1058

2-2-26 Uhrfunktionsbefehle

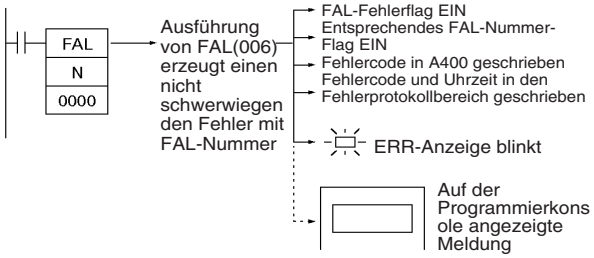
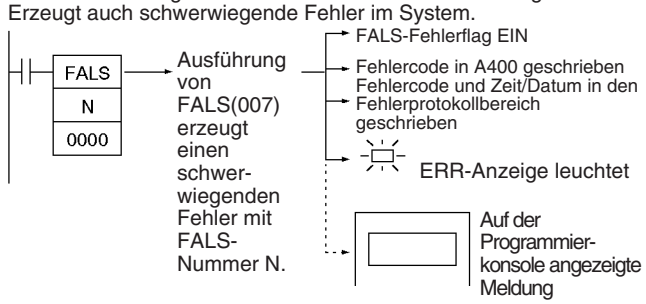
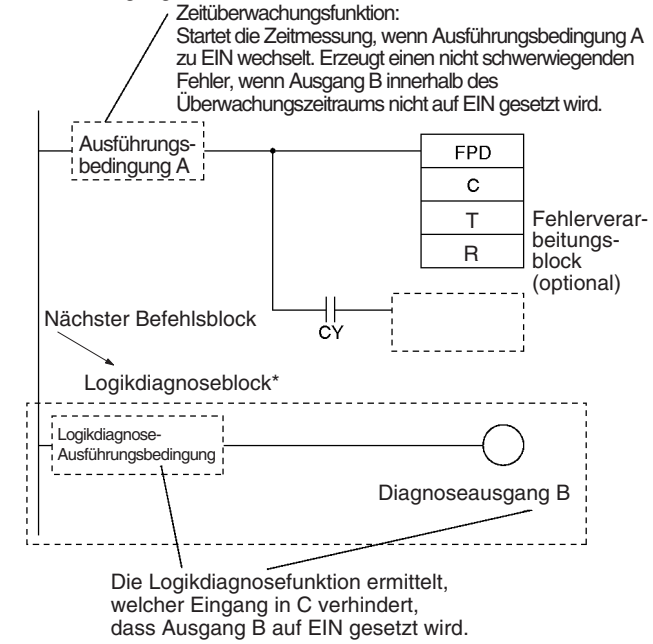
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
CALENDAR ADD (Datums-Addition) CADD @CADD 730	CADD(730) C T R **C**: Erstes Kalenderwort **T**: Erstes Zeit-Wort **R**: Erstes Ergebniswort	Addiert Zeit zu den Kalenderdaten in den spezifizierten Worten. C: 15–8 Minuten, 7–0 Sekunden C+1: Tag / Stunden C+2: Jahr / Monat + T: 15–8 Minuten, 7–0 Sekunden T+1: Stunden ↓ R: 15–8 Minuten, 7–0 Sekunden R+1: Tag / Stunden R+2: Jahr / Monat	Ausgang Erforderlich	1061
CALENDAR SUBTRACT (Datums-Subtraktion) CSUB @CSUB 731	CSUB(731) C T R **C**: Erstes Kalenderwort **T**: Erstes Zeit-Wort **R**: Erstes Ergebniswort	Subtrahiert Zeit von den Kalenderdaten in den spezifizierten Worten. C: 15–8 Minuten, 7–0 Sekunden C+1: Tag / Stunden C+2: Jahr / Monat – T: 15–8 Minuten, 7–0 Sekunden T+1: Stunden ↓ R: 15–8 Minuten, 7–0 Sekunden R+1: Tag / Stunden R+2: Jahr / Monat	Ausgang Erforderlich	1065

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
HOURS TO SECONDS (Stunden zu Sekunden) SEC @SEC 065	SEC(065) S D **S**: Erstes Quellwort **D**: Erstes Zielwort	Wandelt aus Stunden/Minuten/Sekunden bestehende Zeitdaten in Sekunden um. 	Ausgang Erforderlich	1068
SECONDS TO HOURS (Sekunden zu Stunden) HMS @HMS 066	HMS(066) S D **S**: Erstes Quellwort **D**: Erstes Zielwort	Wandelt Sekundendaten in die entsprechende Zeit in Stunden/Minuten/Sekunden um. 	Ausgang Erforderlich	1070
CLOCK ADJUSTMENT (Uhreinstellung) DATE @DATE 735	DATE(735) S **S**: Erstes Quellwort	Ändert die Einstellung der internen Uhr auf die Einstellung in den spezifizierten Worten. 	Ausgang Erforderlich	1073

2-2-27 Fehlersuch-Befehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
TRACE MEMORY SAMPLING (Speicheraufzeichnung) TRSM 045	TRSM(045)	Wenn TRSM(045) ausgeführt wird, wird der Status eines vorausgewählten Bits oder Wortes erfasst und im Aufzeichnungsspeicher gespeichert. TRSM(045) kann überall im Programm beliebig oft verwendet werden.	Ausgang Nicht erforderlich	1075

2-2-28 Fehlerdiagnosebefehle

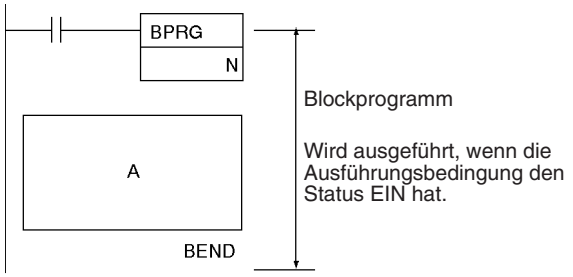
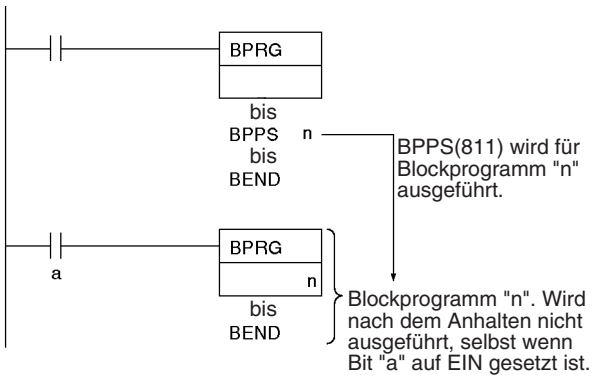
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
FAILURE ALARM (Geringfügiger Fehleralarm) FAL @FAL 006	FAL(006) N S **N**: FAL-Nummer **S**: Erstes Meldungs-Wort oder zu generierender Fehlercode	Erzeugt oder löscht anwenderdefinierte nicht schwerwiegende Fehler. Durch nicht schwerwiegende Fehler wird der SPS-Betrieb nicht angehalten. Erzeugt auch nicht schwerwiegende Fehler im System. 	Ausgang Erforderlich	1079
SEVERE FAILURE ALARM (Schwerwiegender Fehleralarm) FALS 007	FALS(007) N S **N:** FALS-Nummer **S:** Erstes Meldungs-Wort oder zu generierender Fehlercode	Erzeugt schwerwiegende anwenderdefinierte Fehler. Durch schwerwiegende Fehler wird der SPS-Betrieb angehalten. Erzeugt auch schwerwiegende Fehler im System. 	Ausgang Erforderlich	1087
FAILURE POINT DETECTION (Fehlerpunkt-Erfassung) FPD 269	FPD(269) C T R **C**: Steuerwort **T**: Überwachungs-zeit **R**: Erstes Registerwort	Ermittelt einen Fehler in einem Befehlsblock durch Zeitüberwachung zwischen der Ausführung von FPD(269) und der Ausführung eines Diagnose-Ausgangs und Ermittlung, welcher Eingang verhindert, dass der Ausgang auf EIN wechselt. 	Ausgang Erforderlich	1095

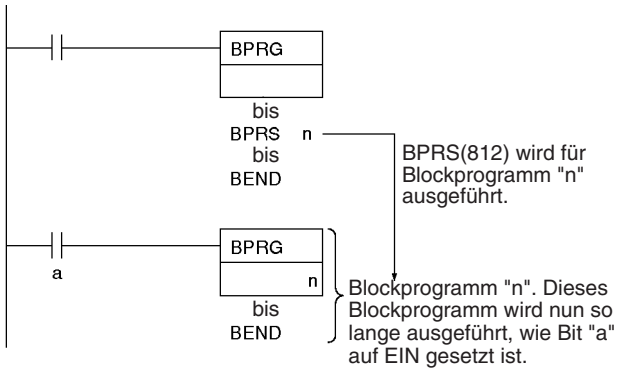
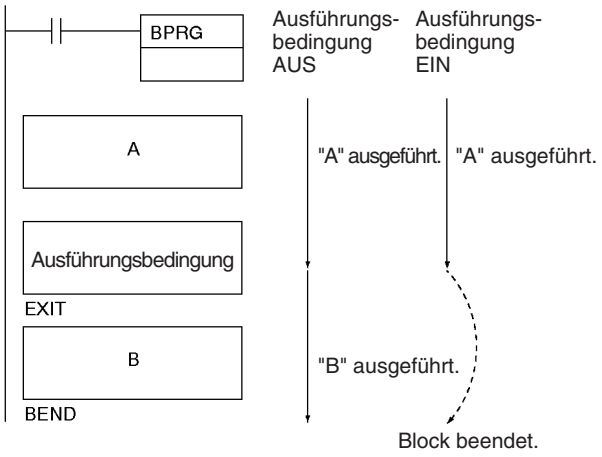
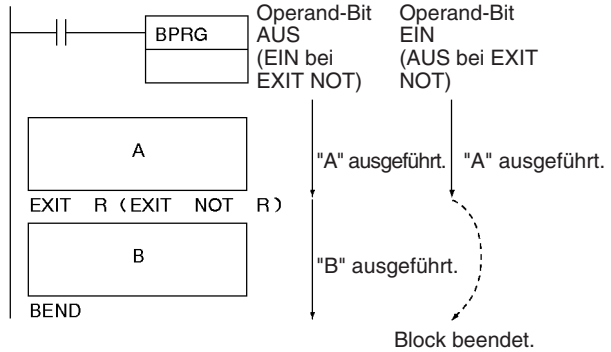
2-2-29 Sonstige Befehle

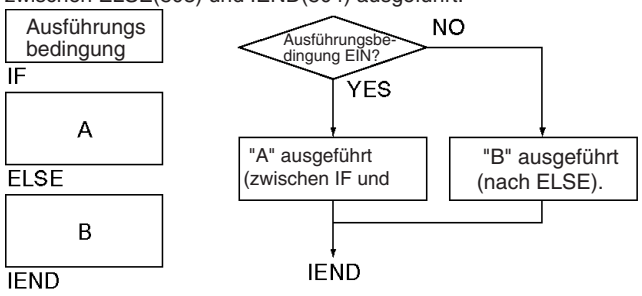
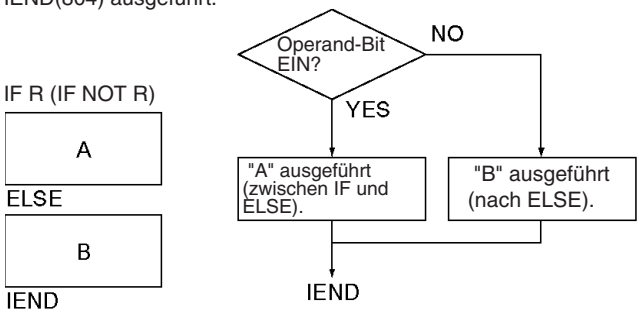
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
SET CARRY (Übertrags-Flag setzen) STC @STC 040	STC(040)	Setzt das Übertrags-Flag (CY).	Ausgang Erforderlich	1104
CLEAR CARRY (Übertrags-Flag löschen) CLC @CLC 041	CLC(041)	Setzt das Übertrags-Flag (CY) auf AUS.	Ausgang Erforderlich	1105
SELECT EM BANK (EM-Bank-Auswahl) EMBC @EMBC 281	EMBC(281) N **N**: Nummer der EM-Bank	Wechselt die aktuelle EM-Bank.	Ausgang Erforderlich	1106
EXTEND MAXIMUM CYCLE TIME (Maximale Zykluszeit erhöhen) WDT @WDT 094	WDT(094) T **T**: Zeitgeber-einstellung	Verlängert die maximale Zykluszeit für den Zyklus, in dem dieser Befehl ausgeführt wird.	Ausgang Erforderlich	1108
SAVE CONDITION FLAGS (Bedingungs-Flags speichern) (nur CS1-H, CJ1-H, CJ1M und CS1D) CCS @CCS 282	CCS(282)	Speichert den Status der Bedingungs-Flags.	Ausgang Erforderlich	1110
LOAD CONDITION FLAGS (Bedingungs-Flags laden) (nur CS1-H, CJ1-H, CJ1M und CS1D) CCL @CCL 283	CCL(283)	Liest den Status der gespeicherten Bedingungs-Flags.	Ausgang Erforderlich	1112
CONVERT ADDRESS FROM CV (Adresskonvertierung von CV) (nur CS1-H, CJ1-H, CJ1M und CS1D) FRMCV @FRMCV 284	FRMCV(284) S D **S**: Wort, das eine Speicheradresse der CV-Serie enthält **D**: Ziel-Indexregister	Konvertiert die Speicheradresse einer SPS der CV-Serie zur entsprechenden Speicheradresse einer SPS der CS/CJ-Serie.	Ausgang Erforderlich	1113
CONVERT ADDRESS TO CV (Adresskonvertierung zu CV) (nur CS1-H, CJ1-H, CJ1M und CS1D) TOCV @TOCV 285	TOCV(285) S D **S**: Indexregister, das eine Speicheradresse der CS-Serie enthält **D**: Zielwort	Konvertiert die Speicheradresse einer SPS der CS/CJ-Serie zur entsprechenden Speicheradresse einer SPS der CV-Serie.	Ausgang Erforderlich	1117

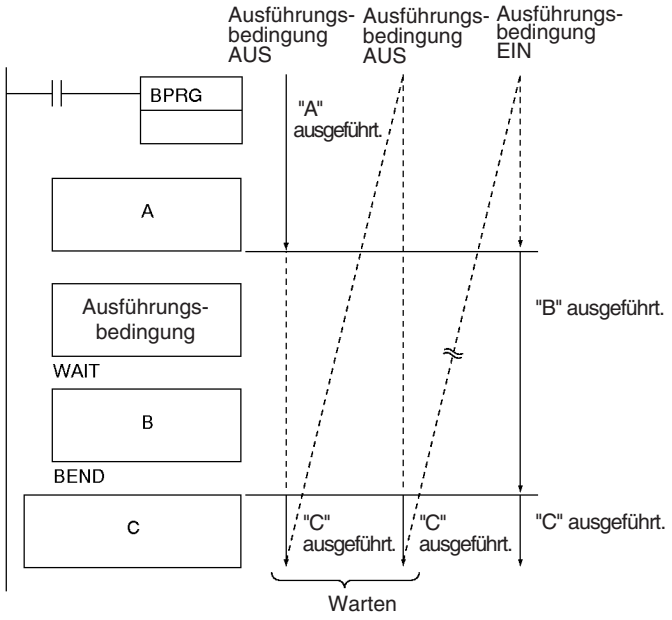
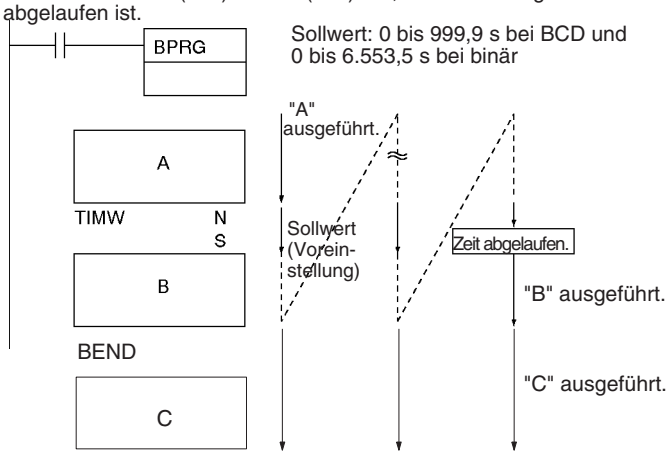
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
DISABLE PERIPHERAL SERVICING (Peripherieservice deaktivieren) (nur CS1D CPU-Baugruppen für Einzel-CPU-Systeme, CS1-H, CJ1-H, und CJ1M) IOSP @IOSP 287	IOSP(287)	Deaktiviert den Peripherieservice während der Programmausführung in einem der Parallelverarbeitungsmodi bzw. im Peripherieprioritätsmodus.	Ausgang Erforderlich	1121
ENABLE PERIPHERAL SERVICING (Peripherieservice aktivieren) (nur CS1D CPU-Baugruppen für Einzel-CPU-Systeme, CS1-H, CJ1-H, und CJ1M) IORS 288	IORS(288)	Aktiviert den Peripherieservice, der durch IOSP(287) für die Programmausführung in einem der Parallelverarbeitungsmodi bzw. im Peripherieprioritätsmodus deaktiviert wurde.	Ausgang Nicht erforderlich	1123

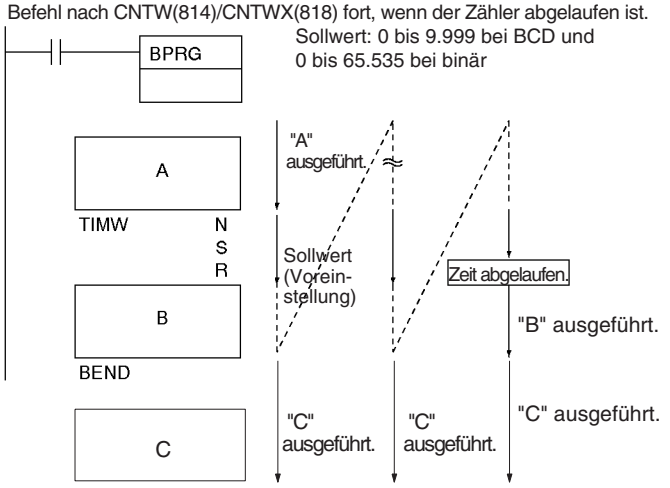
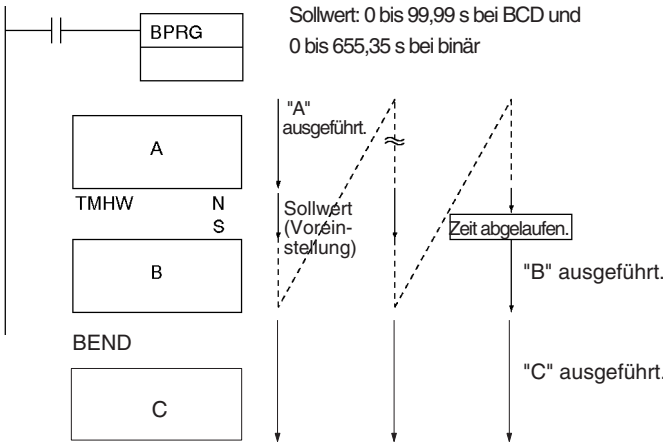
2-2-30 Blockprogrammierbefehle

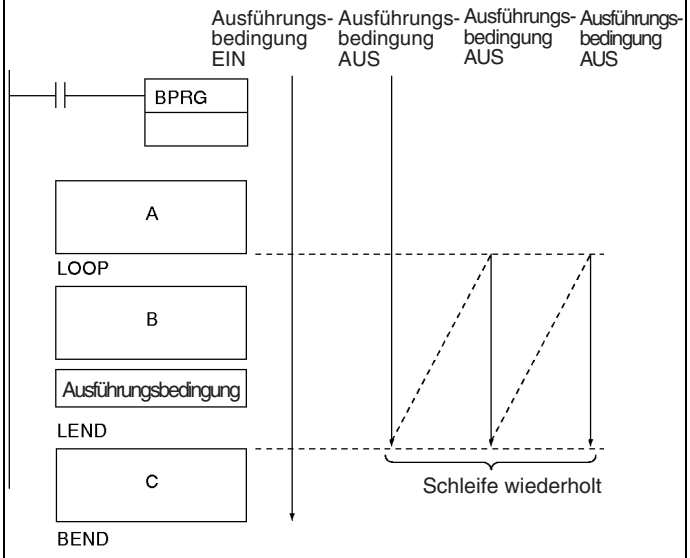
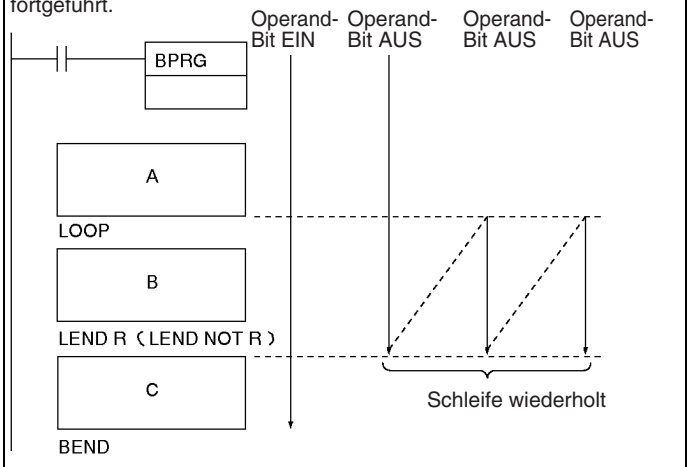
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
BLOCK PROGRAM BEGIN (Blockprogramm-Anfang) BPRG 096	BPRG(096) N **N**: Blockprogramm-Nummer	Definiert einen Blockprogrammierbereich. Für jeden BPRG(096)-Befehl muss es einen entsprechenden BEND(801)-Befehl geben. 	Ausgang Erforderlich	1128
BLOCK PROGRAM END (Blockprogramm-Ende) BEND 801		Definiert einen Blockprogrammierbereich. Für jeden BPRG(096)-Befehl muss es einen entsprechenden BEND(801)-Befehl geben.	Blockprogramm Erforderlich	1128
BLOCK PROGRAM PAUSE (Blockprogramm-Pause) BPPS 811	BPPS (811) N **N**: Blockprogramm-Nummer	Anhalten des spezifizierten Blockprogramms und Neustart des Blockprogramms aus einem anderen Blockprogramm heraus. 	Blockprogramm Erforderlich	1131

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
BLOCK PROGRAM RESTART (Block-programm-Fort-setzung) BPRS 812	BPRS (812) N **N**: Blockprogramm-Nummer	Anhalten des spezifizierten Blockprogramms und Neustart des Blockprogramms aus einem anderen Blockprogramm heraus.  	Blockpro-gramm Erforderlich	1131
CONDITIONAL BLOCK EXIT (Bedingtes Ver-lassen des Blockpro-gramms) EXIT 806	EXIT(806) **B**: Bit-Operand	EXIT(806) ohne ein Operand-Bit verlässt das Programm, wenn die Ausführungsbedingung den Zustand EIN hat.  	Blockpro-gramm Erforderlich	1137
CONDITIONAL BLOCK EXIT (Bedingtes Ver-lassen des Blockpro-gramms) EXIT 806	EXIT(806)B **B**: Bit-Operand	EXIT(806) ohne ein Operand-Bit verlässt das Programm, wenn die Ausführungsbedingung den Zustand EIN hat.  	Blockpro-gramm Erforderlich	1137
CONDITIONAL BLOCK EXIT NOT (Bedingtes Verlassen des Blockpro-gramms inver-tiert) EXIT NOT 806	EXIT NOT(806)B **B**: Bit-Operand	EXIT(806) ohne ein Operanden-Bit verlässt das Programm, wenn die Ausführungsbedingung den Zustand AUS hat.	Blockpro-gramm Erforderlich	1137

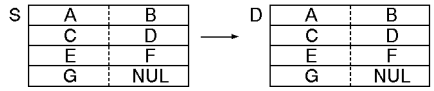
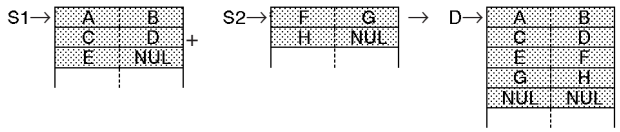
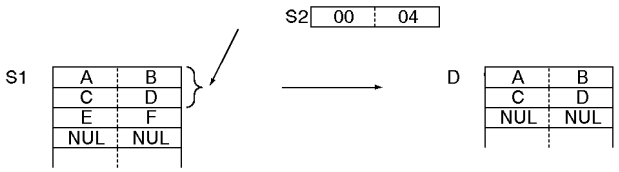
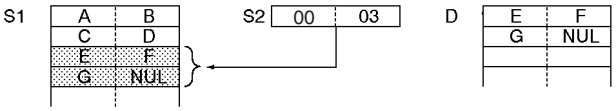
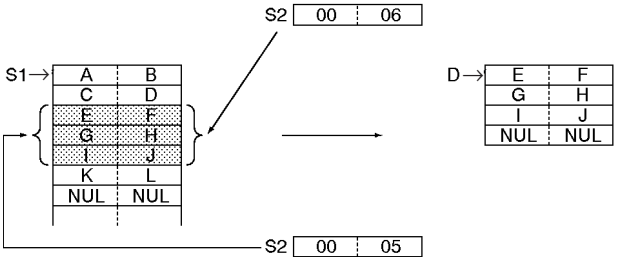
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
CONDITIONAL BLOCK BRANCHING (Bedingte Blockverzweigung) IF 802	IF(802)	Wenn die Ausführungsbedingung auf EIN gesetzt ist, werden die Befehle zwischen IF(802) und ELSE(803) ausgeführt. Wenn die Ausführungsbedingung auf AUS gesetzt ist, werden die Befehle zwischen ELSE(803) und IEND(804) ausgeführt. 	Blockpro-gramm Erforderlich	1133
CONDITIONAL BLOCK BRANCHING (Bedingte Blockverzweigung) IF 802	IF(802) B **B**: Bit-Operand	Wenn das Operand-Bit auf EIN gesetzt ist, werden die Befehle zwischen IF(802) und ELSE(803) ausgeführt. Wenn das Operand-Bit auf AUS gesetzt ist, werden die Befehle zwischen ELSE(803) und IEND(804) ausgeführt. 	Blockpro-gramm Erforderlich	1133
CONDITIONAL BLOCK BRANCHING (NOT) (Bedingte Blockverzweigung invertiert) IF NOT 802	IFNOT (802) B **B**: Bit-Operand	Die Befehle zwischen IF(802) und ELSE(803) werden ausgeführt, wenn das Operand-Bit auf EIN gesetzt ist, die Befehle ELSE(803) und IEND(804) werden ausgeführt, wenn das Operand-Bit auf AUS gesetzt ist.	Blockpro-gramm Erforderlich	1133
CONDITIONAL BLOCK BRANCHING (ELSE) (Bedingte Blockverzweigung (ELSE)) ELSE 803	---	Wenn der Befehl ELSE(803) nicht vorhanden ist und das Operanden-Bit den Zustand EIN hat, werden die Befehle zwischen IF(802) und IEND(804) ausgeführt.	Blockpro-gramm Erforderlich	1133
CONDITIONAL BLOCK BRANCHING END (Bedingte Blockverzweigung Ende) IEND 804	---	Wenn das Operanden-Bit den Zustand AUS hat, werden nur die Befehle hinter IEND(804) ausgeführt.	Blockpro-gramm Erforderlich	1133

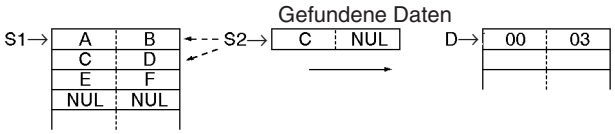
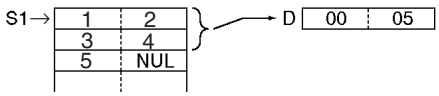
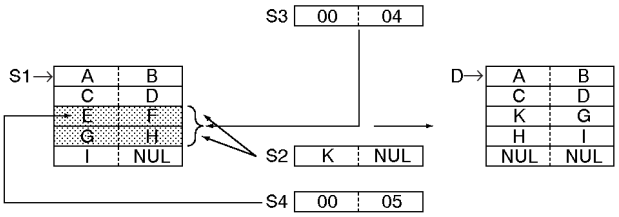
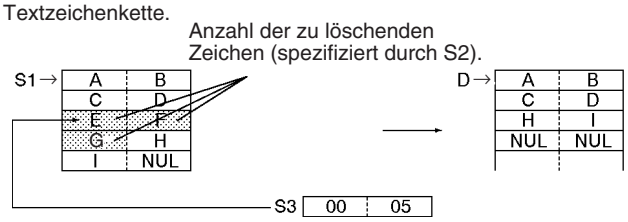
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
ONE CYCLE AND WAIT (Ein Zyklus und warten) WAIT 805	WAIT(805)	Wenn die Ausführungsbedingung für WAIT(805) auf EIN gesetzt ist, werden die restlichen Befehle im Blockprogramm übersprungen. 	Blockpro-gramm Erforderlich	1140
ONE CYCLE AND WAIT (Ein Zyklus und warten) WAIT 805	WAIT(805) B **B**: Bit-Operand	Wenn das Operand-Bit den Zustand AUS hat (EIN bei WAIT NOT(805)), werden die restlichen Befehle im Blockprogramm übersprungen. Im nächsten Zyklus wird kein Teil der Blockprogramms ausgeführt, außer der Teil für die Ausführungsbedingung für WAIT(805) bzw. WAIT NOT(805). Wenn die Ausführungsbedingung auf EIN wechselt (AUS bei WAIT NOT(805)), werden die Befehle von WAIT(805) bzw. WAIT NOT(805) bis zum Ende des Programms ausgeführt.	Blockpro-gramm Erforderlich	1140
ONE CYCLE AND WAIT (NOT) (Ein Zyklus und Warten invertiert) WAIT NOT 805	WAIT(805) NOT B **B**: Bit-Operand	Wenn das Operand-Bit den Zustand AUS hat (EIN bei WAIT NOT(805)), werden die restlichen Befehle im Blockprogramm übersprungen. Im nächsten Zyklus wird kein Teil der Blockprogramms ausgeführt, außer der Teil für die Ausführungsbedingung für WAIT(805) bzw. WAIT NOT(805). Wenn die Ausführungsbedingung auf EIN wechselt (AUS bei WAIT NOT(805)), werden die Befehle von WAIT(805) bzw. WAIT NOT(805) bis zum Ende des Programms ausgeführt.	Blockpro-gramm Erforderlich	1140
TIMER WAIT (Zeitgeber-Warten) TIMW 813 (BCD) TIMWX 816 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	TIMW(813) N SV **N**: Zeitgebernummer **SV**: Einstellwert TIMWX(816) N SV **N**: Zeitgebernummer **SV**: Sollwert	Verzögert die Ausführung des Blockprogramms, bis die spezifizierte Zeitspanne verstrichen ist. Die Ausführung fährt mit dem nächsten Befehl nach TIMW(813)/TIMWX(816) fort, wenn der Zeitgeber abgelaufen ist. 	Blockpro-gramm Erforderlich	1144

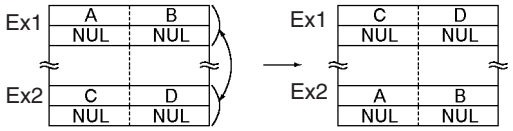
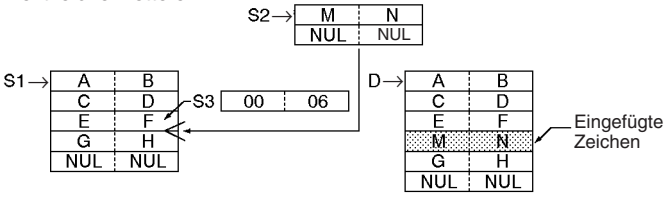
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
COUNTER WAIT (Zähler-Warten) CNTW 814 (BCD) CNTWX 818 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	CNTW(814) N SV **N**: Zähler-nummer **SV**: Sollwert **I**: Zähleingang CNTWX(818) N SV **N**: Zähler-nummer **SV**: Sollwert **I**: Zähleingang	Verzögert die Ausführung des restlichen Blockprogramms, bis der spezifizierte Zählwert erreicht ist. Die Ausführung fährt mit dem nächsten Befehl nach CNTW(814)/CNTWX(818) fort, wenn der Zähler abgelaufen ist. Sollwert: 0 bis 9.999 bei BCD und 0 bis 65.535 bei binär 	Blockpro-gramm Erforderlich	1147
HIGH-SPEED TIMER WAIT (Schneller Zeit-geber warten) TMHW 815 (BCD) TMHWX 817 (Binär) (nur CS1-H, CJ1-H, CJ1M und CS1D)	TMHW(815) N SV **N**: Zeitgebernum-mer **SV**: Sollwert TMHWX(817) N SV **N**: Zeitgebernum-mer **SV**: Sollwert	Verzögert die Ausführung des restlichen Blockprogramms, bis die spezifizierte Zeitspanne verstrichen ist. Die Ausführung fährt mit dem nächsten Befehl nach TMHW(815)/TMHWX(818) fort, wenn der Zeitgeber abgelaufen ist. Sollwert: 0 bis 99,99 s bei BCD und 0 bis 655,35 s bei binär 	Blockpro-gramm Erforderlich	1150

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
LOOP (Schleife) LOOP 809	---	LOOP(809) kennzeichnet den Anfang einer Programmschleife. 	Blockpro-gramm Erforderlich	1153
LEND (Schleifenende) LEND 810	LEND(810)	LEND(810) bzw. LEND(810) NOT spezifizieren das Ende der Schleife. Wenn LEND(810) bzw. LEND(810) NOT erreicht ist, geht die Programmausführung zurück zum vorhergehenden Befehl LOOP(809), bis das Operanden-Bit für LEND(810) bzw. LEND(810) NOT auf EIN bzw. AUS wechselt oder bis die Ausführungsbedingung für LEND(810) auf EIN wechselt.	Blockpro-gramm Erforderlich	1153
LEND (Schleifenende) LEND 810	LEND(810) B **B**: Bit-Operand	Wenn das Operand-Bit für LEND(810) auf AUS gesetzt ist (bzw. auf EIN bei LEND(810) NOT) wird die Ausführung der Schleife beginnend mit dem nächsten Befehl nach LOOP(809) wiederholt. Wenn das Operand-Bit für LEND(810) auf EIN gesetzt ist (bzw. auf AUS bei LEND(810) NOT) wird die Schleife beendet, und die Ausführung wird mit dem nächsten Befehl nach LEND(810) bzw. LEND(810) NOT fortgeführt.  **Hinweis:** Der Status des Operand-Bits ist im Falle von LEND(810) NOT umgekehrt.	Blockpro-gramm Erforderlich	1153
LEND NOT (Schleifenende invertiert) LEND NOT 810	LEND(810) NOT **B**: Bit-Operand	LEND(810) bzw. LEND(810) NOT spezifizieren das Ende der Schleife. Wenn LEND(810) bzw. LEND(810) NOT erreicht ist, geht die Programmausführung zurück zum vorhergehenden Befehl LOOP(809), bis das Operanden-Bit für LEND(810) bzw. LEND(810) NOT auf EIN bzw. AUS wechselt oder bis die Ausführungsbedingung für LEND(810) auf EIN wechselt.	Blockpro-gramm Erforderlich	1153

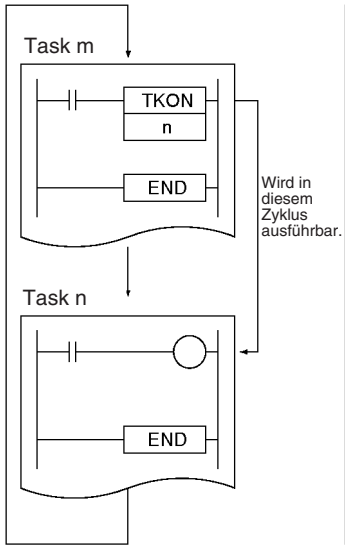
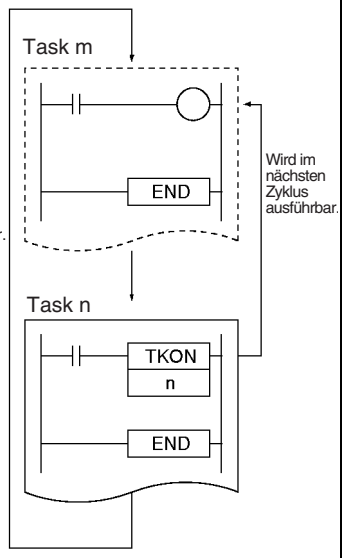
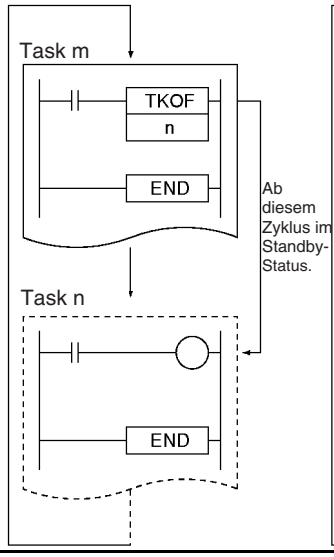
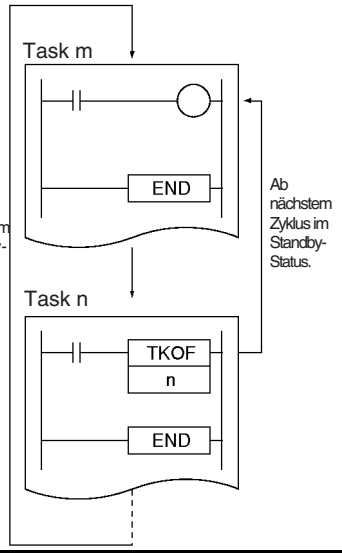
2-2-31 Textverarbeitungsbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
MOV STRING (Zeichenkette übertragen) MOV$ @MOV$ 664	MOV$(664) S D **S**: Erstes Quellwort **D**: Erstes Zielwort	Überträgt eine Textzeichenkette. 	Ausgang Erforderlich	1159
CONCATENATE STRING (Zeichenkette verknüpfen) +$ @+$ 656	+$(656) S1 S2 D **S1**: Textzeichenkette 1 **S2**: Textzeichenkette 2 **D**: Erstes Zielwort	Verknüpft eine Textzeichenkette mit einer anderen Textzeichenkette. 	Ausgang Erforderlich	1161
GET STRING LEFT (Zeichenkette von links lesen) LEFT$ @LEFT$ 652	LEFT$(652) S1 S2 D **S1**: Textzeichenkette, erstes Wort **S2**: Anzahl der Zeichen **D**: Erstes Zielwort	Liest eine angegebene Anzahl von Zeichen von der linken Seite (dem Anfang) einer Textzeichenkette. 	Ausgang Erforderlich	1164
GET STRING RIGHT (Zeichenkette von rechts lesen) RGHT$ @RGHT$ 653	RGHT$(653) S1 S2 D **S1**: Textzeichenkette, erstes Wort **S2**: Anzahl der Zeichen **D**: Erstes Zielwort	Liest eine angegebene Anzahl von Zeichen von der rechten Seite (dem Ende) einer Textzeichenkette. 	Ausgang Erforderlich	1166
GET STRING MIDDLE (Zeichenkette aus Mitte lesen) MID$ @MID$ 654	MID$(654) S1 S2 S3 D **S1**: Textzeichenkette, erstes Wort **S2**: Anzahl der Zeichen **S3**: Anfangsposition **D**: Erstes Zielwort	Liest eine spezifizierte Anzahl von Zeichen ab einer beliebigen Stelle innerhalb einer Textzeichenkette. 	Ausgang Erforderlich	1168

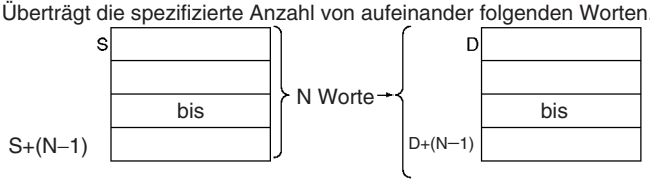
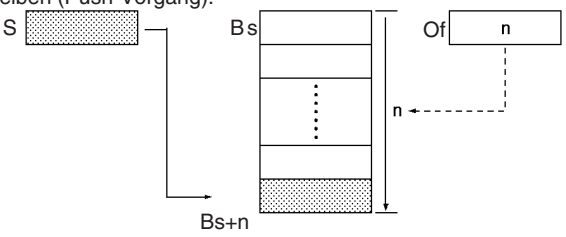
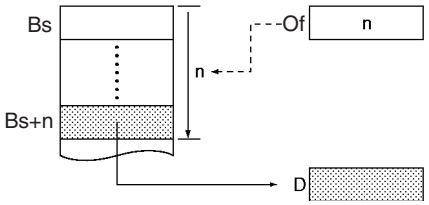
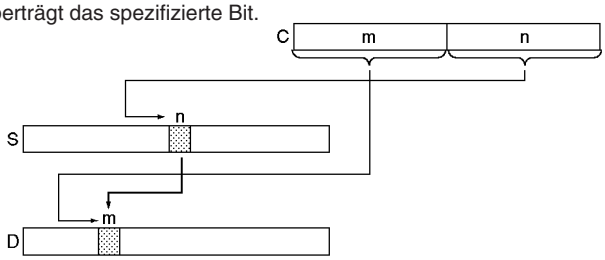
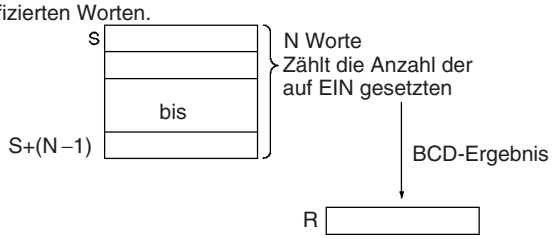
Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
FIND IN STRING (In Zeichenkette suchen) FIND @FIND$ 660	FIND$(660) S1 S2 D **S1**: Quell-Textzeichenkette, erstes Wort **S2**: Erstes Wort der gefundenen Textzeichenkette **D**: Erstes Zielwort	Sucht eine bestimmte Textzeichenkette innerhalb einer Textzeichenkette. 	Ausgang Erforderlich	1171
STRING LENGTH (Zeichenketten-Länge) LEN$ @LEN$ 650	LEN$(650) S D **S**: Textzeichenkette, erstes Wort **D**: Erstes Zielwort	Berechnet die Länge einer Textzeichenkette. 	Ausgang Erforderlich	1173
REPLACE IN STRING (In Zeichenkette ersetzen) RPLC$ @RPLC$ 661	RPLC$(661) S1 S2 S3 S4 D **S1**: Textzeichenkette, erstes Wort **S2**: Ersatztextzeichenkette, erstes Wort **S3**: Anzahl der Zeichen **S4**: Anfangsposition **D**: Erstes Zielwort	Ersetzt eine Textzeichenkette durch eine angegebene Textzeichenkette ab einer bestimmten Position. 	Ausgang Erforderlich	1175
DELETE STRING (In Zeichenkette löschen) DEL$ @DEL$ 658	DEL$(658) S1 S2 S3 D **S1**: Textzeichenkette, erstes Wort **S2**: Anzahl der Zeichen **S3**: Anfangsposition **D**: Erstes Zielwort	Löscht eine spezifizierte Textzeichenkette aus der Mitte einer Textzeichenkette. 	Ausgang Erforderlich	1178

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
EXCHANGE STRING (Zeichenkette austauschen) XCHG$ @XCHG$ 665	XCHG$(665) Ex1 Ex2 **Ex1**: Erstes Austauschwort 1 **Ex2**: Erstes Austauschwort 2	Ersetzt eine spezifizierte Textzeichenkette durch eine andere spezifizierte Textzeichenkette. 	Ausgang Erforderlich	1180
CLEAR STRING (Zeichenkette löschen) CLR$ @CLR$ 666	CLR$(666) S **S**: Textzeichenkette, erstes Wort	Löscht eine komplette Textzeichenkette zu NUL (00 Hex) 	Ausgang Erforderlich	1182
INSERT INTO STRING (In Zeichenkette einfügen) INS$ @INS$ 657	INS$(657) S1 S2 S3 D **S1**: Basistext-Zeichenkette, erstes Wort **S2**: Eingefügte Textzeichenkette, erstes Wort **S3**: Anfangsposition **D**: Erstes Zielwort	Fügt eine spezifizierte Textzeichenkette in die Mitte einer Textzeichenkette ein. 	Ausgang Erforderlich	1184
Zeichenkettenvergleich LD, AND, OR + =$, <>$, <$, <=$, >$, >=$ 670 (=$) 671 (<>$) 672 (<$) 673 (<=$) 674 (>$) 675 (>=$)	**LD** Symbol S1 S2 **AND** Symbol S1 S2 **OR** Symbol S1 S2 **S1**: Textzeichenkette 1 **S2**: Textzeichenkette 2	Zeichenketten-Vergleichsbefehle (=$, <>$, <$, <=$, >$, >=$) vergleichen zwei Textzeichenketten, an deren Anfang beginnend. Dabei werden die Werte der ASCII-Codes verglichen. Wenn das Ergebnis des Vergleichs wahr ist, wird eine EIN-Ausführungsbedingung für LOAD, AND oder OR erzeugt.	LD: Nicht erforderlich AND, OR: Erforderlich	1187

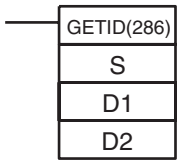
2-2-32 Task-Steuerbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
TASK ON (Programm-Task EIN) TKON @TKON 820	TKON(820) N **N**: Task-Nummer	Macht die spezifizierte Task ausführbar. Die Nummer der spezifizierten Task ist höher als die Nummer der lokalen Task (m<n).  Die Nummer der spezifizierten Task ist niedriger als die Nummer der lokalen Task (m>n). 	Ausgang Erforderlich	1192
TASK OFF (Programm-Task AUS) TKOF @TKOF 821	TKOF(821) N **N**: Task-Nummer	Versetzt die angegebene Task in den Standby-Status. Die Nummer der spezifizierten Task ist höher als die Nummer der lokalen Task (m<n).  Die Nummer der spezifizierten Task ist niedriger als die Nummer der lokalen Task (m>n). 	Ausgang Erforderlich	1196

2-2-33 Befehle zur Modellkonvertierung (nur CPU-Baugruppen ab Ver. 3.0)

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs- bedingung	Seite
BLOCK TRANSFER (Blockübertragung) XFERC @XFERC 565	XFERC(565) N S D **N**: Anzahl der Worte **S**: Erstes Quellwort **D**: Erstes Zielwort	Überträgt die spezifizierte Anzahl von aufeinander folgenden Worten. 	Ausgang Erforderlich	1201
SINGLE WORD DISTRIBUTE (Einzelwortverteilung) DISTC @DISTC 566	DISTC(566) S Bs Of **S**: Quellwort **Bs**: Ziel-Basisadresse **Of**: Offset	Überträgt das Quellwort an das durch Addition eines Offsetwerts zur Basisadresse errechnete Zielwort. Kann auch in einen Stapel schreiben (Push-Vorgang). 	Ausgang Erforderlich	1203
DATA COLLECT (Daten sammeln) COLLC @COLLC 567	COLLC(567) Bs Of D **Bs**: Quellen-Basisadresse **Of**: Offset **D**: Zielwort	Überträgt das Quellwort (errechnet durch Addition eines Offsetwerts zur Basisadresse) an das Zielwort. Kann auch Daten aus einem Stapel in FIFO- oder LIFO-Reihenfolge (Stapel-Leseoperation) lesen. 	Ausgang Erforderlich	1206
MOVE BIT (Bit übertragen) MOVBC @MOVBC 568	MOVBC(568) S C D **S**: Quellwort oder Daten **C**: Steuerwort **D**: Zielwort	Überträgt das spezifizierte Bit. 	Ausgang Erforderlich	1211
BIT COUNTER (Bitzähler) BCNTC @BCNTC 621	BCNTC(621) N S R **N**: Anzahl der Worte (BCD) **S**: Erstes Quellwort **R**: Ergebniswort	Zählt die Gesamtanzahl der auf EIN gesetzten Bits in den spezifizierten Worten. 	Ausgang Erforderlich	1212

2-2-34 Spezielle Funktionsblockbefehle

Befehl AWL Code	Symbol/Operand	Funktion	Position Ausführungs-bedingung	Seite
GET VARIABLE ID (Variablen-ID abrufen) GETID @GETID 286	 **S**: Variable oder Adresse **D1**: ID-Code **D2**: Zielwort	Gibt den FINS-Variablentyp-Code (Datenbereich) und die Wortadresse der angegebenen Variable oder Adresse aus. Dieser Befehl wird im allgemeinen verwendet, um die einer Variable in einem Funktionsblock zugeordnete Adresse zu ermitteln.	Ausgang Erforderlich	1214

2-3 Alphabetische Liste der Befehle nach AWL-Kürzel

A

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
ACC	ACCELERATION CONTROL	888	@ACC	---	---	855
ACOS	ARC COSINE	464	@ACOS	---	---	589
ACOSD	DOUBLE ARC COSINE	855	@ACOSD	---	---	645
AND	AND	---	@AND	%AND	!AND	157
AND <	AND LESS THAN	310	---	---	---	275
AND <$	AND STRING LESS THAN	672	---	---	---	1187
AND <>	AND NOT EQUAL	305	---	---	---	275
AND <>$	AND STRING NOT EQUAL	671	---	---	---	1187
AND <>D	AND DOUBLE FLOATING NOT EQUAL	336	---	---	---	657
AND <> DT	AND TIME NOT EQUAL	342	---	---	---	281
AND <>F	AND FLOATING NOT EQUAL	330	---	---	---	600
AND <>L	AND DOUBLE NOT EQUAL	306	---	---	---	275
AND <>S	AND SIGNED NOT EQUAL	307	---	---	---	275
AND <>SL	AND DOUBLE SIGNED NOT EQUAL	308	---	---	---	275
AND <D	AND DOUBLE FLOATING LESS THAN	337	---	---	---	657
AND < DT	AND TIME LESS THAN	343	---	---	---	281
AND <F	AND FLOATING LESS THAN	331	---	---	---	600
AND <L	AND DOUBLE LESS THAN	311	---	---	---	275
AND <S	AND SIGNED LESS THAN	312	---	---	---	275
AND <SL	AND DOUBLE SIGNED LESS THAN	313	---	---	---	275
AND =	AND EQUAL	300	---	---	---	275
AND =$	AND STRING EQUALS	670	---	---	---	1187
AND =D	AND DOUBLE FLOATING EQUAL	335	---	---	---	657
AND = DT	AND TIME EQUAL	341	---	---	---	281
AND =F	AND FLOATING EQUAL	329	---	---	---	600
AND =L	AND DOUBLE EQUAL	301	---	---	---	275
AND =S	AND SIGNED EQUAL	302	---	---	---	275
AND =SL	AND DOUBLE SIGNED EQUAL	303	---	---	---	275
AND >	AND GREATER THAN	320	---	---	---	275
AND >$	AND STRING GREATER THAN	674	---	---	---	1187
AND >D	AND DOUBLE FLOATING GREATER THAN	339	---	---	---	657
AND >DT	AND TIME GREATER THAN	345	---	---	---	281
AND >F	AND FLOATING GREATER THAN	333	---	---	---	600

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
AND >L	AND DOUBLE GREATER THAN	321	---	---	---	275
AND >S	AND SIGNED GREATER THAN	322	---	---	---	275
AND >SL	AND DOUBLE SIGNED GREATER THAN	323	---	---	---	275
AND LD	AND LOAD	---	---	---	---	164
AND NOT	AND NOT	---	---	---	!AND NOT	159
AND TST	AND BIT TEST	350	---	---	---	174
AND TSTN	AND BIT TEST	351	---	---	---	174
AND <=	AND LESS THAN OR EQUAL	315	---	---	---	275
AND <=$	AND STRING LESS THAN OR EQUAL	673	---	---	---	1187
AND <=D	AND DOUBLE FLOATING LESS THAN OR EQUAL	338	---	---	---	657
AND <=DT	AND TIME LESS THAN OR EQUAL	344	---	---	---	281
AND <=F	AND FLOATING LESS THAN OR EQUAL	332	---	---	---	600
AND <=L	AND DOUBLE LESS THAN OR EQUAL	316	---	---	---	275
AND <=S	AND SIGNED LESS THAN OR EQUAL	317	---	---	---	275
AND <=SL	AND DOUBLE SIGNED LESS THAN OR EQUAL	318	---	---	---	275
AND >=	AND GREATER THAN OR EQUAL	325	---	---	---	275
AND >=$	AND STRING GREATER THAN OR EQUALS	675	---	---	---	1187
AND >=D	AND DOUBLE FLOATING GREATER THAN OR EQUAL	340	---	---	---	657
AND >=DT	AND TIME GREATER THAN OR EQUAL	346	---	---	---	281
AND >=F	AND FLOATING GREATER THAN OR EQUAL	334	---	---	---	600
AND >=L	AND DOUBLE GREATER THAN OR EQUAL	326	---	---	---	275
AND >=S	AND SIGNED GREATER THAN OR EQUAL	327	---	---	---	275
AND >=SL	AND DOUBLE SIGNED GREATER THAN OR EQUAL	328	---	---	---	275
ANDL	DOUBLE LOGICAL AND	610	@ANDL	---	---	519
ANDW	LOGICAL AND	034	@ANDW	---	---	517
APR	ARITHMETIC PROCESS	069	@APR	---	---	540
ASC	ASCII CONVERT	086	@ASC	---	---	486
ASFT	ASYNCHRONOUS SHIFT REGISTER	017	@ASFT	---	---	349
ASIN	ARC SINE	463	@ASIN	---	---	587
ASIND	DOUBLE ARC SINE	854	@ASIND	---	---	643
ASL	ARITHMETIC SHIFT LEFT	025	@ASL	---	---	354
ASLL	DOUBLE SHIFT LEFT	570	@ASLL	---	---	355

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
ASR	ARITHMETIC SHIFT RIGHT	026	@ASR	---	---	357
ASRL	DOUBLE SHIFT RIGHT	571	@ASRL	---	---	358
ATAN	ARC TANGENT	465	@ATAN	---	---	591
ATAND	DOUBLE ARC TANGENT	856	@ATAND	---	---	647
AVG	AVERAGE	195	---	---	---	769

B

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
BAND	DEAD BAND CONTROL	681	@BAND	---	---	743
BCD	BINARY-TO-BCD	024	@BCD	---	---	469
BCDL	DOUBLE BINARY-TO-BCD	059	@BCDL	---	---	470
BCDS	SIGNED BINARY-TO-BCD	471	@BCDS	---	---	505
BCMP	UNSIGNED BLOCK COMPARE	068	@BCMP	---	---	304
BCMP2	EXPANDED BLOCK COMPARE	502	@BCMP2	---	---	306
BCNT	BIT COUNTER	067	@BCNT	---	---	556
BCNTC	BIT COUNTER	621	@BCNTC	---	---	1212
BDSL	DOUBLE SIGNED BINARY-TO-BCD	473	@BDSL	---	---	507
BEND	BLOCK PROGRAM END	801	---	---	---	1128
BIN	BCD-TO-BINARY	023	@BIN	---	---	466
BINL	DOUBLE BCD-TO-DOUBLE BINARY	058	@BINL	---	---	467
BINS	SIGNED BCD-TO-BINARY	470	@BINS	---	---	499
BISL	DOUBLE SIGNED BCD-TO-BINARY	472	@BISL	---	---	502
BPPS	BLOCK PROGRAM PAUSE	811	---	---	---	1131
BPRG	BLOCK PROGRAM BEGIN	096	---	---	---	1128
BPRS	BLOCK PROGRAM RESTART	812	---	---	---	1131
BREAK	BREAK LOOP	514	---	---	---	232
BSET	BLOCK SET	071	@BSET	---	---	331

C

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
CADD	CALENDAR ADD	730	@CADD	---	---	1061
CCL	LOAD CONDITION FLAGS	283	@CCL	---	---	1112
CCS	SAVE CONDITION FLAGS	282	@CCS	---	---	1110
CJP	CONDITIONAL JUMP	510	---	---	---	223
CJPN	CONDITIONAL JUMP	511	---	---	---	223
CLC	CLEAR CARRY	041	@CLC	---	---	1105
CLI	CLEAR INTERRUPT	691	@CLI	---	---	809
CLR$	CLEAR STRING	666	@CLR$	---	---	1182

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
CMND	DELIVER COMMAND	490	@CMND	---	---	1003
CMP	COMPARE	020	---	---	!CMP	287
CMPL	DOUBLE COMPARE	060	---	---	---	290
CNR	RESET TIMER/COUNTER	545	@CNR	---	---	267
CNRX	RESET TIMER/COUNTER	548	@CNRX	---	---	267
CNT	COUNTER	---	---	---	---	260
CNTX	COUNTER	546	---	---	---	260
CNTR	REVERSIBLE COUNTER	012	---	---	---	263
CNTRX	REVERSIBLE COUNTER	548	---	---	---	263
CNTW	COUNTER WAIT	814	---	---	---	1147
CNTWX	COUNTER WAIT	818	---	---	---	1147
COLL	DATA COLLECT	081	@COLL	---	---	338
COLLC	DATA COLLECT	567	@COLLC	---	---	1206
COLM	LINE TO COLUMN	064	@COLM	---	---	496
COM	COMPLEMENT	029	---	---	---	531
COML	DOUBLE COMPLEMENT	614	@COML	---	---	533
COS	COSINE	461	@COS	---	---	583
COSD	DOUBLE COSINE	852	@COSD	---	---	639
CPS	SIGNED BINARY COMPARE	114	---	---	!CPS	293
CPSL	DOUBLE SIGNED BINARY COMPARE	115	---	---	---	296
CSUB	CALENDAR SUBTRACT	731	@CSUB	---	---	1065
CTBL	COMPARISON TABLE LOAD	882	@CTBL	---	---	837

D

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
DATE	CLOCK ADJUSTMENT	735	@DATE	---	---	1073
DBL	16-BIT BINARY TO DOUBLE FLOATING	843	@DBL	---	---	623
DBLL	32-BIT BINARY TO DOUBLE FLOATING	844	@DBLL	---	---	624
DEG	RADIANS-TO DEGREES	459	@DEG	---	---	579
DEGD	DOUBLE RADIANS TO DEGREES	850	@RADD	---	---	634
DEL$	DELETE STRING	658	@DEL$	---	---	1178
DI	DISABLE INTERRUPTS	693	@DI	---	---	814
DIFD	DIFFERENTIATE DOWN	014	---	---	!DIFD	184
DIFU	DIFFERENTIATE UP	013	---	---	!DIFU	184
DIM	DIMENSION RECORD TABLE	631	@DIM	---	---	678
DIST	SINGLE WORD DISTRIBUTE	080	@DIST	---	---	336
DISTC	SINGLE WORD DISTRIBUTE	566	@DISTC	---	---	1203
DLNK	CPU BUS UNIT I/O REFRESH	226	@DLNK	---	---	921

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
DMPX	DATA ENCODER	077	@DMPX	---	---	482
DOWN	CONDITION OFF	522	---	---	---	173
DSW	DIGITAL SWITCH INPUT	210	---	---	---	890

E

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
ECHRD	EXPLICIT WORD READ	723	@ECHRD	---	---	1034
ECHWR	EXPLICIT WORD WRITE	724	@ECHWR	---	---	1038
EGATR	EXPLICIT GET ATTRIBUTE	721	@EGATR	---	---	1021
EI	ENABLE INTERRUPTS	694	---	---	---	816
ELSE	ELSE	803	---	---	---	1133
EMBC	SELECT EM BANK	281	@EMBC	---	---	1106
END	END	001	---	---	---	197
ESATR	EXPLICIT SET ATTRIBUTE	722	@ESATR	---	---	1028
EXIT NOT (Operand)	CONDITIONAL BLOCK EXIT NOT	806	---	---	---	1137
EXIT (Eingangsbedingung)	CONDITIONAL BLOCK EXIT	806	---	---	---	1137
EXIT (Operand)	CONDITIONAL BLOCK EXIT	806	---	---	---	1137
EXP	EXPONENT	467	@EXP	---	---	595
EXPD	DOUBLE EXPONENT	858	@EXPD	---	---	651
EXPLT	EXPLICIT MESSAGE SEND	720	@EXPLT	---	---	1013

F

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
FAL	FAILURE ALARM	006	@FAL	---	---	1079
FALS	SEVERE FAILURE ALARM	007	---	---	---	1087
FCS	FRAME CHECKSUM	180	@FCS	---	---	700
FDIV	FLOATING POINT DIVIDE	079	@FDIV	---	---	552
FIFO	FIRST IN FIRST OUT	633	@FIFO	---	---	672
FIND$	FIND IN STRING	660	@FIND$	---	---	1171
FIX	FLOATING TO 16-BIT	450	@FIX	---	---	563
FIXD	DOUBLE FLOATING TO 16-BIT BINARY	841	@FIXD	---	---	620
FIXL	FLOATING TO 32-BIT	451	@FIXL	---	---	565
FIXLD	DOUBLE FLOATING TO 32-BIT BINARY	842	@FIXLD	---	---	621
FLT	16-BIT TO FLOATING	452	@FLT	---	---	566
FLTL	32-BIT TO FLOATING	453	@FLTL	---	---	568
FOR	FOR-NEXT LOOPS	512	---	---	---	229
FPD	FAILURE POINT DETECTION	269	---	---	---	1095
FREAD	READ DATA FILE	700	@FREAD	---	---	1045

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
FRMCV	CONVERT ADDRESS FROM CV	284	@FRMCV	---	---	1113
FSTR	FLOATING POINT TO ASCII	448	@FSTR	---	---	604
FWRIT	WRITE DATA FILE	701	@FWRIT	---	---	1052
FVAL	ASCII TO FLOATING POINT	449	@FVAL	---	---	609

G

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
GETID	GET VARIABLE ID	286	@GETID	---	---	1214
GETR	GET RECORD NUMBER	636	@GETR	---	---	683
GRET	GLOBAL SUBROUTINE RETURN	752	---	---	---	797
GRY	GRAY CODE CONVERSION	474	@GRY	---	---	511
GSBN	GLOBAL SUBROUTINE ENTRY	751	---	---	---	794
GSBS	GLOBAL SUBROUTINE CALL	750	@GSBS	---	---	786

H

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
HEX	ASCII TO HEX	162	@HEX	---	---	490
HKY	HEXADECIMAL KEY INPUT	212	---	---	---	899
HMS	SECONDS TO HOURS	066	@HMS	---	---	1070

I

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
IEND	IF END	804	---	---	---	1133
IF NOT (Operand)	IF NOT	802	---	---	---	1133
IF (Eingangsbedingung)	IF	802	---	---	---	1133
IF (Operand)	IF	802	---	---	---	1133
IL	INTERLOCK	002	---	---	---	201
ILC	INTERLOCK CLEAR	003	---	---	---	201
INI	MODE CONTROL	880	@INI	---	---	823
INS$	INS$	657	@INS$	---	---	1184
IORD	INTELLIGENT I/O READ	222	@IORD	---	---	913
IORF	I/O REFRESH	097	@IORF	---	---	885
IORS	ENABLE PERIPHERAL SERVICING	288	---	---	---	1123
IOSP	DISABLE PERIPHERAL SERVICING	287	@IOSP	---	---	1121
IOWR	INTELLIGENT I/O WRITE	223	@IOWR	---	---	917

J

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
JME	JUMP END	005	---	---	---	219
JME0	MULTIPLE JUMP END	516	---	---	---	227
JMP	JUMP	004	---	---	---	219
JMP0	MULTIPLE JUMP	515	---	---	---	227

K

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
KEEP	KEEP	011	---	---	!KEEP	180

L

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
LD	LOAD	---	@LD	%LD	!LD	153
LD <	LOAD LESS THAN	310	---	---	---	275
LD <$	LOAD STRING LESS THAN	672	---	---	---	1187
LD <D	LOAD DOUBLE FLOATING LESS THAN	337	---	---	---	657
LD < DT	LOAD TIME LESS THAN	343	---	---	---	281
LD <F	LOAD FLOATING LESS THAN	331	---	---	---	600
LD <>	LOAD NOT EQUAL	305	---	---	---	275
LD <>$	LOAD STRING NOT EQUAL	671	---	---	---	1187
LD <>D	LOAD DOUBLE FLOATING NOT EQUAL	336	---	---	---	657
LD <> DT	LOAD TIME NOT EQUAL	342	---	---	---	281
LD <>F	LOAD FLOATING NOT EQUAL	330	---	---	---	600
LD <>L	LOAD DOUBLE NOT EQUAL	306	---	---	---	275
LD <>S	LOAD SIGNED NOT EQUAL	307	---	---	---	275
LD <>SL	LOAD DOUBLE SIGNED NOT EQUAL	308	---	---	---	275
LD <L	LOAD DOUBLE LESS THAN	311	---	---	---	275
LD <S	LOAD SIGNED LESS THAN	312	---	---	---	275
LD <SL	LOAD DOUBLE SIGNED LESS THAN	313	---	---	---	275
LD =	LOAD EQUAL	300	---	---	---	275
LD =$	LOAD STRING EQUALS	670	---	---	---	1187
LD =D	LOAD DOUBLE FLOATING EQUAL	335	---	---	---	657
LD =DT	LOAD TIME EQUAL	341	---	---	---	281
LD =F	LOAD FLOATING EQUAL	329	---	---	---	600
LD =L	LOAD DOUBLE EQUAL	301	---	---	---	275
LD =S	LOAD SIGNED EQUAL	302	---	---	---	275
LD =SL	LOAD DOUBLE SIGNED EQUAL	303	---	---	---	275

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
LD >	LOAD GREATER THAN	320	---	---	---	275
LD >$	LOAD STRING GREATER THAN	674	---	---	---	1187
LD >D	LOAD DOUBLE FLOATING GREATER THAN	339	---	---	---	657
LD >DT	LOAD TIME GREATER THAN	345	---	---	---	281
LD >F	LOAD FLOATING GREATER THAN	333	---	---	---	600
LD >L	LOAD DOUBLE GREATER THAN	321	---	---	---	275
LD >S	LOAD SIGNED GREATER THAN	322	---	---	---	275
LD >SL	LOAD DOUBLE SIGNED GREATER THAN	323	---	---	---	275
LD NOT	LOAD NOT	---	---	---	!LD NOT	155
LD TST	LOAD BIT TEST	350	---	---	---	174
LD TSTN	LOAD BIT TEST	351	---	---	---	174
LD <=	LOAD LESS THAN OR EQUAL	315	---	---	---	275
LD <=$	LOAD STRING LESS THAN OR EQUAL	673	---	---	---	1187
LD <=D	LOAD DOUBLE FLOATING LESS THAN OR EQUAL	338	---	---	---	657
LD <=DT	LOAD TIME LESS THAN OR EQUAL	344	---	---	---	281
LD <=F	LOAD FLOATING LESS THAN OR EQUAL	332	---	---	---	600
LD <=L	LOAD DOUBLE LESS THAN OR EQUAL	316	---	---	---	275
LD <=S	LOAD SIGNED LESS THAN OR EQUAL	317	---	---	---	275
LD <=SL	LOAD DOUBLE SIGNED LESS THAN OR EQUAL	318	---	---	---	275
LD >=	LOAD GREATER THAN OR EQUAL	325	---	---	---	275
LD >=$	LOAD STRING GREATER THAN OR EQUALS	675	---	---	---	1187
LD >=D	LOAD DOUBLE FLOATING GREATER THAN OR EQUAL	340	---	---	---	657
LD >=DT	LOAD TIME GREATER THAN OR EQUAL	346	---	---	---	281
LD >=F	LOAD FLOATING GREATER THAN OR EQUAL	334	---	---	---	600
LD >=L	LOAD DOUBLE GREATER THAN OR EQUAL	326	---	---	---	275
LD >=S	LOAD SIGNED GREATER THAN OR EQUAL	327	---	---	---	275
LD >=SL	LOAD DOUBLE SIGNED GREATER THAN OR EQUAL	328	---	---	---	275
LEFT$	GET STRING LEFT	652	@LEFT$	---	---	1164
LEN$	STRING LENGTH	650	@LEN$	---	---	1173
LEND NOT (Operand)	LOOP END NOT	810	---	---	---	1153

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
LEND (Eingangsbedingung)	LOOP END	810	---	---	---	1153
LEND (Operand)	LOOP END	810	---	---	---	1153
LIFO	LAST IN FIRST OUT	634	@LIFO	---	---	675
LINE	COLUMN TO LINE	063	@LINE	---	---	494
LMT	LIMIT CONTROL	680	@LMT	---	---	741
LOG	LOGARITHM	468	@LOG	---	---	597
LOGD	DOUBLE LOGARITHM	859	@LOGD	---	---	653
LOOP	LOOP	809	---	---	---	1153

M

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
MAX	FIND MAXIMUM	182	@MAX	---	---	689
MCMP	MULTIPLE COMPARE	019	@MCMP	---	---	299
MCRO	MACRO	099	@MCRO	---	---	779
MID$	GET STRING MIDDLE	654	@MID$	---	---	1168
MILC	MULTI-INTERLOCK CLEAR	519	---	---	---	205
MILH	MULTI-INTERLOCK DIFFERENTIATION HOLD	517	---	---	---	205
MILR	MULTI-INTERLOCK DIFFERENTIATION RELEASE	518	---	---	---	205
MIN	FIND MINIMUM	183	@MIN	---	---	693
MLPX	DATA DECODER	076	@MLPX	---	---	477
MOV	MOVE	021	@MOV	---	!MOV	315
MOV$	MOVE STRING	664	@MOV$	---	---	1159
MOVB	MOVE BIT	082	@MOVB	---	---	321
MOVBC	MOVE BIT	568	@MOVBC	---	---	1211
MOVD	MOVE DIGIT	083	@MOVD	---	---	323
MOVL	DOUBLE MOVE	498	@MOVL	---	---	318
MOVR	MOVE TO REGISTER	560	@MOVR	---	---	340
MOVRW	MOVE TIMER/COUNTER PV TO REGISTER	561	---	---	---	342
MSG	DISPLAY MESSAGE	046	@MSG	---	---	1058
MSKR	READ INTERRUPT MASK	692	@MSKR	---	---	804
MSKS	SET INTERRUPT MASK	690	@MSKS	---	---	798
MTIM	MULTI-OUTPUT TIMER	543	---	---	---	254
MTIMX	MULTI-OUTPUT TIMER	554	---	---	---	254
MTR	MATRIX INPUT	213	---	---	---	904
MVN	MOVE NOT	022	@MVN	---	---	317
MVNL	DOUBLE MOVE NOT	499	@MVNL	---	---	320

N

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
NASL	SHIFT N-BITS LEFT	580	@NASL	---	---	381
NASR	SHIFT N-BITS RIGHT	581	@NASR	---	---	387
NEG	2'S COMPLEMENT	160	@NEG	---	---	472
NEGL	DOUBLE 2'S COMPLEMENT	161	@NEGL	---	---	474
NEXT	FOR-NEXT LOOPS	513	---	---	---	229
NOP	NO OPERATION	000	---	---	---	198
NOT	NOT	520	---	---	---	172
NSFL	SHIFT N-BIT DATA LEFT	578	@NSFL	---	---	377
NSFR	SHIFT N-BIT DATA RIGHT	579	@NSFR	---	---	379
NSLL	DOUBLE SHIFT N-BITS LEFT	582	@NSLL	---	---	384
NSRL	DOUBLE SHIFT N-BITS RIGHT	583	@NSRL	---	---	389

O

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
OR	OR	---	@OR	%OR	!OR	161
OR <	OR LESS THAN	310	---	---	---	275
OR <$	OR STRING LESS THAN	672	---	---	---	1187
OR <>	OR NOT EQUAL	305	---	---	---	275
OR <>$	OR STRING NOT EQUAL	671	---	---	---	1187
OR <>D	OR DOUBLE FLOATING NOT EQUAL	336	---	---	---	657
OR <> DT	OR TIME NOT EQUAL	342	---	---	---	281
OR <>F	OR FLOATING NOT EQUAL	330	---	---	---	600
OR <>L	OR DOUBLE NOT EQUAL	306	---	---	---	275
OR <>S	OR SIGNED NOT EQUAL	307	---	---	---	275
OR <>SL	OR DOUBLE SIGNED NOT EQUAL	308	---	---	---	275
OR <D	OR DOUBLE FLOATING LESS THAN	337	---	---	---	657
OR <DT	OR TIME LESS THAN	343	---	---	---	281
OR <F	OR FLOATING LESS THAN	331	---	---	---	600
OR <L	OR DOUBLE LESS THAN	311	---	---	---	275
OR <S	OR SIGNED LESS THAN	312	---	---	---	275
OR <SL	OR DOUBLE SIGNED LESS THAN	313	---	---	---	275
OR =	OR EQUAL	300	---	---	---	275
OR =$	OR STRING EQUALS	670	---	---	---	1187
OR =D	OR DOUBLE FLOATING EQUAL	335	---	---	---	657
OR =DT	OR TIME EQUAL	341	---	---	---	281
OR =F	OR FLOATING EQUAL	329	---	---	---	600
OR =L	OR DOUBLE EQUAL	301	---	---	---	275
OR =S	OR SIGNED EQUAL	302	---	---	---	275

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
OR =SL	OR DOUBLE SIGNED EQUAL	303	---	---	---	275
OR >	OR GREATER THAN	320	---	---	---	275
OR >$	OR STRING GREATER THAN	674	---	---	---	1187
OR >D	OR DOUBLE FLOATING GREATER THAN	339	---	---	---	657
OR >DT	OR TIME GREATER THAN	345	---	---	---	281
OR >F	OR FLOATING GREATER THAN	333	---	---	---	600
OR >L	OR DOUBLE GREATER THAN	321	---	---	---	275
OR >S	OR SIGNED GREATER THAN	322	---	---	---	275
OR >SL	OR DOUBLE SIGNED GREATER THAN	323	---	---	---	275
OR LD	OR LOAD	---	---	---	---	166
OR NOT	OR NOT	---	---	---	!OR NOT	163
OR TST	OR BIT TEST	350	---	---	---	174
OR TSTN	OR BIT TEST	351	---	---	---	174
OR <=	OR LESS THAN OR EQUAL	315	---	---	---	275
OR <=$	OR STRING LESS THAN OR EQUALS	673	---	---	---	1187
OR <=D	OR DOUBLE FLOATING LESS THAN OR EQUAL	338	---	---	---	657
OR <=DT	OR TIME LESS THAN OR EQUAL	344	---	---	---	281
OR <=F	OR FLOATING LESS THAN OR EQUAL	332	---	---	---	600
OR <=L	OR DOUBLE LESS THAN OR EQUAL	316	---	---	---	275
OR <=S	OR SIGNED LESS THAN OR EQUAL	317	---	---	---	275
OR <=SL	OR DOUBLE SIGNED LESS THAN OR EQUAL	318	---	---	---	275
OR >=	OR GREATER THAN OR EQUAL	325	---	---	---	275
OR >=$	OR STRING GREATER THAN OR EQUALS	675	---	---	---	1187
OR >=D	OR DOUBLE FLOATING GREATER THAN OR EQUAL	340	---	---	---	657
OR >=DT	OR TIME GREATER THAN OR EQUAL	346	---	---	---	281
OR >=F	OR FLOATING GREATER THAN OR EQUAL	334	---	---	---	600
OR >=L	OR DOUBLE GREATER THAN OR EQUAL	326	---	---	---	275
OR >=S	OR SIGNED GREATER THAN OR EQUAL	327	---	---	---	275
OR >=SL	OR DOUBLE SIGNED GREATER THAN OR EQUAL	328	---	---	---	275
ORG	ORIGIN SEARCH	889	@ORG	---	---	862
ORW	LOGICAL OR	035	@ORW	---	---	520
ORWL	DOUBLE LOGICAL OR	611	@ORWL	---	---	522
OUT	OUTPUT	---	---	---	!OUT	177

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
OUTB	SINGLE BIT OUTPUT	534	@OUTB	---	!OUTB	195
OUT NOT	OUTPUT NOT	---	---	---	!OUT NOT	178

P

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
PID	PID CONTROL	190	---	---	---	720
PIDAT	PID CONTROL WITH AUTOTUNING	191	---	---	---	731
PMCR	PROTOCOL MACRO	260	@PMCR	---	---	928
PRV	HIGH-SPEED COUNTER PV READ	881	@PRV	---	---	827
PRV2	COUNTER FREQUENCY CONVERT	883	@PRV2	---	---	833
PULS	SET PULSES	886	@PULS	---	---	846
PLS2	PULSE OUTPUT	887	@PLS2	---	---	849
PUSH	PUSH ONTO STACK	632	@PUSH	---	---	669
PWM	PULSE WITH VARIABLE DUTY FACTOR	891	@PWM	---	---	865
PWR	EXPONENTIAL POWER	840	@PWR	---	---	599
PWRD	DOUBLE EXPONENTIAL POWER	860	@PWRD	---	---	655

R

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
RAD	DEGREES TO RADIANS	458	@RAD	---	---	597
RADD	DOUBLE DEGREES TO RADIANS	849	@RADD	---	---	634
RECV	NETWORK RECEIVE	098	@RECV	---	---	997
RET	SUBROUTINE RETURN	093	---	---	---	786
RGHT$	GET STRING RIGHT	653	@RGHT$	---	---	1166
RLNC	ROTATE LEFT WITHOUT CARRY	574	@RLNC	---	---	367
RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	576	@RLNL	---	---	369
ROL	ROTATE LEFT	027	@ROL	---	---	360
ROLL	DOUBLE ROTATE LEFT	572	@ROLL	---	---	362
ROOT	BCD SQUARE ROOT	072	@ROOT	---	---	536
ROR	ROTATE RIGHT	028	@ROR	---	---	364
RORL	DOUBLE ROTATE RIGHT	573	@RORL	---	---	365
ROTB	BINARY ROOT	620	@ROTB	---	---	534
RPLC$	REPLACE IN STRING	661	@RPLC$	---	---	1175
RRNC	ROTATE RIGHT WITHOUT CARRY	575	@RRNC	---	---	371
RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	577	@RRNL	---	---	372
RSET	RESET	---	@RSET	%RSET	!RSET	187
RSTA	MULTIPLE BIT RESET	531	@RSTA	---	---	189
RSTB	SINGLE BIT RESET	533	@RSTB	---	!RSTB	192

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
RXD	RECEIVE	235	@RXD	---	---	944
RXDU	RECEIVE VIA SERIAL COMMUNICATIONS UNIT	255	@RXDU	---	---	960

S

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
SBN	SUBROUTINE ENTRY	092	---	---	---	783
SBS	SUBROUTINE CALL	091	@SBS	---	---	773
SCL	SCALING	194	@SCL	---	---	757
SCL2	SCALING 2	486	@SCL2	---	---	762
SCL3	SCALING 3	487	@SCL3	---	---	766
SDEC	7-SEGMENT DECODER	078	@SDEC	---	---	928
SDEL	STACK DATA DELETE	642	@SDEL	---	---	716
SEC	HOURS TO SECONDS	065	@SEC	---	---	1068
SEND	NETWORK SEND	090	@SEND	---	---	991
SET	SET	---	@SET	%SET	!SET	187
SETA	MULTIPLE BIT SET	530	@SETA	---	---	189
SETB	SINGLE BIT SET	532	@SETB	---	!SETB	192
SETR	SET RECORD LOCATION	635	@SETR	---	---	681
SFT	SHIFT REGISTER	010	---	---	---	345
SFTR	REVERSIBLE SHIFT REGISTER	084	@SFTR	---	---	346
SIGN	16-BIT TO 32-BIT SIGNED BINARY	600	@SIGN	---	---	476
SIN	SINE	460	@SIN	---	---	581
SIND	DOUBLE SINE	851	@SIND	---	---	637
SINS	STACK DATA INSERT	641	@SINS	---	---	713
SLD	ONE DIGIT SHIFT LEFT	074	@SLD	---	---	374
SNUM	STACK SIZE READ	638	@SNUM	---	---	704
SNXT	STEP START	009	---	---	---	868
SPED	SPEED OUTPUT	885	@SPED	---	---	841
SQRT	SQUARE ROOT	466	@SQRT	---	---	593
SQRTD	DOUBLE SQUARE ROOT	857	@SQRTD	---	---	649
SRCH	DATA SEARCH	181	@SRCH	---	---	685
SRD	ONE DIGIT SHIFT RIGHT	075	@SRD	---	---	376
SREAD	STACK DATA READ	639	@SREAD	---	---	707
SSET	SET STACK	630	@SSET	---	---	666
STC	SET CARRY	040	@STC	---	---	1104
STEP	STEP DEFINE	008	---	---	---	868
STUP	CHANGE SERIAL PORT SETUP	237	@STUP	---	---	968
SUM	SUM	184	@SUM	---	---	697
SWAP	SWAP BYTES	637	@SWAP	---	---	687
SWRIT	STACK DATA WRITE	640	@SWRIT	---	---	710

T

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
TAN	TANGENT	462	@TAN	---	---	585
TAND	DOUBLE TANGENT	853	@TAND	---	---	641
TCMP	TABLE COMPARE	085	@TCMP	---	---	301
TIM	TIMER	---	---	---	---	235
TIMH	HIGH-SPEED TIMER	015	---	---	---	240
TIMHX	HIGH-SPEED TIMER	551	---	---	---	240
TIML	LONG TIMER	542	---	---	---	251
TIMLX	LONG TIMER	553	---	---	---	251
TIMW	TIMER WAIT	813	---	---	---	1144
TIMWX	TIMER WAIT	816	---	---	---	1144
TIMX	TIMER	505	---	---	---	235
TKOF	TASK OFF	821	@TKOF	---	---	1196
TKON	TASK ON	820	@TKON	---	---	1192
TKY	TEN KEY INPUT	211	@TKY	---	---	896
TMHH	ONE-MS TIMER	540	---	---	---	244
TMHHX	ONE-MS TIMER	552	---	---	---	244
TMHW	HIGH-SPEED TIMER WAIT	815	---	---	---	1150
TMHWX	HIGH-SPEED TIMER WAIT	817	---	---	---	1150
TOCV	CONVERT ADDRESS TO CV	285	@TOCV	---	---	1117
TPO	TIME-PROPORTIONAL OUTPUT	685	---	---	---	749
TRSM	TRACE MEMORY SAMPLING	045	---	---	---	1075
TTIM	ACCUMULATIVE TIMER	087	---	---	---	247
TTIMX	ACCUMULATIVE TIMER	555	---	---	---	247
TXD	TRANSMIT	236	@TXD	---	---	937
TXDU	TRANSMIT VIA SERIAL COMMUNICATIONS UNIT	256	@TXDU	---	---	952

U

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
UP	CONDITION ON	521	---	---	---	173

W

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
WAIT NOT (Operand)	ONE CYCLE AND WAIT NOT	805	---	---	---	1140
WAIT (Eingangsbedingung)	ONE CYCLE AND WAIT	805	---	---	---	1140
WAIT (Operand)	ONE CYCLE AND WAIT	805	---	---	---	1140
WDT	EXTEND MAXIMUM CYCLE TIME	094	@WDT	---	---	1108
WSFT	WORD SHIFT	016	@WSFT	---	---	352

X

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
XCGL	DOUBLE DATA EXCHANGE	562	@XCGL	---	---	334
XCHG	DATA EXCHANGE	073	@XCHG	---	---	333
XCHG$	EXCHANGE STRING	665	@XCHG$	---	---	1180
XFER	BLOCK TRANSFER	070	@XFER	---	---	328
XFERC	BLOCK TRANSFER	565	@XFERC	---	---	1201
XFRB	MULTIPLE BIT TRANSFER	062	@XFRB	---	---	326
XNRL	DOUBLE EXCLUSIVE NOR	613	@XNRL	---	---	529
XNRW	EXCLUSIVE NOR	037	@XNRW	---	---	528
XORL	DOUBLE EXCLUSIVE OR	612	@XORL	---	---	526
XORW	EXCLUSIVE OR	036	@XORW	---	---	524

Z

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
ZCP	AREA RANGE COMPARE	088	---	---	---	310
ZCPL	DOUBLE AREA RANGE COMPARE	116	---	---	---	313
ZONE	DEAD ZONE CONTROL	682	@ZONE	---	---	746

Symbole

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
7SEG	7-SEGMENT DISPLAY OUTPUT	214	---	---	---	908
+	SIGNED BINARY ADD WITHOUT CARRY	400	@+	---	---	410
+$	CONCATENATE STRING	656	@+$	---	---	1161
++	INCREMENT BINARY	590	@++	---	---	393
++B	INCREMENT BCD	594	@++B	---	---	401
++BL	DOUBLE INCREMENT BCD	595	@++BL	---	---	403
++L	DOUBLE INCREMENT BINARY	591	@++L	---	---	395
+B	BCD ADD WITHOUT CARRY	404	@+B	---	---	418
+BC	BCD ADD WITH CARRY	406	@+BC	---	---	421
+BCL	DOUBLE BCD ADD WITH CARRY	407	@+BCL	---	---	423
+BL	DOUBLE BCD ADD WITHOUT CARRY	405	@+BL	---	---	419
+C	SIGNED BINARY ADD WITH CARRY	402	@+C	---	---	414
+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	403	@+CL	---	---	416
+D	DOUBLE FLOATING-POINT ADD	845	@+D	---	---	626
+F	FLOATING-POINT ADD	454	@+F	---	---	570

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	401	@+L	---	---	412
–	SIGNED BINARY SUBTRACT WITHOUT CARRY	410	@–	---	---	424
– –	DECREMENT BINARY	592	@– –	---	---	397
– –B	DECREMENT BCD	596	@– –B	---	---	405
– –BL	DOUBLE DECREMENT BCD	597	@– –BL	---	---	407
– –L	DOUBLE DECREMENT BINARY	593	@– –L	---	---	399
–B	BCD SUBTRACT WITHOUT CARRY	414	@–B	---	---	435
–BC	BCD SUBTRACT WITH CARRY	416	@–BC	---	---	440
–BCL	DOUBLE BCD SUBTRACT WITH CARRY	417	@–BCL	---	---	441
–BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	415	@–BL	---	---	436
–C	SIGNED BINARY SUBTRACT WITH CARRY	412	@–C	---	---	430
–CL	DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	413	@–CL	---	---	432
–D	DOUBLE FLOATING-POINT SUBTRACT	846	@–D	---	---	628
–F	FLOATING-POINT SUBTRACT	455	@–F	---	---	572
*	SIGNED BINARY MULTIPLY	420	@*	---	---	443
*B	BCD MULTIPLY	424	@*B	---	---	450
*BL	DOUBLE BCD MULTIPLY	425	@*BL	---	---	452
*D	DOUBLE FLOATING-POINT MULTIPLY	847	@*D	---	---	630
*F	FLOATING-POINT MULTIPLY	456	@*F	---	---	574
*L	DOUBLE SIGNED BINARY MULTIPLY	421	@*L	---	---	445
*U	UNSIGNED BINARY MULTIPLY	422	@*U	---	---	447
*UL	DOUBLE UNSIGNED BINARY MULTIPLY	423	@*UL	---	---	449
–L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	411	@–L	---	---	426
/	SIGNED BINARY DIVIDE	430	@/	---	---	454
/B	BCD DIVIDE	434	@/B	---	---	462
/BL	DOUBLE BCD DIVIDE	435	@/BL	---	---	464
/D	DOUBLE FLOATING-POINT DIVIDE	848	@/D	---	---	632
/F	FLOATING-POINT DIVIDE	457	@/F	---	---	576
/L	DOUBLE SIGNED BINARY DIVIDE	431	@/L	---	---	456

AWL	Befehl	Funktionscode	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
/U	UNSIGNED BINARY DIVIDE	432	@/U	---	---	458
/UL	DOUBLE UNSIGNED BINARY DIVIDE	433	@/UL	---	---	460

2-4 Liste der Befehl nach Funktionscode

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
---	LD	LOAD	@LD	%LD	!LD	153
---	LD NOT	LOAD NOT	---	---	!LD NOT	155
---	AND	AND	@AND	%AND	!AND	157
---	AND NOT	AND NOT	---	---	!AND NOT	159
---	OR	OR	@OR	%OR	!OR	161
---	OR NOT	OR NOT	---	---	!OR NOT	163
---	AND LD	AND LOAD	---	---	---	164
---	OR LD	OR LOAD	---	---	---	166
---	OUT	OUTPUT	---	---	!OUT	177
---	OUT NOT	OUTPUT NOT	---	---	!OUT NOT	178
---	SET	SET	@SET	%SET	!SET	187
---	RSET	RESET	@RSET	%RSET	!RSET	187
---	TIM	TIMER	---	---	---	235
---	TIMX	TIMER	---	---	---	235
---	CNT	COUNTER	---	---	---	260
000	NOP	NO OPERATION	---	---	---	198
001	END	END	---	---	---	197
002	IL	INTERLOCK	---	---	---	201
003	ILC	INTERLOCK CLEAR	---	---	---	201
004	JMP	JUMP	---	---	---	219
005	JME	JUMP END	---	---	---	219
006	FAL	FAILURE ALARM	@FAL	---	---	1079
007	FALS	SEVERE FAILURE ALARM	---	---	---	1087
008	STEP	STEP DEFINE	---	---	---	868
009	SNXT	STEP START	---	---	---	868
010	SFT	SHIFT REGISTER	---	---	---	345
011	KEEP	KEEP	---	---	!KEEP	180
012	CNTR	REVERSIBLE COUNTER	---	---	---	263
013	DIFU	DIFFERENTIATE UP	---	---	!DIFU	184
014	DIFD	DIFFERENTIATE DOWN	---	---	!DIFD	184
015	TIMH	HIGH-SPEED TIMER	---	---	---	240
016	WSFT	WORD SHIFT	@WSFT	---	---	352
017	ASFT	ASYNCHRONOUS SHIFT REGISTER	@ASFT	---	---	349
019	MCMP	MULTIPLE COMPARE	@MCMP	---	---	299
020	CMP	UNSIGNED COMPARE	---	---	!CMP	287
021	MOV	MOVE	@MOV	---	!MOV	315
022	MVN	MOVE NOT	@MVN	---	---	317
023	BIN	BCD-TO-BINARY	@BIN	---	---	466
024	BCD	BINARY-TO-BCD	@BCD	---	---	469
025	ASL	ARITHMETIC SHIFT LEFT	@ASL	---	---	354
026	ASR	ARITHMETIC SHIFT RIGHT	@ASR	---	---	357
027	ROL	ROTATE LEFT	@ROL	---	---	360
028	ROR	ROTATE RIGHT	@ROR	---	---	364
029	COM	COMPLEMENT	@COM	---	---	531
034	ANDW	LOGICAL AND	@ANDW	---	---	517
035	ORW	LOGICAL OR	@ORW	---	---	520

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
036	XORW	EXCLUSIVE OR	@XORW	---	---	524
037	XNRW	EXCLUSIVE NOR	@XNRW	---	---	528
040	STC	SET CARRY	@STC	---	---	1104
041	CLC	CLEAR CARRY	@CLC	---	---	1105
045	TRSM	TRACE MEMORY SAMPLING	---	---	---	1075
046	MSG	DISPLAY MESSAGE	@MSG	---	---	1058
058	BINL	DOUBLE BCD-TO-DOUBLE BINARY	@BINL	---	---	467
059	BCDL	DOUBLE BINARY-TO-BCD	@BCDL	---	---	470
060	CMPL	DOUBLE UNSIGNED COMPARE	---	---	---	290
062	XFRB	MULTIPLE BIT TRANSFER	@XFRB	---	---	326
063	LINE	COLUMN TO LINE	@LINE	---	---	494
064	COLM	LINE TO COLUMN	@COLM	---	---	496
065	SEC	HOURS TO SECONDS	@SEC	---	---	1068
066	HMS	SECONDS TO HOURS	@HMS	---	---	1070
067	BCNT	BIT COUNTER	@BCNT	---	---	556
068	BCMP	UNSIGNED BLOCK COMPARE	@BCMP	---	---	304
069	APR	ARITHMETIC PROCESS	@APR	---	---	540
070	XFER	BLOCK TRANSFER	@XFER	---	---	328
071	BSET	BLOCK SET	@BSET	---	---	331
072	ROOT	BCD SQUARE ROOT	@ROOT	---	---	536
073	XCHG	DATA EXCHANGE	@XCHG	---	---	333
074	SLD	ONE DIGIT SHIFT LEFT	@SLD	---	---	374
075	SRD	ONE DIGIT SHIFT RIGHT	@SRD	---	---	376
076	MLPX	DATA DECODER	@MLPX	---	---	477
077	DMPX	DATA ENCODER	@DMPX	---	---	482
078	SDEC	7-SEGMENT DECODER	@SDEC	---	---	928
079	FDIV	FLOATING POINT DIVIDE	@FDIV	---	---	552
080	DIST	SINGLE WORD DISTRIBUTE	@DIST	---	---	336
081	COLL	DATA COLLECT	@COLL	---	---	338
082	MOVB	MOVE BIT	@MOVB	---	---	321
083	MOVD	MOVE DIGIT	@MOVD	---	---	323
084	SFTR	REVERSIBLE SHIFT REGISTER	@SFTR	---	---	346
085	TCMP	TABLE COMPARE	@TCMP	---	---	301
086	ASC	ASCII CONVERT	@ASC	---	---	486
087	TTIM	ACCUMULATIVE TIMER	---	---	---	247
088	ZCP	AREA RANGE COMPARE	---	---	---	310
090	SEND	NETWORK SEND	@SEND	---	---	991
091	SBS	SUBROUTINE CALL	@SBS	---	---	773
092	SBN	SUBROUTINE ENTRY	---	---	---	783
093	RET	SUBROUTINE RETURN	---	---	---	786

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
094	WDT	EXTEND MAXIMUM CYCLE TIME	@WDT	---	---	1108
096	BPRG	BLOCK PROGRAM BEGIN	---	---	---	1128
097	IORF	I/O REFRESH	@IORF	---	---	885
098	RECV	NETWORK RECEIVE	@RECV	---	---	997
099	MCRO	MACRO	@MCRO	---	---	779
114	CPS	SIGNED BINARY COMPARE	---	---	!CPS	293
115	CPSL	DOUBLE SIGNED BINARY COMPARE	---	---	---	296
116	ZCPL	DOUBLE AREA RANGE COMPARE	---	---	---	313
160	NEG	2'S COMPLEMENT	@NEG	---	---	472
161	NEGL	DOUBLE 2'S COMPLEMENT	@NEGL	---	---	474
162	HEX	ASCII TO HEX	@HEX	---	---	490
180	FCS	FRAME CHECKSUM	@FCS	---	---	700
181	SRCH	DATA SEARCH	@SRCH	---	---	685
182	MAX	FIND MAXIMUM	@MAX	---	---	689
183	MIN	FIND MINIMUM	@MIN	---	---	693
184	SUM	SUM	@SUM	---	---	697
190	PID	PID CONTROL	---	---	---	720
191	PIDAT	PID CONTROL WITH AUTOTUNING	---	---	---	731
194	SCL	SCALING	@SCL	---	---	757
195	AVG	AVERAGE	---	---	---	769
210	DSW	DIGITAL SWITCH INPUT	---	---	---	890
211	TKY	TEN KEY INPUT	@TKY	---	---	896
212	HKY	HEXADECIMAL KEY INPUT	---	---	---	899
213	MTR	MATRIX INPUT	---	---	---	904
214	7SEG	7-SEGMENT DISPLAY OUTPUT	---	---	---	908
222	IORD	INTELLIGENT I/O READ	@IORD	---	---	913
223	IOWR	INTELLIGENT I/O WRITE	@IOWR	---	---	917
226	DLNK	CPU BUS UNIT I/O REFRESH	@DLNK	---	---	921
235	RXD	RECEIVE	@RXD	---	---	944
236	TXD	TRANSMIT	@TXD	---	---	937
255	RXDU	RECEIVE VIA SERIAL COMMUNICATIONS UNIT	@RXDU	---	---	960
256	TXDU	TRANSMIT VIA SERIAL COMMUNICA-TIONS UNIT	@TXDU	---	---	952
237	STUP	CHANGE SERIAL PORT SETUP	@STUP	---	---	968
260	PMCR	PROTOCOL MACRO	@PMCR	---	---	928
269	FPD	FAILURE POINT DETECTION	---	---	---	1095
281	EMBC	SELECT EM BANK	@EMBC	---	---	1106
282	CCS	SAVE CONDITION FLAGS	@CCS	---	---	1110
283	CCL	LOAD CONDITION FLAGS	@CCL	---	---	1112

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
284	FRMCV	CONVERT ADDRESS FROM CV	@FRMCV	---	---	1113
285	TOCV	CONVERT ADDRESS TO CV	@TOCV	---	---	1117
286	GETID	GET VARIABLE ID	@GETID	---	---	1214
287	IOSP	DISABLE PERI-PHERAL SERVICING	@IOSP	---	---	1121
288	IORS	ENABLE PERI-PHERAL SERVICING	---	---	---	1123
300	AND =	AND EQUAL	---	---	---	275
300	LD =	LOAD EQUAL	---	---	---	275
300	OR =	OR EQUAL	---	---	---	275
301	AND =L	AND DOUBLE EQUAL	---	---	---	275
301	LD =L	LOAD DOUBLE EQUAL	---	---	---	275
301	OR =L	OR DOUBLE EQUAL	---	---	---	275
302	AND =S	AND SIGNED EQUAL	---	---	---	275
302	LD =S	LOAD SIGNED EQUAL	---	---	---	275
302	OR =S	OR SIGNED EQUAL	---	---	---	275
303	AND =SL	AND DOUBLE SIGNED EQUAL	---	---	---	275
303	LD =SL	LOAD DOUBLE SIGNED EQUAL	---	---	---	275
303	OR =SL	OR DOUBLE SIGNED EQUAL	---	---	---	275
305	AND <>	AND NOT EQUAL	---	---	---	275
305	LD <>	LOAD NOT EQUAL	---	---	---	275
305	OR <>	OR NOT EQUAL	---	---	---	275
306	AND <>L	AND DOUBLE NOT EQUAL	---	---	---	275
306	LD <>L	LOAD DOUBLE NOT EQUAL	---	---	---	275
306	OR <>L	OR DOUBLE NOT EQUAL	---	---	---	275
307	AND <>S	AND SIGNED NOT EQUAL	---	---	---	275
307	LD <>S	LOAD SIGNED NOT EQUAL	---	---	---	275
307	OR <>S	OR SIGNED NOT EQUAL	---	---	---	275
308	AND <>SL	AND DOUBLE SIGNED NOT EQUAL	---	---	---	275
308	LD <>SL	LOAD DOUBLE SIGNED NOT EQUAL	---	---	---	275
308	OR <>SL	OR DOUBLE SIGNED NOT EQUAL	---	---	---	275
310	AND <	AND LESS THAN	---	---	---	275
310	LD <	LOAD LESS THAN	---	---	---	275
310	OR <	OR LESS THAN	---	---	---	275
311	AND <L	AND DOUBLE LESS THAN	---	---	---	275
311	LD <L	LOAD DOUBLE LESS THAN	---	---	---	275
311	OR <L	OR DOUBLE LESS THAN	---	---	---	275
312	AND <S	AND SIGNED LESS THAN	---	---	---	275
312	LD <S	LOAD SIGNED LESS THAN	---	---	---	275

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
312	OR <S	OR SIGNED LESS THAN	---	---	---	275
313	AND <SL	AND DOUBLE SIGNED LESS THAN	---	---	---	275
313	LD <SL	LOAD DOUBLE SIGNED LESS THAN	---	---	---	275
313	OR <SL	OR DOUBLE SIGNED LESS THAN	---	---	---	275
315	AND <=	AND LESS THAN OR EQUAL	---	---	---	275
315	LD <=	LOAD LESS THAN OR EQUAL	---	---	---	275
315	OR <=	OR LESS THAN OR EQUAL	---	---	---	275
316	AND <=L	AND DOUBLE LESS THAN OR EQUAL	---	---	---	275
316	LD <=L	LOAD DOUBLE LESS THAN OR EQUAL	---	---	---	275
316	OR <=L	OR DOUBLE LESS THAN OR EQUAL	---	---	---	275
317	AND <=S	AND SIGNED LESS THAN OR EQUAL	---	---	---	275
317	LD <=S	LOAD SIGNED LESS THAN OR EQUAL	---	---	---	275
317	OR <=S	OR SIGNED LESS THAN OR EQUAL	---	---	---	275
318	AND <=SL	AND DOUBLE SIGNED LESS THAN OR EQUAL	---	---	---	275
318	LD <=SL	LOAD DOUBLE SIGNED LESS THAN OR EQUAL	---	---	---	275
318	OR <=SL	OR DOUBLE SIGNED LESS THAN OR EQUAL	---	---	---	275
320	AND >	AND GREATER THAN	---	---	---	275
320	LD >	LOAD GREATER THAN	---	---	---	275
320	OR >	OR GREATER THAN	---	---	---	275
321	AND >L	AND DOUBLE GREATER THAN	---	---	---	275
321	LD >L	LOAD DOUBLE GREATER THAN	---	---	---	275
321	OR >L	OR DOUBLE GREATER THAN	---	---	---	275
322	AND >S	AND SIGNED GREATER THAN	---	---	---	275
322	LD >S	LOAD SIGNED GREATER THAN	---	---	---	275
322	OR >S	OR SIGNED GREATER THAN	---	---	---	275
323	AND >SL	AND DOUBLE SIGNED GREATER THAN	---	---	---	275
323	LD >SL	LOAD DOUBLE SIGNED GREATER THAN	---	---	---	275
323	OR >SL	OR DOUBLE SIGNED GREATER THAN	---	---	---	275
325	AND >=	AND GREATER THAN OR EQUAL	---	---	---	275
325	LD >=	LOAD GREATER THAN OR EQUAL	---	---	---	275

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
325	OR >=	OR GREATER THAN OR EQUAL	---	---	---	275
326	AND >=L	AND DOUBLE GREATER THAN OR EQUAL	---	---	---	275
326	LD >=L	LOAD DOUBLE GREATER THAN OR EQUAL	---	---	---	275
326	OR >=L	OR DOUBLE GREATER THAN OR EQUAL	---	---	---	275
327	AND >=S	AND SIGNED GREATER THAN OR EQUAL	---	---	---	275
327	LD >=S	LOAD SIGNED GREATER THAN OR EQUAL	---	---	---	275
327	OR >=S	OR SIGNED GREATER THAN OR EQUAL	---	---	---	275
328	AND >=SL	AND DOUBLE SIGNED GREATER THAN OR EQUAL	---	---	---	275
328	LD >=SL	LOAD DOUBLE SIGNED GREATER THAN OR EQUAL	---	---	---	275
328	OR >=SL	OR DOUBLE SIGNED GREATER THAN OR EQUAL	---	---	---	275
329	AND =F	AND FLOATING EQUAL	---	---	---	600
329	LD =F	LOAD FLOATING EQUAL	---	---	---	600
329	OR =F	OR FLOATING EQUAL	---	---	---	600
330	AND <>F	AND FLOATING NOT EQUAL	---	---	---	600
330	LD <>F	LOAD FLOATING NOT EQUAL	---	---	---	600
330	OR <>F	OR FLOATING NOT EQUAL	---	---	---	600
331	AND <F	AND FLOATING LESS THAN	---	---	---	600
331	LD <F	LOAD FLOATING LESS THAN	---	---	---	600
331	OR <F	OR FLOATING LESS THAN	---	---	---	600
332	AND <=F	AND FLOATING LESS THAN OR EQUAL	---	---	---	600
332	LD <=F	LOAD FLOATING LESS THAN OR EQUAL	---	---	---	600
332	OR <=F	OR FLOATING LESS THAN OR EQUAL	---	---	---	600
333	AND >F	AND FLOATING GREATER THAN	---	---	---	600
333	LD >F	LOAD FLOATING GREATER THAN	---	---	---	600
333	OR >F	OR FLOATING GREATER THAN	---	---	---	600
334	AND >=F	AND FLOATING GREATER THAN OR EQUAL	---	---	---	600
334	LD >=F	LOAD FLOATING GREATER THAN OR EQUAL	---	---	---	600
334	OR >=F	OR FLOATING GREATER THAN OR EQUAL	---	---	---	600
335	AND =D	AND DOUBLE FLOATING EQUAL	---	---	---	657

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
335	LD =D	LOAD DOUBLE FLOATING EQUAL	---	---	---	657
335	OR =D	OR DOUBLE FLOATING EQUAL	---	---	---	657
336	AND <>D	AND DOUBLE FLOATING NOT EQUAL	---	---	---	657
336	LD <>D	LOAD DOUBLE FLOATING NOT EQUAL	---	---	---	657
336	OR <>D	OR DOUBLE FLOATING NOT EQUAL	---	---	---	657
337	AND <D	AND DOUBLE FLOATING LESS THAN	---	---	---	657
337	LD <D	LOAD DOUBLE FLOATING LESS THAN	---	---	---	657
337	OR <D	OR DOUBLE FLOATING LESS THAN	---	---	---	657
338	AND <=D	AND DOUBLE FLOATING LESS THAN OR EQUAL	---	---	---	657
338	LD <=D	LOAD DOUBLE FLOATING LESS THAN OR EQUAL	---	---	---	657
338	OR <=D	OR DOUBLE FLOATING LESS THAN OR EQUAL	---	---	---	657
339	AND >D	AND DOUBLE FLOATING GREATER THAN	---	---	---	657
339	LD >D	LOAD DOUBLE FLOATING GREATER THAN	---	---	---	657
339	OR >D	OR DOUBLE FLOATING GREATER THAN	---	---	---	657
340	AND >=D	AND DOUBLE FLOATING GREATER THAN OR EQUAL	---	---	---	657
340	LD >=D	LOAD DOUBLE FLOATING GREATER THAN OR EQUAL	---	---	---	657
340	OR >=D	OR DOUBLE FLOATING GREATER THAN OR EQUAL	---	---	---	657
341	AND = DT	AND TIME EQUAL	---	---	---	281
341	LD = DT	LOAD TIME EQUAL	---	---	---	281
341	OR = DT	OR TIME EQUAL	---	---	---	281
342	AND <> DT	AND TIME NOT EQUAL	---	---	---	281
342	LD <> DT	LOAD TIME NOT EQUAL	---	---	---	281
342	OR <> DT	OR TIME NOT EQUAL	---	---	---	281
343	AND < DT	AND TIME LESS THAN	---	---	---	281
343	LD < DT	LOAD TIME LESS THAN	---	---	---	281
343	OR < DT	OR TIME LESS THAN	---	---	---	281
344	AND <= DT	AND TIME LESS THAN OR EQUAL	---	---	---	281
344	LD <= DT	LD TIME LESS THAN OR EQUAL	---	---	---	281
344	OR <= DT	OR TIME LESS THAN OR EQUAL	---	---	---	281
345	AND > DT	AND TIME GREATER THAN	---	---	---	281
345	LD > DT	LOAD TIME GREATER THAN	---	---	---	281

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
345	OR > DT	OR TIME GREATER THAN	---	---	---	281
346	AND >= DT	AND TIME GREATER THAN OR EQUAL	---	---	---	281
346	LD >= DT	LOAD TIME GREATER THAN OR EQUAL	---	---	---	281
346	OR >= DT	OR TIME GREATER THAN OR EQUAL	---	---	---	281
350	AND TST	AND BIT TEST	---	---	---	174
350	LD TST	LOAD BIT TEST	---	---	---	174
350	OR TST	OR BIT TEST	---	---	---	174
351	AND TSTN	AND BIT TEST NOT	---	---	---	174
351	LD TSTN	LOAD BIT TEST NOT	---	---	---	174
351	OR TSTN	OR BIT TEST NOT	---	---	---	174
400	+	SIGNED BINARY ADD WITHOUT CARRY	@+	---	---	410
401	+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	@+L	---	---	412
402	+C	SIGNED BINARY ADD WITH CARRY	@+C	---	---	414
403	+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	@+CL	---	---	416
404	+B	BCD ADD WITHOUT CARRY	@+B	---	---	421
405	+BL	DOUBLE BCD ADD WITHOUT CARRY	@+BL	---	---	419
406	+BC	BCD ADD WITH CARRY	@+BC	---	---	421
407	+BCL	DOUBLE BCD ADD WITH CARRY	@+BCL	---	---	423
410	−	SIGNED BINARY SUBTRACT WITHOUT CARRY	@−	---	---	424
411	−L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	@−L	---	---	426
412	−C	SIGNED BINARY SUBTRACT WITH CARRY	@−C	---	---	430
413	−CL	DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	@−CL	---	---	432
414	−B	BCD SUBTRACT WITHOUT CARRY	@−B	---	---	435
415	−BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	@−BL	---	---	436
416	−BC	BCD SUBTRACT WITH CARRY	@−BC	---	---	440
417	−BCL	DOUBLE BCD SUBTRACT WITH CARRY	@−BCL	---	---	441
420	*	SIGNED BINARY MULTIPLY	@*	---	---	443
421	*L	DOUBLE SIGNED BINARY MULTIPLY	@*L	---	---	445
422	*U	UNSIGNED BINARY MULTIPLY	@*U	---	---	447
423	*UL	DOUBLE UNSIGNED BINARY MULTIPLY	@*UL	---	---	449
424	*B	BCD MULTIPLY	@*B	---	---	450

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
425	*BL	DOUBLE BCD MULTIPLY	@*BL	---	---	452
430	/	SIGNED BINARY DIVIDE	@/	---	---	454
431	/L	DOUBLE SIGNED BINARY DIVIDE	@/L	---	---	456
432	/U	UNSIGNED BINARY DIVIDE	@/U	---	---	458
433	/UL	DOUBLE UNSIGNED BINARY DIVIDE	@/UL	---	---	460
434	/B	BCD DIVIDE	@/B	---	---	462
435	/BL	DOUBLE BCD DIVIDE	@/BL	---	---	464
448	FSTR	FLOATING POINT TO ASCII	@FSTR	---	---	604
449	FVAL	ASCII TO FLOATING POINT	@FVAL	---	---	609
450	FIX	FLOATING TO 16-BIT	@FIX	---	---	563
451	FIXL	FLOATING TO 32-BIT	@FIXL	---	---	565
452	FLT	16-BIT TO FLOATING	@FLT	---	---	566
453	FLTL	32-BIT TO FLOATING	@FLTL	---	---	568
454	+F	FLOATING-POINT ADD	@+F	---	---	570
455	−F	FLOATING-POINT SUBTRACT	@−F	---	---	572
456	*F	FLOATING-POINT MULTIPLY	@*F	---	---	574
457	/F	FLOATING-POINT DIVIDE	@/F	---	---	576
458	RAD	DEGREES TO RADIANS	@RAD	---	---	597
459	DEG	RADIANS-TO DEGREES	@DEG	---	---	579
460	SIN	SINE	@SIN	---	---	581
461	COS	COSINE	@COS	---	---	583
462	TAN	TANGENT	@TAN	---	---	585
463	ASIN	ARC SINE	@ASIN	---	---	587
464	ACOS	ARC COSINE	@ACOS	---	---	589
465	ATAN	ARC TANGENT	@ATAN	---	---	591
466	SQRT	SQUARE ROOT	@SQRT	---	---	593
467	EXP	EXPONENT	@EXP	---	---	595
468	LOG	LOGARITHM	@LOG	---	---	597
470	BINS	SIGNED BCD-TO-BINARY	@BINS	---	---	499
471	BCDS	SIGNED BINARY-TO-BCD	@BCDS	---	---	505
472	BISL	DOUBLE SIGNED BCD-TO-BINARY	@BISL	---	---	502
473	BDSL	DOUBLE SIGNED BINARY-TO-BCD	@BDSL	---	---	507
474	GRY	GRAY CODE CONVERSION	@GRY	---	---	511
486	SCL2	SCALING 2	@SCL2	---	---	762
487	SCL3	SCALING 3	@SCL3	---	---	766
490	CMND	DELIVER COMMAND	@CMND	---	---	1003
498	MOVL	DOUBLE MOVE	@MOVL	---	---	318
499	MVNL	DOUBLE MOVE NOT	@MVNL	---	---	320
502	BCMP2	EXPANDED BLOCK COMPARE	@BCMP2	---	---	306

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
510	CJP	CONDITIONAL JUMP	---	---	---	223
511	CJPN	CONDITIONAL JUMP NOT	---	---	---	223
512	FOR	FOR-NEXT LOOPS	---	---	---	229
513	NEXT	FOR-NEXT LOOPS	---	---	---	229
514	BREAK	BREAK LOOP	---	---	---	232
515	JMP0	MULTIPLE JUMP	---	---	---	227
516	JME0	MULTIPLE JUMP END	---	---	---	227
517	MILH	MULTI-INTERLOCK DIFFERENTIATION HOLD	---	---	---	205
518	MILR	MULTI-INTERLOCK DIFFERENTIATION RELEASE	---	---	---	205
519	MILC	MULTI-INTERLOCK CLEAR	---	---	---	205
520	NOT	NOT	---	---	---	172
521	UP	CONDITION ON	---	---	---	173
522	DOWN	CONDITION OFF	---	---	---	173
530	SETA	MULTIPLE BIT SET	@SETA	---	---	189
531	RSTA	MULTIPLE BIT RESET	@RSTA	---	---	189
532	SETB	SINGLE BIT SET	@SETB	---	!SETB	192
533	RSTB	SINGLE BIT RESET	@RSTB	---	!RSTB	192
534	OUTB	SINGLE BIT OUTPUT	@OUTB	---	!OUTB	195
540	TMHH	ONE-MS TIMER	---	---	---	244
542	TIML	LONG TIMER	---	---	---	251
543	MTIM	MULTI-OUTPUT TIMER	---	---	---	254
545	CNR	RESET TIMER/ COUNTER	@CNR	---	---	267
546	CNTX	COUNTER	---	---	---	260
547	CNRX	RESET TIMER/ COUNTER	---	---	---	267
548	CNTRX	REVERSIBLE COUNTER	---	---	---	263
550	TIMX	TIMER	---	---	---	235
551	TIMHX	HIGH-SPEED TIMER	---	---	---	240
552	TMHHX	ONE-MS TIMER	---	---	---	244
553	TIMLX	LONG TIMER	---	---	---	251
554	MTIMX	MULTI-OUTPUT TIMER	---	---	---	254
555	TTIMX	ACCUMULATIVE TIMER	---	---	---	247
560	MOVR	MOVE TO REGISTER	@MOVR	---	---	340
561	MOVRW	MOVE TIMER/COUNTER PV TO REGISTER	@MOVRW	---	---	342
562	XCGL	DOUBLE DATA EXCHANGE	@XCGL	---	---	334
565	XFERC	BLOCK TRANSFER	@XFERC	---	---	1201
566	DISTC	SINGLE WORD DISTRIBUTE	@DISTC	---	---	1203
567	COLLC	DATA COLLECT	@COLLC	---	---	1206
568	MOVBC	MOVE BIT	@MOVBC	---	---	1211
570	ASLL	DOUBLE SHIFT LEFT	@ASLL	---	---	355
571	ASRL	DOUBLE SHIFT RIGHT	@ASRL	---	---	358

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
572	ROLL	DOUBLE ROTATE LEFT	@ROLL	---	---	362
573	RORL	DOUBLE ROTATE RIGHT	@RORL	---	---	365
574	RLNC	ROTATE LEFT WITHOUT CARRY	@RLNC	---	---	367
575	RRNC	ROTATE RIGHT WITHOUT CARRY	@RRNC	---	---	371
576	RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	@RLNL	---	---	369
577	RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	@RRNL	---	---	372
578	NSFL	SHIFT N-BIT DATA LEFT	@NSFL	---	---	377
579	NSFR	SHIFT N-BIT DATA RIGHT	@NSFR	---	---	379
580	NASL	SHIFT N-BITS LEFT	@NASL	---	---	381
581	NASR	SHIFT N-BITS RIGHT	@NASR	---	---	387
582	NSLL	DOUBLE SHIFT N-BITS LEFT	@NSLL	---	---	384
583	NSRL	DOUBLE SHIFT N-BITS RIGHT	@NSRL	---	---	389
590	++	INCREMENT BINARY	@++	---	---	393
591	++L	DOUBLE INCREMENT BINARY	@++L	---	---	395
592	– –	DECREMENT BINARY	@– –	---	---	397
593	– –L	DOUBLE DECREMENT BINARY	@– –L	---	---	399
594	++B	INCREMENT BCD	@++B	---	---	401
595	++BL	DOUBLE INCREMENT BCD	@++BL	---	---	403
596	– –B	DECREMENT BCD	@– –B	---	---	405
597	– –BL	DOUBLE DECREMENT BCD	@– –BL	---	---	407
600	SIGN	16-BIT TO 32-BIT SIGNED BINARY	@SIGN	---	---	476
610	ANDL	DOUBLE LOGICAL AND	@ANDL	---	---	519
611	ORWL	DOUBLE LOGICAL OR	@ORWL	---	---	522
612	XORL	DOUBLE EXCLUSIVE OR	@XORL	---	---	526
613	XNRL	DOUBLE EXCLUSIVE NOR	@XNRL	---	---	529
614	COML	DOUBLE COMPLEMENT	@COML	---	---	533
620	ROTB	BINARY ROOT	@ROTB	---	---	534
621	BCNTC	BIT COUNTER	@BCNTC	---	---	1212
630	SSET	SET STACK	@SSET	---	---	666
631	DIM	DIMENSION RECORD TABLE	@DIM	---	---	678
632	PUSH	PUSH ONTO STACK	@PUSH	---	---	669
633	FIFO	FIRST IN FIRST OUT	@FIFO	---	---	672
634	LIFO	LAST IN FIRST OUT	@LIFO	---	---	675
635	SETR	SET RECORD LOCATION	@SETR	---	---	681
636	GETR	GET RECORD NUMBER	@GETR	---	---	683
637	SWAP	SWAP BYTES	@SWAP	---	---	687

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
638	SNUM	STACK SIZE READ	@SNUM	---	---	704
639	SREAD	STACK DATA READ	@SREAD	---	---	707
640	SWRIT	STACK DATA WRITE	@SWRIT	---	---	710
641	SINS	STACK DATA INSERT	@SINS	---	---	713
642	SDEL	STACK DATA DELETE	@SDEL	---	---	716
650	LEN$	STRING LENGTH	@LEN$	---	---	1173
652	LEFT$	GET STRING LEFT	@LEFT$	---	---	1164
653	RGHT$	GET STRING RIGHT	@RGHT$	---	---	1166
654	MID$	GET STRING MIDDLE	@MID$	---	---	1168
656	+$	CONCATENATE STRING	@+$	---	---	1161
657	INS$	INS$	@INS$	---	---	1184
658	DEL$	DELETE STRING	@DEL$	---	---	1178
660	FIND$	FIND IN STRING	@FIND$	---	---	1171
661	RPLC$	REPLACE IN STRING	@RPLC$	---	---	1175
664	MOV$	MOV STRING	@MOV$	---	---	1159
665	XCHG$	EXCHANGE STRING	@XCHG$	---	---	1180
666	CLR$	CLEAR STRING	@CLR$	---	---	1182
670	AND =$	AND STRING EQUALS	---	---	---	1187
670	LD =$	LOAD STRING EQUALS	---	---	---	1187
670	OR =$	OR STRING EQUALS	---	---	---	1187
671	AND <>$	AND STRING NOT EQUAL	---	---	---	1187
671	LD <>$	LOAD STRING NOT EQUAL	---	---	---	1187
671	OR <>$	OR STRING NOT EQUAL	---	---	---	1187
672	AND <$	AND STRING LESS THAN	---	---	---	1187
672	LD <$	LOAD STRING LESS THAN	---	---	---	1187
672	OR <$	OR STRING LESS THAN	---	---	---	1187
673	AND <=$	AND STRING LESS THAN OR EQUALS	---	---	---	1187
673	LD <=$	LOAD STRING LESS THAN OR EQUAL	---	---	---	1187
673	OR <=$	OR STRING LESS THAN OR EQUALS	---	---	---	1187
674	AND >$	AND STRING GREATER THAN	---	---	---	1187
674	LD >$	LOAD STRING GREATER THAN	---	---	---	1187
674	OR >$	OR STRING GREATER THAN	---	---	---	1187
675	AND >=$	AND STRING GREATER THAN OR EQUALS	---	---	---	1187
675	LD >=$	LOAD STRING GREATER THAN OR EQUALS	---	---	---	1187
675	OR >=$	OR STRING GREATER THAN OR EQUALS	---	---	---	1187
680	LMT	LIMIT CONTROL	@LMT	---	---	741
681	BAND	DEAD BAND CONTROL	@BAND	---	---	743
682	ZONE	DEAD ZONE CONTROL	@ZONE	---	---	746

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
685	TPO	TIME-PROPORTIONAL OUTPUT	---	---	---	749
690	MSKS	SET INTERRUPT MASK	@MSKS	---	---	798
691	CLI	CLEAR INTERRUPT	@CLI	---	---	809
692	MSKR	READ INTERRUPT MASK	@MSKR	---	---	804
693	DI	DISABLE INTERRUPTS	@DI	---	---	814
694	EI	ENABLE INTERRUPTS	---	---	---	816
700	FREAD	READ DATA FILE	@FREAD	---	---	1045
701	FWRIT	WRITE DATA FILE	@FWRIT	---	---	1052
720	EXPLT	EXPLICIT MESSAGE SEND	@EXPLT	---	---	1013
721	EGATR	EXPLICIT GET ATTRIBUTE	@EGATR	---	---	1021
722	ESATR	EXPLICIT SET ATTRIBUTE	@ESATR	---	---	1028
723	ECHRD	EXPLICIT WORD READ	@ECHRD	---	---	1034
724	ECHWR	EXPLICIT WORD CLEAR	@ECHWR	---	---	1038
730	CADD	CALENDAR ADD	@CADD	---	---	1061
731	CSUB	CALENDAR SUBTRACT	@CSUB	---	---	1065
735	DATE	CLOCK ADJUSTMENT	@DATE	---	---	1073
750	GSBS	GLOBAL SUBROUTINE CALL	@GSBS	---	---	786
751	GSBN	GLOBAL SUBROUTINE ENTRY	---	---	---	794
752	GRET	GLOBAL SUBROUTINE RETURN	---	---	---	797
801	BEND	BLOCK PROGRAM END	---	---	---	1128
802	IF	CONDITIONAL BRANCHING BLOCK	---	---	---	1133
802	IF	CONDITIONAL BRANCHING BLOCK	---	---	---	1133
802	IF NOT	CONDITIONAL BRANCHING BLOCK NOT	---	---	---	1133
803	ELSE	ELSE	---	---	---	1133
804	IEND	IF END	---	---	---	1133
805	WAIT	ONE CYCLE AND WAIT	---	---	---	1140
805	WAIT	ONE CYCLE AND WAIT	---	---	---	1140
805	WAIT NOT	ONE CYCLE AND WAIT NOT	---	---	---	1140
806	EXIT	CONDITIONAL BLOCK EXIT	---	---	---	1137
806	EXIT	CONDITIONAL BLOCK EXIT	---	---	---	1137
806	EXIT NOT	CONDITIONAL BLOCK EXIT NOT	---	---	---	1137
809	LOOP	LOOP	---	---	---	1153
810	LEND	LOOP END	---	---	---	1153
810	LEND	LOOP END	---	---	---	1153
810	LEND NOT	LOOP END NOT	---	---	---	1153

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
811	BPPS	BLOCK PROGRAM PAUSE	---	---	---	1131
812	BPRS	BLOCK PROGRAM RESTART	---	---	---	1131
813	TIMW	TIMER WAIT	---	---	---	1144
814	CNTW	COUNTER WAIT	---	---	---	1147
815	TMHW	HIGH-SPEED TIMER WAIT	---	---	---	1150
816	TIMWX	TIMER WAIT	---	---	---	1144
817	TMHWX	HIGH-SPEED TIMER WAIT	---	---	---	1150
818	CNTWX	COUNTER WAIT	---	---	---	1147
820	TKON	TASK ON	@TKON	---	---	1192
821	TKOF	TASK OFF	@TKOF	---	---	1196
840	PWR	EXPONENTIAL POWER	@PWR	---	---	599
841	FIXD	DOUBLE FLOATING TO 16-BIT BINARY	@FIXD	---	---	620
842	FIXLD	DOUBLE FLOATING TO 32-BIT BINARY	@FIXLD	---	---	621
843	DBL	16-BIT BINARY TO DOUBLE FLOATING	@DBL	---	---	623
844	DBLL	32-BIT BINARY TO DOUBLE FLOATING	@DBLL	---	---	624
845	+D	DOUBLE FLOATING-POINT ADD	@+D	---	---	626
846	−D	DOUBLE FLOATING-POINT SUBTRACT	@−D	---	---	628
847	✻D	DOUBLE FLOATING-POINT MULTIPLY	@✻D	---	---	630
848	/D	DOUBLE FLOATING-POINT DIVIDE	@/D	---	---	632
849	RADD	DOUBLE DEGREES TO RADIANS	@RADD	---	---	634
850	DEGD	DOUBLE RADIANS TO DEGREES	@DEGD	---	---	636
851	SIND	DOUBLE SINE	@SIND	---	---	637
852	COSD	DOUBLE COSINE	@COSD	---	---	639
853	TAND	DOUBLE TANGENT	@TAND	---	---	641
854	ASIND	DOUBLE ARC SINE	@ASIND	---	---	643
855	ACOSD	DOUBLE ARC COSINE	@ACOSD	---	---	645
856	ATAND	DOUBLE ARC TAN-GENT	@ATAND	---	---	647
857	SQRTD	DOUBLE SQUARE ROOT	@SQRTD	---	---	649
858	EXPD	DOUBLE EXPONENT	@EXPD	---	---	651
859	LOGD	DOUBLE LOGARITHM	@LOGD	---	---	653
860	PWRD	DOUBLE EXPONEN-TIAL POWER	@PWRD	---	---	655
880	INI	MODE CONTROL	@INI	---	---	823
881	PRV	HIGH-SPEED COUN-TER PV READ	@PRV	---	---	827
882	CTBL	COMPARISON TABLE LOAD	@CTBL	---	---	837
883	PRV2	COUNTER FRE-QUENCY CONVERT	@PRV2	---	---	833
885	SPED	SPEED OUTPUT	@SPED	---	---	841
886	PULS	SET PULSES	@PULS	---	---	846

Funktionscode	AWL	Befehl	Ausführung bei steigender Flanke	Ausführung bei fallender Flanke	Spezifikation der direkten Aktualisierung	Seite
887	PLS2	PULSE OUTPUT	@PLS2	---	---	849
888	ACC	ACCELERATION CONTROL	@ACC	---	---	855
889	ORG	ORIGIN SEARCH	@ORG	---	---	862
891	PWN	PULSE WITH VARIABLE DUTY FACTOR	@PWN	---	---	865

KAPITEL 3
Befehle

In diesem Kapitel werden die einzelnen Befehle beschrieben, die bei der Programmierung der SPS-Systeme der CS/CJ-Serie verwendet werden können. Die Befehle werden nach Funktionen geordnet beschrieben. Die Klassifizierung der Befehle finden Sie in *Kapitel 2, Zusammenfassung der Befehle*.

3-1 Notation und Layout von Befehlsbeschreibungen 148
3-2 Befehlserweiterungen und neue Befehle. 151
3-2-1 Erweiterungen für CS-Serie CPU-Baugruppen der Version 1 151
3-2-2 Funktionserweiterungen für CS1-H/CJ1-H CPU-Baugruppen 151
3-3 Sequenzeingabebefehle. 153
3-3-1 LOAD: LD 153
3-3-2 LOAD NOT: LD NOT 155
3-3-3 AND: AND. 157
3-3-4 AND NOT: AND NOT. 159
3-3-5 OR: OR. 161
3-3-6 OR NOT: OR NOT. 163
3-3-7 AND LOAD: AND LD. 164
3-3-8 OR LOAD: OR LD. 166
3-3-9 Befehle mit Flankenausführung und Befehle für direkte Aktualisierung. 169
3-3-10 Zeitliches Verhalten von E/A-Befehlen 170
3-3-11 TR-Bits 170
3-3-12 NOT: NOT(520) 172
3-3-13 CONDITION ON/OFF: UP(521) und DOWN(522) 173
3-3-14 BIT TEST: TST(350) und TSTN(351). 174
3-4 Sequenzausgabebefehle. 177
3-4-1 OUTPUT: OUT 177
3-4-2 OUTPUT NOT: OUT NOT 178
3-4-3 KEEP: KEEP(011) 180
3-4-4 DIFFERENTIATE UP/DOWN: DIFU(013) und DIFD(014) 184
3-4-5 SET and RESET: SET und RSET 187
3-4-6 MULTIPLE BIT SET/RESET: SETA(530)/RSTA(531) 189
3-4-7 SINGLE BIT SET/RESET: SETB(532)/RSTB(533). 192
3-4-8 SINGLE BIT OUTPUT: OUTB(534) 195
3-5 Sequenzsteuerbefehle 197
3-5-1 END: END(001). 197
3-5-2 NO OPERATION: NOP(000). 198
3-5-3 Übersicht über Verriegelungsbefehle 199
3-5-4 INTERLOCK und INTERLOCK CLEAR: IL(002) und ILC(003). 201
3-5-5 MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE und MULTI-INTERLOCK CLEAR: MILH(517), MILR(518) und MILC(519) 205
3-5-6 JUMP und JUMP END: JMP(004) und JME(005) 219
3-5-7 CONDITIONAL JUMP: CJP(510)/CJPN(511) 223
3-5-8 MULTIPLE JUMP und JUMP END: JMP0(515) und JME0(516) 227
3-5-9 FOR-NEXT-Programmschleife: FOR(512)/NEXT(513). 229
3-5-10 BREAK LOOP: BREAK(514). 232
3-6 Zeitgeber- und Zählerbefehle 233
3-6-1 TIMER: TIM/TIMX(550) 235
3-6-2 HIGH-SPEED TIMER: TIMH(015)/TIMHX(551) 240
3-6-3 ONE-MS TIMER: TMHH(540)/TMHHX(552). 244
3-6-4 ACCUMULATIVE TIMER: TTIM(087)/TTIMX(555) 247
3-6-5 LONG TIMER: TIML(542)/TIMLX(553). 251

3-6-6 MULTI-OUTPUT TIMER: MTIM(543)/MTIMX(554) 254
3-6-7 COUNTER: CNT/CNTX(546) 260
3-6-8 REVERSIBLE COUNTER: CNTR(012)/CNTRX(548) 263
3-6-9 RESET TIMER/COUNTER: CNR(545)/CNRX(547) 267
3-6-10 Beispiele für Zeitgeber- und Zähleranwendungen 269
3-6-11 Indirekte Adressierung von Zeitgeber-/Zählernummern 272
3-7 Vergleichsbefehle 275
3-7-1 Symbolvergleichsbefehle (300 bis 328) 275
3-7-2 Zeitvergleichsbefehle (341 bis 346) 281
3-7-3 COMPARE: CMP(020) 287
3-7-4 DOUBLE COMPARE: CMPL(060) 290
3-7-5 SIGNED BINARY COMPARE: CPS(114) 293
3-7-6 DOUBLE SIGNED BINARY COMPARE: CPSL(115) 296
3-7-7 MULTIPLE COMPARE: MCMP(019) 299
3-7-8 TABLE COMPARE: TCMP(085) 301
3-7-9 BLOCK COMPARE: BCMP(068) 304
3-7-10 EXPANDED BLOCK COMPARE: BCMP2(502) 306
3-7-11 AREA RANGE COMPARE: ZCP(088) 310
3-7-12 DOUBLE AREA RANGE COMPARE: ZCPL(116) 313
3-8 Datenübertragungsbefehle 315
3-8-1 MOVE: MOV(021) 315
3-8-2 MOVE NOT: MVN(022) 317
3-8-3 DOUBLE MOVE: MOVL(498) 318
3-8-4 DOUBLE MOVE NOT: MVNL(499) 320
3-8-5 MOVE BIT: MOVB(082) 321
3-8-6 MOVE DIGIT: MOVD(083) 323
3-8-7 MULTIPLE BIT TRANSFER: XFRB(062) 326
3-8-8 BLOCK TRANSFER: XFER(070) 328
3-8-9 BLOCK SET: BSET(071) 331
3-8-10 DATA EXCHANGE: XCHG(073) 333
3-8-11 DOUBLE DATA EXCHANGE: XCGL(562) 334
3-8-12 SINGLE WORD DISTRIBUTE: DIST(080) 336
3-8-13 DATA COLLECT: COLL(081) 338
3-8-14 MOVE TO REGISTER: MOVR(560) 340
3-8-15 MOVE TIMER/COUNTER PV TO REGISTER: MOVRW(561) 342
3-9 Datenverschiebungsbefehle 344
3-9-1 SHIFT REGISTER: SFT(010) 345
3-9-2 REVERSIBLE SHIFT REGISTER: SFTR(084) 346
3-9-3 ASYNCHRONOUS SHIFT REGISTER: ASFT(017) 349
3-9-4 WORD SHIFT: WSFT(016) 352
3-9-5 ARITHMETIC SHIFT LEFT: ASL(025) 354
3-9-6 DOUBLE SHIFT LEFT: ASLL(570) 355
3-9-7 ARITHMETIC SHIFT RIGHT: ASR(026) 357
3-9-8 DOUBLE SHIFT RIGHT: ASRL(571) 358
3-9-9 ROTATE LEFT: ROL(027) 360
3-9-10 DOUBLE ROTATE LEFT: ROLL(572) 362
3-9-11 ROTATE RIGHT: ROR(028) 364
3-9-12 DOUBLE ROTATE RIGHT: RORL(573) 365
3-9-13 ROTATE LEFT WITHOUT CARRY: RLNC(574) 367
3-9-14 DOUBLE ROTATE LEFT WITHOUT CARRY: RLNL(576) 369
3-9-15 ROTATE RIGHT WITHOUT CARRY: RRNC(575) 371
3-9-16 DOUBLE ROTATE RIGHT WITHOUT CARRY: RRNL(577) 372
3-9-17 ONE DIGIT SHIFT LEFT: SLD(074) 374
3-9-18 ONE DIGIT SHIFT RIGHT: SRD(075) 376
3-9-19 SHIFT N-BIT DATA LEFT: NSFL(578) 377
3-9-20 SHIFT N-BIT DATA RIGHT: NSFR(579) 379
3-9-21 SHIFT N-BITS LEFT: NASL(580) 381

3-9-22 DOUBLE SHIFT N-BITS LEFT: NSLL(582) 384
3-9-23 SHIFT N-BITS RIGHT: NASR(581) 387
3-9-24 DOUBLE SHIFT N-BITS RIGHT: NSRL(583) 389
3-10 Inkrementier-/Dekrementierbefehle 393
3-10-1 INCREMENT BINARY: ++(590) 393
3-10-2 DOUBLE INCREMENT BINARY: ++L(591) 395
3-10-3 DECREMENT BINARY: – –(592) 397
3-10-4 DOUBLE DECREMENT BINARY: – –L(593) 399
3-10-5 INCREMENT BCD: ++B(594) 401
3-10-6 DOUBLE INCREMENT BCD: ++BL(595) 403
3-10-7 DECREMENT BCD: – –B(596) 405
3-10-8 DOUBLE DECREMENT BCD: – –BL(597) 407
3-11 Symbolarithmetische Befehle 409
3-11-1 SIGNED BINARY ADD WITHOUT CARRY: +(400) 410
3-11-2 DOUBLE SIGNED BINARY ADD WITHOUT CARRY: +L(401) 412
3-11-3 SIGNED BINARY ADD WITH CARRY: +C(402) 414
3-11-4 DOUBLE SIGNED BINARY ADD WITH CARRY: +CL(403) 416
3-11-5 BCD ADD WITHOUT CARRY: +B(404) 418
3-11-6 DOUBLE BCD ADD WITHOUT CARRY: +BL(405) 419
3-11-7 BCD ADD WITH CARRY: +BC(406) 421
3-11-8 DOUBLE BCD ADD WITH CARRY: +BCL(407) 423
3-11-9 SIGNED BINARY SUBTRACT WITHOUT CARRY: –(410) 424
3-11-10 DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: –L(411) 426
3-11-11 SIGNED BINARY SUBTRACT WITH CARRY: –C(412) 430
3-11-12 DOUBLE SIGNED BINARY SUBTRACT WITH CARRY: –CL(413) 432
3-11-13 BCD SUBTRACT WITHOUT CARRY: –B(414) 435
3-11-14 DOUBLE BCD SUBTRACT WITHOUT CARRY: –BL(415) 436
3-11-15 BCD SUBTRACT WITH CARRY: –BC(416) 440
3-11-16 DOUBLE BCD SUBTRACT WITH CARRY: –BCL(417) 441
3-11-17 SIGNED BINARY MULTIPLY: *(420) 443
3-11-18 DOUBLE SIGNED BINARY MULTIPLY: *L(421) 445
3-11-19 UNSIGNED BINARY MULTIPLY: *U(422) 447
3-11-20 DOUBLE UNSIGNED BINARY MULTIPLY: *UL(423) 449
3-11-21 BCD MULTIPLY: *B(424) 450
3-11-22 DOUBLE BCD MULTIPLY: *BL(425) 452
3-11-23 SIGNED BINARY DIVIDE: /(430) 454
3-11-24 DOUBLE SIGNED BINARY DIVIDE: /L(431) 456
3-11-25 UNSIGNED BINARY DIVIDE: /U(432) 458
3-11-26 DOUBLE UNSIGNED BINARY DIVIDE: /UL(433) 460
3-11-27 BCD DIVIDE: /B(434) 462
3-11-28 DOUBLE BCD DIVIDE: /BL(435) 464
3-12 Konvertierungsbefehle 465
3-12-1 BCD-TO-BINARY: BIN(023) 466
3-12-2 DOUBLE BCD-TO-DOUBLE BINARY: BINL(058) 467
3-12-3 BINARY-TO-BCD: BCD(024) 469
3-12-4 DOUBLE BINARY-TO-DOUBLE BCD: BCDL(059) 470
3-12-5 2’S COMPLEMENT: NEG(160) 472
3-12-6 DOUBLE 2’S COMPLEMENT: NEGL(161) 474
3-12-7 16-BIT TO 32-BIT SIGNED BINARY: SIGN(600) 476
3-12-8 DATA DECODER: MLPX(076) 477
3-12-9 DATA ENCODER: DMPX(077) 482
3-12-10 ASCII CONVERT: ASC(086) 486
3-12-11 ASCII TO HEX: HEX(162) 490
3-12-12 COLUMN TO LINE: LINE(063) 494
3-12-13 LINE TO COLUMN: COLM(064) 496
3-12-14 SIGNED BCD-TO-BINARY: BINS(470) 499
3-12-15 DOUBLE SIGNED BCD-TO-BINARY: BISL(472) 502

3-12-16 SIGNED BINARY-TO-BCD: BCDS(471) 505
3-12-17 DOUBLE SIGNED BINARY-TO-BCD: BDSL(473) 507
3-12-18 GRAY CODE CONVERT: GRY(474) 511
3-13 Logikbefehle 517
3-13-1 LOGICAL AND: ANDW(034) 517
3-13-2 DOUBLE LOGICAL AND: ANDL(610) 519
3-13-3 LOGICAL OR: ORW(035) 520
3-13-4 DOUBLE LOGICAL OR: ORWL(611) 522
3-13-5 EXCLUSIVE OR: XORW(036) 524
3-13-6 DOUBLE EXCLUSIVE OR: XORL(612) 526
3-13-7 EXCLUSIVE NOR: XNRW(037) 528
3-13-8 DOUBLE EXCLUSIVE NOR: XNRL(613) 529
3-13-9 COMPLEMENT: COM(029) 531
3-13-10 DOUBLE COMPLEMENT: COML(614) 533
3-14 Spezielle arithmetische Befehle 534
3-14-1 BINARY ROOT: ROTB(620) 534
3-14-2 BCD SQUARE ROOT: ROOT(072) 536
3-14-3 ARITHMETIC PROCESS: APR(069) 540
3-14-4 FLOATING POINT DIVIDE: FDIV(079) 552
3-14-5 BIT COUNTER: BCNT(067) 556
3-15 Gleitkomma-Arithmetikbefehle 558
3-15-1 FLOATING TO 16-BIT: FIX(450) 563
3-15-2 FLOATING TO 32-BIT: FIXL(451) 565
3-15-3 16-BIT TO FLOATING: FLT(452) 566
3-15-4 32-BIT TO FLOATING: FLTL(453) 568
3-15-5 FLOATING-POINT ADD: +F(454) 570
3-15-6 FLOATING-POINT SUBTRACT: –F(455) 572
3-15-7 FLOATING-POINT MULTIPLY: *F(456) 574
3-15-8 FLOATING-POINT DIVIDE: /F(457) 576
3-15-9 DEGREES TO RADIANS: RAD(458) 578
3-15-10 RADIANS TO DEGREES: DEG(459) 579
3-15-11 SINE: SIN(460) 581
3-15-12 COSINE: COS(461) 583
3-15-13 TANGENT: TAN(462) 585
3-15-14 ARC SINE: ASIN(463) 587
3-15-15 ARC COSINE: ACOS(464) 589
3-15-16 ARC TANGENT: ATAN(465) 591
3-15-17 SQUARE ROOT: SQRT(466) 593
3-15-18 EXPONENT: EXP(467) 595
3-15-19 LOGARITHM: LOG(468) 597
3-15-20 EXPONENTIAL POWER: PWR(840) 599
3-15-21 Vergleichsbefehle für Gleitkommadaten einfacher Genauigkeit 600
3-15-22 FLOATING-POINT TO ASCII: FSTR(448) 604
3-15-23 ASCII TO FLOATING-POINT: FVAL(449) 609
3-16 Befehle für Gleitkommadaten doppelter Genauigkeit (nur CS1-H, CJ1-H, CJ1M und CS1D) . . 613
3-16-1 DOUBLE FLOATING TO 16-BIT: FIXD(841) 620
3-16-2 DOUBLE FLOATING TO 32-BIT: FIXLD(842) 621
3-16-3 16-BIT TO DOUBLE FLOATING: DBL(843) 623
3-16-4 32-BIT TO DOUBLE FLOATING: DBLL(844) 624
3-16-5 DOUBLE FLOATING-POINT ADD: +D(845) 626
3-16-6 DOUBLE FLOATING-POINT SUBTRACT: –D(846) 628
3-16-7 DOUBLE FLOATING-POINT MULTIPLY: *D(847) 630
3-16-8 DOUBLE FLOATING-POINT DIVIDE: /D(848) 632
3-16-9 DOUBLE DEGREES TO RADIANS: RADD(849) 634
3-16-10 DOUBLE RADIANS TO DEGREES: DEGD(850) 636
3-16-11 DOUBLE SINE: SIND(851) 637
3-16-12 DOUBLE COSINE: COSD(852) 639

3-16-13 DOUBLE TANGENT: TAND(853) . . . 641
3-16-14 DOUBLE ARC SINE: ASIND(854) . . . 643
3-16-15 DOUBLE ARC COSINE: ACOSD(855) . . . 645
3-16-16 DOUBLE ARC TANGENT: ATAND(856) . . . 647
3-16-17 DOUBLE SQUARE ROOT: SQRTD(857) . . . 649
3-16-18 DOUBLE EXPONENT: EXPD(858) . . . 651
3-16-19 DOUBLE LOGARITHM: LOGD(859) . . . 653
3-16-20 DOUBLE EXPONENTIAL POWER: PWRD(860) . . . 655
3-16-21 Symbolvergleichsbefehle für Gleitkommadaten doppelter Genauigkeit . . . 657
3-17 Tabellendatenverarbeitungsbefehle . . . 660
3-17-1 SET STACK: SSET(630) . . . 666
3-17-2 PUSH ONTO STACK: PUSH(632) . . . 669
3-17-3 FIRST IN FIRST OUT: FIFO(633) . . . 672
3-17-4 LAST IN FIRST OUT: LIFO(634) . . . 675
3-17-5 DIMENSION RECORD TABLE: DIM(631) . . . 678
3-17-6 SET RECORD LOCATION: SETR(635) . . . 681
3-17-7 GET RECORD NUMBER: GETR(636) . . . 683
3-17-8 DATA SEARCH: SRCH(181) . . . 685
3-17-9 SWAP BYTES: SWAP(637) . . . 687
3-17-10 FIND MAXIMUM: MAX(182) . . . 689
3-17-11 FIND MINIMUM: MIN(183) . . . 693
3-17-12 SUM: SUM(184) . . . 697
3-17-13 FRAME CHECKSUM: FCS(180) . . . 700
3-17-14 STACK SIZE READ: SNUM(638) . . . 704
3-17-15 STACK DATA READ: SREAD(639) . . . 707
3-17-16 STACK DATA OVERWRITE: SWRIT(640) . . . 710
3-17-17 STACK DATA INSERT: SINS(641) . . . 713
3-17-18 STACK DATA DELETE: SDEL(642) . . . 716
3-18 Datensteuerungsbefehle . . . 720
3-18-1 PID CONTROL: PID(190) . . . 720
3-18-2 PID CONTROL WITH AUTOTUNING: PIDAT(191) . . . 731
3-18-3 LIMIT CONTROL: LMT(680) . . . 741
3-18-4 DEAD BAND CONTROL: BAND(681) . . . 743
3-18-5 DEAD ZONE CONTROL: ZONE(682) . . . 746
3-18-6 TIME-PROPORTIONAL OUTPUT: TPO(685) . . . 749
3-18-7 SCALING: SCL(194) . . . 757
3-18-8 SCALING 2: SCL2(486) . . . 762
3-18-9 SCALING 3: SCL3(487) . . . 766
3-18-10 AVERAGE: AVG(195) . . . 769
3-19 Unterprogramme . . . 773
3-19-1 SUBROUTINE CALL: SBS(091) . . . 773
3-19-2 MACRO: MCRO(099) . . . 779
3-19-3 SUBROUTINE ENTRY: SBN(092) . . . 783
3-19-4 SUBROUTINE RETURN: RET(093) . . . 786
3-19-5 GLOBAL SUBROUTINE CALL: GSBS(750) . . . 786
3-19-6 GLOBAL SUBROUTINE ENTRY: GSBN(751) . . . 794
3-19-7 GLOBAL SUBROUTINE RETURN: GRET(752) . . . 797
3-20 Interrupt-Steuerungsbefehle . . . 798
3-20-1 SET INTERRUPT MASK: MSKS(690) . . . 798
3-20-2 READ INTERRUPT MASK: MSKR(692) . . . 804
3-20-3 CLEAR INTERRUPT: CLI(691) . . . 809
3-20-4 DISABLE INTERRUPTS: DI(693) . . . 814
3-20-5 ENABLE INTERRUPTS: EI(694) . . . 816
3-20-6 Übersicht über die Interrupt-Steuerung . . . 818
3-21 Befehle für schnelle Zähler und Impulsausgaben . . . 823
3-21-1 MODE CONTROL: INI(880) (nur CJ1M-CPU21/22/23) . . . 823
3-21-2 HIGH-SPEED COUNTER PV READ: PRV(881) (nur CJ1M-CPU21/22/23) . . . 827

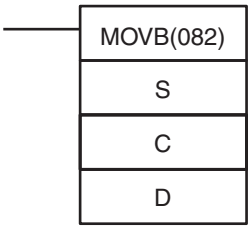
3-21-3 COUNTER FREQUENCY CONVERT: PRV2(883) . . . 833
3-21-4 REGISTER COMPARISON TABLE: CTBL(882) (nur CJ1M-CPU21/22/23) . . . 837
3-21-5 SPEED OUTPUT: SPED(885) (nur CJ1M-CPU21/22/23) . . . 841
3-21-6 SET PULSES: PULS(886) (nur CJ1M-CPU21/22/23) . . . 846
3-21-7 PULSE OUTPUT: PLS2(887) (nur CJ1M-CPU21/22/23) . . . 849
3-21-8 ACCELERATION CONTROL: ACC(888) (nur CJ1M-CPU21/22/23) . . . 855
3-21-9 ORIGIN SEARCH: ORG(889) (nur CJ1M-CPU21/22/23) . . . 862
3-21-10 PULSE WITH VARIABLE DUTY FACTOR: PWM(891) (nur CJ1M-CPU21/22/23) 865
3-22 Schritt-Befehle . . . 867
3-22-1 STEP DEFINE und STEP START: STEP(008)/SNXT(009) . . . 868
3-23 E/A-Baugruppenbefehle . . . 885
3-23-1 I/O REFRESH: IORF(097) . . . 885
3-23-2 7-SEGMENT DECODER: SDEC(078) . . . 888
3-23-3 DIGITAL SWITCH INPUT: DSW(210) . . . 890
3-23-4 TEN KEY INPUT: TKY(211) . . . 896
3-23-5 HEXADECIMAL KEY INPUT: HKY(212) . . . 899
3-23-6 MATRIX INPUT: MTR(213) . . . 904
3-23-7 7-SEGMENT DISPLAY OUTPUT: 7SEG(214) . . . 908
3-23-8 INTELLIGENT I/O READ: IORD(222) . . . 913
3-23-9 INTELLIGENT I/O WRITE: IOWR(223) . . . 917
3-23-10 CPU BUS UNIT I/O REFRESH: DLNK(226) . . . 921
3-24 Befehle für serielle Kommunikation . . . 926
3-24-1 Serielle Kommunikation . . . 926
3-24-2 PROTOCOL MACRO: PMCR(260) . . . 928
3-24-3 TRANSMIT: TXD(236) . . . 937
3-24-4 RECEIVE: RXD(235) . . . 944
3-24-5 TRANSMIT VIA SERIAL COMMUNICATIONS UNIT: TXDU(256) . . . 952
3-24-6 RECEIVE VIA SERIAL COMMUNICATIONS UNIT: RXDU(255) . . . 960
3-24-7 CHANGE SERIAL PORT SETUP: STUP(237) . . . 968
3-25 Netzwerkbefehle . . . 973
3-25-1 Informationen zur Funktionsweise von SYSMAC NET Link/SYSMAC LINK . . . 973
3-25-2 Informationen zu Explizite-Meldung-Befehlen . . . 986
3-25-3 NETWORK SEND: SEND(090) . . . 991
3-25-4 NETWORK RECEIVE: RECV(098) . . . 997
3-25-5 DELIVER COMMAND: CMND(490) . . . 1003
3-25-6 EXPLICIT MESSAGE SEND: EXPLT(720) . . . 1013
3-25-7 EXPLICIT GET ATTRIBUTE: EGATR(721) . . . 1021
3-25-8 EXPLICIT SET ATTRIBUTE: ESATR(722) . . . 1028
3-25-9 EXPLICIT WORD READ: ECHRD(723) . . . 1034
3-25-10 EXPLICIT WORD WRITE: ECHWR(724) . . . 1038
3-26 Dateispeicherbefehle . . . 1042
3-26-1 Vorsichtsmaßnahmen bei der Verwendung von Speicherkarten . . . 1042
3-26-2 READ DATA FILE: FREAD(700) . . . 1045
3-26-3 WRITE DATA FILE: FWRIT(701) . . . 1052
3-27 Anzeigebefehle: DISPLAY MESSAGE: MSG(046) . . . 1058
3-28 Datum- und Uhrzeitbefehle . . . 1061
3-28-1 CALENDAR ADD: CADD(730) . . . 1061
3-28-2 CALENDAR SUBTRACT: CSUB(731) . . . 1065
3-28-3 HOURS TO SECONDS: SEC(065) . . . 1068
3-28-4 SECONDS TO HOURS: HMS(066) . . . 1070
3-28-5 CLOCK ADJUSTMENT: DATE(735) . . . 1073
3-29 Fehlersuchebefehle . . . 1075
3-29-1 TRACE MEMORY SAMPLING: TRSM(045) . . . 1075
3-30 Fehlerdiagnosebefehle . . . 1079
3-30-1 FAILURE ALARM: FAL(006) . . . 1079
3-30-2 SEVERE FAILURE ALARM: FALS(007) . . . 1087
3-30-3 FAILURE POINT DETECTION: FPD(269) . . . 1095

3-31 Sonstige Befehle 1104
3-31-1 SET CARRY: STC(040) 1104
3-31-2 CLEAR CARRY: CLC(041) 1105
3-31-3 SELECT EM BANK: EMBC(281) 1106
3-31-4 EXTEND MAXIMUM CYCLE TIME: WDT(094) 1108
3-31-5 SAVE CONDITION FLAGS: CCS(282) 1110
3-31-6 LOAD CONDITION FLAGS: CCL(283) 1112
3-31-7 CONVERT ADDRESS FROM CV: FRMCV(284) 1113
3-31-8 CONVERT ADDRESS TO CV: TOCV(285) 1117
3-31-9 DISABLE PERIPHERAL SERVICING: IOSP(287) (nur CS1-H/CJ1-H/CJ1M) 1121
3-31-10 ENABLE PERIPHERAL SERVICING: IOSP(288) (nur CS1-H/CJ1-H/CJ1M) 1123
3-32 Blockprogrammierbefehle 1124
3-32-1 Einführung 1124
3-32-2 BLOCK PROGRAM BEGIN/END: BPRG(096)/BEND(801) 1128
3-32-3 BLOCK PROGRAM PAUSE/RESTART: BPPS(811)/BPRS(812) 1131
3-32-4 Verzweigung: IF(802), ELSE(803) und IEND(804) 1133
3-32-5 CONDITIONAL BLOCK EXIT (NOT): EXIT (NOT)(806) 1137
3-32-6 ONE CYCLE AND WAIT (NOT): WAIT(805)/WAIT(805) NOT 1140
3-32-7 TIMER WAIT: TIMW(813) und TIMWX(816) 1144
3-32-8 COUNTER WAIT: CNTW(814) und CNTWX(818) 1147
3-32-9 HIGH-SPEED TIMER WAIT: TMHW(815) and TMHWX(817) 1150
3-32-10 Schleifensteuerung: LOOP(809)/LEND(810)/LEND(810) NOT 1153
3-33 Textverarbeitungsbefehle 1158
3-33-1 Übersicht über Textzeichenkettenverarbeitung 1158
3-33-2 MOV STRING: MOV$(664) 1159
3-33-3 CONCATENATE STRING: +$(656) 1161
3-33-4 GET STRING LEFT: LEFT$(652) 1164
3-33-5 GET STRING RIGHT: RGHT$(653) 1166
3-33-6 GET STRING MIDDLE: MID$(654) 1168
3-33-7 FIND IN STRING: FIND$(660) 1171
3-33-8 STRING LENGTH: LEN$(650) 1173
3-33-9 REPLACE IN STRING: RPLC$(661) 1175
3-33-10 DELETE STRING: DEL$(658) 1178
3-33-11 EXCHANGE STRING: XCHG$(665) 1180
3-33-12 CLEAR STRING: CLR$(666) 1182
3-33-13 INSERT INTO STRING: INS$(657) 1184
3-33-14 Zeichenketten-Vergleichsbefehle (670 bis 675) 1187
3-34 Task-Steuerbefehle 1192
3-34-1 TASK ON: TKON(820) 1192
3-34-2 TASK OFF: TKOF(821) 1196
3-35 Befehle zur Modellkonvertierung (nur CPU-Baugruppen ab Ver. 3.0) 1199
3-35-1 BLOCK TRANSFER: XFERC(565) 1201
3-35-2 SINGLE WORD DISTRIBUTE: DISTC(566) 1203
3-35-3 DATA COLLECT: COLLC(567) 1206
3-35-4 MOVE BIT: MOVBC(568) 1211
3-35-5 BIT COUNTER: BCNTC(621) 1212
3-35-6 GET VARIABLE ID: GETID(286) 1214

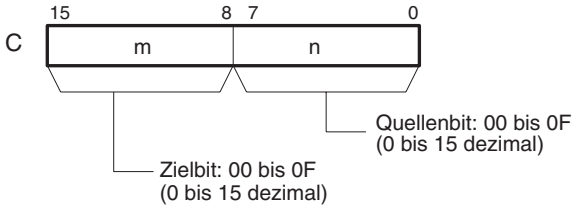
3-1 Notation und Layout von Befehlsbeschreibungen

Befehle werden nach Funktion gruppiert beschrieben. Eine Liste der Befehle nach AWL mit der Angabe der Seitennummer in diesem Kapitel für jeden Befehl finden Sie unter *2-3 Alphabetische Liste der Befehle nach AWL-Kürzel*.

Die Beschreibung der einzelnen Befehle ist entsprechend der Darstellung in der folgenden Tabelle organisiert.

<table>
<tr><th colspan="2">Eigenschaft</th><th>Inhalt</th></tr>
<tr><td colspan="2">Name und AWL</td><td>Die Überschrift der einzelnen Abschnitte besteht aus dem Namen des Befehls gefolgt von dem AWL-Kürzel mit dem Funktionscode in Klammern. Beispiel: MOVE BIT: MOVB(082)</td></tr>
<tr><td colspan="2">Funktion</td><td>Auf die Abschnittsüberschrift folgt eine Beschreibung der grundlegenden Verwendung des jeweiligen Befehls.</td></tr>
<tr><td colspan="2">Kontaktplansymbol und Operandennamen</td><td>Es wird das zur Darstellung des Befehls im CX-Programmer verwendete Kontaktplansymbol gezeigt, wie in dem nachstehenden Beispiel für den MOVE BIT-Befehl. Im Kontaktplansymbol sind zudem die Namen der einzelnen Operanden angegeben.
 

S: Quellwort oder -daten
C: Steuerwort
D: Zielwort</td></tr>
<tr><td rowspan="2">Variationen</td><td>Variationen</td><td>Die Variationen, die verwendet werden können, um die Ausführung des Befehls unter bestimmten Bedingungen steuern zu können, werden unter Verwendung der AWL-Form angegeben. Jede Variation, die von einem Befehl nicht unterstützt wird, ist als "Nicht unterstützt" angegeben.
• Ausführung in jedem Zyklus bei EIN-Bedingung: Der Befehl wird so lange ausgeführt, wie er eine EIN-Ausführungsbedingung empfängt.
• Ausführung einmal bei steigender Flanke: Der Befehl wird nur dann während des nächsten Zyklus ausgeführt, nachdem der Zustand der Ausführungsbedingung von AUS zu EIN gewechselt ist.
• Ausführung einmal bei fallender Flanke: Der Befehl wird nur dann während des nächsten Zyklus ausgeführt, nachdem der Zustand der Ausführungsbedingung von EIN zu AUS gewechselt ist.
• Stets ausgeführt: Der Befehl erfordert keine Ausführungsbedingung und wird in jedem Zyklus ausgeführt.
• Erzeugt EIN-Bedingung ... Der Befehl wird in jedem Zyklus ausgeführt, um eine Ausführungsbedingung für den nächsten Befehl zu erzeugen.
<table><tr><td rowspan="3">Variationen</td><td>Ausführung in jedem Zyklus bei EIN-Bedingung</td><td>MOVB(082)</td></tr><tr><td>Ausführung einmal bei steigender Flanke</td><td>@MOVB(082)</td></tr><tr><td>Ausführung einmal bei fallender Flanke</td><td>Nicht unterstützt</td></tr></table></td></tr>
<tr><td rowspan="2">Variationen</td><td>Variationen</td></tr>
<tr><td>Spezifikation der direkten Aktualisierung</td><td>Die direkte Aktualisierung kann für einige Befehle spezifiziert werden, damit die E/A-Aktualisierung unmittelbar bei Ausführung des Befehl erfolgt. Wenn die direkte Aktualisierung unterstützt wird, erfolgt die Angabe unter Verwendung der AWL-Form. Wenn die direkte Aktualisierung nicht von einem Befehl unterstützt wird, ist "Nicht unterstützt" angegeben.
<table><tr><td>Spezifikation der direkten Aktualisierung</td><td>Nicht unterstützt</td></tr></table></td></tr>
<tr><td colspan="2">Anwendbarkeit in Programmbereichen</td><td>Die Programmbereiche, in denen der Befehl verwendet werden kann, werden angegeben. "OK" kennzeichnet die Bereiche, in denen der Befehl verwendet werden.
<table><tr><th>Blockprogrammbereiche</th><th>Schrittprogrammbereiche</th><th>Unterprogramme</th><th>Interrupt-Tasks</th></tr><tr><td>OK</td><td>OK</td><td>OK</td><td>OK</td></tr></table></td></tr>
</table>

<table>
<tr><th>Eigenschaft</th><th>Inhalt</th></tr>
<tr><td>Operanden</td><td>In bestimmten Operanden, beispielsweise Steuerworten, haben die Bits in den Operanden eine spezielle Bedeutung.
 
</td></tr>
<tr><td>Operanden-Spezifikationen</td><td>Die Adressen der Speicherbereiche, die für die einzelnen Operanden verwendet werden können, sind in einer Tabelle wie der folgenden aufgelistet. Die in den Spaltenüberschriften verwendeten Buchstaben links sind mit den im Kontaktplansymbol verwendeten identisch. "---" wird verwendet, wenn ein Bereich nicht für einen Operanden spezifisch sein kann.
<table>
<tr><th>Bereich</th><th>S</th><th>C</th><th>D</th></tr>
<tr><td>CIO-Bereich</td><td colspan="3">CIO 0000 bis CIO 6143</td></tr>
<tr><td>Arbeitsbereich nicht remanent</td><td colspan="3">W000 bis W511</td></tr>
<tr><td>Arbeitsbereich remanent</td><td colspan="3">H000 bis H511</td></tr>
<tr><td>Systembereich</td><td colspan="2">A000 bis A959</td><td>A448 bis A959</td></tr>
<tr><td>Zeitgeberbereich</td><td colspan="3">T0000 bis T4095</td></tr>
<tr><td>Zählerbereich</td><td colspan="3">C0000 bis C4095</td></tr>
<tr><td>DM-Bereich</td><td colspan="3">D00000 bis D32767</td></tr>
<tr><td>EM-Bereich ohne Bank</td><td colspan="3">E00000 bis E32767</td></tr>
</table></td></tr>
<tr><td>Beschreibung</td><td>Die Funktion des Befehls und der im Befehl verwendeten Operanden wird beschrieben.</td></tr>
<tr><td>Flags</td><td>Die Flag-Tabelle gibt den Status der Bedingungs-Flags unmittelbar nach der Ausführung des Befehls an. Der Befehl hat keine Auswirkungen auf nicht aufgeführte Flags. "AUS" gibt an, dass ein Flag unmittelbar nach der Ausführung des Befehls auf AUS gesetzt ist, und zwar unabhängig von den Ergebnissen der Ausführung des Befehls.
<table>
<tr><th>Name</th><th>Bezeichnung</th><th>Funktion</th></tr>
<tr><td>Fehler-Flag</td><td>ER</td><td>EIN, wenn die Steuerdaten innerhalb der Bereiche liegen.
In allen anderen Fällen AUS.</td></tr>
<tr><td>Gleich-Flag</td><td>=</td><td>AUS</td></tr>
<tr><td>Negativ-Flag</td><td>N</td><td>AUS</td></tr>
</table></td></tr>
<tr><td>Vorsichtsmaßnahmen</td><td>Es werden spezielle Vorsichtsmaßnahmen bei der Verwendung des Befehls angegeben. Lesen Sie diese Vorsichtsmaßnahmen durch, und beachten Sie sie.</td></tr>
<tr><td>Beispiel</td><td>Es wird ein Beispiel für die Verwendung des Befehls mit bestimmten Operanden angeführt, um die Funktion des Befehls näher zu erläutern.</td></tr>
</table>

Konstanten

Die Konstanteneingabe für Operanden wird wie im Folgenden aufgeführt angegeben.

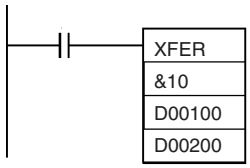
Operanden-Beschreibungen und Operanden-Spezifikationen

- Operanden, die Bit-Zeichenketten spezifizieren (Normalerweise als Hexadezimaldaten eingegeben):
  Für Operanden, die Bit-Zeichenketten spezifizieren, wird nur die Hexadezimalform angegeben, beispielsweise wird für den Befehl MOV(021) nur „#0000 bis #FFFF“ für den Operanden spezifiziert. In CX-Programmer können Bit-Zeichenketten jedoch in Dezimalform unter Verwendung des Präfix „&“ eingegeben werden.
- Operanden, die numerische Werte spezifizieren (normalerweise in Dezimalform eingegeben, einschließlich Sprungnummern):
  Operanden, die numerische Werte spezifizieren, werden sowohl in Dezimal- als auch Hexadezimalform angegeben, beispielsweise werden die Werte „#0000 bis #FFFF“ und „&0 bis &65535“ für den Operanden „N“ des Befehls XFER(070) angegeben.

- Operanden, die Steuerwerte angeben (mit Ausnahme von Sprungnummern): Steuerwerte werden in Dezimalform angegeben, z. B. wird „0 bis 1023“ für den Operanden „N“ des Befehls SBS(091) angegeben.

Beispiele

In den Beispielen werden Konstanten in der CX-Programmer-Notation angegeben, z. B. werden Operanden, die numerische Werte spezifizieren, in Dezimalform und dem Präfix „&“ angegeben, wie in dem folgenden Beispiel gezeigt.



Die Eingabemethoden für Konstanten bei Programmiergeräten sind in der folgenden Tabelle aufgeführt.

Operand	CX-Programmer	Programmierkonsole
Operanden, die Bit-Zeichenketten spezifizieren (normalerweise als Hexadezimalwert eingegeben): Operanden, die numerische Werte spezifizieren (normalerweise als Dezimalwert eingegeben)	Eingabe in Dezimalform mit „&“ als Präfix oder Eingabe in Hexadezimalform mit „#“ als Präfix (siehe Hinweis).	Die Taste „Cont/#“ kann gedrückt werden, um Hexadezimalwerte standardmäßig mit „#“ als Präfix einzugeben. Die Taste „CHG“ kann anschließend gedrückt werden, um zwischen Hexadezimal (mit dem Präfix „#“), Dezimal mit Vorzeichen (mit +/–) und Dezimal ohne Vorzeichen (mit dem Präfix "&") zu wechseln.
Operanden, die Steuerwerte spezifizieren (mit Ausnahme von Sprungnummern)	Eingabe in Dezimalform mit „#“ als Präfix (siehe Hinweis).	Direkte Eingabe in Dezimalform. Wenn das Präfix „&“ automatisch hinzugefügt wird, kann die Taste "CHG" gedrückt werden, um zwischen der Dezimal ohne Vorzeichen (mit „&“ als Präfix), Hexadezimal (mit „#“ als Präfix) und Dezimal mit Vorzeichen (mit +/–) zu wechseln. Wenn keine Präfix angezeigt wird, muss der Wert in Dezimalform eingegeben werden.

Hinweis Wenn Operanden bei CX-Programmer eingegeben werden, werden die Eingabebereiche zusammen mit den entsprechenden Präfixen angezeigt.

Bedingungs-Flags

Für Bedingungs-Flags werden in diesem Kapitel die Bezeichnungen für die Programmierkonsole verwendet. Bei CX-Programmer werden die Bedingungs-Flags im Vorfeld als globale Symbole mit „P_“ vor dem Symbolnamen registriert.

Flag	Bezeichnung bei Programmierkonsole	Bezeichnung bei CX-Programmer
Fehler-Flag	ER	P_ER
Zugriffsfehler-Flag	AER	P_AER
Übertrags-Flag	CY	P_CY
Größer-als-Flag	>	P_GT
Gleich-Flag	=	P_EQ
Kleiner-als-Flag	<	P_LT
Negativ-Flag	N	P_N
Überlauf-Flag	OF	P_OF
Unterlauf-Flag	UF	P_UF
Größer-oder-gleich-Flag	>=	P_GE
Ungleich-Flag	<>	P_NE

Flag	Bezeichnung bei Programmierkonsole	Bezeichnung bei CX-Programmer
Kleiner-oder-gleich-Flag	<=	P_LE
Immer-EIN-Flag	ON	P_On
Immer-AUS-Flag	OFF	P_Off

3-2 Befehlserweiterungen und neue Befehle

In diesem Abschnitt sind die Befehlserweiterungen für CS1 CPU-Baugruppen mit dem Suffix „-EV1“ und für CS1-H/CJ1-H CPU-Baugruppen aufgeführt.

3-2-1 Erweiterungen für CS-Serie CPU-Baugruppen der Version 1

Die folgenden Befehle wurden für CPU-Baugruppen der Version 1 in ihrer Funktionalität erweitert. Weitere Informationen finden Sie auf den angegebenen Seiten.

Funktionen, die nur von CPU-Baugruppen der Version 1 unterstützt werden, sind mit „(nur -EV1)“ gekennzeichnet.

Name	AWL	Funktionscode	Funktion	Erweiterung	Seite
READ DATA FILE	FREAD	700	Jetzt werden die Datenformate CSV und Text (.txt) unterstützt. (Zuvor wurden nur Binärdaten unterstützt.)	Der Inhalt von Steuerdaten wurde geändert, indem Spezifikationen wie Datenformat, vorhandene Wagenrückläufe und Wagenrücklaufpositionen hinzugefügt wurden.	1045
WRITE DATA FILE	FWRIT	701			1052
DELIVER COMMAND	CMND	490	Die CPU-Baugruppe kann nun den FINS-Befehl an sich selbst senden. (Dies war zuvor nicht möglich.)	Die Fähigkeit, FINS-Befehle an die CPU-Baugruppe durch die Ausführung von CMND(490) zu senden, wurde hinzugefügt.	1003

3-2-2 Funktionserweiterungen für CS1-H/CJ1-H CPU-Baugruppen

Neue Befehle

Die folgenden Befehle sind für die CS1-H und CJ1-H CPU-Baugruppen neu hinzugekommen.

Sequenzausgabebefehle

SINGLE BIT SET, SETB(532)
SINGLE BIT RESET, RSTB(533)
SINGLE BIT OUTPUT, OUTB(534)

Datenvergleichsbefehle

AREA RANGE COMPARE, ZCP(088)
DOUBLE AREA RANGE COMPARE, ZCPL(116)

Befehle für die Berechnung und Konvertierung von Gleitkommadaten

Gleitkommadaten-Vergleichsbefehle: =F, <>F, <F, <=F, >F und >=F (329 bis 334)
FLOATING POINT TO ASCII, FSTR(448)
ASCII TO FLOATING POINT, VAL(449)

Befehle für die Berechnung und Konvertierung von Gleitkommadaten doppelter Genauigkeit

Vergleichsbefehle doppelter Genauigkeit: =D, <>D, <D, <=D, >D und >=D (335 bis 340)
DOUBLE FLOATING TO 16-BIT BINARY, FIXD(841)
DOUBLE FLOATING TO 32-BIT BINARY, FIXLD(8420)
16-BIT BINARY TO DOUBLE FLOATING, DBL(843)
32-BIT BINARY TO DOUBLE FLOATING, DBLL(844)
DOUBLE FLOATING-POINT ADD, +D(845)
DOUBLE FLOATING-POINT SUBTRACT, −D(846)
DOUBLE FLOATING-POINT MULTIPLY, ∗D(847)

DOUBLE FLOATING-POINT DIVIDE, /D(848)
DOUBLE DEGREES TO RADIANS, RADD(849)
DOUBLE RADIANS TO DEGREES, DEGD(850)
DOUBLE SINE, SIND(851)
DOUBLE COSINE, COSD(852)
DOUBLE TANGENT, TAND(853)
DOUBLE ARC SINE, ASIND(854)
DOUBLE ARC COSINE, ACOSD(855)
DOUBLE ARC TANGENT, ATAND(856)
DOUBLE SQUARE ROOT, SQRTD(857)
DOUBLE EXPONENT, EXPD(858)
DOUBLE LOGARITHM, LOGD(859)
DOUBLE EXPONENTIAL POWER, PWRD(860)

Tabellendaten-Verarbeitungsbefehle

STACK SIZE READ, SNUM(638)
STACK DATA READ, SREAD(639)
STACK DATA WRITE, SWRIT(640)
STACK DATA INSERT, SINS(641)
STACK DATA DELETE, SDEL(642)

Regelungsbefehle

PID CONTROL WITH AUTOTUNING, PIDAT(191)

Unterprogrammbefehle

GLOBAL SUBROUTINE CALL, GSBS(750)
GLOBAL SUBROUTINE ENTRY, GSBN(751)
GLOBAL SUBROUTINE RETURN, GRET(752)

Befehle für E/A-Baugruppen

CPU BUS UNIT I/O REFRESH, DLNK(226)

Sonstige Befehle

SAVE CONDITION FLAGS, CCS(282)
LOAD CONDITION FLAGS, CCL(283)
CONVERT ADDRESS FROM CV, FRMCV(284)
CONVERT ADDRESS TO CV, TOCV(285)
DISABLE PERIPHERAL SERVICING, IOSP(287)
ENABLE PERIPHERAL SERVICING, IORS(288)

Neue Befehle

Die folgenden Befehle wurden für die CS1-H und CJ1-H CPU-Baugruppen in ihrer Funktionalität erweitert.

Spezielle arithmetische Befehle

ARITHMETIC PROCESS, APR(069)

Fehlerdiagnosebefehle

FAILURE ALARM, FAL(006)
SEVERE FAILURE ALARM, FALS(007)

3-3 Sequenzeingabebefehle

3-3-1 LOAD: LD

Funktion

Bildet den Anfang der logischen Verknüpfung und erzeugt eine EIN/AUS-Ausführungsbedingung auf der Grundlage des EIN/AUS-Status des spezifizierten Operanden-Bits.

Kontaktplansymbol



Variationen

Variationen	**Startet die Logik neu und erzeugt in jedem Zyklus eine EIN-Bedingung, in dem das Operand-Bit auf EIN gesetzt ist**	LD
	Startet die Logik neu und erzeugt eine EIN-Bedingung einmalig bei steigender Flanke	@LD
	die Logik neu und erzeugt eine EIN-Bedingung einmalig bei fallender Flanke	%LD
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!LD
Kombinierte Variationen	**Aktualisiert das Eingangsbit, Startet die Logik neu und erzeugt eine EIN-Bedingung einmalig bei steigender Flanke (siehe Hinweis)**	!@LD
	Aktualisiert das Eingangsbit, Startet die Logik neu und erzeugt eine EIN-Bedingung einmalig bei fallender Flanke (siehe Hinweis)	!%LD

Hinweis Direkte Aktualisierung wird von CS1D CPU-Baugruppen für Duplex-CPU-Systeme nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	LD-Operand-Bit
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Taskflagbereich	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, A1, A0
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 Min
TR-Bereich	TR0 bis TR15
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---

Bereich	LD-Operand-Bit
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

LD wird für das erste Schließer-Bit an der Stromschiene oder für das erste Schließer-Bit eines logischen Blocks verwendet. Wenn keine Spezifikation der direkten Aktualisierung vorhanden ist, wird das spezifizierte Bit im E/A-Speicher gelesen. Wenn eine Spezifikation der direkten Aktualisierung vorhanden ist, wird der Status der Eingangsklemmen der Eingangsbaugruppe gelesen und verwendet.

LD wird unter folgenden Umständen als Befehl für die Bildung des Anfangs einer logischen Verknüpfung verwendet.

- Bei direkter Verbindung mit der Stromschiene.
- Bei der Verbindung logischer Blöcke durch AND LD oder OR LD, also am Anfang eines logischen Blocks.

Die Befehle AND LOAD und OR LOAD werden für die Reihen- oder Parallelverbindung von Logikblöcken verwendet, die mit LD oder LD NOT beginnen.

Für die Ausführungsbedingung ist mindestens ein LOAD- oder LOAD NOT-Befehl erforderlich, wenn Ausgabebefehle nicht direkt mit der Stromschiene verbunden werden können. Wenn kein LOAD- oder LOAD NOT-Befehl vorhanden ist, tritt bei der Programmprüfung durch das Peripheriegerät ein Programmfehler auf.

Wenn logische Blöcke über AND LOAD- oder OR LOAD-Befehle verbunden werden, muss die Gesamtzahl der AND LOAD/OR LOAD-Befehle der Gesamtzahl der LOAD/LOAD NOT-Befehle minus 1 entsprechen. Stimmen die Gesamtzahlen nicht überein, tritt ein Programmfehler auf. Weitere Informationen dazu finden Sie unter *3-3-7 AND LOAD: AND LD* und *3-3-8 OR LOAD: OR LD*.

Flags

Dieser Befehl hat keinen Einfluss auf Flags.

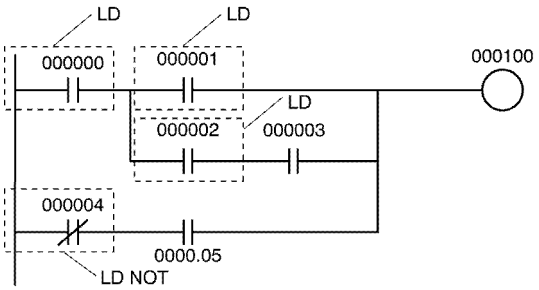
Vorsichtsmaßnahmen

Ausführung bei steigender Flanke (@) oder Ausführung bei fallender Flanke (%) kann für LD spezifiziert werden. Ist Ausführung bei steigender Flanke (@) spezifiziert, wird die Ausführungsbedingung erst für einen Zyklus auf EIN gesetzt, nachdem der Status des Operanden-Bits von AUS zu EIN gewechselt ist. Wenn Ausführung bei fallender Flanke (%) spezifiziert wurde, wird die Ausführungsbedingung erst für einen auf EIN gesetzt, nachdem der Status des Operanden-Bits von EIN zu AUS gewechselt ist.

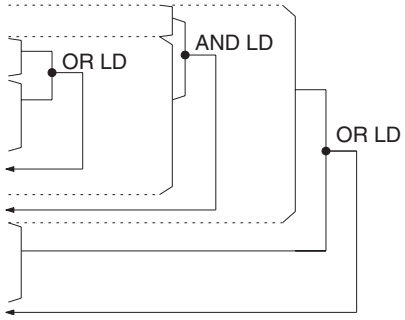
Direkte Aktualisierung (!) kann für LD spezifiziert werden. Durch einen Befehl mit direkter Aktualisierung wird der Status des Eingangsbits bei Basis-Eingangsbaugruppen kurz vor der Ausführung des Befehls aktualisiert (dies gilt jedoch nicht für Basis-Eingangsbaugruppen auf Remote-Slave-Baugruppenträgern oder für C200H Multipunkt-Eingangsbaugruppen der Gruppe 2).

Für LD können direkte Aktualisierung und Ausführung bei steigender oder fallender Flanke (!@ oder !%) kombiniert werden. Bei der Festlegung einer dieser Optionen wird der Eingangsstatus der Basis-Eingangsbaugruppe kurz vor der Ausführung des Befehls aktualisiert und die Ausführungsbedingung wird nur dann einen Zyklus lang auf EIN gesetzt, nachdem der Status von AUS auf EIN oder von EIN auf AUS gewechselt ist.

Beispiel



Befehl	Operand
LD	000000
LD	000001
LD	000002
AND	000003
OR LD	---
AND LD	---
LD NOT	000004
AND	000005
OR LD	---
OUT	000100

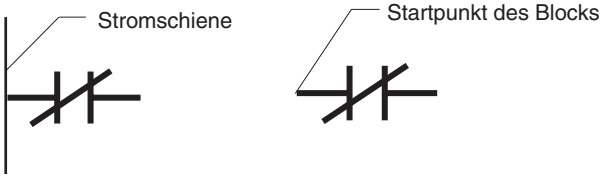


3-3-2 LOAD NOT: LD NOT

Funktion

Bildet den Anfang der logischen Verknüpfung und erzeugt eine EIN/AUS-Ausführungsbedingung auf der Grundlage des invertierten EIN/AUS-Status des spezifizierten Operanden-Bits.

Kontaktplansymbol



Variationen

Variationen	**Startet die Logik neu und erzeugt eine EIN-Bedingung in jedem Zyklus, in dem das Operand-Bit auf AUS gesetzt ist**	LD NOT
	Startet die Logik neu und erzeugt eine EIN-Bedingung einmalig bei steigender Flanke (siehe Hinweis 1)	@LD NOT
	Startet die Logik neu und erzeugt eine EIN-Bedingung einmalig bei fallender Flanke (siehe Hinweis 1)	%LD NOT
Spezifikation der direkten Aktualisierung (siehe Hinweis 2)		!LD NOT
Kombinierte Variationen	**Aktualisiert das Eingangsbit, startet die Logik neu und erzeugt eine EIN-Bedingung einmalig bei steigender Flanke (siehe Hinweis 3)**	!@LD NOT
	Aktualisiert das Eingangsbit, startet die Logik neu und erzeugt eine EIN-Bedingung einmalig bei fallender Flanke (siehe Hinweis 3)	!%LD NOT

Hinweis

1. Die folgenden Variationen werden nur von den CS1-H, CJ1-H, CJ1M oder CS1D CPU-Baugruppen unterstützt: @LD NOT, %LD NOT, !@LD NOT und !%LD NOT.
2. Direkte Aktualisierung wird von CS1D CPU-Baugruppen für Duplex-CPU-Systeme nicht unterstützt.
3. Kombinierte Varianten werden nur von CS1D CPU-Baugruppen für Einzel-CPU-Systeme sowie CS1-H, CJ1-H und CJ1M CPU-Baugruppen unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	LD NOT Bit-Operand
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Taskflagbereich	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 Min
TR-Bereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

LD NOT wird für das erste Öffner-Bit an der Stromschiene oder für das erste Öffner-Bit eines logischen Blocks verwendet. Wenn keine Spezifikation der direkten Aktualisierung vorhanden ist, wird das angegebene Bit im E/A-Speicher gelesen und invertiert. Wenn eine Spezifikation der direkten Aktualisierung vorhanden ist, wird der Status der Eingangsklemmen der Basis-Eingangsbaugruppe gelesen, invertiert und verwendet.

LD NOT wird unter folgenden Bedingungen als Befehl für die Bildung des Anfangs einer logischen Verknüpfung verwendet.

- Bei direkter Verbindung mit der Stromschiene.
- Bei der Verbindung logischer Blöcke durch AND LD oder OR LD (am Anfang eines logischen Blocks verwendet).

Die Befehle AND LOAD und OR LOAD werden für die Reihen- oder Parallelverbindung von Logikblöcken verwendet, die mit LD oder LD NOT beginnen.

Für die Ausführungsbedingung ist mindestens ein LOAD- oder LOAD NOT-Befehl erforderlich, wenn Ausgabebefehle nicht direkt mit der Stromschiene verbunden werden können. Wenn kein LOAD- oder LOAD NOT-Befehl vorhanden ist, tritt bei der Programmprüfung durch das Peripheriegerät ein Programmfehler auf.

Wenn logische Blöcke über AND LOAD- oder OR LOAD-Befehle verbunden werden, muss die Gesamtzahl der AND LOAD/OR LOAD-Befehle der Gesamtzahl der LOAD/LOAD NOT-Befehle minus 1 entsprechen. Stimmen die Gesamtzahlen nicht überein, tritt ein Programmfehler auf.

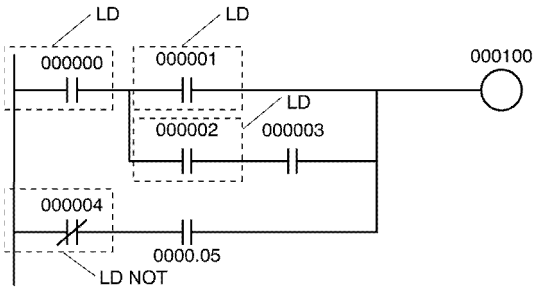
Flags

Dieser Befehl hat keinen Einfluss auf Flags.

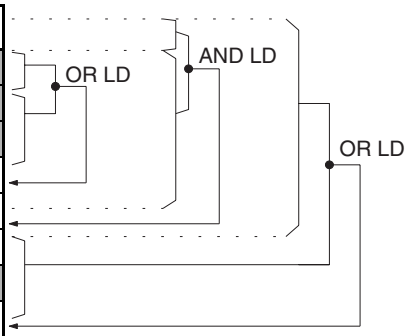
Vorsichtsmaßnahmen

Für den Befehl LD NOT kann direkte Aktualisierung (!) spezifiziert werden. Durch einen Befehl mit direkter Aktualisierung wird der Status des Eingangsbits bei Basis-Eingangsbaugruppen kurz vor der Ausführung des Befehls aktualisiert (dies gilt jedoch nicht für Basis-Eingangsbaugruppen auf Remote-Slave-Baugruppenträgern oder für C200H Multipunkt-Eingangsbaugruppen der Gruppe 2).

Beispiel



Befehl	Operand
LD	000000
LD	000001
LD	000002
AND	000003
OR LD	---
AND LD	---
LD NOT	000004
AND	000005
OR LD	---
OUT	000100



3-3-3 AND: AND

Funktion

Bildet eine logische UND-Verknüpfung zwischen dem Status des spezifizierten Operanden-Bits und der aktuellen Ausführungsbedingung.

Kontaktplansymbol



Variationen

Variationen	**Erzeugt in jedem Zyklus eine EIN-Bedingung, in dem das AND-Ergebnis EIN ist.**	AND
	Erzeugt eine EIN-Bedingung einmalig bei steigender Flanke	@AND
	Erzeugt eine EIN-Bedingung einmalig bei fallender Flanke	%AND
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!AND
Kombinierte Variationen	**Aktualisiert das Eingangsbit und erzeugt eine EIN-Bedingung einmalig bei steigender Flanke (siehe Hinweis)**	!@AND
	Aktualisiert das Eingangsbit und erzeugt eine EIN-Bedingung einmalig bei fallender Flanke (siehe Hinweis)	!%AND

Hinweis Direkte Aktualisierung wird von CS1D CPU-Baugruppen für Duplex-CPU-Systeme nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	AND Bit-Operand
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Taskflagbereich	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
TR-Bereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

AND wird für die Reihenschaltung eines Schließer-Bits verwendet. AND kann nicht direkt mit der Stromschiene verbunden und nicht zu Beginn eines logischen Blocks verwendet werden. Wenn keine Spezifikation der direkten Aktualisierung vorhanden ist, wird das spezifizierte Bit im E/A-Speicher gelesen. Wenn eine Spezifikation der direkten Aktualisierung vorhanden ist, wird der Status der Eingangsklemmen der Basis-Eingangsbaugruppe gelesen.

Flags

Dieser Befehl hat keinen Einfluss auf Flags.

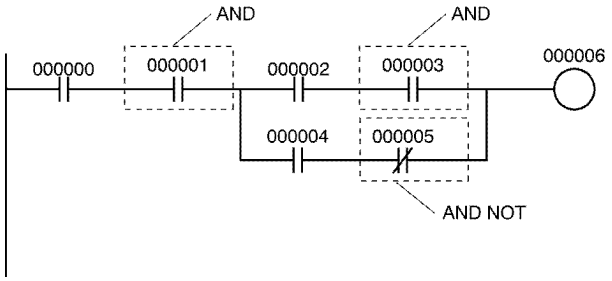
Vorsichtsmaßnahmen

Für den Befehl AND kann die Ausführung bei steigender Flanke (@) oder bei fallender Flanke (%) spezifiziert werden. Wenn Ausführung bei steigender Flanke (@) spezifiziert wurde, wird die Ausführungsbedingung erst für einen Zyklus auf EIN gesetzt, nachdem der Status des Operanden-Bits von AUS zu EIN gewechselt ist. Wenn Ausführung bei fallender Flanke (%) spezifiziert wurde, wird die Ausführungsbedingung erst für einen auf EIN gesetzt, nachdem der Status des Operanden-Bits von EIN zu AUS gewechselt ist.

Für AND kann direkte Aktualisierung (!) spezifiziert werden. Durch einen Befehl mit direkter Aktualisierung wird der Status des Eingangsbits bei Basis-Eingangsbaugruppen kurz vor der Ausführung des Befehls aktualisiert (dies gilt jedoch nicht für Basis-Eingangsbaugruppen auf Remote-Slave-Baugruppenträgern oder für C200H Multipunkt-Eingangsbaugruppen der Gruppe 2).

Für den Befehl AND können direkte Aktualisierung und Ausführung bei steigender oder fallender Flanke (!@ oder !%) kombiniert werden. Bei der Festlegung einer dieser Optionen wird der Eingangsstatus der Basis-Eingangsbaugruppe kurz vor der Ausführung des Befehls aktualisiert und die Ausführungsbedingung wird nur dann einen Zyklus lang auf EIN gesetzt, nachdem der Status von AUS auf EIN oder von EIN auf AUS gewechselt ist.

Beispiel



Befehl	Operand
LD	000000
AND	000001
LD	000002
AND	000003
LD	000004
AND NOT	000005
OR LD	---
AND LD	---
OUT	000006

3-3-4 AND NOT: AND NOT

Funktion

Kehrt den Status des spezifizierten Operanden-Bits um und bildet ein logisches UND mit der aktuellen Ausführungsbedingung.

Kontaktplansymbol



Variationen

Variationen	**Erzeugt eine EIN-Bedingung in jedem Zyklus, in dem das AND NOT-Ergebnis EIN ist.**	AND NOT
	Erzeugt eine EIN-Bedingung einmalig bei steigender Flanke (siehe Hinweis 1)	@AND NOT
	Erzeugt eine EIN-Bedingung einmalig bei fallender Flanke (siehe Hinweis 1)	%AND NOT
Spezifikation der direkten Aktualisierung (siehe Hinweis 2)		!AND NOT
Kombinierte Variationen	**Aktualisiert das Eingangsbit und erzeugt eine EIN-Bedingung einmalig bei steigender Flanke (siehe Hinweis 3)**	!@AND NOT
	Aktualisiert das Eingangsbit und erzeugt eine EIN-Bedingung einmalig bei fallender Flanke (siehe Hinweis 3)	!%AND NOT

Hinweis

1. Die folgenden Variationen werden nur von den CS1-H, CJ1-H, CJ1M oder CS1D CPU-Baugruppen unterstützt: @AND NOT, %AND NOT, !@AND NOT und !%AND NOT.
2. Direkte Aktualisierung wird von CS1D CPU-Baugruppen für Duplex-CPU-Systeme nicht unterstützt.
3. Kombinierte Varianten werden nur von CS1D CPU-Baugruppen für Einzel-CPU-Systeme sowie CS1-H, CJ1-H und CJ1M CPU-Baugruppen unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	AND NOT Bit-Operand
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115

Bereich	AND NOT Bit-Operand
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Taskflagbereich	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
TR-Bereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

AND NOT wird für ein in Reihe geschaltetes Öffner-Bit verwendet. AND NOT kann nicht direkt mit der Stromschiene verbunden und nicht zu Beginn eines logischen Blocks verwendet werden. Wenn keine Spezifikation der direkten Aktualisierung vorhanden ist, wird das spezifizierte Bit im E/A-Speicher gelesen. Wenn eine Spezifikation der direkten Aktualisierung vorhanden ist, wird der Status der Eingangsklemmen des Basiseingangsmoduls gelesen.

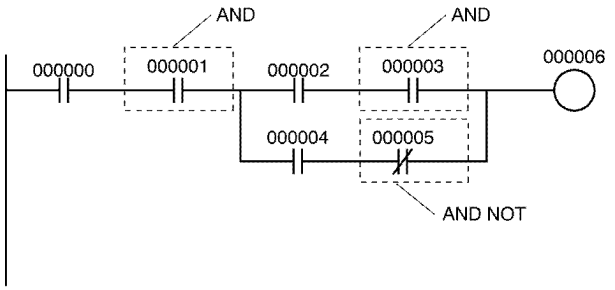
Flags

Dieser Befehl hat keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

Für den Befehl AND NOT kann direkte Aktualisierung (!) spezifiziert werden. Durch einen Befehl mit direkter Aktualisierung wird der Status des Eingangsbits bei Basis-Eingangsbaugruppen kurz vor der Ausführung des Befehls aktualisiert (dies gilt jedoch nicht für Basis-Eingangsbaugruppen auf Remote-Slave-Baugruppenträgern oder für C200H Multipunkt-Eingangsbaugruppen der Gruppe 2).

Beispiel



Befehl	Operand
LD	000000
AND	000001
LD	000002
AND	000003
LD	000004
AND NOT	000005

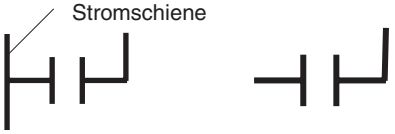
Befehl	Operand
OR LD	---
AND LD	---
OUT	000006

3-3-5 OR: OR

Funktion

Bildet eine logische ODER-Verknüpfung zwischen dem EIN/AUS-Status des spezifizierten Operanden-Bits und der aktuellen Ausführungsbedingung.

Kontaktplansymbol



Variationen

Variationen	**Erzeugt eine EIN-Bedingung in jedem Zyklus, in dem das OR-Ergebnis EIN ist.**	OR
	Erzeugt eine EIN-Bedingung einmalig bei steigender Flanke	@OR
	Erzeugt eine EIN-Bedingung einmalig bei fallender Flanke	%OR
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!OR
Kombinierte Variationen	**Aktualisiert das Eingangsbit und erzeugt eine EIN-Bedingung einmalig bei steigender Flanke (siehe Hinweis)**	!@OR
	Aktualisiert das Eingangsbit und erzeugt eine EIN-Bedingung einmalig bei fallender Flanke (siehe Hinweis)	!%OR

Hinweis Direkte Aktualisierung wird von CS1D CPU-Baugruppen für Duplex-CPU-Systeme nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	OR Bit-Operand
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Taskflagbereich	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---

Bereich	OR Bit-Operand
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

OR wird für ein parallel geschaltetes Schließer-Bit verwendet. Ein Schließer-Bit wird konfiguriert, um eine logische ODER-Verknüpfung mit einem logischen Block zu bilden, der mit einem LOAD- oder LOAD NOT-Befehl beginnt (mit der Stromschiene verbunden oder am Anfang des logischen Blocks). Wenn keine Spezifikation der direkten Aktualisierung vorhanden ist, wird das spezifizierte Bit im E/A-Speicher gelesen. Wenn eine Spezifikation der direkten Aktualisierung vorhanden ist, wird der Status der Eingangsklemmen der Basis-Eingangsbaugruppe gelesen.

Flags

Dieser Befehl hat keinen Einfluss auf Flags.

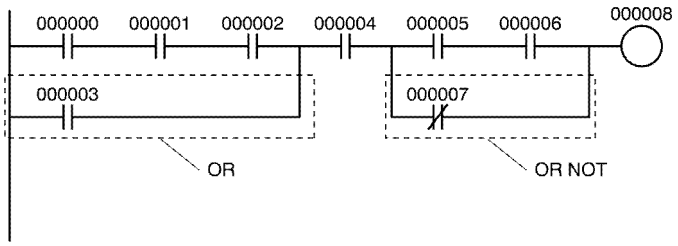
Vorsichtsmaßnahmen

Für den Befehl OR kann die Ausführung bei steigender Flanke (@) oder Ausführung bei fallender Flanke (%) spezifiziert werden. Wenn Ausführung bei steigender Flanke (@) spezifiziert wurde, wird die Ausführungsbedingung erst für einen Zyklus auf EIN gesetzt, nachdem der Status des Operanden-Bits von AUS zu EIN gewechselt ist. Wenn Ausführung bei fallender Flanke (%) spezifiziert wurde, wird die Ausführungsbedingung erst für einen auf EIN gesetzt, nachdem der Status des Operanden-Bits von EIN zu AUS gewechselt ist.

Für den Befehl OR kann direkte Aktualisierung (!) spezifiziert werden. Durch einen Befehl mit direkter Aktualisierung wird der Status des Eingangsbits bei Basis-Eingangsbaugruppen kurz vor der Ausführung des Befehls aktualisiert (dies gilt jedoch nicht für Basis-Eingangsbaugruppen auf Remote-Slave-Baugruppenträgern oder für C200H Multipunkt-Eingangsbaugruppen der Gruppe 2).

Für den Befehl OR können direkte Aktualisierung und Ausführung bei steigender oder fallender Flanke (!@ oder !%) kombiniert werden. Bei Spezifizierung einer dieser Optionen wird der Eingang der Basis-Eingangsbaugruppe unmittelbar vor der Ausführung des Befehls aktualisiert und die Ausführungsbedingung wird nur dann einen Zyklus lang auf EIN gesetzt, wenn der Status des Operand-Bits von AUS zu EIN bzw. von EIN zu AUS gewechselt ist.

Beispiel



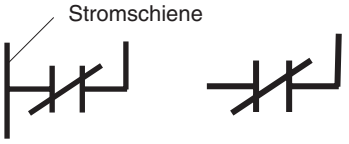
Befehl	Operand
LD	000000
AND	000001
AND	000002
OR	000003
AND	000004
LD	000005
AND	000006
OR NOT	000007
AND LD	---
OUT	000008

3-3-6 OR NOT: OR NOT

Funktion

Kehrt den Status des spezifizierten Operanden-Bits um und bildet ein logisches ODER mit der aktuellen Ausführungsbedingung.

Kontaktplansymbol



Variationen

Variationen	**Erzeugt eine EIN-Bedingung in jedem Zyklus, in dem das OR NOT-Ergebnis EIN ist.**	OR NOT
	Erzeugt eine EIN-Bedingung einmalig bei steigender Flanke (siehe Hinweis 1)	@OR NOT
	Erzeugt eine EIN-Bedingung einmalig bei fallender Flanke (siehe Hinweis 1)	%OR NOT
Spezifikation der direkten Aktualisierung (siehe Hinweis 2)		!OR NOT
Kombinierte Variationen	**Aktualisiert das Eingangsbit und erzeugt eine EIN-Bedingung einmalig bei steigender Flanke (siehe Hinweis 3)**	!@OR NOT
	Aktualisiert das Eingangsbit und erzeugt eine EIN-Bedingung einmalig bei fallender Flanke (siehe Hinweis 3)	!%OR NOT

Hinweis

1. Die folgenden Variationen werden nur von den CS1-H, CJ1-H, CJ1M oder CS1D CPU-Baugruppen unterstützt: @OR NOT, %OR NOT, !@OR NOT und !%OR NOT.
2. Direkte Aktualisierung wird von CS1D CPU-Baugruppen für Duplex-CPU-Systeme nicht unterstützt.
3. Kombinierte Varianten werden nur von CS1D CPU-Baugruppen für Einzel-CPU-Systeme sowie CS1-H, CJ1-H und CJ1M CPU-Baugruppen unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	OR NOT-Bit-Operand
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Taskflagbereich	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, A1, A0
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
TR-Bereich	---
DM-Bereich	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---

Bereich	OR NOT-Bit-Operand
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

OR NOT wird für ein parallel geschaltetes Öffner-Bit verwendet. Ein Öffner-Bit wird konfiguriert, um eine logische OR-Verknüpfung mit einem logischen Block zu bilden, beginnend mit einem LOAD- oder LOAD NOT-Befehl (mit der Stromschiene verbunden oder am Anfang des logischen Blocks). Wenn keine Spezifikation der direkten Aktualisierung vorhanden ist, wird das spezifizierte Bit im E/A-Speicher gelesen. Wenn eine Spezifikation der direkten Aktualisierung vorhanden ist, wird der Status der Eingangsklemmen der Basis-Eingangsbaugruppe gelesen.

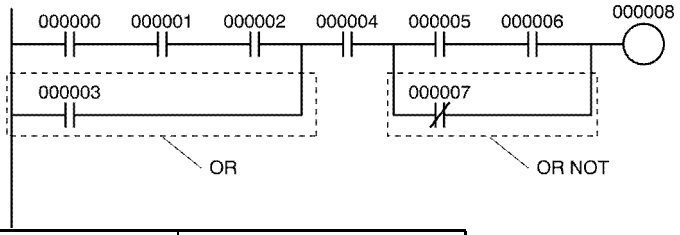
Flags

Dieser Befehl hat keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

Für OR NOT kann direkte Aktualisierung (!) spezifiziert werden. Durch einen Befehl mit direkter Aktualisierung wird der Status des Eingangsbits bei Basis-Eingangsbaugruppen unmittelbar vor der Ausführung des Befehls aktualisiert (dies gilt jedoch nicht für Basis-Eingangsbaugruppen auf Remote-Slave-Baugruppenträgern oder für C200H Multipunkt-Eingangsbaugruppen der Gruppe 2).

Beispiel



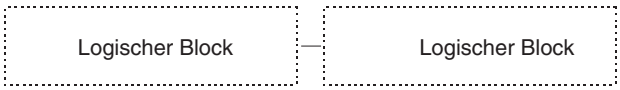
Befehl	Operand
LD	000000
AND	000001
AND	000002
OR	000003
AND	000004
LD	000005
AND	000006
OR NOT	000007
AND LD	---
OUT	000008

3-3-7 AND LOAD: AND LD

Funktion

Führt eine logische AND-Bedingung zwischen logischen Blöcken ein.

Kontaktplansymbol



Variationen

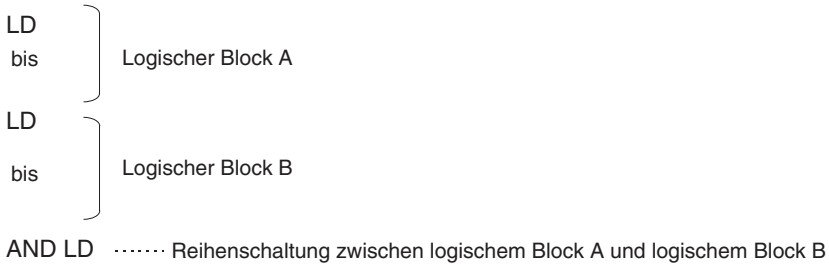
Variationen	Erzeugt in jedem Zyklus eine EIN-Bedingung, in dem das AND-Ergebnis EIN ist.	AND LD
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

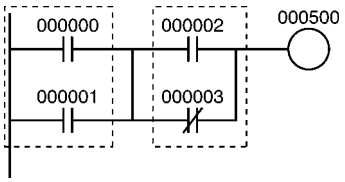
Beschreibung

AND LD verbindet den logischen Block unmittelbar vor Ausführung dieses Befehls mit einem anderen logischen Block in Reihe.



Der logische Block besteht aus allen Befehlen von einem LOAD oder LOAD NOT Befehl bis unmittelbar vor dem nächsten LOAD oder LOAD NOT Befehl im selben Netzwerk.

In der nachfolgenden Abbildung werden die beiden logischen Blöcke durch gepunktete Linien gekennzeichnet. Dieses Beispiel zeigt, dass die EIN-Ausführungsbedingung erzeugt wird, wenn eine der Ausführungsbedingungen im linken Logikblock auf EIN gesetzt ist (also wenn entweder CIO 000000 oder CIO 000001 auf EIN gesetzt ist) **und** eine der Ausführungsbedingungen im rechten Logikblock auf EIN gesetzt ist (also wenn entweder CIO 000002 auf EIN oder CIO 000003 auf AUS gesetzt ist).



Flags

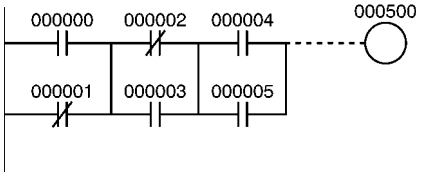
Dieser Befehl hat keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

Mit diesem Befehl können drei oder mehr Logikblöcke in Reihe geschaltet werden, indem zuerst zwei der logischen Blöcke und dann die nächsten in Reihenfolge verbunden werden. Dieser Befehl kann auch hinter drei oder mehr Logikblöcken platziert werden und diese in Reihe schalten.

Wenn ein logischer Block über AND LOAD- oder OR LOAD-Befehle verbunden wird, muss die Gesamtzahl der AND LOAD/OR LOAD-Befehle der Gesamtzahl der LOAD/LOAD NOT-Befehle minus 1 entsprechen. Wenn diese nicht übereinstimmen, tritt ein Programmfehler auf.

Beispiel



Kodierungsbeispiel (1)

Befehl	Operand
LD	000000
OR NOT	000001
LD NOT	000002
OR	000003
AND LD	---
LD	000004
OR	000005

Befehl	Operand
AND LD	---
. .	. .
OUT	000500

Kodierungsbeispiel (2)

Befehl	Operand
LD	000000
OR NOT	000001
LD NOT	000002
OR	000003
LD	000004
OR	000005
. .	. .
AND LD	---
AND LD	---
. .	. .
OUT	000500

Der AND LOAD-Befehl kann wiederholt verwendet werden. Bei der oben aufgeführten Programmiermethode (2) wird die Anzahl der AND LOAD-Befehle jedoch gegenüber der Anzahl vor dem Befehl vorhandener LOAD und LOAD NOT-Befehle um einen Befehl reduziert.

Bei Methode (2) muss sichergestellt werden, dass die Gesamtzahl der LOAD und LOAD NOT-Befehle vor dem AND LOAD-Befehl nicht mehr als acht beträgt. Wenn die Anzahl neun oder mehr beträgt, muss Programmiermethode (1) gewählt werden. Wenn bei Methode (2) neun oder mehr Befehle verwendet werden, tritt während der Programmprüfung durch das Peripheriegerät ein Programmfehler auf.

Kodierung

Adresse	Befehl	Operand
000000	LD	000000
000001	OR	000001
000002	LD	000002
000003	OR NOT	000003
000004	AND LD	---
000005	OUT	000500

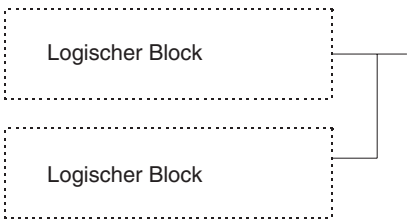
Zweiter LD: Für das erste Bit des nächsten mit dem vorherigen Block in Reihe geschalteten Blocks verwendet.

3-3-8 OR LOAD: OR LD

Funktion

Führt eine logische ODER-Bedingung zwischen logischen Blöcken ein.

Kontaktplansymbol



Variationen

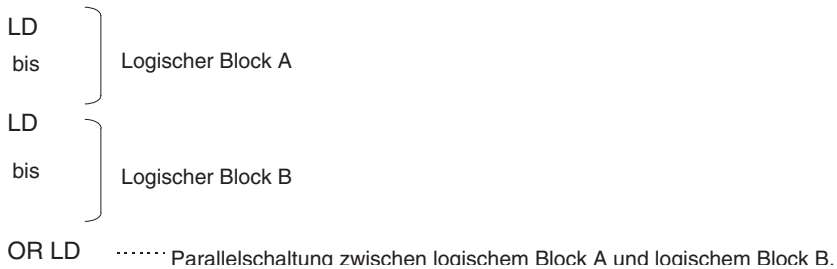
Variationen	Erzeugt in jedem Zyklus eine OR-Bedingung, in dem das OR-Ergebnis EIN ist.	OR LD
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

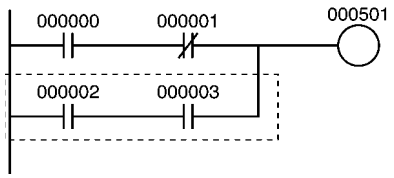
Beschreibung

Der Befehl OR LD bildet eine Parallelschaltung zwischen dem logischen Block unmittelbar vor dem Befehl und einem anderen logischen Block.



Der logische Block besteht aus allen Befehlen von einem LOAD oder LOAD NOT Befehl bis unmittelbar vor dem nächsten LOAD oder LOAD NOT Befehl im selben Netzwerk .

Die folgende Abbildung erfordert einen OR LOAD-Befehl zwischen oberem und unterem Logikblock. Eine EIN-Ausführungsbedingung wird erzeugt, wenn entweder CIO 000000 auf EIN gesetzt ist und CIO 000001 auf AUS gesetzt ist oder wenn CIO 000002 und CIO 000003 auf EIN gesetzt sind. Funktion und AWL-Code des OR LOAD-Befehls sind identisch mit denen für einen AND LOAD-Befehl, außer, dass die aktuelle Ausführungsbedingung eine ODER-Verknüpfung mit der letzten ungebrauchten Ausführungsbedingung bildet.



Flags

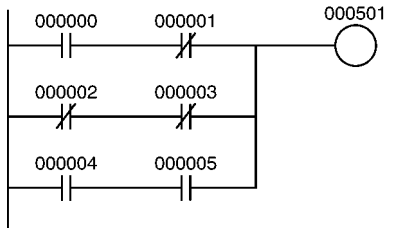
Dieser Befehl hat keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

Mit diesem Befehl können drei oder mehr Logikblöcke parallel geschaltet werden, indem zuerst zwei der logischen Blöcke und dann die nächsten in Reihenfolge verbunden werden. Dieser Befehl kann auch hinter drei oder mehr Logikblöcken platziert und diese parallel schalten.

Wenn ein logischer Block über AND LOAD- oder OR LOAD-Befehle verbunden wird, muss die Gesamtzahl der AND LOAD/OR LOAD-Befehle der Gesamtzahl der LOAD/LOAD NOT-Befehle minus 1 entsprechen. Wenn diese nicht übereinstimmen, tritt ein Programmfehler auf.

Beispiel



Kodierungsbeispiel (1)

Befehl	Operand
LD	000000
AND NOT	000001
LD NOT	000002
AND NOT	000003
OR LD	---
LD	000004
AND	000005
OR LD	---
. .	. .
OUT	000501

Kodierungsbeispiel (2)

Befehl	Operand
LD	000000
AND NOT	000001
LD NOT	000002
AND NOT	000003
LD	000004
AND	000005
. .	. .
OR LD	---
OR LD	---
. .	. .
OUT	000501

Der OR LOAD-Befehl kann wiederholt verwendet werden. Bei der oben genannten Programmiermethode (2) wird die Anzahl der OR LOAD-Befehle jedoch gegenüber der Anzahl vor dem Befehl vorhandener LOAD und LOAD NOT-Befehle um einen Befehl reduziert.

Bei Methode (2) muss sichergestellt werden, dass die Gesamtzahl der LOAD und LOAD NOT-Befehle vor dem OR LOAD-Befehl nicht mehr als acht beträgt. Wenn die Anzahl neun oder mehr beträgt, muss Programmiermethode (1) gewählt werden. Wenn bei Methode (2) neun oder mehr Befehle verwendet werden, tritt während der Programmprüfung durch das Peripheriegerät ein Programmfehler auf.

Kodierung

Adresse	Befehl	Operand
000100	LD	000000
000101	AND NOT	000001
000102	LD	000002
000103	AND	000003
000104	OR LD	---
000105	OUT	000501

Zweiter LD: Für das erste Bit des nächsten mit dem vorherigen Block in Reihe geschalteten Blocks verwendet.

3-3-9 Befehle mit Flankenausführung und Befehle für direkte Aktualisierung

LOAD-, AND- und OR-Befehle besitzen zusätzlich zu ihren regulären Formen Varianten mit Flankenausführung und direkter Aktualisierung. Zudem sind auch zwei Kombinationen möglich.

Für die Befehle LOAD NOT, AND NOT, OR NOT, OUT und OUT NOT gibt es zusätzlich zu ihren regulären Formen Varianten für direkte Aktualisierung.

Der E/A-Aktualisierungszeitpunkt der Daten, die von Befehlen verarbeitet werden, unterscheidet sich zwischen regulären Befehlen, Befehlen mit Flankenauswertung, Befehlen mit direkter Aktualisierung und Befehlen mit direkter Aktualisierung und Flankenüberwachung.

Reguläre Befehle und Befehle mit Flankenauswertung werden unter Verwendung von Daten vorheriger E/A-Aktualisierungen ausgeführt und die Ergebnisse werden bei der nächsten E/A-Verarbeitung ausgegeben. Hier bedeutet „E/A-Aktualisierung“, dass die Daten zwischen dem internen Speicher der CPU und der E/A-Baugruppe ausgetauscht werden.

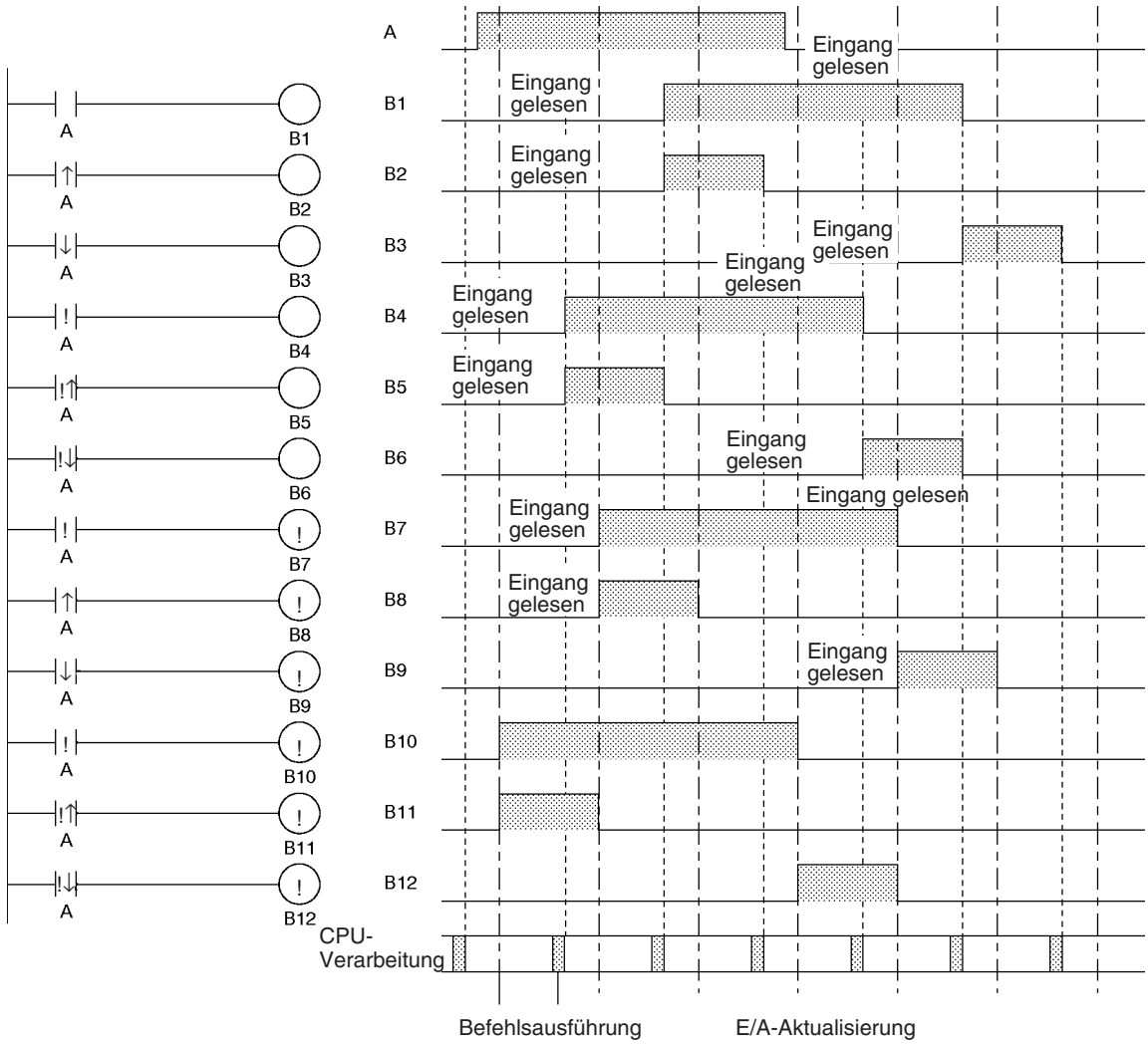
Zusätzlich zu der oben beschriebenen E/A-Aktualisierung tauscht ein Befehl mit direkter Aktualisierung mit der E/A-Baugruppe die Daten solcher Worte aus, auf die mit dem Befehl zugegriffen wird. Ein Befehl mit direkter Aktualisierung aktualisiert zusätzlich zum spezifizierten Bit acht Bits gleichzeitig (höchstwertige oder niederwertigste acht Bits).

Befehle mit direkter Aktualisierung können nicht für Baugruppen auf Remote-Slave-Baugruppenträgern verwendet werden.

Befehlsvariante	AWL	Funktion	E/A-Aktualisierung
Regulär	LD, AND, OR, LD NOT, AND NOT, OR NOT	Der EIN/AUS-Status des spezifizierten Bits wird von der CPU-Baugruppe im Rahmen der zyklischen Aktualisierung ermittelt und spiegelt sich bei der nächsten Befehlsausführung wider.	Zyklische Aktualisierung
	OUT, OUT NOT	Nach Ausführung des Befehls wird der EIN/AUS-Status des spezifizierten Bits bei der nächsten zyklischen Aktualisierung ausgegeben.	
Ausführung bei steigender Flanke	@LD, @AND, @OR	Der Befehl wird einmal ausgeführt, wenn das spezifizierte Bit von AUS zu EIN wechselt und der EIN-Status über einen Zyklus gehalten wird.	
Ausführung bei fallender Flanke	%LD, %AND, %OR	Der Befehl wird einmal ausgeführt, wenn das spezifizierte Bit von EIN zu AUS wechselt und der AUS-Status über einen Zyklus gehalten wird.	
Direkte Aktualisierung	!LD, !AND, !OR, !LD NOT, !AND NOT, !OR NOT	Die Eingangsdaten für das spezifizierte Bit werden von der CPU ermittelt und der Befehl wird ausgeführt.	Vor Ausführung des Befehls
	!OUT, !OUT NOT	Nach Ausführung des Befehls werden die Daten für das spezifizierte Bit ausgegeben.	Nach Ausführung des Befehls
Ausführung bei steigender Flanke / direkte Aktualisierung	!@LD, !@AND, !@OR	Die Eingangsdaten für das spezifizierte Bit werden von der CPU aktualisiert und der Befehl wird einmal ausgeführt, wenn das Bit von AUS zu EIN wechselt und der EIN-Status über einen Zyklus gehalten wird.	Vor Ausführung des Befehls
Ausführung bei fallender Flanke / direkte Aktualisierung	!%LD, !%AND, !%OR	Die Eingangsdaten für das spezifizierte Bit werden von der CPU aktualisiert und der Befehl wird einmal ausgeführt, wenn das Bit von EIN zu AUS wechselt und der AUS-Status über einen Zyklus gehalten wird.	

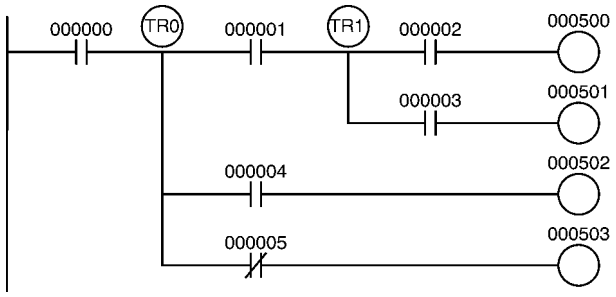
3-3-10 Zeitliches Verhalten von E/A-Befehlen

Die nachfolgende Tabelle zeigt die Unterschiede im zeitlichen Verhalten von Befehlen in einem Programm, dass durch LD und OUT konfiguriert ist.



3-3-11 TR-Bits

TR-Bits werden bei der Programmierung in AWL-Code zur vorübergehenden Beibehaltung des EIN/AUS-Status von Ausführungsbedingungen in einem Programm verwendet. Sie werden bei der direkten Programmierung im Kontaktplanprogramm nicht angezeigt, da die Verwendung automatisch vom Programmiertool ausgeführt wird. Die nachfolgende Abbildung zeigt eine einfache Anwendung, bei der zwei TR-Bits verwendet werden.

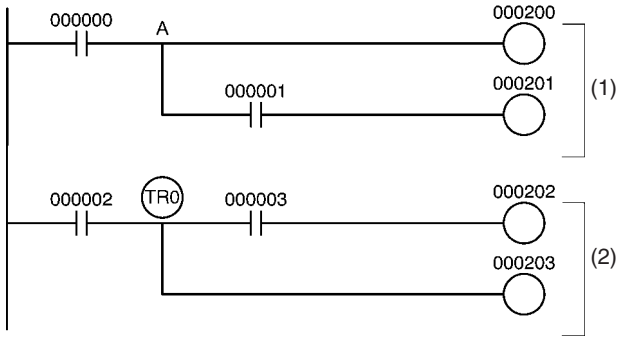


Adresse	Befehl	Operanden
000000	LD	000000
000001	OUT	TR0
000002	AND	000001
000003	OUT	TR1
000004	AND	000002
000005	OUT	000500
000006	LD	TR1
000007	AND	000003
000008	OUT	000501
000009	LD	TR0
000010	AND	000004
000011	OUT	000502
000012	LD	TR0
000013	AND NOT	000005
000014	OUT	000503

Verwendung von TR0 bis TR15

TR0 bis TR15 werden nur zusammen mit den Befehlen LOAD und OUTPUT verwendet. Es gibt keine Beschränkungen für die Reihenfolge, in der die Bitadressen verwendet werden.

Manchmal kann ein Programm durch Umschreiben so vereinfacht werden, dass TR-Bits nicht erforderlich sind. Die nachfolgende Abbildung zeigt einen Fall, in dem ein TR-Bit unnötig ist und einen, in dem ein TR-Bit erforderlich ist.



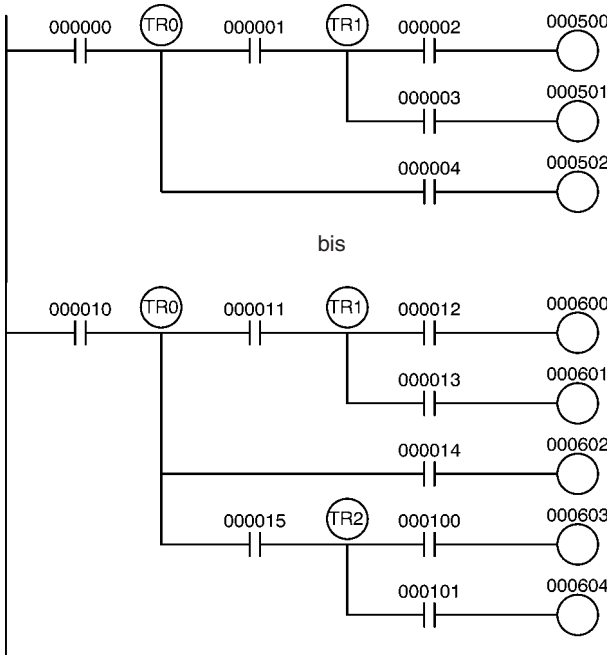
In Befehlsblock (1) ist der EIN/AUS-Status an Punkt A identisch mit dem von Ausgang CIO 00200, deshalb können AND 000001 und OUT 000201 ohne Erfordernis eines TR-Bits kodiert werden. In Befehlsblock (2) ist der Status des Verzweigungspunkts mit dem des Ausgangs CIO 000202 nicht unbedingt gleich, so dass ein TR-Bit verwendet werden muss. In diesem Fall kann die Anzahl von Steps im Programm durch Verwendung von Befehlsblock (1) anstelle von Befehlsblock (2) reduziert werden.

Hinweise zu TR0 bis TR15

TR-Bits werden in Programmen mit vielen Verzweigungen nur zur Erhaltung (OUT TR0 bis TR15) und Wiederherstellung (LD TR0 bis TR15) des EIN/AUS-Status von Verzweigungspunkten verwendet. Sie unterscheiden sich daher von allgemeinen Bits und können nicht zusammen mit den Befehlen AND oder OR sowie mit Befehlen, die NOT beinhalten, verwendet werden.

Ausgangsduplizierung mit TR0 bis TR15

Eine TR-Bitadresse kann nicht innerhalb desselben Blocks in einem Programm mit vielen Ausgangsverzweigungen wiederholt werden, wie in der nachfolgenden Abbildung dargestellt. Sie kann allerdings in einem anderen Block erneut verwendet werden.



3-3-12 NOT: NOT(520)

Funktion

Kehrt die Ausführungsbedingung um.

Kontaktplansymbol



Variationen

Variationen	Kehrt die Ausführungsbedingung in jedem Zyklus um	NOT(520)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Beschreibung

NOT(520) wird zwischen einer Ausführungsbedingung und einem anderen Befehl platziert, um die Ausführungsbedingung umzukehren.

Flags

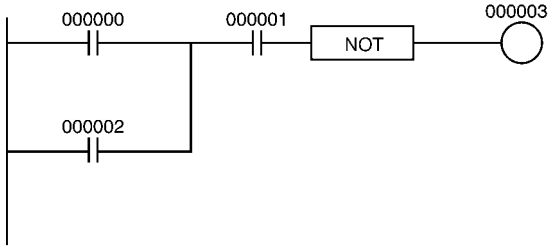
NOT(520) hat keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

NOT(520) ist ein Zusatzbefehl. Er kann also nicht als Befehl auf der rechten Seite verwendet werden. Stellen Sie sicher, dass nach NOT(520) ein Befehl auf der rechten Seite programmiert wird.

Beispiel

Im folgenden Beispiel kehrt NOT(520) die Ausführungsbedingung um.



Die folgende Tabelle zeigt die Funktion dieses Programmabschnitts.

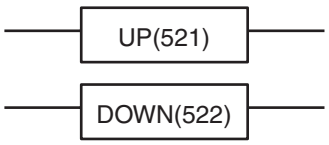
Eingangsbit-Status			Ausgangsbit-Status
CIO 000000	**CIO 000001**	**CIO 000002**	**CIO 000003**
1	1	1	0
1	1	0	0
1	0	1	1
0	1	1	0
1	0	0	1
0	1	0	1
0	0	1	1
0	0	0	1

3-3-13 CONDITION ON/OFF: UP(521) und DOWN(522)

Funktion

UP(521) setzt die Ausführungsbedingung für den nächsten Befehl einen Zyklus lang auf EIN, wenn die empfangene Ausführungsbedingung von AUS zu EIN wechselt. DOWN(522) setzt die Ausführungsbedingung für den nächsten Befehl einen Zyklus lang auf EIN, wenn die empfangene Ausführungsbedingung von EIN zu AUS wechselt.

Kontaktplansymbole



Variationen

Variationen	Erzeugt eine EIN-Bedingung einmalig bei steigender Flanke	UP(521)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Variationen	Erzeugt eine EIN-Bedingung einmalig bei fallender Flanke	UP(522)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Beschreibung

UP(521) wird zwischen einer Ausführungsbedingung und einem anderen Befehl platziert, um die Ausführungsbedingung in eine Bedingung mit steigender Flanke umzuwandeln. UP(521) bewirkt, dass der Verbindungsbefehl nur einmal ausgeführt wird, wenn die Ausführungsbedingung von AUS zu EIN wechselt.

UP(522) wird zwischen einer Ausführungsbedingung und einem anderen Befehl platziert, um die Ausführungsbedingung in eine Bedingung mit fallender Flanke umzuwandeln. DOWN(522) bewirkt, dass der Verbindungsbefehl nur einmal ausgeführt wird, wenn die Ausführungsbedingung von EIN zu AUS wechselt.

Die Befehle DIFU(013) und DIFD(014) können für den selben Zweck verwendet werden, benötigen aber Arbeits-Bits. UP(521) und DOWN(522) vereinfachen die Programmierung durch Reduzierung der benötigten Anzahl an Arbeits-Bits und Programmadressen.

Flags

UP(521) und DOWN(522) haben keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

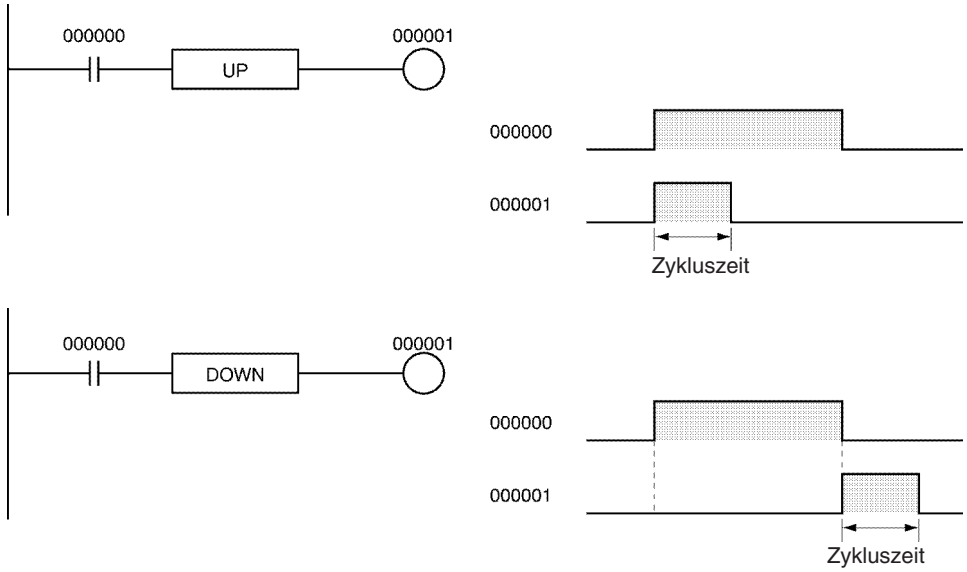
UP(521) und DOWN(522) sind Zwischenbefehle, sie können also nicht als Befehle auf der rechten Seite verwendet werden. Stellen Sie sicher, dass nach NOT(521) oder DOWN(522) ein Befehl auf der rechten Seite programmiert wird.

Die Funktion von UP(521) und DOWN(522) hängt sowohl von der Ausführungsbedingung für den Befehl als auch, bei der Programmierung in einem verriegel-

ten Programmabschnitt, einem übersprungenen Programmabschnitt oder einem Unterprogramm, von der Ausführungsbedingung für den Programmabschnitt ab. Detaillierte Informationen hierzu finden Sie unter *3-5-4 INTERLOCK und INTERLOCK CLEAR: IL(002) und ILC(003), 3-5-6 JUMP und JUMP END: JMP(004) und JME(005)* und *3-20 Interrupt-Steuerungsbefehle*.

Beispiele

Wenn der Status von CIO 000000 im folgenden Beispiel von AUS zu EIN wechselt, wird CIO 000001 nur einen Zyklus lang auf EIN gesetzt.



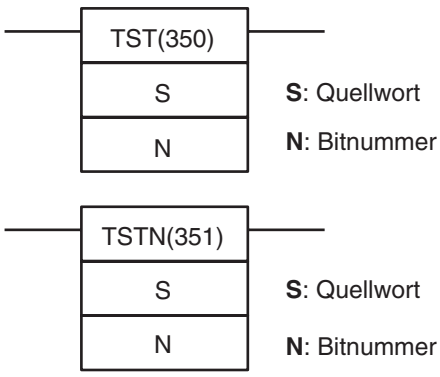
3-3-14 BIT TEST: TST(350) und TSTN(351)

Funktion

LD TST(350), AND TST(350) sowie OR TST(350) werden im Programm wie LD, AND und OR verwendet. Die Ausführungsbedingung ist EIN, wenn das spezifizierte Bit im spezifizierten Wort den Status EIN hat. Entsprechend ist sie AUS, wenn das Bit den Status AUS hat.

LD TSTN(351), AND TSTN(351) sowie OR TSTN(351) werden im Programm wie LD NOT, AND NOT und OR NOT verwendet. Die Ausführungsbedingung ist AUS, wenn das spezifizierte Bit im spezifizierten Wort den Zustand EIN hat. Entsprechend ist sie EIN, wenn das Bit den Zustand AUS hat.

Kontaktplansymbole



Variationen

Variationen	Ausführung bei jedem Zyklus	TST(350)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Variationen	Ausführung bei jedem Zyklus	TSTN(351)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Bitnummer

Die Bit-Nummer muss zwischen 0000 und 000F hexadezimal oder zwischen &0000 und &0015 dezimal liegen. Wenn eine Wortadresse spezifiziert wird, ist nur das niederwertigste Bit (0 bis F hexadezimal) des Wortinhalts gültig.

Operanden-Spezifikationen

Bereich	S	N
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	#0000 bis #000F (binär) oder &0 bis &15
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

LD TST(350), AND TST(350) sowie OR TST(350) können im Programm wie LD, AND und OR verwendet werden. Die Ausführungsbedingung ist EIN, wenn das spezifizierte Bit im spezifizierten Wort den Status EIN hat. Entsprechend ist sie AUS, wenn das Bit den Status AUS hat. Anders als bei LD, AND und OR können in TST(350) Bits in den DM- und EM-Bereichen als Operanden verwendet werden.

LD TSTN(351), AND TSTN(351) sowie OR TSTN(351) können im Programm wie LD NOT, AND NOT und OR NOT verwendet werden. Die Ausführungsbedingung ist AUS, wenn das spezifizierte Bit im spezifizierten Wort den Zustand EIN hat. Entsprechend ist sie EIN, wenn das Bit den Zustand AUS hat. Anders als bei LD NOT, AND NOT und OR NOT können in TSTN(351) Bits in den DM- und EM-Bereichen als Operanden verwendet werden.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

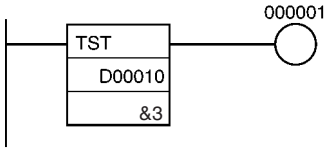
Vorsichtsmaßnahmen

TST(350) and TSTN(351) sind Zwischenbefehle, sie können also nicht als Befehle auf der rechten Seite verwendet werden. Stellen Sie sicher, dass nach TST(350) oder TSTN(351) ein Befehl auf der rechten Seite programmiert wird.

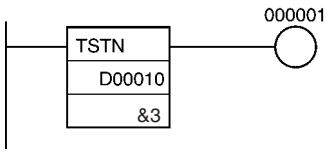
Beispiele

LD TST(350) und LD TSTN(351)

Im folgenden Beispiel wird CIO 000001 auf EIN gesetzt, wenn Bit 3 von D00010 auf EIN gesetzt ist.

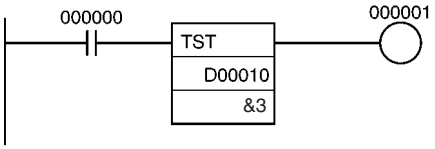


Im folgenden Beispiel wird CIO 000001 auf EIN gesetzt, wenn Bit 3 von D00010 auf AUS gesetzt ist.

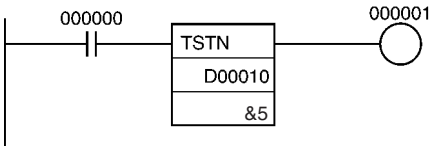


AND TST(350) und AND TSTN(351)

Im folgenden Beispiel wird CIO 000001 auf EIN gesetzt, wenn CIO 000000 und Bit 3 von D00010 auf EIN gesetzt sind.

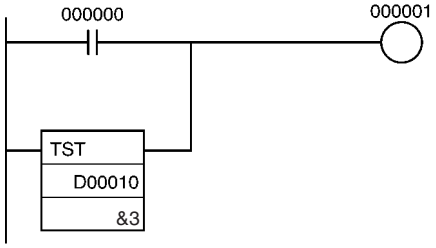


Im folgenden Beispiel wird CIO 000001 auf EIN gesetzt, wenn CIO 000000 auf EIN gesetzt ist und Bit 5 von D00010 auf AUS gesetzt ist.

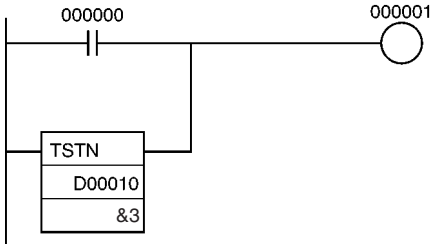


OR TST(350) und OR TSTN(351)

Im folgenden Beispiel wird CIO 000001 auf EIN gesetzt, wenn CIO 000000 oder Bit 3 von D00010 auf EIN gesetzt ist.



Im folgenden Beispiel wird CIO 000001 auf EIN gesetzt, wenn CIO 000000 auf EIN gesetzt ist oder Bit 3 von D00010 auf AUS gesetzt ist.



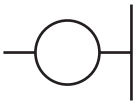
3-4 Sequenzausgabebefehle

3-4-1 OUTPUT: OUT

Funktion

Gibt das Ergebnis (Ausführungsbedingung) der logischen Verarbeitung in das spezifizierte Bit aus.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	OUT
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!OUT

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Operanden-Spezifikationen

Bereich	OUT Bit-Operand
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A44800 bis A95915
Zeitgeberbereich	---
Zählerbereich	---

Bereich	OUT Bit-Operand
TR-Bereich	TR0 bis TR15
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Falls keine Spezifikation für eine direkte Aktualisierung vorhanden ist, wird der Status der Ausführungsbedingung (Stromfluss) in das spezifizierte Bit des E/A-Speichers geschrieben. Falls eine Spezifikation für eine direkte Aktualisierung vorhanden ist, wird der Status der Ausführungsbedingung (Stromfluss) nicht nur in das Ausgangs-Bit im E/A-Speicher geschrieben, sondern auch zusätzlich für die Ausgangsklemme der Ausgangsbaugruppe gesetzt.

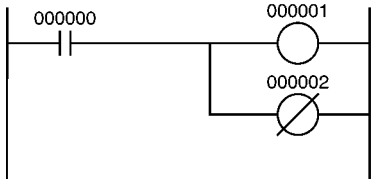
Flags

Dieser Befehl hat keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

Direkte Aktualisierung (!) kann für OUT und OUT NOT spezifiziert werden. Ein Befehl mit direkter Aktualisierung aktualisiert den Status der Ausgangsklemmen der Ausgangsbaugruppe (jedoch nicht bei Ausgangbaugruppen auf Remote-Slave-Baugruppenträgern oder bei C200H Multipunkt-Eingangsbaugruppen der Gruppe 2) unmittelbar nach Ausführung des Befehls. Gleichzeitig wird der Status der Ausführungsbedingung (Stromfluss) in das spezifizierte Ausgangs-Bit im E/A-Speicher geschrieben.

Beispiel



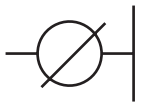
Befehl	Operand
LD	000000
OUT	000001
OUT NOT	000002

3-4-2 OUTPUT NOT: OUT NOT

Funktion

Kehrt das Ergebnis (Ausführungsbedingung) der logischen Verarbeitung um und gibt es in das spezifizierte Bit aus.

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	OUT NOT
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!OUT NOT

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Operanden-Spezifikationen

Bereich	OUT-Bit-Operand
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A44800 bis A95915
Zeitgeberbereich	---
Zählerbereich	---
TR-Bereich	TR0 bis TR15
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

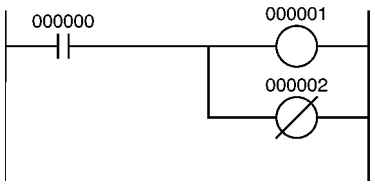
Beschreibung

Falls keine Spezifikation für eine direkte Aktualisierung vorhanden ist, wird der Status der Ausführungsbedingung (Stromfluss) invertiert und in ein spezifiziertes Bit im E/A-Speicher geschrieben. Falls eine Spezifikation für eine direkte Aktualisierung vorhanden ist, wird der Status der Ausführungsbedingung (Stromfluss) invertiert und nicht nur in das Ausgangs-Bit im E/A-Speicher geschrieben, sondern auch zusätzlich für die Ausgangsklemme der Ausgangsbaugruppe gesetzt.

Flags

Dieser Befehl hat keinen Einfluss auf Flags.

Beispiel



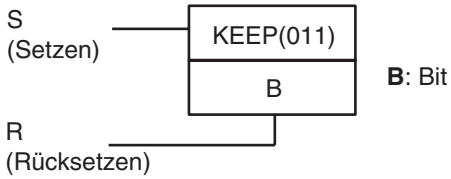
Befehl	Operand
LD	000000
OUT	000001
OUT NOT	000002

3-4-3 KEEP: KEEP(011)

Funktion

Arbeitet als Halterelais.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	KEEP(011)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!KEEP(011)

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Anwendbarkeit in Programmbereichen

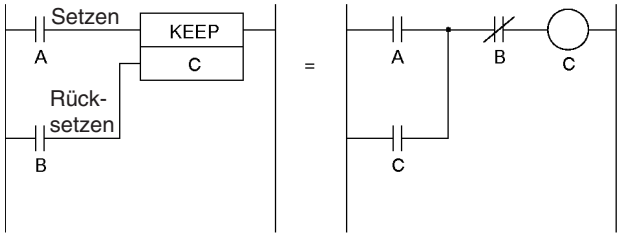
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

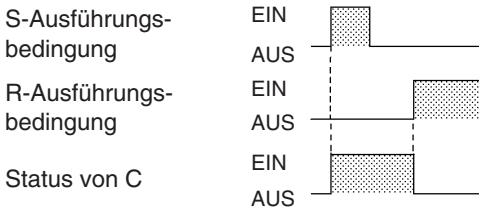
Operanden-Spezifikationen

Bereich	B
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A44800 bis A95915
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

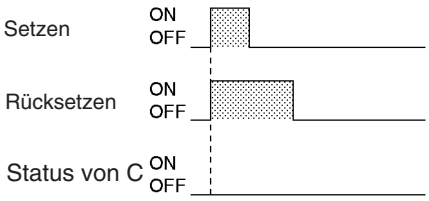
Beschreibung

Wenn S auf EIN gesetzt wird, wird das angegebene Bit auf EIN gesetzt und bleibt bis zur Rücksetzung in diesem Status, unabhängig davon, ob S auf EIN gesetzt bleibt oder auf AUS gesetzt wird. Wenn R auf EIN gesetzt wird, wird das angegebene Bit auf AUS gesetzt. Die Beziehung zwischen Ausführungsbedingung und Bit-Staus von KEEP(011) wird nachstehend gezeigt.

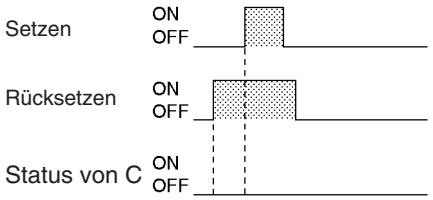




Wenn S und R gleichzeitig auf EIN gesetzt sind, hat der Rücksetzeingang Vorrang.

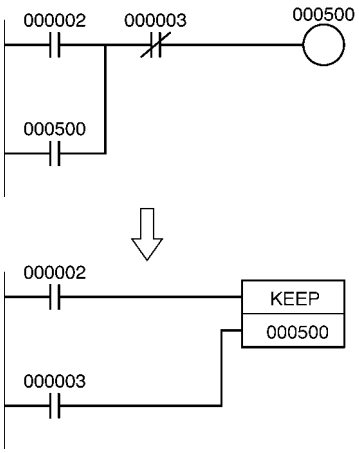


Der Setzen-Eingang (S) kann nicht empfangen werden, solange R auf EIN gesetzt ist.

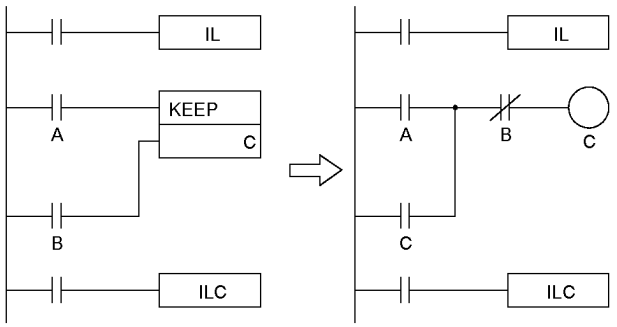


Für den Befehl KEEP(011) gibt es eine Vatiante mit direkter Aktualisierung (!KEEP(011)). Wenn für B in einem !KEEP(011)-Befehl ein externes Ausgangs-Bit spezifiziert wurde, werden alle Änderungen von B bei der Ausführung von !KEEP(011) aktualisiert und unmittelbar im Ausgabe-Bit reflektiert. (Änderungen werden nicht unmittelbar reflektiert, wenn das Bit einer Multipoint-E/A-Baugruppen der Gruppe 2, einer Multipoint-Spezial-E/A-Baugruppe oder einer Baugruppe zugewiesen ist, die auf einem Remote-Slave-Baugruppenträger montiert ist.)

KEEP(011) funktioniert wie das selbsthaltende Bit, aber ein mit KEEP(011) programmiertes selbsthaltendes Bit erfordert einen Befehl weniger.



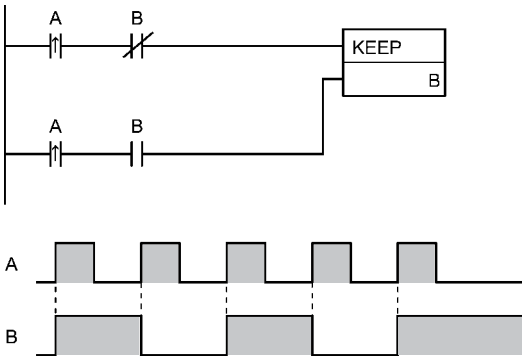
Mit KEEP(011) programmierte selbsthaltende Bits behalten ihren Status sogar in einem verriegelten Programmabschnitt bei, es sei denn, das selbsthaltende Bit wurde ohne KEEP(011) programmiert.



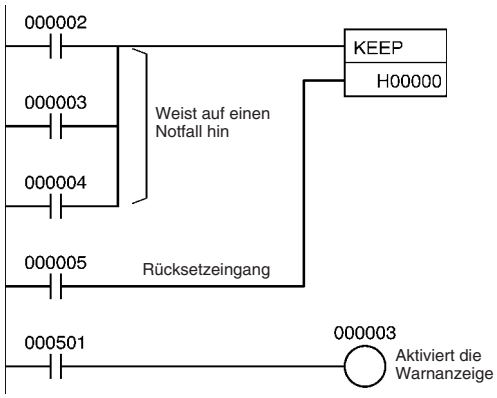
Ausgangs-Bit C behält seinen vorherigen Status bei einer Verriegelung bei.

Ausgangs-Bit C wird bei einer Verriegelung auf AUS gesetzt.

KEEP(011) kann, wie nachstehend dargestellt, zur Erzeugung von Flip-Flops verwendet werden.



Bei Verwendung eines remanenten Arbeitsbits für B bleibt der Bit-Status sogar bei einer Unterbrechung der Spannungsversorgung erhalten. KEEP(011) kann somit zur Programmierung von Bits verwendet werden, die nach dem Neustart der SPS aufgrund einer Unterbrechung der Versorgungsspannung ihren Status beibehalten. Nachstehend ist ein Beispiel gezeigt, wie auf diese Weise im Notfall nach einer Systemabschaltung eine Warnanzeige erzeugt werden kann.



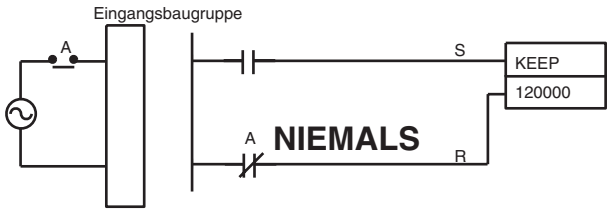
Der Status von E/A-Bereichs-Bits kann im Fall einer Unterbrechung der Versorgungsspannung durch ein im SPS-Setup auf EIN gesetztes E/A-Haltebit aufrecht erhalten werden. In diesem Fall behalten die in KEEP(011) verwendeten E/A-Bereichs-Bits (wie remanente Arbeitsbits) ihren Status nach einem Neustart der SPS aufgrund einer Unterbrechung der Versorgungsspannung bei. Stellen Sie sicher, dass die SPS nach Änderung des SPS-Setup neu gestartet wird. Anderenfalls werden die neuen Einstellungen nicht verwendet.

Flags

Flags sind von KEEP(011) nicht betroffen.

Vorsichtsmaßnahmen

Verwenden Sie keinesfalls ein Eingangs-Bit in einer Öffner-Bedingung zur Rücksetzung (R) für KEEP(011), wenn das Eingabegerät eine AC-Spannungsversorgung besitzt. Die Verzögerung beim Ausschalten der DC-Spannungsversorgung der SPS (relativ zur AC-Spannungsversorgung des Eingangsgeräts) kann eine Rücksetzung des Operand-Bits von KEEP(011) bewirken. Diese Situation wird nachstehend gezeigt.



Die Operanden für KEEP(011) werden bei Kontaktplanprogrammen und AWL-Codes in einer anderen Reihenfolge eingegeben.

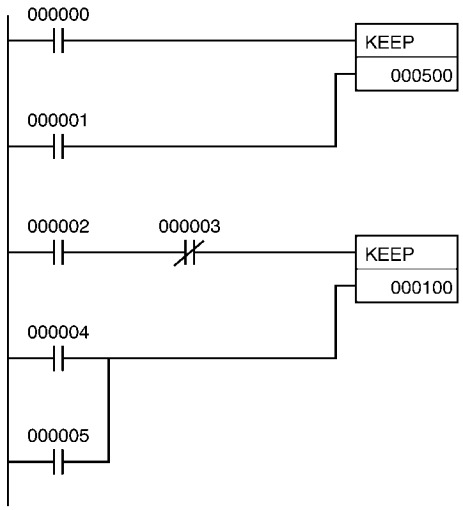
Kontaktplanprogramm-Reihenfolge:
Setz-Eingang → KEEP(011) → Rücksetz-Eingang

AWL-Reihenfolge: Setz-Eingang → Rücksetz-Eingang → KEEP(011)

Beispiel

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, wird CIO 00500 auf EIN gesetzt. CIO 00500 bleibt auf EIN gesetzt, bis CIO 000001 auf EIN gesetzt wird.

Wenn CIO 000002 auf EIN gesetzt wird und CIO 000003 im folgenden Beispiel auf AUS gesetzt wird, wird CIO 00100 auf EIN gesetzt. CIO 00100 bleibt auf EIN gesetzt, bis CIO 000004 oder CIO 000005 auf EIN gesetzt wird.



Kodierung

Adresse	Befehl	Operand
000100	LD	000000
000101	LD	000001
000102	KEEP(011)	000500
000103	LD	000002
000104	AND NOT	000003
000105	LD	000004
000106	OR	000005
000107	KEEP(011)	000100

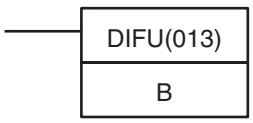
Hinweis KEEP(011) wird in Kontaktplan- oder AWL-Form in unterschiedlichen Reihenfolgen eingegeben. In der Kontaktplan-Form wird der Setzeingang, KEEP(011) und dann der Rücksetzeingang eingegeben. In der AWL-Form wird Setzeingang, der Rücksetzeingang und dann KEEP(011) eingegeben.

3-4-4 DIFFERENTIATE UP/DOWN: DIFU(013) und DIFD(014)

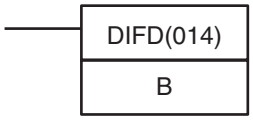
Funktion

DIFU(013) setzt das angegebene Bit für einen Zyklus auf EIN, wenn die Ausführungsbedingung von AUS zu EIN wechselt (steigende Flanke).
DIFD(014) setzt das angegebene Bit für einen Zyklus auf EIN, wenn die Ausführungsbedingung von EIN zu AUS wechselt (fallende Flanke).

Kontaktplansymbole



B: Bit



B: Bit

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	Nicht unterstützt
	Ausführung einmal bei steigender Flanke	DIFU(013)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!DIFU(013)

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	Nicht unterstützt
	Ausführung einmal bei steigender Flanke	DIFD(014)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!DIFD(014)

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Anwendbarkeit in Programmbereichen

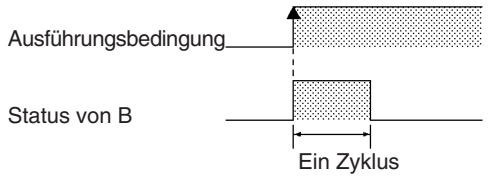
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Operanden-Spezifikationen

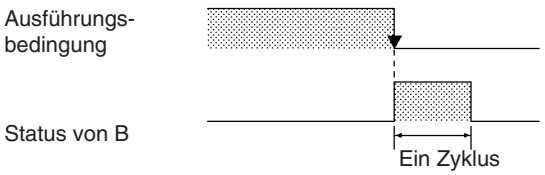
Bereich	B
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A44800 bis A95915
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Wenn die Ausführungsbedingung von AUS auf EIN wechselt, setzt DIFU(013) B auf EIN. Wenn DIFU(013) im nächsten Zyklus erreicht wird, wird B auf AUS gesetzt.



Wenn die Ausführungsbedingung von EIN auf AUS wechselt, setzt DIFD(014) B auf EIN. Wenn DIFD(014) im nächsten Zyklus erreicht wird, wird B auf AUS gesetzt.



DIFU(013) und DIFD(014) haben Varianten für direkte Aktualisierung (!DIFU(013) und !DIFD(014)). Wenn für B in einem dieser Befehle ein externes Ausgangs-Bit spezifiziert wurde, werden alle Änderungen an B bei der Ausführung des Befehls aktualisiert und sofort im Ausgabe-Bit reflektiert.

(Änderungen werden nicht unmittelbar reflektiert, wenn das Bit einer Multipoint-E/A-Baugruppen der Gruppe 2, einer Multipoint-Spezial-E/A-Baugruppe oder einer Baugruppe zugewiesen ist, die auf einem Remote-Slave-Baugruppenträger montiert ist.)

UP(521) und DOWN(522) können zur Ausführung eines Befehls in nur einem Zyklus verwendet werden, wenn die Ausführungsbedingung von AUS → EIN oder EIN → AUS wechselt. Ausführliche Informationen hierzu finden Sie unter *3-3-13 CONDITION ON/OFF: UP(521) und DOWN(522)*.

Flags

DIFU(013) und DIFD(014) haben keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

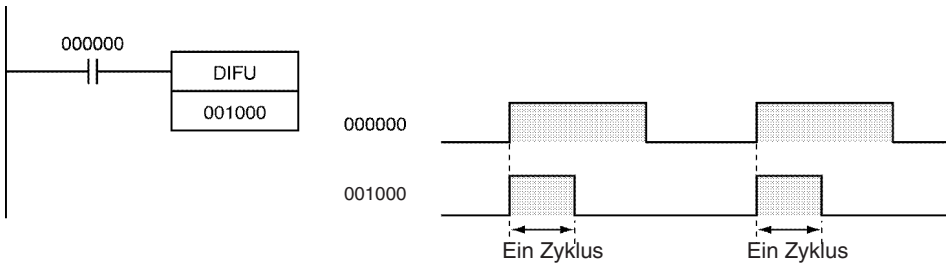
Die Funktion von DIFU(013) und DIFD(014) hängt sowohl von der Ausführungsbedingung für den Befehl selbst als auch, bei der Programmierung in einem verriegelten Programmabschnitt, einem übersprungenen Programmabschnitt oder einem Unterprogramm, von der Ausführungsbedingung für den Programmabschnitt ab. Weitere Informationen hierzu finden Sie unter *3-5-4 INTERLOCK und INTERLOCK CLEAR: IL(002) und ILC(003), 3-5-6 JUMP und JUMP END: JMP(004) und JME(005)* und *3-20 Interrupt-Steuerungsbefehle*.

Wenn DIFU(013) in einer FOR-NEXT-Schleife verwendet wird und sich die Schleife in einem Zyklus wiederholt, ist das gesteuerte Bit innerhalb dieser Schleife immer auf EIN oder immer auf AUS gesetzt.

Beispiele

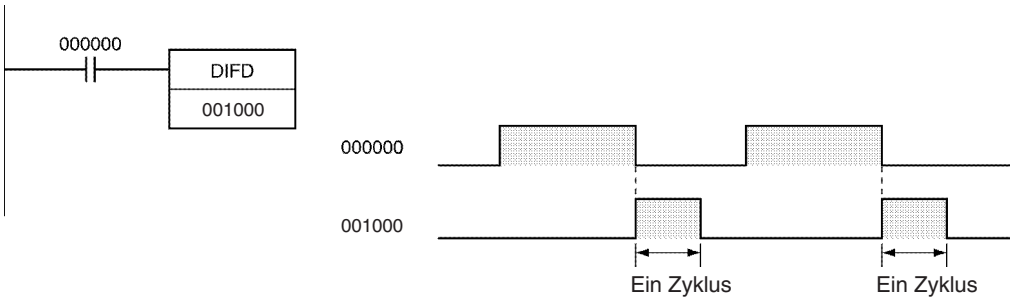
Funktion von DIFU(013)

Wenn CIO 000000 im folgenden Beispiel von AUS zu EIN wechselt, wird CIO 001000 einen Zyklus lang auf EIN gesetzt.



Funktion von DIFD(014)

Wenn CIO 000000 im folgenden Beispiel von EIN zu AUS wechselt, wird CIO 001000 einen Zyklus lang auf EIN gesetzt.

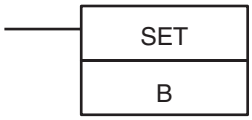


3-4-5 SET and RESET: SET und RSET

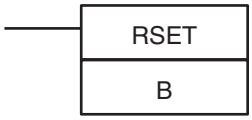
Funktion

SET setzt das Operanden-Bit auf EIN, wenn die Ausführungsbedingung den Zustand EIN hat.
RSET setzt das Operanden-Bit auf AUS, wenn die Ausführungsbedingung den Zustand EIN hat.

Kontaktplansymbole



B: Bit



B: Bit

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SET
	Ausführung einmal bei steigender Flanke	@SET
	Ausführung einmal bei fallender Flanke	%SET
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!SET
Kombinierte Variationen	**Einmalige Ausführung und unmittelbare Bit-Aktualisierung bei steigender Flanke (siehe Hinweis)**	!@SET
	Einmalige Ausführung und unmittelbare Bit-Aktualisierung bei fallender Flanke (siehe Hinweis)	!%SET

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	RSET
	Ausführung einmal bei steigender Flanke	@RSET
	Ausführung einmal bei fallender Flanke	%RSET
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!RSET
Kombinierte Variationen	**Einmalige Ausführung mit direkter Aktualisierung bei steigender Flanke (siehe Hinweis)**	!@RSET
	Einmalige Ausführung mit direkter Aktualisierung bei fallender Flanke (siehe Hinweis)	!%RSET

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

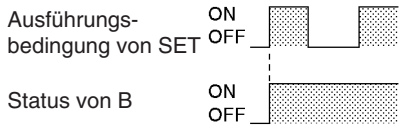
Operanden-Spezifikationen

Bereich	B
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A44800 bis A95915
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---

Bereich	B
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

SET setzt das Operand-Bit auf EIN, wenn die Ausführungsbedingung auf EIN gesetzt ist und hat keinen Einfluss auf den Status des Operand-Bits, wenn die Ausführungsbedingung auf AUS gesetzt ist. Verwenden Sie RSET, um ein Bit auf AUS zu setzen, dass mit Hilfe von SET auf EIN gesetzt wurde.



RSET setzt das Operand-Bit auf AUS, wenn die Ausführungsbedingung auf EIN gesetzt ist und hat keinen Einfluss auf den Status des Operand-Bits, wenn die Ausführungsbedingung auf AUS gesetzt ist. Verwenden Sie SET, um ein Bit auf EIN zu setzen, dass mit Hilfe von RSET auf AUS gesetzt wurde.



SET und RSET haben Varianten mit direkter Aktualisierung (!SET und !RSET). Wenn für B in einem dieser Befehle ein externes Ausgangs-Bit spezifiziert wurde, werden alle Änderungen an B bei der Ausführung des Befehls aktualisiert und sofort im Ausgabe-Bit reflektiert. (Änderungen werden nicht unmittelbar reflektiert, wenn das Bit einer Multipoint-E/A-Baugruppen der Gruppe 2, einer Multipoint-Spezial-E/A-Baugruppe oder einer Baugruppe zugewiesen ist, die auf einem Remote-Slave-Baugruppenträger montiert ist.)

Der Setzeingang und der Rücksetzeingang für einen KEEP(011)-Befehl müssen mit dem Befehl programmiert werden, die Befehle SET und RSET können jedoch vollständig unabhängig programmiert werden. Außerdem kann dasselbe Bit als Operand in einer beliebigen Anzahl von SET- oder RSET-Befehle verwendet werden.

Flags

Die Befehle SET und RSET haben keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

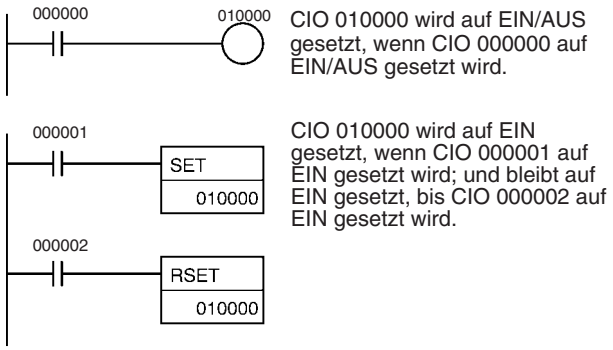
SET und RSET können nicht zum Setzen und Rücksetzen von Zeitgebern und Zählern verwendet werden.

Wenn SET oder RSET zwischen IL(002) und ILC(003) oder JMP(004) und JME(005) programmiert wird, ändert sich der Status des spezifizierten Bits bei einem verriegelten oder übersprungenen Programmabschnitt nicht.

Beispiel

Unterschiede zwischen OUT/OUT NOT und SET/RSET

Die Funktion von SET unterscheidet sich von der Funktion von OUT, da der Befehl OUT das Operand-Bit auf AUS setzt, wenn die Ausführungsbedingung auf AUS gesetzt ist. Außerdem unterscheidet sich RSET von OUT NOT, da OUT NOT das Operand-Bit auf EIN setzt, wenn dessen Ausführungsbedingung den Zustand AUS hat.

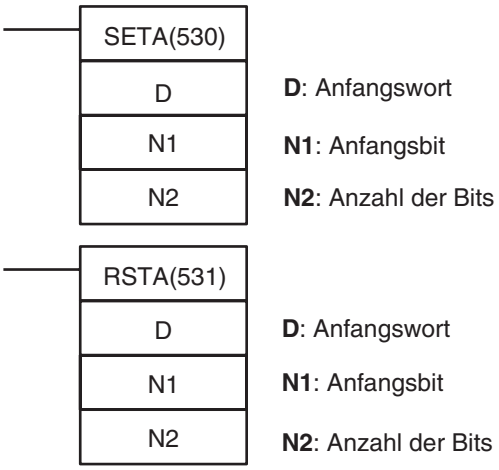


3-4-6 MULTIPLE BIT SET/RESET: SETA(530)/RSTA(531)

Funktion

SETA(530) setzt die spezifizierte Anzahl von aufeinander folgenden Bits auf EIN. RSTA(531) setzt die spezifizierte Anzahl von aufeinander folgenden Bits auf AUS.

Kontaktplansymbole



Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SETA(530)
	Ausführung einmal bei steigender Flanke	@SETA(530)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	RSTA(531)
	Ausführung einmal bei steigender Flanke	@RSTA(531)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

D: Anfangswort

Spezifiziert das erste Wort, in dem Bits auf EIN oder AUS gesetzt werden.

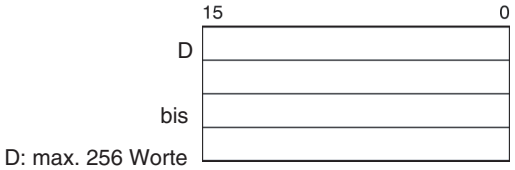
N1: Anfangsbit

Spezifiziert das erste Bit, das auf EIN oder AUS gesetzt wird. N1 muss #0000 bis #000F sein (&0 bis &15).

N2: Anzahl der Bits

Spezifiziert die Anzahl von Bits, die auf EIN oder AUS gesetzt werden. N2 muss #0000 bis #FFFF sein (&0 bis &65535).

Hinweis Die auf EIN oder AUS gesetzten Bits müssen im gleichen Datenbereich liegen. (Der Wortebereich erstreckt sich grob von D bis D+N2÷16).



Operanden-Spezifikationen

Bereich	D	N1	N2
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A448 bis A959	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0000 bis #000F (binär) oder &0 bis &15	#0000 bis #FFFF (binär) oder &0 bis &65535
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

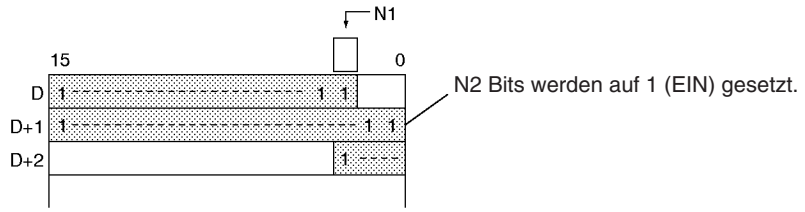
Beschreibung

Die Funktionen von SETA(530) und RSTA(531) werden nachfolgend separat beschrieben.

Funktion von SETA(530)

SETA(530) setzt N2 Bits auf EIN, beginnend mit Bit N1 von D und weiter mit den linken (höherwertigen) Bits. Alle anderen Bits bleiben unverändert. (Wenn N2 auf 0 gesetzt ist, erfolgen keine Änderungen.)

Von SETA(530) auf EIN gesetzte Bits können auch durch andere Befehle (nicht nur durch RSTA(531)) auf AUS gesetzt werden.

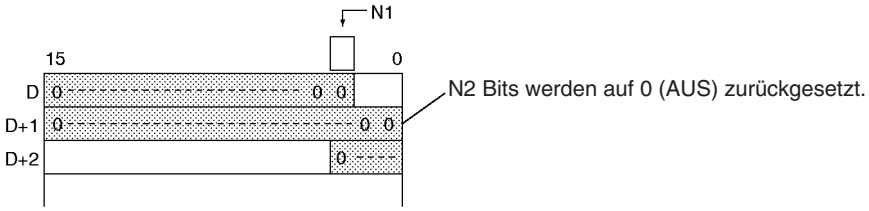


Mit SETA(530) können Bits in Datenbereichen auf EIN gesetzt werden, auf die normalerweise nur wortweise zugegriffen werden kann, wie z.B. die DM- und EM-Bereiche.

Funktion von RSTA(531)

RSTA(531) setzt N2 Bits auf AUS, beginnend mit Bit N1 von D und weiter mit den linken (höherwertigen) Bits. Alle anderen Bits bleiben unverändert. (Wenn N2 auf 0 gesetzt ist, erfolgen keine Änderungen.)

Durch RSTA(531) auf AUS gesetzte Bits können auch durch andere Befehle (nicht nur durch SETA(530)) auf EIN gesetzt werden.



Mit RSTA(531) können Bits in Datenbereichen auf AUS gesetzt werden, auf die normalerweise nur wortweise zugegriffen werden kann, wie z.B. die DM- und EM-Bereiche.

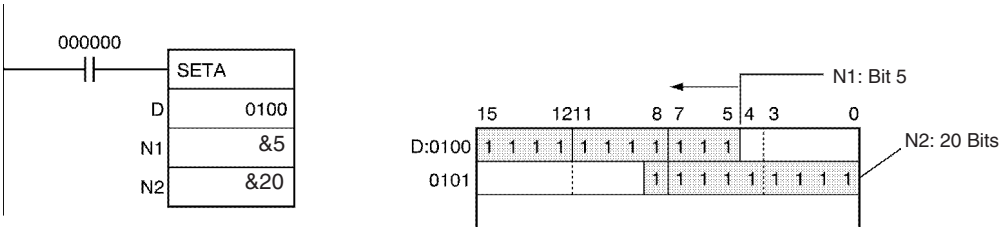
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N1 nicht innerhalb des spezifizierten Bereich von 0000 bis 000F liegt. In allen anderen Fällen AUS.

Beispiele

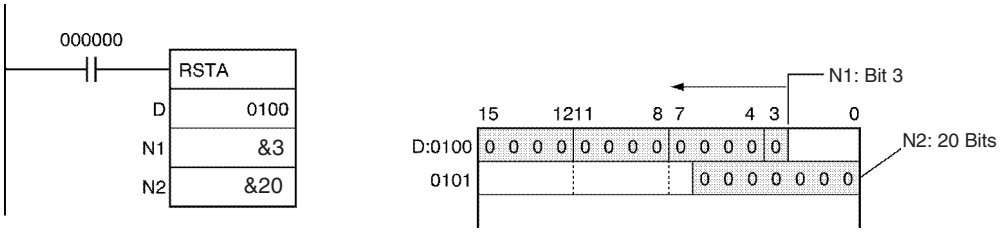
Bespiel für SETA(530)

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, werden die 20 Bits (0014 hexadezimal), beginnend mit Bit 5 von CIO 0100, auf EIN gesetzt.



Beispiel für RSTA(531)

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, werden die 20 Bits (0014 hexadezimal), beginnend mit Bit 3 von CIO 0100, auf AUS gesetzt.



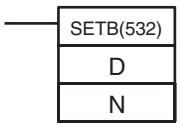
3-4-7 SINGLE BIT SET/RESET: SETB(532)/RSTB(533)

Funktion

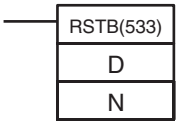
SETB(532) setzt das spezifizierte Bit auf EIN.
RSTB(533) setzt das spezifizierte Bit auf AUS.

Diese Befehle werden nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbole



D: Wortadresse
N: Bitnummer



D: Wortadresse
N: Bitnummer

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SETB(532)
	Ausführung einmal bei steigender Flanke	@SETB(532)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!SETB(532)
Kombinierte Variationen	**Einmalige Ausführung und unmittelbare Bit-Aktualisierung bei steigender Flanke (siehe Hinweis).**	!@SETB(532)
	Einmalige Ausführung und unmittelbare Bit-Aktualisierung bei fallender Flanke	Nicht unterstützt

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	RSTB(533)
	Ausführung einmal bei steigender Flanke	@RSTB(533)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!RSTB(533)
Kombinierte Variationen	**Einmalige Ausführung und unmittelbare Bit-Aktualisierung bei steigender Flanke (siehe Hinweis).**	!@RSTB(533)
	Einmalige Ausführung und unmittelbare Bit-Aktualisierung bei fallender Flanke	Nicht unterstützt

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

D: Wort-Adresse

Spezifiziert das Wort, in dem Bits auf EIN oder AUS gesetzt werden.

N: Anfangsbit

Spezifiziert das Bit, das auf EIN oder AUS gesetzt wird. N muss #0000 bis #000F sein (&0 bis &15).

Operanden-Spezifikationen

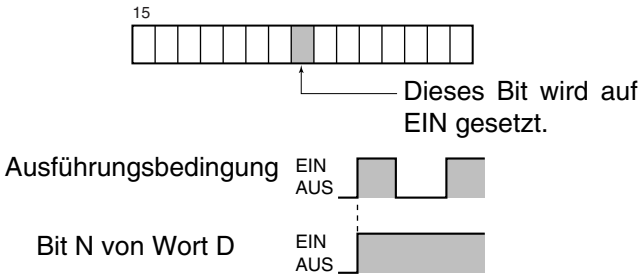
Bereich	D	N
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A448 bis A959	A000 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	#0000 bis #000F (binär) oder &0 bis &15
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Die Funktionen von SETB(532) und RSTB(533) werden nachfolgend separat beschrieben.

Funktion von SETB(532)

SETB(532) setzt Bit N von Wort D auf EIN, wenn die Ausführungsbedingung auf EIN gesetzt ist. Der Status des Bits wird nicht verändert, wenn die Ausführungsbedingung auf AUS gesetzt ist. Anders als SET kann SETB(532) ein Bit im DM- oder EM-Bereich auf EIN setzen.

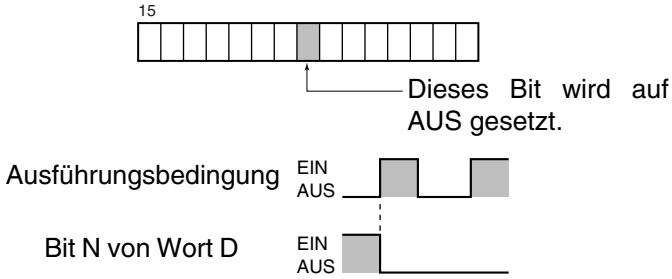


Durch SETB(532) auf EIN gesetzte Bits können auch durch andere Befehle (nicht nur durch RSTB(533)) auf AUS gesetzt werden.

SETB(532) wird nur unterstützt durch CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M.

Funktion von RSTB(533)

RSTB(533) setzt Bit N von Wort D auf AUS, wenn die Ausführungsbedingung auf EIN gesetzt ist. Der Status des Bits wird nicht verändert, wenn die Ausführungsbedingung auf AUS gesetzt ist. (Verwenden Sie SETB(532), um das Bit auf EIN zu setzen.) Anders als RST kann RSTB(533) ein Bit im DM- oder EM-Bereich auf AUS setzen.



Durch RSTB(533) auf AUS gesetzte Bits können auch durch andere Befehle (nicht nur durch SETB(532)) auf EIN gesetzt werden.

RSTB(533) wird nur durch CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M unterstützt.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N nicht im spezifizierten Bereich von 0000 bis 000F (&0 bis &15) liegt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

SETB(532) und RSTB(533) können keine Zeitgeber und Zähler setzen oder zurücksetzen.

Wenn SETB(532) oder RSTB(533) zwischen IL(002) und ILC(003) oder JMP(004) und JME(005) programmiert wird, ändert sich der Status des spezifizierten Bits bei einem verriegelten oder übersprungenen Programmabschnitt nicht (d.h. wenn die Verriegelungsbedingung oder Sprungbedingung auf AUS gesetzt ist).

SETB(532) und RSTB(533) haben Varianten für direkte Aktualisierung (!SETB(532) und !RSTB(533)). Wenn ein externes Ausgangs-Bit in einem dieser Befehle spezifiziert wurde, werden alle Änderungen des spezifizierten Bits bei Ausführung des Befehls aktualisiert und sofort im Ausgabe-Bit reflektiert. (Änderungen werden nicht unmittelbar reflektiert, wenn das Bit einer Multipoint-E/A-Baugruppen der Gruppe 2, einer Multipoint-Spezial-E/A-Baugruppe oder einer Baugruppe zugewiesen ist, die auf einem Remote-Slave-Baugruppenträger montiert ist.)

Unterschiede zwischen SET/RSET und SETB(532)/RSTB(533)

Die Befehle SET und RSET funktionieren etwas anders als SETB(532) und RSTB(533).

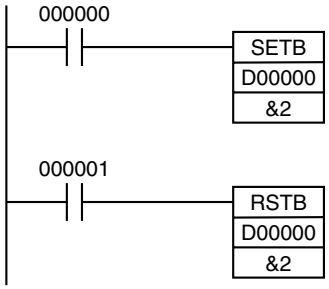
1. Die Befehle funktionieren auf die selbe Weise, wenn sich das spezifizierte Bit im CIO-, W-, H- oder A-Bereich befindet.
2. Im Gegensatz zu SET und RSET können die Befehle SETB(532) und RSTB(533) die Bits in DM- und EM-Bereichen steuern.

Unterschiede zwischen den Befehlen OUTB(532) und SETB(532)/ RSTB(533)

Der Befehl OUTB(534) funktioniert etwas anders als die Befehle SETB(532) und RSTB(533).

1. Die Befehle SETB(532) und RSTB(533) ändern den Status des spezifizierten Bits nur dann, wenn ihre Ausführungsbedingung auf EIN gesetzt ist. Diese Befehle haben keinen Einfluss auf den Status des spezifizierten Bits, wenn ihre Ausführungsbedingung auf AUS gesetzt ist.

2. Der Befehl OUTB(534) setzt das spezifizierte Bit auf EIN, wenn seine Ausführungsbedingung auf EIN gesetzt ist und setzt es auf AUS, wenn die Ausführungsbedingung auf AUS gesetzt ist.
3. Der Setzeingang und der Rücksetzeingang für einen KEEP(011)-Befehl müssen mit dem Befehl programmiert werden, die Befehle SETB(532) und RSTB(533) können jedoch vollständig unabhängig programmiert werden. Außerdem kann dasselbe Bit als Operand in einer beliebigen Anzahl von SETB(532)- und RSTB(533)-Befehlen verwendet werden.



Bit 02 von D00000 wird auf EIN gesetzt, wenn CIO 000000 den Zustand EIN hat.

Bit 02 von D00000 wird auf AUS gesetzt, wenn CIO 000001 den Zustand EIN hat.

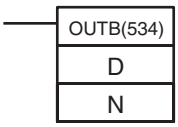
3-4-8 SINGLE BIT OUTPUT: OUTB(534)

Funktion

OUTB(534) gibt den Status der Ausführungsbedingung des Befehls in das spezifizierte Bit aus. OUTB(534) kann im Gegensatz zu OUT ein Bit im DM- oder EM-Bereich steuern.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbole



D: Wortadresse
N: Bitnummer

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	OUTB(534)
	Ausführung einmal bei steigender Flanke	@OUTB(534)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!OUTB(534)

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Operanden

D: Wort-Adresse

Spezifiziert das Wort, welches das zu steuernde Bit enthält.

N: Anfangsbit

Spezifiziert das zu steuernde Bit. N muss #0000 bis #000F sein (&0 bis &15).

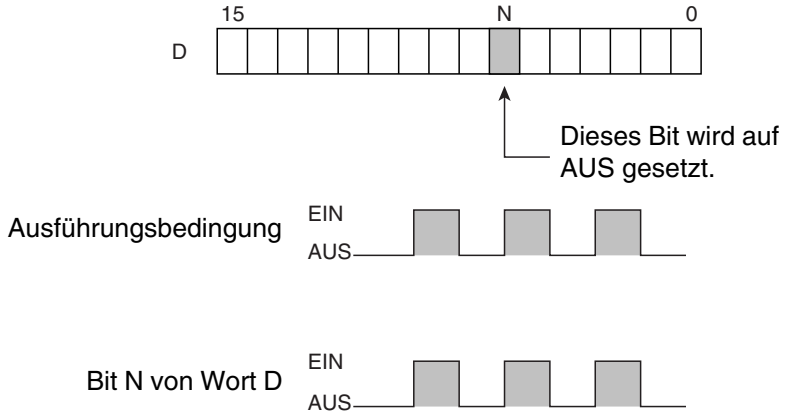
Operanden-Spezifikationen

Bereich	D	N
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A448 bis A959	A000 bis A959

Bereich	D	N
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	#0000 bis #000F (binär) oder &0 bis &15
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Wenn die Ausführungsbedingung auf EIN gesetzt ist, setzt OUTB(534) Bit N von Wort D auf EIN. Wenn die Ausführungsbedingung auf AUS gesetzt ist, setzt OUTB(534) Bit N von Wort D auf AUS.



Falls die Variation mit der direkten Aktualisierung nicht verwendet wird, wird der Status der Ausführungsbedingung (Stromfluss) in das spezifizierte Bit im E/A-Speicher geschrieben. Falls die Variante mit direkter Aktualisierung verwendet wird, wird der Status der Ausführungsbedingung (Stromfluss) für die Ausgangsklemme der Ausgangsbaugruppe sowie in das Ausgangs-Bit im E/A-Speicher geschrieben.

OUTB(534) wird nur durch CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M unterstützt.

Flags

Dieser Befehl hat keinen Einfluss auf Flags.

Vorsichtsmaßnahmen

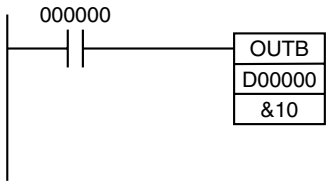
Direkte Aktualisierung (!OUTB(534)) kann spezifiziert werden. Ein Befehl mit direkter Aktualisierung aktualisiert den Status der Ausgangsklemmen unmittelbar nach Ausführung des Befehls für ein Ausgangsbit, das einer Ausgangs-

baugruppe (jedoch nicht bei Ausgangbaugruppen auf Remote-Slave-Baugruppenträgern oder bei C200H Multipunkt-Eingangsbaugruppen der Gruppe 2) zugeordnet ist. Gleichzeitig wird der Status der Ausführungsbedingung (Stromfluss) in das spezifizierte Ausgangs-Bit im E/A-Speicher geschrieben.

Wenn OUTB(534) zwischen IL(002) und ILC(003) programmiert wird, wird das spezifizierte Bit bei Verriegelung des Programmabschnitts auf AUS gesetzt. (Das entspricht einem OUT-Befehl in einem verriegelten Programmabschnitt.)

Wenn ein Wort für die Bit-Nummer (N) spezifiziert ist, werden nur die Bits 00 bis 03 von N verwendet. Wenn N z.B. FFFA hex enthält, steuert OUTB(534) Bit 10 von Wort D.

Beispiel



Bit 10 von D00000 wird auf AUS gesetzt, wenn CIO 000000 den Zustand AUS hat.

3-5 Sequenzsteuerbefehle

3-5-1 END: END(001)

Funktion

Kennzeichnet das Ende eines Programms.

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	END(001)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

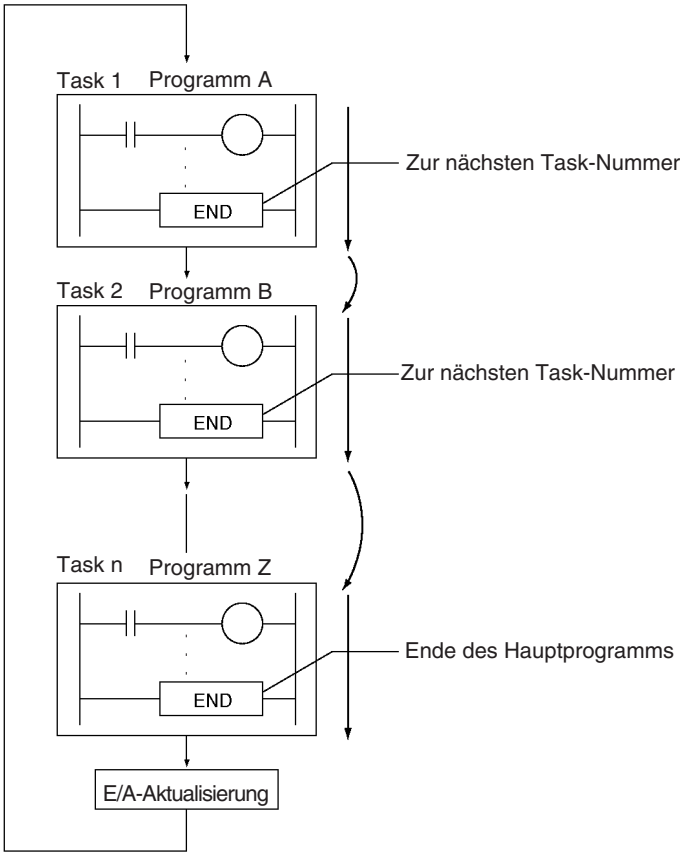
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	Nicht zulässig	Nicht zulässig	OK

Beschreibung

END(001) schließt die Ausführung eines Programms für diesen Zyklus ab. Nach END(001) geschriebene Befehle werden nicht ausgeführt.

Die Ausführung fährt mit dem Programm mit der nächsten Task-Nummer fort. Wenn das aktuell ausgeführte Programm die höchste Task-Nummer im Programm hat, markiert END(001) das Ende des gesamten Hauptprogramms.



Vorsichtsmaßnahmen

Am Ende jedes Programms muss stets der Befehl END(001) stehen. Wenn der Befehl END(001) in einem Programm fehlt, tritt ein Programmfehler auf.

3-5-2 NO OPERATION: NOP(000)

Funktion

Dieser Befehl hat keine Funktion. (Durch NOP(000) erfolgt keine Verarbeitung.)

Kontaktplansymbol

Es gibt zum Befehl NOP(000) kein zugehörigen Kontaktplansymbol.

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	NOP(000)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Beschreibung

Es findet keine Verarbeitung für NOP(000) statt, aber dieser Befehl kann Programmzeilen reservieren, in die später Befehle eingefügt werden können. Wenn die Befehle später eingefügt werden, gibt es keine Änderung in den Programmadressen.

Flags

Flags sind von NOP(000) nicht betroffen.

Vorsichtsmaßnahmen

NOP(000) kann nur bei AWL-Anzeigen und nicht mit Kontaktplanprogrammen verwendet werden.

3-5-3 Übersicht über Verriegelungsbefehle

Verriegelungsbefehle

Folgende Befehlskombinationen können zum Verriegeln von Ausgaben in einem Programmabschnitt verwendet werden.

- INTERLOCK und INTERLOCK CLEAR (IL(002) und ILC(003))
- MULTI-INTERLOCK DIFFERENTIATION HOLD und MULTI-INTERLOCK CLEAR (MILH(517) und MILC(519))*

  Hinweis MILH(517) behält den Status des Flankenauswertungs-Flags bei, so dass verriegelte Befehle mit Flankenauswertung nach dem Aufheben der Verriegelung ausgeführt werden.
- MULTI-INTERLOCK DIFFERENTIATION RELEASE und MULTI-INTERLOCK CLEAR (MILR(518) und MILC(519))*

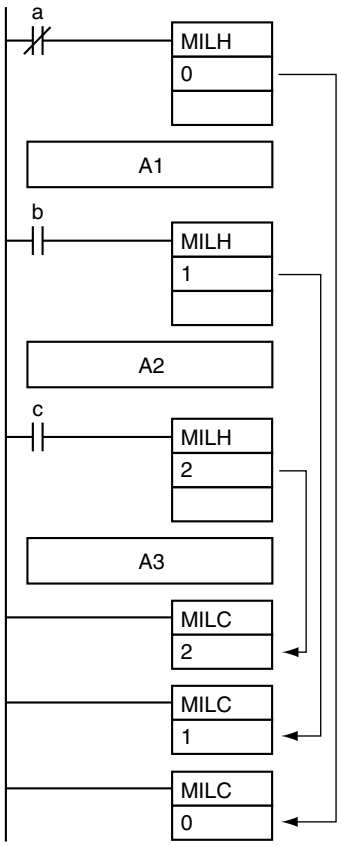
  Hinweis MILH(518) behält den Status des Flankenauswertungs-Flags nicht bei, so dass verriegelte Befehle mit Flankenauswertung nach dem Aufheben der Verriegelung nicht ausgeführt werden.

  * Diese Befehle werden nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

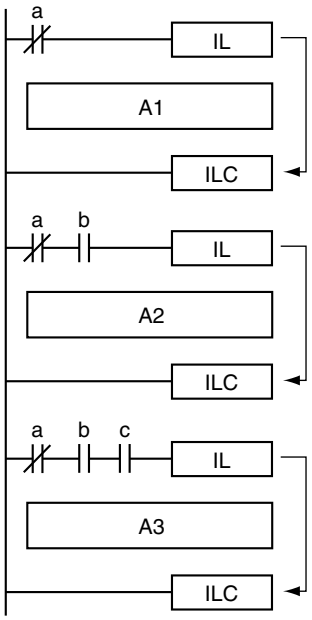
Unterschiede zwischen Verriegelungen und Mehrfachverriegelungen

Reguläre Verriegelungen (IL(002) und ILC(003)) können im Gegensatz zu Mehrfachverriegelungen (MILH(517), MILR(518) und MILC(519)) nicht verschachtelt werden. Die Kontaktplanprogrammierung kann durch Verschachteln von Mehrfachverriegelungen, wie in der nachstehenden Abbildung gezeigt, vereinfacht werden.

Verriegelungen mit MILH und MILC



Verriegelungen mit IL und ILC



Unterschiede zwischen MILH(517) und MILR(518)

Befehle mit Flankenauswertung (DIFU, DIFD oder Befehle mit dem Präfix @ oder %) funktionieren in mit Hilfe von MILH(517) und MILR(518) erzeugten Verriegelungen anders.

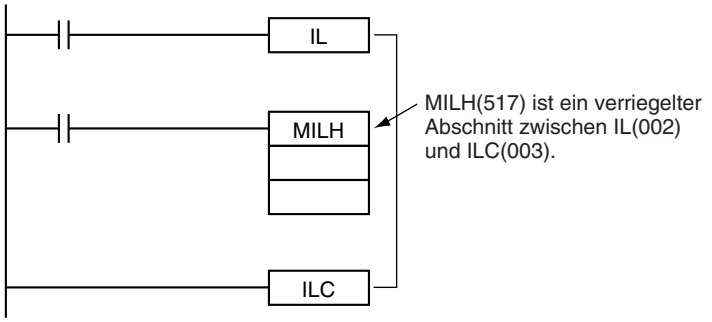
Die Funktion von Befehlen mit Flankenauswertung in einer mit MILH(517) erzeugten Verriegelung ist identisch mit der Funktion in einer mit IL(002) erzeugten Verriegelung.

Weitere Informationen dazu finden Sie unter *3-5-5 MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE und MULTI-INTERLOCK CLEAR: MILH(517), MILR(518) und MILC(519)*.

Vorsichtsmaßnahmen

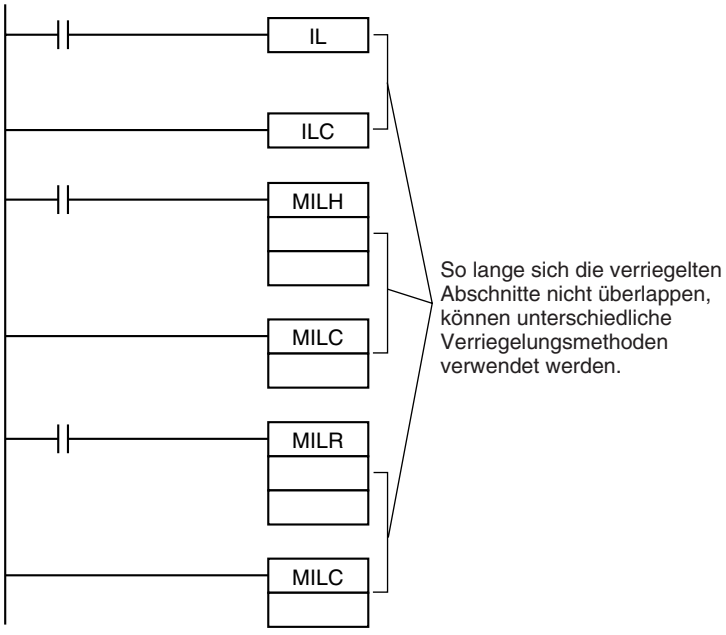
Verriegelungen, die mit unterschiedlichen Verriegelungsbefehlen (IL-ILC, MILH-MILC und MILR-MILC) erzeugt wurden, dürfen nicht kombiniert werden. Wenn verschiedene Verriegelungsmethoden zusammen verwendet werden, funktionieren die Verriegelungen evtl. nicht ordnungsgemäß. Einzelheiten über das Kombinieren von Befehlen finden Sie unter *3-5-5 MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE und MULTI-INTERLOCK CLEAR: MILH(517), MILR(518) und MILC(519)*.)

Beispielsweise kann ein MILH(517)-Befehl nicht zwischen IL(002) und ILC(003) eingefügt werden.



Hinweis Die unterschiedlichen Verriegelungen (IL-ILC, MILH-MILC und MILR-MILC) können zusammen verwendet werden, so lange die verriegelten Programmabschnitte nicht überlappen.

Es können z.B. alle drei Verriegelungsmethoden wie in der nachstehenden Abbildung gezeigt verwendet werden.



Unterschiede zwischen Verriegelungen und Sprüngen

Die folgende Tabelle zeigt die Unterschiede zwischen Verriegelungen (erzeugt mit IL(002)/ILC(003), MILH(517)/MILC(519) oder MILR(518)/ MILC(519)) und mit JMP(004)/JME(005) erzeugten Sprüngen.

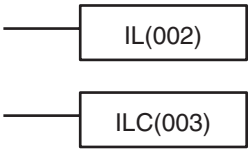
Eigenschaft	Behandlung in IL(002)/ILC(003), MILH(517)/ MILC(519) oder MILR(518)/MILC(519))	Behandlung in JMP(004)/JME(005)
Befehlsausführung	Befehle mit Ausnahme von OUT, OUT NOT, OUTB(534) und Zeitgeberbefehlen werden nicht ausgeführt.	Es werden keine Befehle ausgeführt.
Ausgangsstatus in Befehlen	Mit Ausnahme der Ausgänge von OUT, OUT NOT, OUTB(534) und Zeitgeberbefehlen behalten alle Ausgänge ihren vorherigen Status bei.	Alle Ausgänge behalten ihren vorherigen Status bei.
Bits in OUT, OUT NOT, OUTB(534)	AUS	Alle Ausgänge behalten ihren vorherigen Status bei.
Status der Zeitgeberbefehle (außer (TTIM(087), TTIMX(555), MTIM(543) und MTIMX(554))	Rücksetzen	Der Zeitgeber (nur TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552)) führt die Zeitmessung fort, da die Istwerte auch dann aktualisiert werden, wenn der Zeitgeberbefehl nicht ausgeführt wird.

3-5-4 INTERLOCK und INTERLOCK CLEAR: IL(002) und ILC(003)

Funktion

Verriegelt alle Ausgänge zwischen IL(002) und ILC(003), wenn die Ausführungsbedingung für IL(002) den Zustand AUS hat. IL(002) und ILC(003) werden normalerweise als Paar verwendet.

Kontaktplansymbole



Variationen

Variationen	Verriegelung, wenn auf AUS gesetzt / Keine Verriegelung, wenn auf EIN gesetzt	IL(002)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

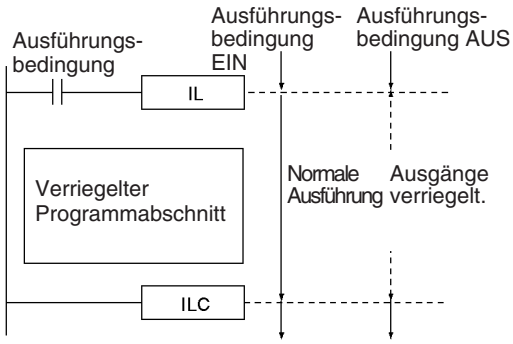
Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	ILC(003)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	Nicht zulässig	OK	OK

Beschreibung

Wenn die Ausführungsbedingung für IL(002) auf AUS gesetzt ist, sind die Ausgänge für alle Befehle zwischen IL(002) und ILC(003) verriegelt. Wenn die Ausführungsbedingung für IL(002) auf EIN gesetzt ist, werden die Befehle zwischen IL(002) und ILC(003) normal ausgeführt.



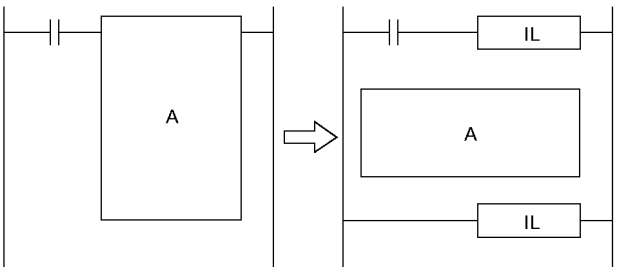
Die folgende Tabelle zeigt die Behandlung verschiedener Ausgänge in einem verriegelten Abschnitt zwischen IL(002) und ILC(003).

Befehl		Behandlung
In OUT, OUT NOT oder OUTB(534) spezifizierte Bits		AUS
TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIML(542) und TIMXL(553)	Fertig-Bit	AUS (Rücksetzung)
	Istwert	Zeitsollwert (Rücksetzung)
In allen anderen Befehlen spezifizierte Bits/Worte (siehe Hinweis)		Vorheriger Status bleibt erhalten.

Hinweis Bits und Worte in allen anderen Befehlen, einschließlich TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, RSET, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT und KEEP(011), behalten ihren vorherigen Status bei.

Wenn es Bits gibt, die in einem verriegelten Programmabschnitt auf EIN gesetzt bleiben sollen, müssen diese Bits kurz vor IL(002) mit dem Befehl SET auf EIN gesetzt werden.

Die Umschaltung eines Programmabschnitts ist mit IL(002) und ILC(003) oft effizienter. Wenn mehrere Prozesse mit der selben Ausführungsbedingung gesteuert werden, sind weniger Programmschritte erforderlich, um diese Prozesse zwischen IL(002) und ILC(003) einzufügen.



Die nachfolgende Tabelle zeigt die Unterschiede zwischen IL(002)/ILC(003) und JMP(004)/JME(005).

Eigenschaft	Behandlung in IL(002)/ILC(003)	Behandlung in JMP(004)/JME(005)
Befehlsausführung	Befehle mit Ausnahme von OUT, OUT NOT, OUTB(534) und Zeitgeberbefehlen werden nicht ausgeführt.	Es werden keine Befehle ausgeführt.
Ausgangsstatus in Befehlen	Mit Ausnahme der Ausgänge von OUT, OUT NOT, OUTB(534) und Zeitgeberbefehlen behalten alle Ausgänge ihren vorherigen Status bei.	Alle Ausgänge behalten ihren vorherigen Status bei.
Bits in OUT, OUT NOT, OUTB(534)	AUS	Alle Ausgänge behalten ihren vorherigen Status bei.
Status der Zeitgeberbefehle (außer (TTIM(087), TTIMX(555), MTIM(543) und MTIMX(554))	Rücksetzung	Der Zeitgeber (nur TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552)) führt die Zeitmessung fort, da die Istwerte auch dann aktualisiert werden, wenn der Zeitgeberbefehl nicht ausgeführt wird.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden Gleich- und Negativ-Flag auf AUS gesetzt.

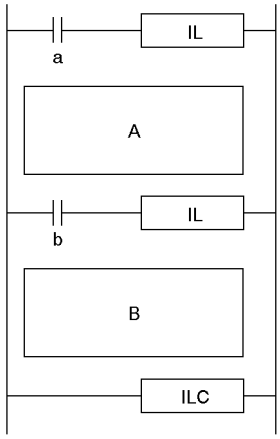
In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben Gleich- und Negativ-Flag unverändert.

Vorsichtsmaßnahmen

Die Zykluszeit wird bei Verriegelung eines Programmabschnitts nicht verkürzt, weil die verriegelten Befehle intern ausgeführt werden.

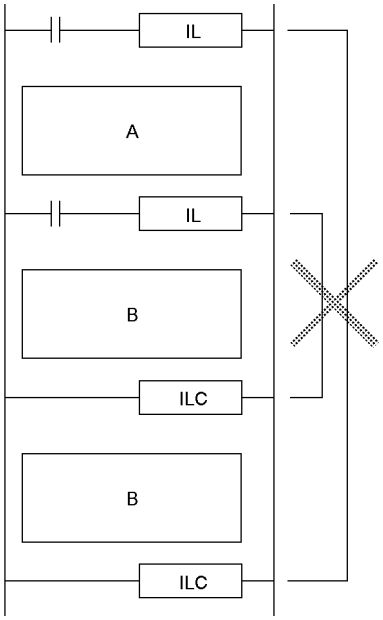
Die Funktion von DIFU(013), DIFD(014) und Befehlen mit Flankenauswertung hängt nicht nur vom Status der Ausführungsbedingung ab, wenn diese zwischen IL(002) und ILC(003) programmiert werden. Änderungen der Ausführungsbedingung für DIFU(013), DIFD(014) oder eines Befehls mit Flankenauswertung werden nicht aufgezeichnet, wenn sich DIFU(013) oder DIFD(014) in einem verriegelten Programmabschnitt befindet und die Ausführungsbedingung für IL(002) auf AUS gesetzt ist.

Allgemein werden IL(002) und ILC(003) paarweise verwendet, obwohl es auch möglich ist, mehr als einen IL(002) mit einem einzelnen ILC(003) zu verwenden. Siehe nachfolgende Abbildung. Wenn IL(002) und ILC(003) nicht als Paar verwendet werden, wird bei der Durchführung einer Programmprüfung eine Fehlermeldung angezeigt, aber das Programm wird ordnungsgemäß ausgeführt.



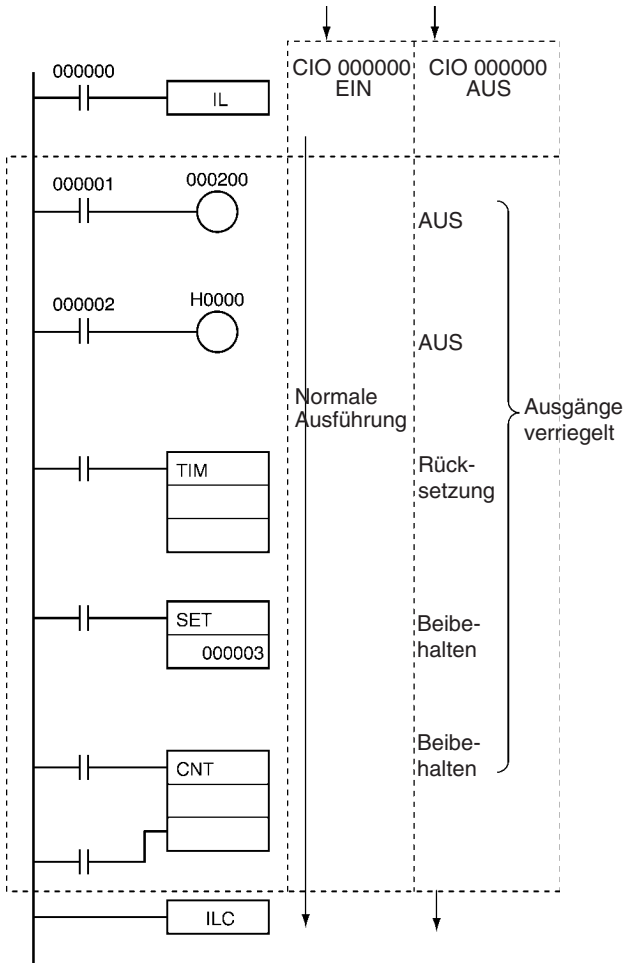
Ausführungsbedingung		Programmabschnitt	
a	**b**	**A**	**B**
AUS	EIN	Verriegelt	Verriegelt
AUS	AUS	Verriegelt	Verriegelt
EIN	AUS	Nicht verriegelt	Verriegelt
EIN	EIN	Nicht verriegelt	Nicht verriegelt

IL(002) und ILC(003) können, wie in der nachfolgenden Abbildung ersichtlich, nicht verschachtelt werden. (Falls Verriegelungen verschachtelt werden müssen, verwenden Sie die Befehle MILH(517)/MILR(518) und MILC(519).)



Beispiele

Wenn CIO 000000 im folgenden Beispiel auf AUS gesetzt ist, werden alle Ausgänge zwischen IL(002) und ILC(003) verriegelt. Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden alle Befehle zwischen IL(002) und ILC(003) normal ausgeführt.



3-5-5 MULTI-INTERLOCK DIFFERENTIATION HOLD, MULTI-INTERLOCK DIFFERENTIATION RELEASE und MULTI-INTERLOCK CLEAR: MILH(517), MILR(518) und MILC(519)

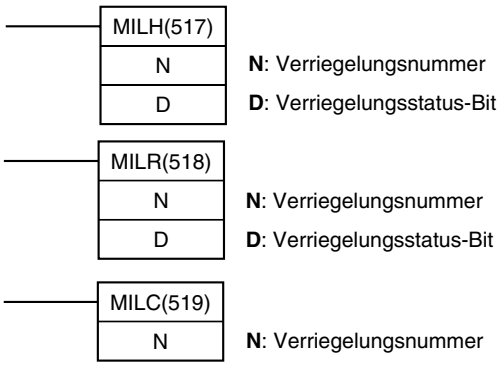
Funktion

Verriegelt alle Ausgänge zwischen MILH(517) (oder MILR(518)) und MILC(519), wenn die Ausführungsbedingung für MILH(517) (oder MILR(518)) auf AUS gesetzt ist. MILH(517) (oder MILR(518)) und MILC(519) werden gewöhnlich paarweise verwendet.

Anders als die Verriegelungen mit IL(002)/ILC(003) können die Verriegelungen mit MILH(517)/MILC(519) und MILR(518)/MILC(519) verschachtelt werden. Die Funktion von Befehlen mit Flankenauswertung ist bei Verriegelungen, die mit MILH(517) und MILR(518) erzeugt wurden, unterschiedlich.

Diese Befehle werden nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbole



Operanden

N: Verriegelungsnummer

Die Verriegelungsnummer muss eine Zahl zwischen 0 und 15 sein. Die Verriegelungsnummer des Befehl MILH(517) (oder MILR(518)) muss der Nummer im entsprechenden MILC(519)-Befehl entsprechen.

Die Verriegelungsnummern können in jeder beliebigen Reihenfolge verwendet werden.

D: Verriegelungsstatus-Bit

- Auf EIN gesetzt, wenn der Programmabschnitt nicht verriegelt ist.
- Auf AUS gesetzt, wenn der Programmabschnitt verriegelt ist.

Wenn die Verriegelung aktiviert ist, kann das Verriegelungsstatus-Bit zwangsweise zum Aufheben der Verriegelung gesetzt werden. Wenn die Verriegelung umgekehrt nicht aktiviert ist, kann das Verriegelungsstatus-Bit zwangsweise zur Aktivierung der Verriegelung zurückgesetzt werden.

Operanden-Spezifikationen

Bereich	N	D
CIO-Bereich	---	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	---	W00000 bis W51115
Arbeitsbereich remanent	---	H00000 bis H51115
Systembereich	---	A00000 bis A95915
Zeitgeberbereich	---	---
Zählerbereich	---	---
DM-Bereich	---	---
EM-Bereich ohne Bank	---	---

Bereich	N	D
EM-Bereich mit Bank	---	---
Indirekte DM/EM-Binäradressen	---	---
Indirekte DM/EM-Adressen in BCD	---	---
Konstanten	0 bis 15	---
Datenregister	---	---
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15

Variationen

Variationen	Verriegelung, wenn auf AUS gesetzt / Keine Verriegelung, wenn auf EIN gesetzt	MILH(517) und MILR(518)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	MILC(519)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Die nachfolgende Tabelle zeigt die geeigneten Programmabschnitte für MILH(517), MILR(518) und MILC(519).

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	Nicht zulässig	OK	OK

Beschreibung

Wenn die Ausführungsbedingung für MILH(517) (oder MILR(518) mit Verriegelungsnummer N den Zustand AUS hat, werden die Ausgaben aller Befehle zwischen diesem MILH(517)/MILR(518)-Befehl und dem nächsten MILC(519)-Befehl mit Verriegelungsnummer N verriegelt.

Wenn die Ausführungsbedingung für MILH(517) (oder MILR(518) mit Verriegelungsnummer N den Zustand EIN hat, werden die Ausgaben aller Befehle zwischen diesem MILH(517)/MILR(518)-Befehl und dem nächsten MILC(519)-Befehl mit Verriegelungsnummer N normal ausgeführt.

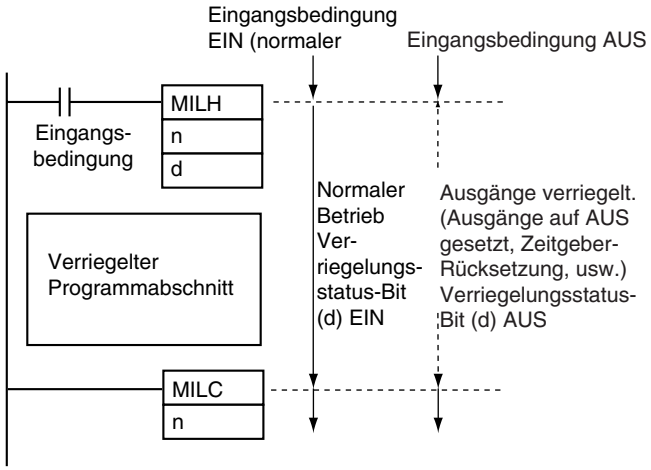
Verriegelungsstatus

Die folgende Tabelle zeigt die Behandlung verschiedener Ausgänge in einem verriegelten Abschnitt zwischen dem Befehl MILH(517)/MILR(518) und dem nächsten MILC(519).

Befehl		Behandlung
In OUT, OUT NOT oder OUTB(534) spezifizierte Bits		AUS
TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIML(542) und TIMXL(553)	Fertig-Bit	AUS (Rücksetzung)
	Istwert	Zeitsollwert (Rücksetzung)
In allen anderen Befehlen spezifizierte Bits/Worte (siehe Hinweis)		Vorheriger Status bleibt erhalten.

Hinweis Bits und Worte in allen anderen Befehlen, einschließlich TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, RSET, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT und KEEP(011), behalten ihren vorherigen Status bei.

Der Befehl MILH(517)/MILR(518) setzt das Verriegelungsstatus-Bit (Operand D) auf AUS, wenn die Verriegelung aktiviert ist und setzt das Bit auf EIN, wenn die Verriegelung nicht aktiviert ist. Folglich kann das Verriegelungsstatus-Bit überwacht werden, um zu prüfen, ob die Verriegelung einer gegebenen Verriegelungsnummer aktiviert ist oder nicht.



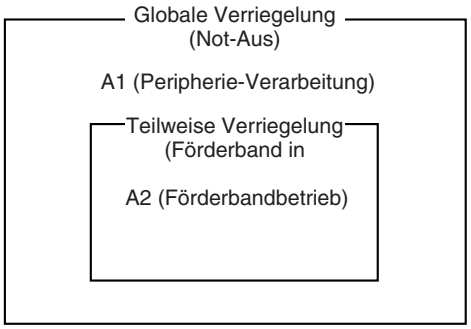
Schachtelung

Verriegelungen werden verschachtelt, wenn ein verriegelter Programmabschnitt (MILH(517)/MILR(518) und MILC(519) Kombination) in einem anderen verriegelten Programmabschnitt (MILH(517)/MILR(518) und MILC(519) Kombination) platziert wird. Verriegelungen können in bis zu 16 Ebenen verschachtelt werden.

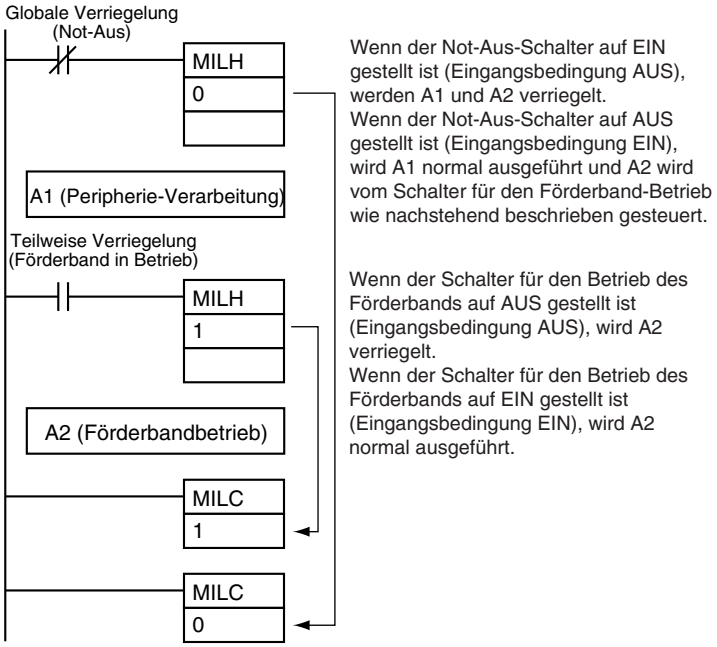
Die Schachtelung kann für die folgenden Arten von Anwendungen verwendet werden.

- Beispiel 1

  Verriegelung des gesamten Programms mit einer Bedingung und Verriegelung eines Abschnitts des Programms mit einer anderen Bedingung (1 Schachtelebene)

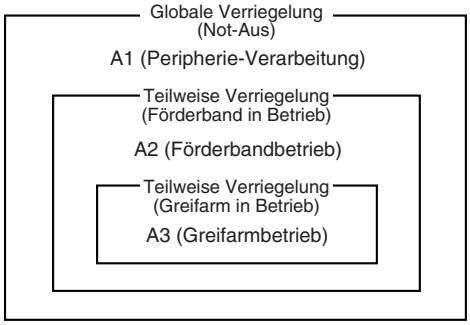


  - A1 und A2 werden bei Betätigung des Not-Aus-Schalters verriegelt.
  - A2 wird verriegelt, wenn das Signal für den Betrieb des Förderbands den Zustand AUS hat.

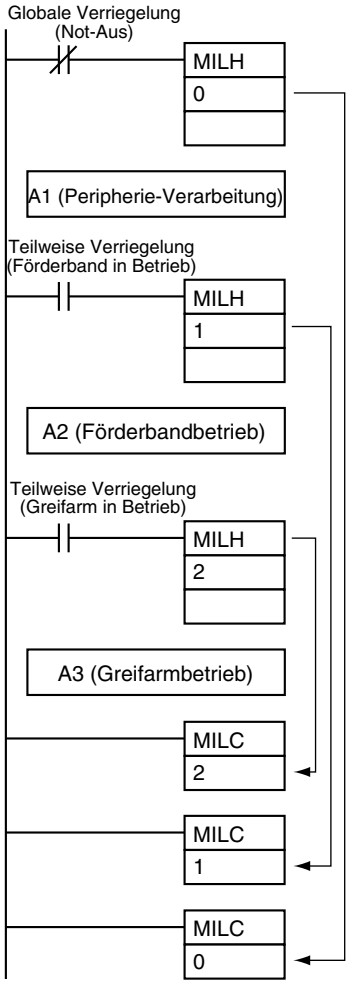


- Beispiel 2

  Verriegelung des gesamten Programms mit einer Bedingung und Verriegelung von zwei überlappenden Abschnitten des Programms durch andere Bedingungen (2 Schachtelebenen)



  - A1, A2 und A3 werden bei Betätigung des Not-Aus-Schalters verriegelt.
  - A2 und A3 werden verriegelt, wenn das Signal für den Betrieb des Förderbands den Zustand AUS hat.
  - A3 wird verriegelt, wenn das Signal für den Betrieb des Greifarms den Zustand AUS hat.



Wenn der Not-Aus-Schalter auf EIN gestellt ist (Eingangsbedingung AUS), werden A1, A2 und A3 verriegelt.
Wenn der Not-Aus-Schalter auf AUS gestellt ist (Eingangsbedingung EIN), wird A1 normal ausgeführt, während A2 und A3 von den Schaltern für den Förderband- und Greifarmbetrieb wie nachstehend beschrieben gesteuert werden.

Wenn der Schalter für den Betrieb des Förderbands auf AUS gestellt ist (Eingangsbedingung AUS), werden A2 und A3 verriegelt.
Wenn der Schalter für den Betrieb des Förderbands auf EIN gestellt ist (Eingangsbedingung EIN), wird A2 normal ausgeführt und A3 wird über den Schalter für den Betrieb des Greifarms wie nachstehend beschrieben gesteuert.

Wenn der Schalter für den Betrieb des Greifarms auf AUS gestellt ist (Eingangsbedingung AUS), wird A3 verriegelt.
Wenn der Schalter für den Betrieb des Greifarms auf EIN gestellt ist (Eingangsbedingung EIN), wird A3 normal ausgeführt.

Unterschiede zwischen MILH(517) und MILR(518)

Befehle mit Flankenauswertung (DIFU, DIFD oder Befehle mit dem Präfix @ oder %) funktionieren in mit Hilfe von MILH(517) und MILR(518) erzeugten Verriegelungen unterschiedlich.

Wenn ein Programmabschnitt durch MILR(518) verriegelt ist, wird ein Befehl mit Flankenauswertung bei Aufhebung der Verriegelung **nicht** ausgeführt, auch wenn die Flankenüberwachungsbedingung während der Verriegelung aktiviert war (Vergleich des Status der Ausführungsbedingung bei Verriegelung mit deren Status bei Aufhebung der Verriegelung).

Wenn ein Programmabschnitt durch MILH(517) verriegelt ist, wird ein Befehl mit Flankenauswertung bei Aufhebung der Verriegelung **ausgeführt**, wenn die Flankenüberwachungsbedingung während der Verriegelung aktiviert war (Vergleich des Status der Ausführungsbedingung bei Verriegelung mit deren Status bei Aufhebung der Verriegelung).

Befehl	Funktion von Befehlen mit Flankenauswertung
MILH(517) MULTI-INTERLOCK DIFFE-RENTIATION HOLD	Ein Befehl mit Flankenauswertung (DIFU, DIFD oder ein Befehl mit Präfix @ oder %) **wird** nach Aufhebung der Verriegelung ausgeführt, wenn die Flankenauswertungsbedingung während der Verriegelung des Befehls hergestellt wurde. (Der Status der Ausführungsbedingung zu Beginn der Verriegelung wird mit dem Status beim Aufheben der Verriegelung verglichen.)
MILR(518) MULTI-INTERLOCK DIFFE-RENTIATION RELEASE	Ein Befehl mit Flankenauswertung (DIFU, DIFD oder ein Befehl mit Präfix @ oder %) wird nach Aufhebung der Verriegelung **nicht** ausgeführt, selbst wenn die Flankenauswertungsbedingung während der Verriegelung des Befehls hergestellt wurde.

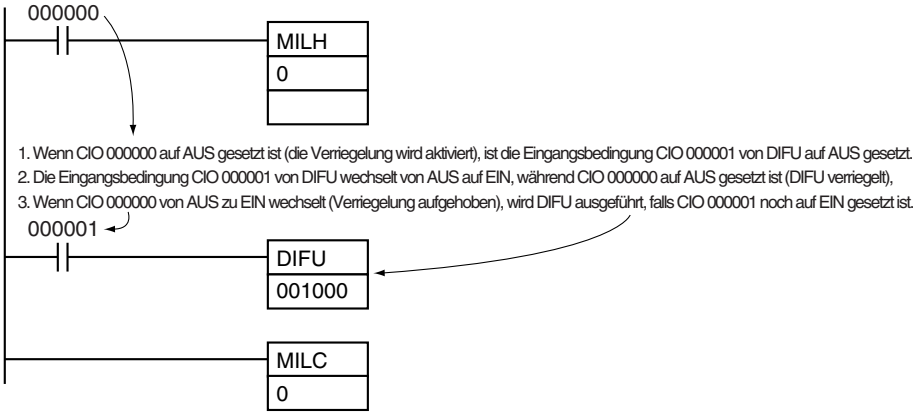
- Funktion von Befehlen mit Flankenauswertung in einer Verriegelung durch MILH(517)

  Bei einem Befehl mit Flankenauswertung (DIFU, DIFD bzw. ein Befehl mit einem Präfix @ oder %) zwischen MILH(517) und dem entsprechenden MILC(519)-Befehl **wird** dieser Befehl nach Aufhebung der Verriegelung ausgeführt, wenn die Flankenauswertungsbedingung des Befehls eingetreten war. (Das System vergleicht den Status der Ausführungsbedingung zu Beginn der Verriegelung mit dem Status beim Aufheben der Verriegelung.)

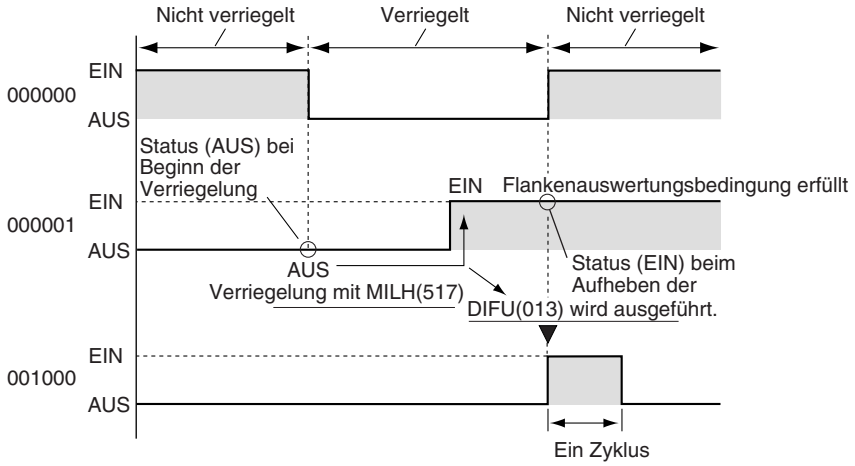
  Ebenso wird ein Befehl mit Flankenauswertung ausgeführt, wenn dessen Ausführungsbedingung gleichzeitig mit dem Beginn oder dem Ende der Verriegelung hergestellt ist.

  Viele andere Bedingungen im Programm können eine Rücksetzung der Flankenauswertungsbedingung verursachen, auch wenn diese während der Verriegelung hergestellt wurde. In diesem Fall wird der Befehl mit Flankenauswertung bei Aufhebung der Verriegelung nicht ausgeführt.

  - Beispiel
    Wenn der Befehl DIFFERENTIATE UP (DIFU(013)) verwendet wird und die Eingangsbedingung bei Aktivierung der Verriegelung auf AUS und beim Aufheben der Verriegelung auf EIN gesetzt ist, wird DIFU(013) nach Aufhebung der Verriegelung **ausgeführt**. (Befehle mit Flankenauswertung funktionieren in der Verriegelung mit MILH(517) genauso wie in einer Verriegelung mit IL(002).)



Zeitablaufdiagramm

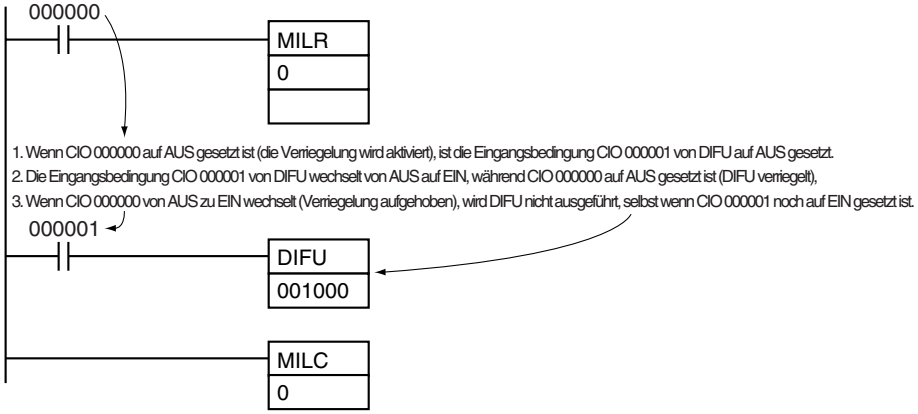


- Funktion von Befehlen mit Flankenauswertung in einer Verriegelung durch MILR(518)

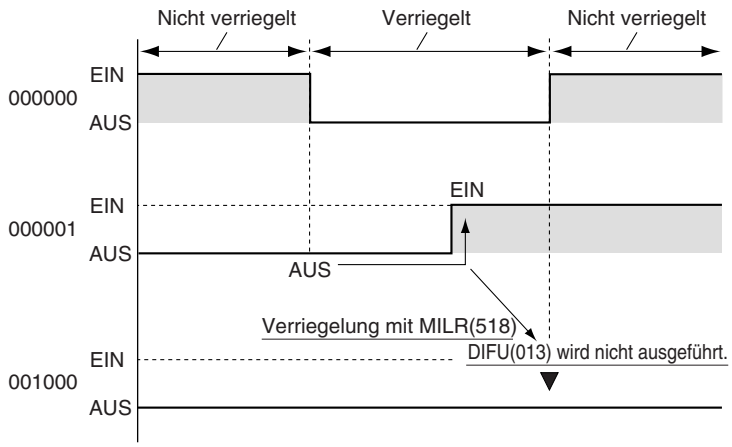
  Bei einem Befehl mit Flankenauswertung (DIFU, DIFD bzw. ein Befehl mit einem Präfix @ oder %) zwischen MILR(518) und dem entsprechenden MILC(519)-Befehl wird dieser Befehl nach Aufhebung der Verriegelung **nicht** ausgeführt, selbst wenn die Flankenauswertungsbedingung des Befehls eingetreten war. (Das System vergleicht den Status der Ausführungsbedingung im Zyklus zu Beginn der Verriegelung mit deren Status im Zyklus bei Aufhebung der Verriegelung.)

  Ebenso wird ein Befehl mit Flankenauswertung nicht ausgeführt, wenn dessen Ausführungsbedingung gleichzeitig mit dem Beginn oder dem Ende der Verriegelung hergestellt war.

  - Beispiel
    Wenn der Befehl DIFFERENTIATE UP (DIFU(013)) verwendet wird und die Eingangsbedingung bei Aktivierung der Verriegelung auf AUS und beim Aufheben der Verriegelung auf EIN gesetzt ist, wird DIFU(013) nach Aufhebung der Verriegelung **nicht** ausgeführt.



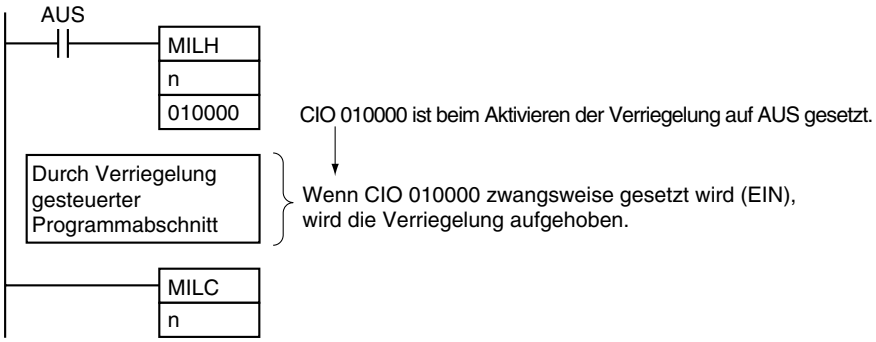
Zeitablaufdiagramm



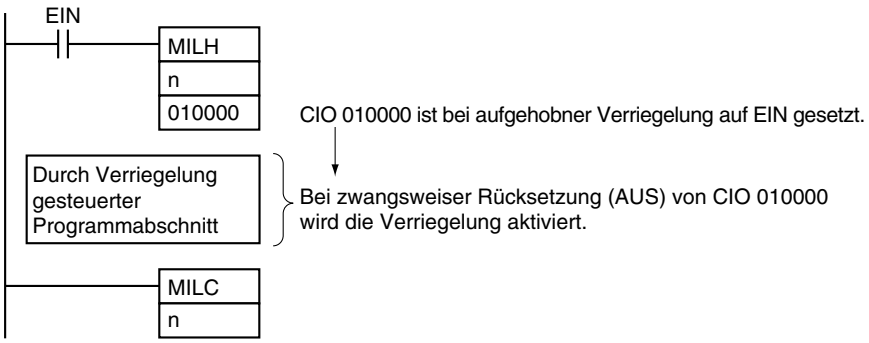
Steuerung des Verriegelungsstatus von einem Programmiergerät aus

Von einem Programmiergerät aus kann eine Verriegelung durch zwangsweises Rücksetzen oder zwangsweises Setzen des Verriegelungsstatus-Bits manuell aktiviert oder aufgehoben werden (spezifiziert durch Operand D von MILH(517) und MILR(518)). Der Zwangsstatus des Verriegelungsstatus-Bits hat Priorität und überschreibt den durch die Programmausführung berechneten Verriegelungsstatus.

Zwangsweises Setzen: Aufheben der Verriegelung.

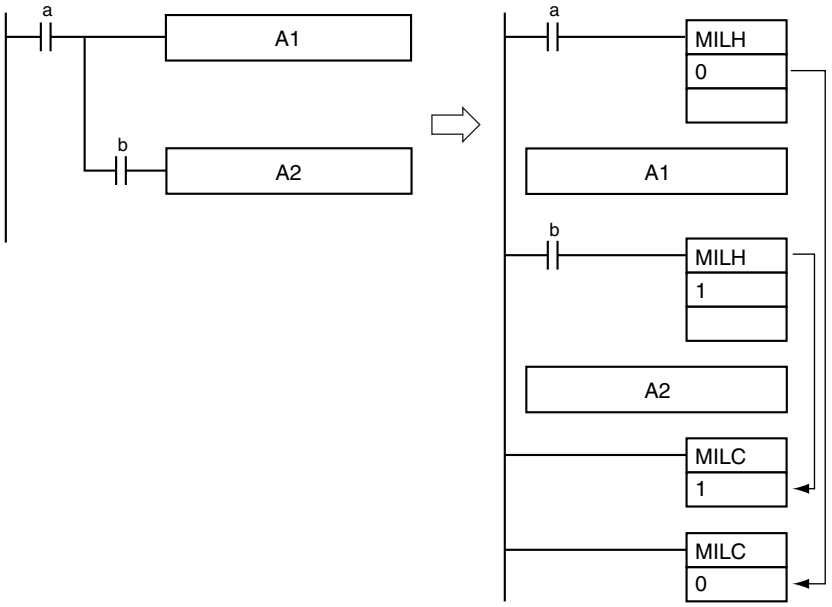


Zwangsweises Rücksetzen: Aktivierung der Verriegelung.



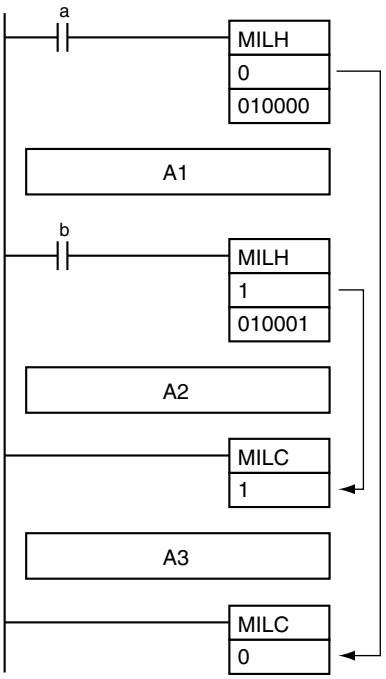
Hinweis Die Programmfunktion kann durch Verwendung von Verriegelungen mit MILH(517) oder MILR(518) effizienter umgeschaltet werden.

Anstatt die Verarbeitung mit Hilfe von komplexen Bedingungen umzuschalten, wird ein MILH(517)- oder MILR(518)-Befehl vor jedem Prozess und ein MILC(519)-Befehl nach jedem Prozess eingefügt.



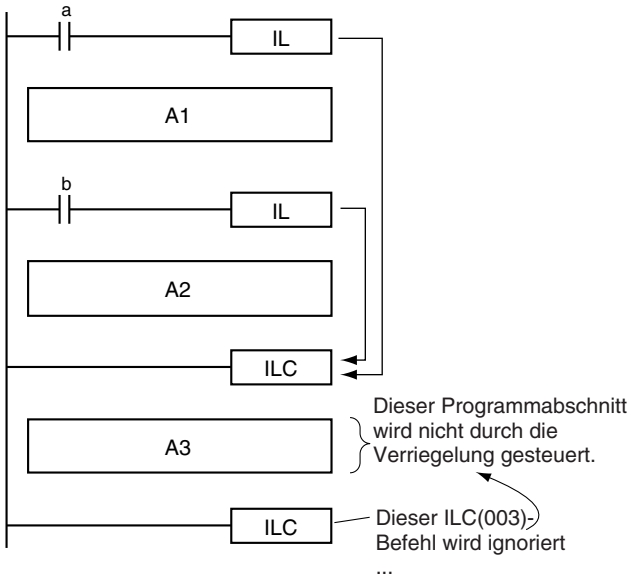
Anders als die Verriegelungen mit IL(002) können die Verriegelungen mit MILH(517) und MILR(518) geschachtelt werden, so dass die Funktion von gleichen Programmen unterschiedlich ist, wenn MILH(517) oder MILR(518) anstelle von ILC(002) verwendet wird.

Programm mit MILH(517)/MILC(519)-Verriegelungen



Ausführungs-bedingung		Programmabschnitt		
a	**b**	**A1**	**A2**	**A3**
AUS	EIN	Verriegelt	Verriegelt	Nicht verriegelt
	AUS			
EIN	AUS	Nicht verriegelt	Verriegelt	Nicht verriegelt
EIN	EIN	Nicht verriegelt	Nicht verriegelt	Nicht verriegelt

Programm mit IL(002)/ILC(003)-Verriegelungen



Ausführungs-bedingung		Programmabschnitt		
a	**b**	**A1**	**A2**	**A3**
AUS	EIN	Verriegelt	Verriegelt	Nicht verriegelt (Nicht von der IL(002)/ILC(003)-Verriegelung gesteuert.)
	AUS			
EIN	AUS	Nicht verriegelt	Verriegelt	
EIN	EIN	Nicht verriegelt	Nicht verriegelt	

Wenn Bits in einem durch MILH(517) oder MILR(518) verriegelten Programmabschnitt auf EIN gesetzt bleiben sollen, setzen Sie diese unmittelbar vor dem Befehl MILH(517) oder MILR(518) mit SET auf EIN.

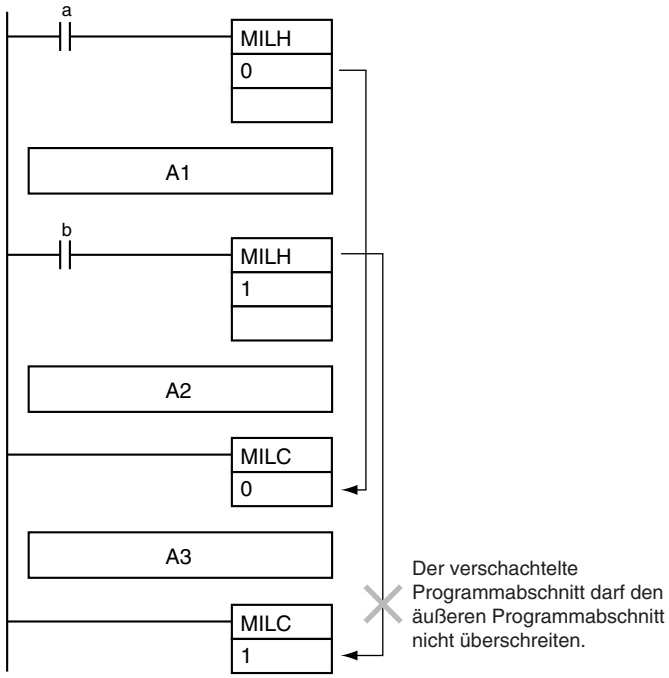
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS

Vorsichtsmaßnahmen

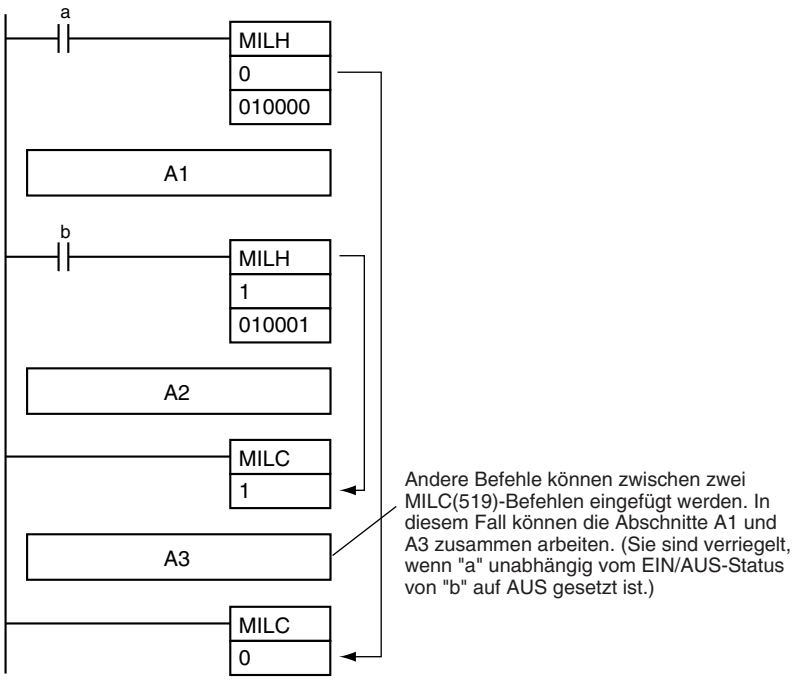
Die Zykluszeit wird bei Verriegelung eines Programmabschnitts durch MILH(517) oder MILR(518) nicht verkürzt, weil die verriegelten Befehle intern ausgeführt werden.

Wenn Verriegelungen verschachtelt werden, müssen Verriegelungsnummern zugewiesen werden, damit der verschachtelte Programmabschnitt den äußeren Programmabschnitt nicht überschreitet.

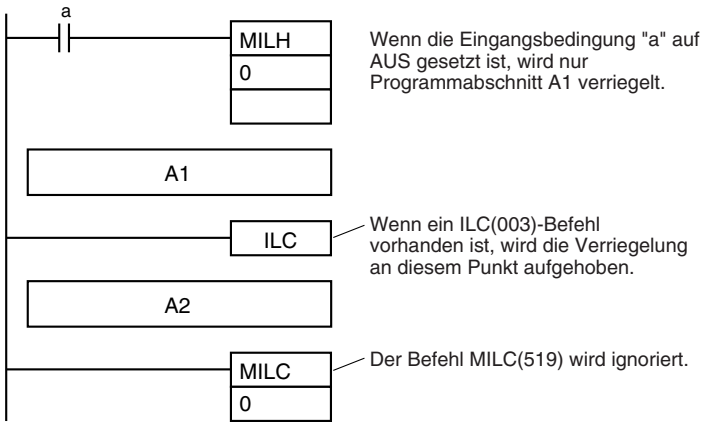


Ausführungs-bedingung		Programmabschnitt		
a	**b**	**A1**	**A2**	**A3**
AUS	EIN	Verriegelt	Verriegelt	Nicht verriegelt
	AUS			
EIN	AUS	Nicht verriegelt	Verriegelt	Verriegelt
	EIN	Nicht verriegelt	Nicht verriegelt	Nicht verriegelt

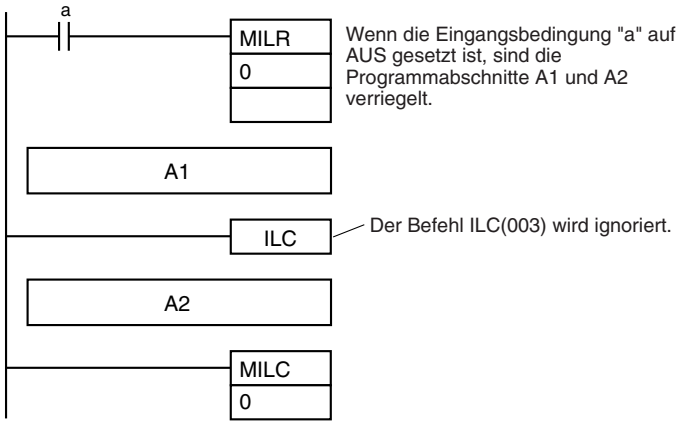
Wie in der folgenden Abbildung ersichtlich, können zwischen den MILC(519)-Befehlen andere Befehle eingegeben werden.



Wenn sich zwischen einem MILH(517)/MILC(519)-Paar ein ILC(003)-Befehl befindet, werden die Programmabschnitte zwischen MILH(517) und ILC(003) verriegelt.

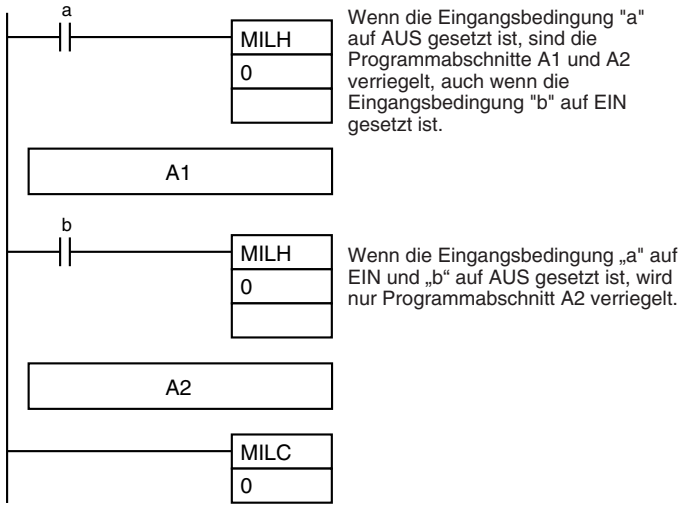


Wenn sich zwischen einem MILR(518)/MILC(519)-Paar ein ILC(003)-Befehl befindet, wird der ILC(003)-Befehl ignoriert und der gesamte Programmabschnitt zwischen MILR(518) und MILC(519) wird verriegelt.



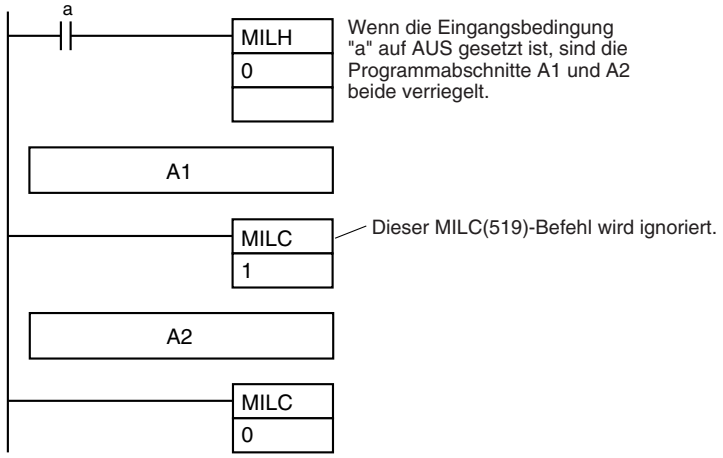
Wenn sich ein weiterer MILH(517) oder MILR(518)-Befehl mit der selben Verriegelungsnummer zwischen einem MILH(517)/MILC(519)-Paar befindet und die Verriegelung des ersten MILH(517)-Befehls aktiviert ist, wird der zweite MILH(517)/MILR(518)-Befehl nicht ausgeführt.

Wenn sich ein weiterer MILH(517) oder MILR(518)-Befehl mit der selben Verriegelungsnummer zwischen einem MILH(517)/MILC(519)-Paar befindet und die Verriegelung des ersten MILH(517)-Befehls aufgehoben ist, wird der zweite MILH(517)/MILR(518)-Befehl normal ausgeführt.



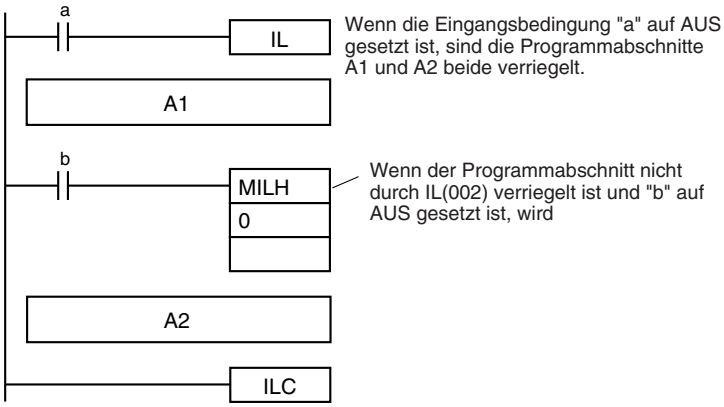
Hinweis MILR(518)-Verriegelungen funktionieren auf die selbe Weise, wenn es zwischen einem MILR(518)/MILC(519)-Paar einen weiteren MILH(517)- oder MILR(518)-Befehl mit der selben Verriegelungsnummer gibt.

Wenn es einen MILC(519)-Befehl mit einer anderen Verriegelungsnummer zwischen einem MILH(517)/MILR(518)- und MILC(519)-Paar gibt, dann wird dieser MILC(519)-Befehl ignoriert.

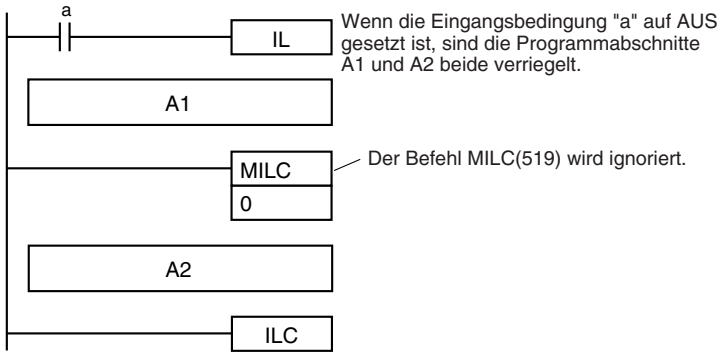


Wenn zwischen dem IL(002)/ILC(003)-Paar ein MILH(517)-Befehl vorhanden ist und die IL(002)-Verriegelung aktiviert ist, so ist der Befehl MILH(517) wirkungslos. In diesem Fall wird der Programmabschnitt zwischen IL(002) und ILC(003) verriegelt.

Wenn die IL(002)-Verriegelung nicht aktiviert ist und die Ausführungsbedingung des Befehls MILH(517) (in diesem Fall b) auf AUS gesetzt ist, wird der Programmabschnitt zwischen MILH(517) und ILC(003) verriegelt.



Wenn sich zwischen einem IL(002)/ILC(003)-Paar ein MILC(519)-Befehl befindet, wird dieser MILC(519)-Befehl ignoriert und der gesamte Programmabschnitt zwischen IL(002) und ILC(003) wird verriegelt.

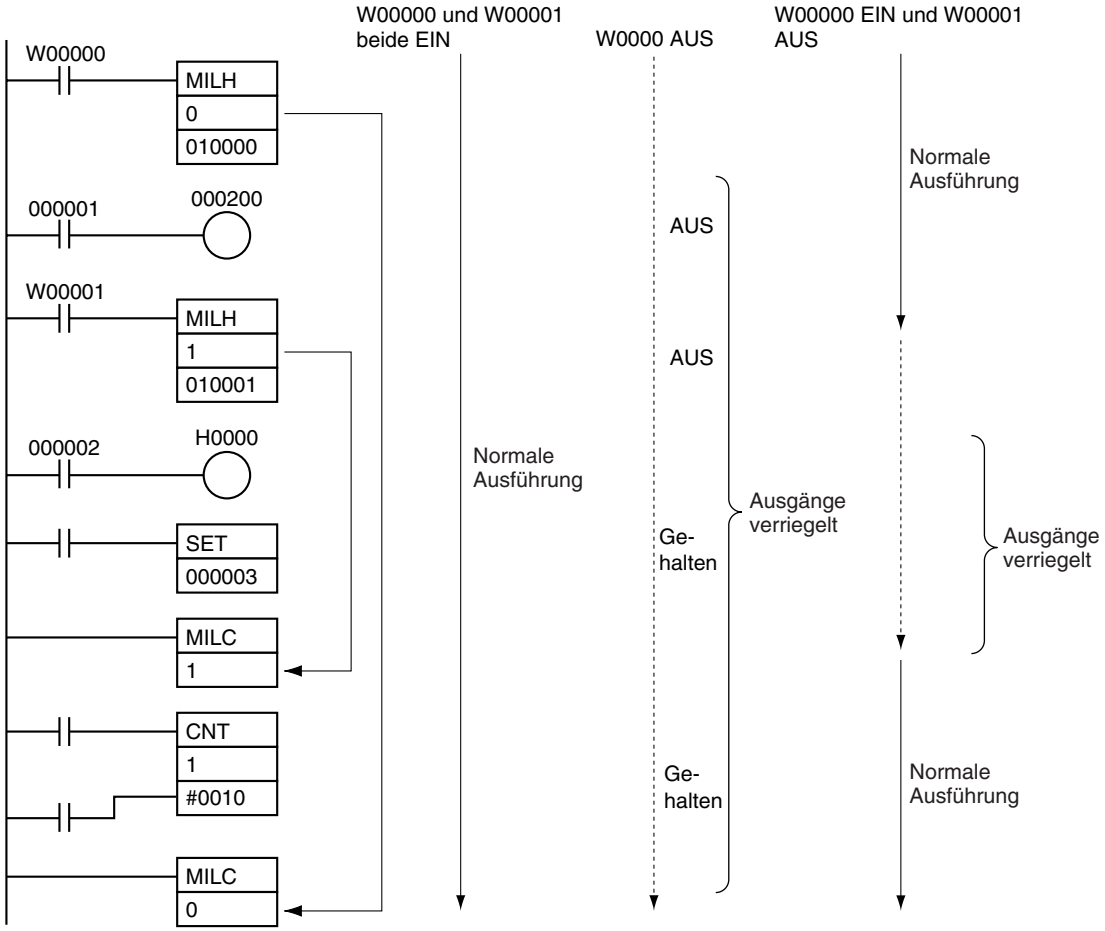


Beispiele

Wenn W00000 und W00001 beide auf EIN gesetzt sind, werden die Befehle zwischen MILH(517) mit Verriegelungsnummer 0 und MILC(519) mit Verriegelungsnummer 0 normal ausgeführt.

Wenn W00000 auf AUS gesetzt ist, werden Befehle zwischen MILH(517) mit Verriegelungsnummer 0 und MILC(519) mit Verriegelungsnummer 0 verriegelt.

Wenn W00000 auf EIN und W00001 auf AUS gesetzt ist, werden Befehle zwischen MILH(517) mit Verriegelungsnummer 1 und MILC(519) mit Verriegelungsnummer 1 verriegelt. Die übrigen Befehle werden normal ausgeführt.

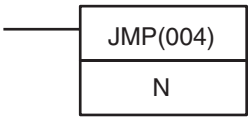


3-5-6 JUMP und JUMP END: JMP(004) und JME(005)

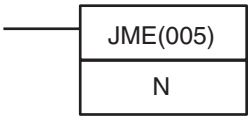
Funktion

Wenn die Ausführungsbedingung für JMP(004) den Zustand AUS hat, springt die Programmausführung direkt zum ersten JME(005) mit derselben Sprung-Nummer im Programm. JMP(004) und JME(005) werden als Paar verwendet.

Kontaktplansymbole



N: Sprungnummer



N: Sprungnummer

Variationen

Variationen	Sprung, wenn auf AUS gesetzt / Kein Sprung, wenn auf EIN gesetzt	JMP(004)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	JME(005)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	Nicht zulässig	OK	OK

Operanden

N: Sprungnummer

Die Sprungnummer muss ein Wert von 0000 bis 03FF (&0 bis &1.023 dezimal) sein.

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 muss die Sprungnummer im Bereich zwischen 0000 und 00FF hex bzw. &0 bis &255 dezimal liegen.

Operanden-Spezifikationen

Bereich	N	
	JMP(004)	JME(005)
CIO-Bereich	CIO 0000 bis CIO 6143	---
Arbeitsbereich nicht remanent	W000 bis W511	---
Arbeitsbereich remanent	H000 bis H511	---
Systembereich	A000 bis A959	---
Zeitgeberbereich	T0000 bis T4095	---
Zählerbereich	C0000 bis C4095	---
DM-Bereich	D00000 bis D32767	---
EM-Bereich ohne Bank	E00000 bis E32767	---
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	---
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	---
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	---
Konstanten	#0000 to #03FF (binär) bzw. &0 bis &1023 (siehe Hinweis)	#0000 to #03FF (binär) bzw. &0 bis &1023 (siehe Hinweis)
Datenregister	DR0 bis DR15	---
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	---

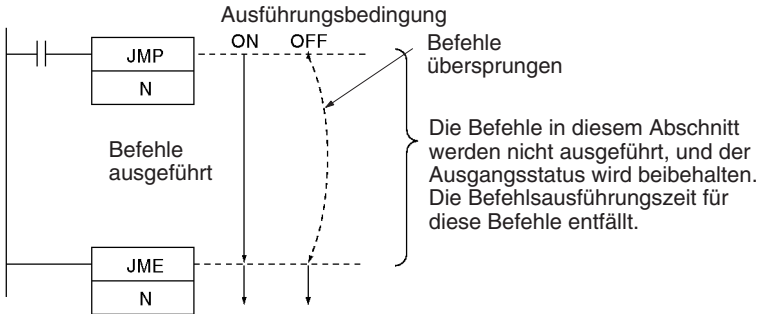
Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 liegt der gültige Bereich zwischen #0000 und #00FF (binär) bzw. &0 bis &255 (dezimal).

Beschreibung

Wenn die Ausführungsbedingung für JMP(004) auf EIN gesetzt ist, wird kein Sprung ausgeführt und das Programm fortlaufend wie geschrieben ausgeführt.

Wenn die Ausführungsbedingung für JMP(004) den Zustand AUS hat, springt die Programmausführung direkt zum ersten JME(005) mit derselben Sprung-Nummer im Programm. Die Befehle zwischen JMP(004) und JME(005) wer-

den nicht ausgeführt, so dass der Status der Ausgänge zwischen JMP(004) und JME(005) beibehalten wird. In Blockprogrammen werden die Befehle zwischen JMP(004) und JME(005) unabhängig vom Status der Ausführungsbedingung übersprungen.



Da alle Befehle zwischen JMP(004) und JME(005) übersprungen werden, wenn die Ausführungsbedingung für JMP(004) auf AUS gesetzt ist, wird die Zykluszeit um die Gesamtausführungszeit der übersprungenen Befehle reduziert. Im Gegensatz dazu findet eine NOP(000)-Verarbeitung für Befehle zwischen JMP0(515) und JME0(516) statt, so dass die Zykluszeit mit diesen Sprungbefehlen nicht so stark reduziert wird.

Die nachfolgende Tabelle bietet einen Vergleich der verschiedenen Sprungbefehle.

Eigenschaft	JMP(004) JME(005)	CJP(510) JME(005)	CJPN(511) JME(005)	JMP0(515) JME0(516)
Ausführungsbedingung für Sprung	AUS	EIN	AUS	AUS
Zulässige Anzahl	1.024 gesamt			Keine Begrenzung
Verarbeitung übersprungener Befehle	Nicht ausgeführt			NOP(000)-Verarbeitung
Ausführungszeit übersprungener Befehle	Keine			Identisch mit NOP(000)-Befehlen
Status übersprungener Ausgänge (Bits und Worte)	Bits und Worte behalten ihren vorherigen Status bei.			
Status von übersprungenen Zeitgebern in Betrieb	Zeitgeber in Betrieb setzen die Zeitmessung fort.			
Verarbeitung in Blockprogrammen	Stets Sprung	Sprung bei EIN	Sprung bei AUS	Nicht zulässig

Flags (JMP)

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N nicht im spezifizierten Bereich von 0000 bis 03FF liegt. (Siehe Hinweis.) EIN, wenn sich ein JMP(004) ohne ein JME(005) mit der selben Sprungnummer im Programm befindet. EIN, wenn sich ein JMP(004) ohne ein JME(005) mit der selben Sprungnummer in einer Task befindet. In allen anderen Fällen AUS.

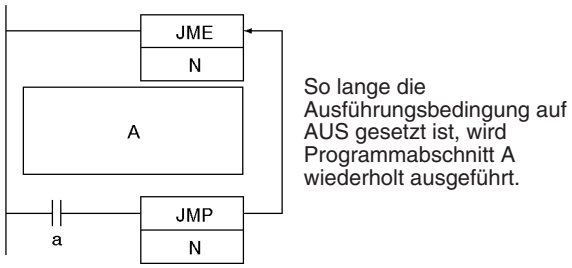
Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 liegt der gültige Bereich zwischen 0 und 255 (0000 bis 00FF hex).

Vorsichtsmaßnahmen

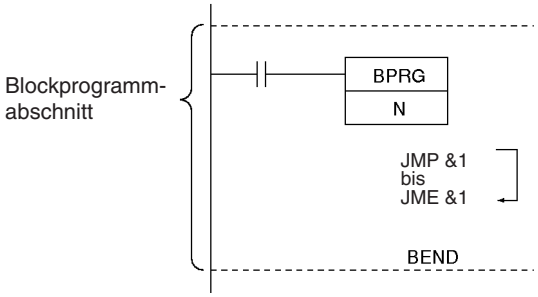
In übersprungenen Befehlen behalten alle Ausgänge (Bits und Worte) ihren vorherigen Status. Zeitgeber in Betrieb (TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552)) führen die Zeitmessung fort, da die Istwerte auch dann aktualisiert werden, wenn der Zeitgeberbefehl nicht ausgeführt wird.

Wenn zwei oder mehr JME(005)-Befehle mit der selben Sprungnummer vorhanden sind, ist nur der Befehl mit der niedrigsten Adresse gültig. Der JME(005) mit der höheren Programmadresse wird ignoriert.

Wenn der JME(005)-Befehl dem JMP(004)-Befehl im Programm vorangeht, werden die Befehle zwischen JME(005) und JMP(004) wiederholt so lange ausgeführt, wie die Ausführungsbedingung für JMP(004) auf AUS gesetzt ist. Ein Zykluszeit-zu-lang-Fehler tritt auf, wenn die Ausführungsbedingung nicht auf EIN gesetzt wird oder END(001) nicht innerhalb der maximalen Zykluszeit ausgeführt wird.



In Blockprogrammen werden die Befehle zwischen JMP(004) und JME(005) unabhängig vom Status der Ausführungsbedingung für JMP(004) stets übersprungen.



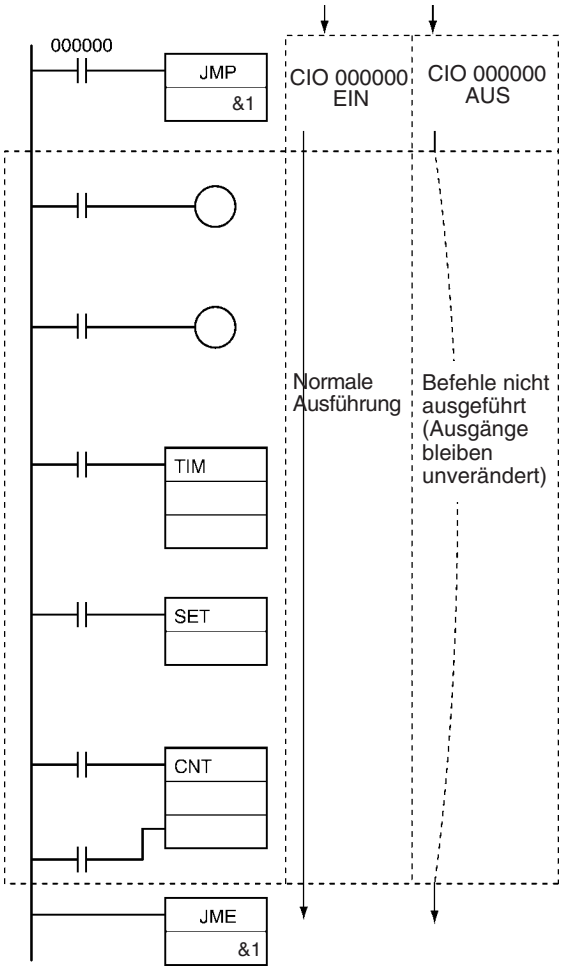
JMP(004)/JME(005)-Paare müssen in der selben Task enthalten sein, da Sprünge zwischen Tasks nicht zulässig sind. Ein Fehler tritt auf, wenn ein JME(005)-Befehl nicht in der selben Task programmiert ist wie sein zugehöriger JMP(004)-Befehl.

Die Funktion von DIFU(013), DIFD(014) und Befehlen mit Flankenauswertung hängt nicht nur vom Status der Ausführungsbedingung ab, wenn diese Befehle zwischen JMP(004) und JME(005) programmiert werden. Wenn DIFU(013), DIFD(014) oder ein Befehl mit Flankenauswertung unmittelbar nachdem die Ausführungsbedingung für JMP(004) auf EIN gesetzt wurde in einem übersprungenem Abschnitt ausgeführt wird, wird die Ausführungsbedingung für DIFU(013), DIFD(014) oder der Befehl mit Flankenauswertung mit der Ausführungsbedingung verglichen, die vor Ausführung des Sprungs gültig war (also bevor die Ausführungsbedingung für JMP(004) auf AUS gesetzt wurde).

Beispiele

Grundfunktion

Wenn im folgenden Beispiel CIO 000000 auf AUS gesetzt ist, werden die Befehle zwischen JMP(004) und JME(005) nicht ausgeführt und die Ausgänge behalten ihren vorherigen Status bei.
Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden die Befehle zwischen JME(005) und JME(005) normal ausgeführt.



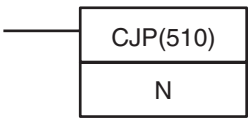
3-5-7 CONDITIONAL JUMP: CJP(510)/CJPN(511)

Funktion

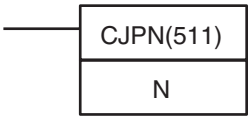
Die Funktion von CJP(510) ist im Prinzip die Umkehrung der Funktion von JMP(004). Wenn die Ausführungsbedingung für CJP(510) den Zustand EIN hat, springt die Programmausführung direkt zum ersten JME(005) mit derselben Sprungnummer im Programm. CJP(510) und JME(005) werden als Paar verwendet.

Die Funktion von CJPN(511) ist annähernd dieselbe wie die von JMP(004). Wenn die Ausführungsbedingung für CJP(510) den Zustand AUS hat, springt die Programmausführung direkt zum ersten JME(005) mit derselben Sprungnummer im Programm. CJPN(511) und JME(005) werden als Paar verwendet.

Kontaktplansymbole



N: Sprungnummer



N: Sprungnummer

Variationen

Variationen	Sprung, wenn auf EIN gesetzt / Kein Sprung, wenn auf AUS gesetzt	CJP(510)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Variationen	**Sprung, wenn auf AUS gesetzt / Kein Sprung, wenn auf EIN gesetzt**	CJPN(511)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	JME(005)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	Nicht zulässig	OK	OK

Operanden

N: Sprungnummer

Die Sprungnummer muss ein Wert von 0000 bis 03FF (0 bis 1.023 dezimal) sein.

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 muss die Sprungnummer im Bereich zwischen 0000 und 00FF hex bzw. &0 bis &255 dezimal liegen.

Operanden-Spezifikationen

Bereich	N		
	CJP(510)	CJPN(511)	JME(005)
CIO-Bereich	CIO 0000 bis CIO 6143		---
Arbeitsbereich nicht remanent	W000 bis W511		---
Arbeitsbereich remanent	H000 bis H511		---
Systembereich	A000 bis A959		---
Zeitgeberbereich	T0000 bis T4095		---
Zählerbereich	C0000 bis C4095		---
DM-Bereich	D00000 bis D32767		---
EM-Bereich ohne Bank	E00000 bis E32767		---
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		---
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		---
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		---
Konstanten	#0000 to #03FF (binär) bzw. &0 bis &1023 (siehe Hinweis)		#0000 to #03FF (binär) bzw. &0 bis &1023 (siehe Hinweis)
Datenregister	DR0 bis DR15		---
Indexregister	---		---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15		---

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 liegt der gültige Bereich zwischen #0000 und #00FF (binär) bzw. &0 bis &255 (dezimal).

Beschreibung

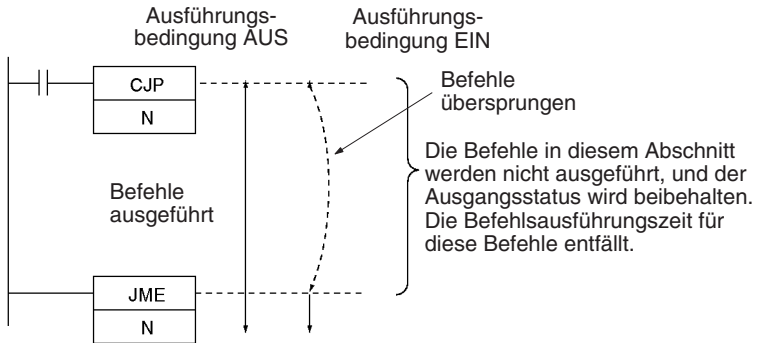
Die Funktionen von CJP(510) und CJPN(511) unterscheiden sich nur in der Ausführungsbedingung. CJP(510) springt zum ersten JME(005), wenn die Ausführungsbedingung auf EIN gesetzt ist und CJPN(511) springt zum ersten JME(005), wenn die Ausführungsbedingung auf AUS gesetzt ist.

Da die Sprungbefehle nicht ausgeführt werden, wird die Zykluszeit um die Gesamtausführungszeit der übersprungenen Befehle reduziert.

Funktion von CJP(510)

Wenn die Ausführungsbedingung für CJP(510) auf AUS gesetzt ist, wird kein Sprung ausgeführt und das Programm fortlaufend wie geschrieben ausgeführt.

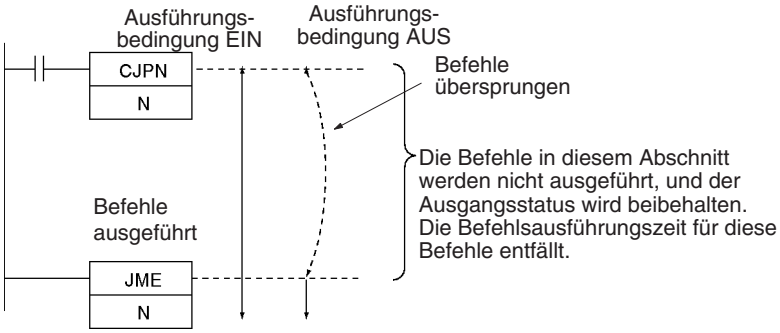
Wenn die Ausführungsbedingung für CJP(510) den Zustand EIN hat, springt die Programmausführung direkt zum ersten JME(005) mit derselben Sprungnummer im Programm.



Funktion von CJPN(511)

Wenn die Ausführungsbedingung für CJPN(511) auf EIN gesetzt ist, wird kein Sprung ausgeführt und das Programm fortlaufend wie geschrieben ausgeführt.

Wenn die Ausführungsbedingung für CJPN(511) den Zustand AUS hat, springt die Programmausführung direkt zum ersten JME(005) im Programm mit derselben Sprungnummer.



Flags

In der folgenden Tabelle sind die von CJP(510) und CJPN(511) betroffenen Flags aufgeführt.

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn kein JME(005) mit der selben Sprungnummer wie CJP(510) bzw. CJPN(511) vorhanden ist. (Siehe Hinweis.) EIN, wenn N nicht im spezifizierten Bereich von 0000 bis 03FF liegt. EIN, wenn sich ein CJP(510)- bzw. CJPN(511)-Befehl in einer Task ohne einen JME(005) mit der selben Sprungnummer befindet. In allen anderen Fällen AUS.

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 muss die Sprungnummer im Bereich zwischen 0 und 255 (0000 und 00FF hex) liegen.

Vorsichtsmaßnahmen

In übersprungenen Befehlen behalten alle Ausgänge (Bits und Worte) ihren vorherigen Status. Zeitgeber in Betrieb (TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540) und TMHHX(552)) führen die Zeitmessung fort, da die Istwerte auch dann aktualisiert werden, wenn der Zeitgeberbefehl nicht ausgeführt wird.

Wenn zwei oder mehr JME(005)-Befehle mit der selben Sprungnummer vorhanden sind, ist nur der Befehl mit der niedrigsten Adresse gültig. Der JME(005) mit der höheren Programmadresse wird ignoriert.

Wenn JME(005) dem Befehl CJP(510) bzw. CJPN(511) im Programm vorangeht, werden die dazwischen liegenden Befehle wiederholt ausgeführt, so lange die Ausführungsbedingung auf AUS (CJP(510)) bzw. EIN (CJPN(511)) gesetzt bleibt. Ein Zykluszeit-zu-lang-Fehler tritt auf, wenn der Sprung bei Änderung der Ausführungsbedingung durch Ausführung von END(001) innerhalb der maximalen Zykluszeit nicht abgeschlossen ist.

Die Befehle CJP(510) und CJPN(511) funktionieren in Blockprogrammen normal.

Wenn die Ausführungsbedingung für CJP(510) auf EIN gesetzt ist bzw. die Ausführungsbedingung für CJPN(511) auf AUS gesetzt ist, springt die Programmausführung ohne Ausführung von Befehlen zwischen CJP(510)/ CJPN(511) und JME direkt zum JME-Befehl. Für diese Befehle ist keine Ausführungszeit erforderlich und daher reduziert sich die Zykluszeit.

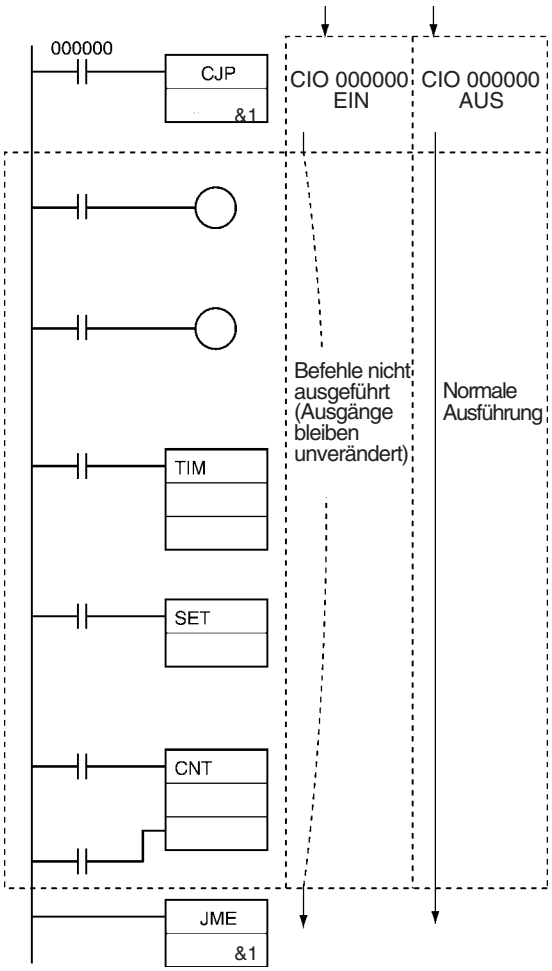
Wenn die Ausführungsbedingung für JMP0 auf AUS gesetzt ist, findet eine NOP-Verarbeitung zwischen JMP0 und JME0 statt, für die eine Ausführungszeit erforderlich ist. Aus diesem Grund wird die Zykluszeit nicht reduziert.

Wenn ein CJP(510)- oder CJPN(511)-Befehl in einer Task programmiert wird, muss ein JME(005)-Befehl mit derselben Sprungsnummer vorhanden sein, da Sprünge zwischen Tasks nicht zulässig sind. Ein Fehler tritt auf, wenn ein entsprechender JME(005)-Befehl nicht in derselben Task programmiert ist.

Die Funktion von DIFU(013), DIFD(014) und Befehlen mit Flankenausführung hängt nicht nur vom Status der Ausführungsbedingung ab, wenn diese in einem übersprungenen Programmabschnitt programmiert sind. Wenn DIFU(013), DIFD(014) oder ein Befehl mit Flankenauswertung unmittelbar nachdem die Ausführungsbedingung für CJP(510) auf AUS gesetzt wurde (EIN bei CJPN(511)) in einem übersprungenem Abschnitt ausgeführt wird, wird die Ausführungsbedingung für DIFU(013), DIFD(014) oder der Befehl mit Flankenauswertung mit der Ausführungsbedingung verglichen, die vor Ausführung des Sprungs gültig war.

Beispiel

Wenn im folgenden Beispiel CIO 000000 auf EIN gesetzt ist, werden die Befehle zwischen CJP(510) und JME(005) nicht ausgeführt und die Ausgänge behalten ihren vorherigen Status bei.
Wenn CIO 000000 im folgenden Beispiel auf AUS gesetzt ist, werden die Befehle zwischen CJP(510) und JME(005) normal ausgeführt.



Hinweis Bei CJPN(511) würde der EIN/AUS-Status von CIO 000000 invertiert.

3-5-8 MULTIPLE JUMP und JUMP END: JMP0(515) und JME0(516)

Funktion

Wenn die Ausführungsbedingung für JMP0(515) den Zustand AUS hat, werden alle Befehle zwischen JMP0(515) und dem nächsten JME0(516) im Programm als NOP(000) verarbeitet, also nicht verarbeitet. JMP0(515) und JME0(516) sind als Paare zu verwenden. Es gibt keine Begrenzung der Anzahl von Paaren, die im Programm verwendet werden können.

Kontaktplansymbole



Variationen

Variationen	Sprung, wenn auf AUS gesetzt / Kein Sprung, wenn auf EIN gesetzt	JMP0(515)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	JME0(516)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

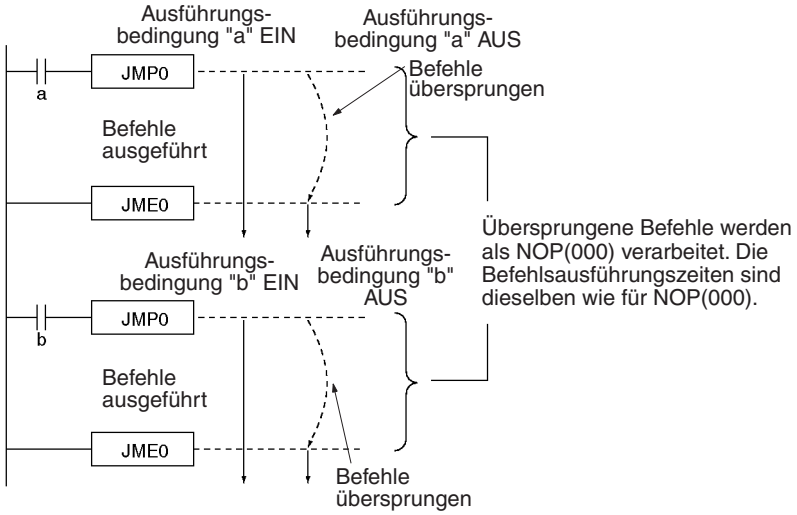
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	Nicht zulässig	OK	OK

Beschreibung

Wenn die Ausführungsbedingung für JMP0(515) auf EIN gesetzt ist, wird kein Sprung ausgeführt und das Programm fortlaufend wie geschrieben ausgeführt.

Wenn die Ausführungsbedingung für JMP0(515) den Zustand AUS hat, werden alle Befehle zwischen JMP0(515) und dem nächsten JME0(516) im Programm als NOP(000) verarbeitet, also nicht verarbeitet. Anders als JMP(004), CJP(510) und CJPN(511) verwendet JMP0(515) keine Sprungnummern, so dass diese Befehle an beliebiger Stelle im Programm platziert werden können.



Anders als bei den Befehlen JMP(004), CJP(510) und CJPN(511), die direkt zum ersten JME(005)-Befehl des Programms springen, werden alle Befehle zwischen JMP0(515) und JME0(516) als NOP(000) ausgeführt. Die Ausführungszeit der übersprungenen Befehle wird reduziert, entfällt jedoch nicht völlig. Die übersprungenen Befehle selbst werden nicht ausgeführt und deren Ausgänge (Bits und Worte) behalten ihren vorherigen Status bei.

Vorsichtsmaßnahmen

Im Programm können mehrere Paare der Befehle JMP0(515) und JME0(516) verwendet werden, aber die Paare können nicht verschachtelt werden.

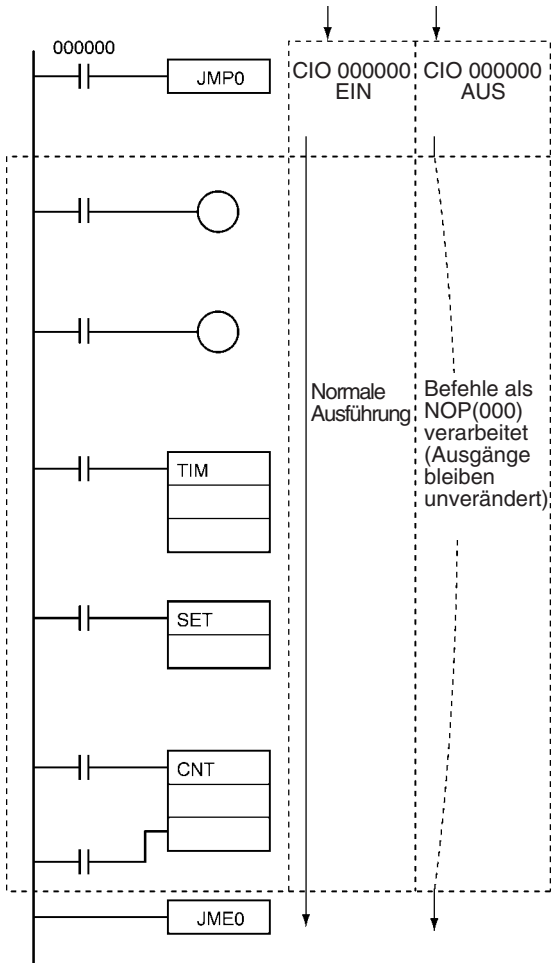
JMP0(515) und JME0(516) können nicht in Blockprogrammen verwendet werden.

Paare von JMP0(515) und JME0(516) müssen in der selben Task enthalten sein, da Sprünge zwischen Tasks nicht zulässig sind.

Die Funktion von DIFU(013), DIFD(014) und Befehlen mit Flankenauswertung hängt nicht nur vom Status der Ausführungsbedingung ab, wenn diese Befehle zwischen JMP0(515) und JME0(516) programmiert sind. Wenn DIFU(013), DIFD(014) oder ein Befehl mit Flankenauswertung unmittelbar nachdem die Ausführungsbedingung für JMP0(515) auf EIN gesetzt wurde in einem übersprungenem Abschnitt ausgeführt wird, wird die Ausführungsbedingung für DIFU(013), DIFD(014) oder der Befehl mit Flankenauswertung mit der Ausführungsbedingung verglichen, die vor Ausführung des Sprungs gültig war (also bevor die Ausführungsbedingung für JMP0(515) auf AUS gesetzt wurde).

Beispiel

Wenn im folgenden Beispiel CIO 000000 auf AUS gesetzt ist, werden die Befehle zwischen JMP0(515) und JME0(516) als NOP(000)-Befehle ausgeführt und die Ausgänge behalten ihren vorherigen Status bei.
Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden die Befehle zwischen JMP0(515) und JME0(516) normal ausgeführt.

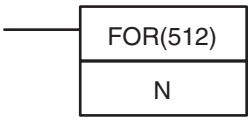


3-5-9 FOR-NEXT-Programmschleife: FOR(512)/NEXT(513)

Funktion

Die Befehle zwischen FOR(512) und NEXT(513) werden für die spezifizierte Anzahl von Ausführungen wiederholt. FOR(512) und NEXT(513) werden als Paar verwendet.

Kontaktplansymbole



N: Anzahl der Schleifen



Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	FOR(512)
	Ausführung in jedem Zyklus bei EIN-Bedingung	NEXT(513)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Operanden

N: Anzahl der Schleifen

Die Anzahl der Schleifen muss im Bereich zwischen 0000 und FFFF hex liegen (0 bis 65.535 dezimal).

Operanden-Spezifikationen

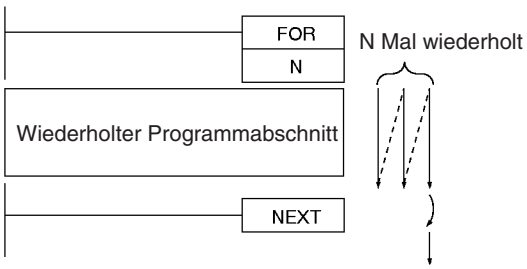
Bereich	N
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A000 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	#0000 bis #FFFF (binär) oder &0 bis &65.535
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

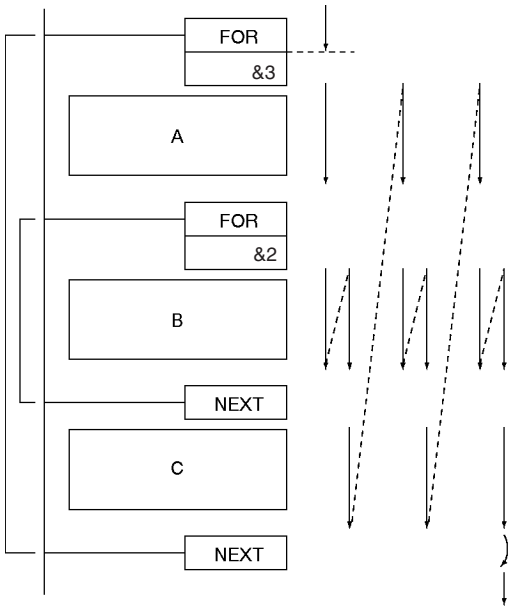
Die Befehle zwischen FOR(512) und NEXT(513) werden N Mal ausgeführt und anschließend wird die Programmausführung mit dem Befehl nach NEXT(513) fortgesetzt. Der Befehl BREAK(515) kann zur Unterbrechung der Schleife verwendet werden.

Wenn N auf 0 gesetzt ist, werden die Befehle zwischen FOR(512) und NEXT(513) als NOP(000)-Befehle verarbeitet.

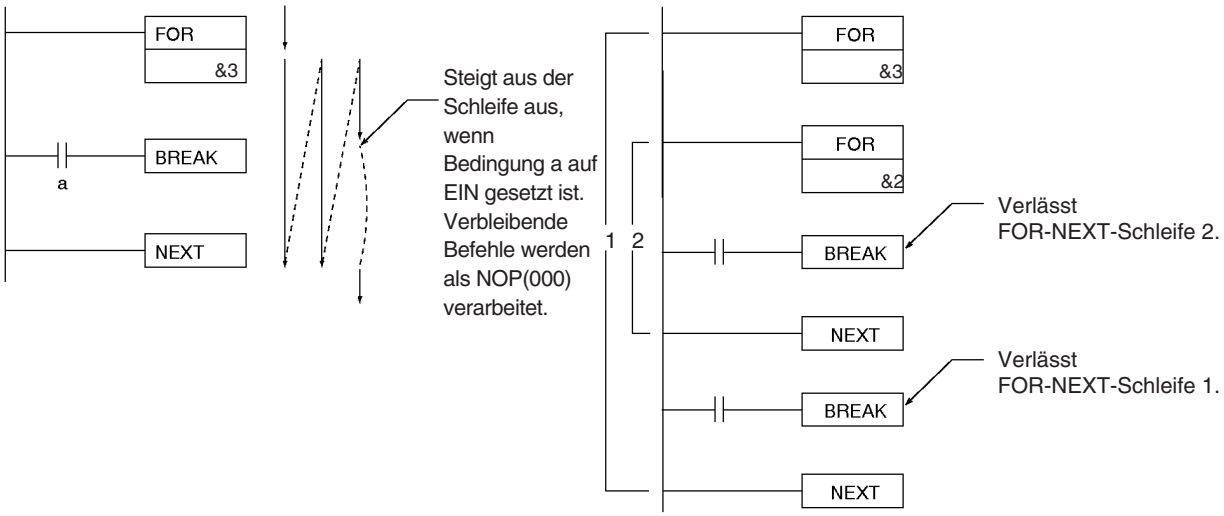
Schleifen können mit geringem Programmierungsaufwand zur Verarbeitung von Datentabellen verwendet werden.



FOR-NEXT-Schleifen können in bis zu 15 Ebenen verschachtelt werden. Im nachstehenden Beispiel werden die Programmabschnitte A, B und C wie folgt ausgeführt:
A → B → B → C, A → B → B → C und A → B → B → C



Verwenden Sie den Befehl BREAK(514), um mit dem Programm aus einer FOR-NEXT-Schleife auszusteigen. Für den Ausstieg aus geschachtelten Schleifen sind mehrere BREAK(514)-Befehle (Anzahl der verschachtelten Ebenen) erforderlich. Die nach Ausführung von BREAK(514) in der Schleife verbliebenen Befehle werden als NOP(000)-Befehle verarbeitet.



Alternative Schleifenmethoden

Ein Programmabschnitt kann auf zwei Arten wiederholt werden, bis eine bestimmte Ausführungsbedingung eingegeben wird.

1,2,3... 1. FOR-NEXT-Schleife mit BREAK

Starten Sie eine FOR-NEXT-Schleife mit maximal N Wiederholungen. Programmieren Sie BREAK(514) mit der gewünschten Ausführungsbedingung innerhalb der Schleife. Die Schleife endet vor N Wiederholungen, wenn die Ausführungsbedingung eingegeben wird.

2. JME(005)-JMP(004)-Schleife

Programmieren Sie eine Schleife mit JMP(005) vor JME(004). Die Befehle zwischen JME(005) und JMP(004) werden wiederholt ausgeführt, so lange die Ausführungsbedingung für JMP(004) auf AUS gesetzt ist. (Ein Zykluszeit-zu-lang-Fehler tritt auf, wenn die Ausführungsbedingung nicht auf EIN gesetzt wird oder END(001) nicht innerhalb der maximalen Zykluszeit ausgeführt wird.)

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn mehr als 15 Schleifen verschachtelt sind. In allen anderen Fällen AUS.
Gleich-Flag	=	AUS
Negativ-Flag	N	AUS

Vorsichtsmaßnahmen

Programmieren Sie FOR(512) und NEXT(513) in derselben Task. Die Ausführung wird nicht wiederholt, wenn sich diese Befehle nicht in derselben Task befinden.

Ein Sprungbefehl (z.B. JMP(004)) kann innerhalb einer FOR-NEXT-Schleife ausgeführt werden, kann aber keinen Sprung über die FOR-NEXT-Schleife hinaus ausführen.

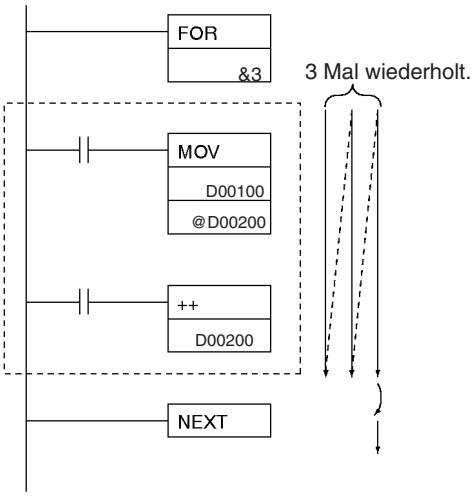
Folgende Befehle können nicht innerhalb von FOR-NEXT-Schleifen verwendet werden:

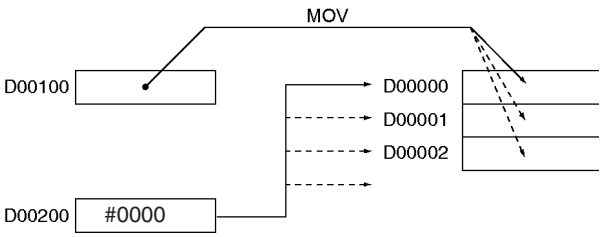
- Blockprogrammierungsbefehle
- MULTIPLE JUMP und JUMP END: JMP(515) und JME(516)
- STEP DEFINE und STEP START: STEP(008)/SNXT(009)

Hinweis Wenn sich eine Schleife in einem Zyklus wiederholt und ein Bit mit Flankenauswertung für die FOR-NEXT-Schleife verwendet wird, ist dieses Bit innerhalb der Schleife immer auf EIN oder immer auf AUS gesetzt.

Beispiel

Im folgenden Beispiel überträgt der Schleifen-Programmabschnitt den Inhalt von D00100 an die in D00200 angegebene Adresse und inkrementiert den Inhalt von D00200 dann um 1.





3-5-10 BREAK LOOP: BREAK(514)

Funktion

In einer FOR-NEXT-Schleife programmiert, um die Schleife bei einer bestimmten Ausführungsbedingung zu unterbrechen. Die verbliebenen Befehle in der Schleife werden als NOP(000)-Befehle verarbeitet.

Kontaktplansymbol



Variationen

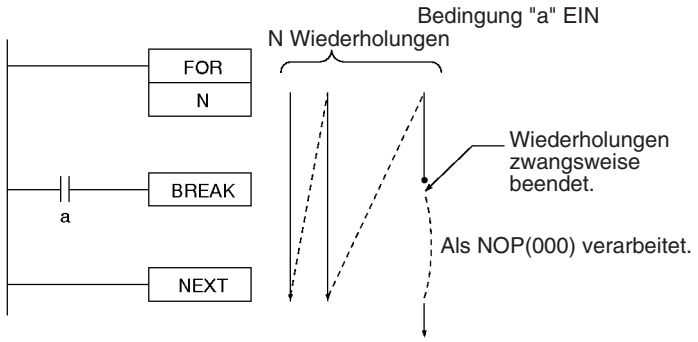
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	BREAK(514)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Beschreibung

Programmieren Sie BREAK(514) zwischen FOR(512) und NEXT(513), um die FOR-NEXT-Schleife bei Ausführung von BREAK(514) zu unterbrechen. Wenn BREAK(514) ausgeführt wird, wird der Rest der Befehle bis NEXT(513) als NOP(000) verarbeitet.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	AUS
Negativ-Flag	N	AUS

Vorsichtsmaßnahmen

Ein BREAK(514)-Befehl unterbricht nur eine Schleife, so dass zum Unterbrechen von verschachtelten Schleifen mehrere BREAK(514)-Befehle (Anzahl der verschachtelten Ebenen) erforderlich sind.

BREAK(514) kann nur in einer FOR-NEXT-Schleife verwendet werden.

3-6 Zeitgeber- und Zählerbefehle

In diesem Abschnitt werden die zur Definition und Handhabung von Zeitgebern und Zählern verwendeten Befehle beschrieben.

Befehl	AWL	Funktionscode	Seite
TIMER	TIM/TIMX	---/551	235
HIGH-SPEED TIMER	TIMH/TIMHX	015/551	240
ONE-MS TIMER	TMHH/TIMHHX	540/552	244
ACCUMULATIVE TIMER	TTIM/TTIMX	087/555	247
LONG TIMER	TIML/TIMLX	542/553	251
MULTI-OUTPUT TIMER	MTIM/MTIMX	543/554	254
COUNTER	CNT/CNTX	---/546	260
REVERSIBLE COUNTER	CNTR/CNTRX	012/548	263
RESET TIMER/COUNTER	CNR/CNRX	545/547	267

Aktualisierungsmethoden für den Zeitgeber-/Zähler-Istwert

■ Übersicht

Die von CS1 und CJ1 CPU-Baugruppen unterstützten Zählerbefehle verwenden sämtlich BCD-Daten und alle Sollwerte werden als BCD-Daten eingegeben. Die Aktualisierungsmethode kann für andere CPU-Baugruppen der CS- und CJ-Serie entweder auf BCD oder binär eingestellt werden (z.B. CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen; siehe Hinweise 1 und 2).

Die Verwendung von Binärdaten anstelle von BCD-Daten ermöglicht eine Vergrößerung des gültigen Sollwertbereichs von Zeitgebern und Zählern von 0 bis 9999 auf 0 bis 65535. Außerdem können mit anderen Befehlen berechnete Binärdaten direkt als Zeitgeber-/Zählersollwert verwendet werden. Die Aktualisierungsmethode ist auch bei indirektem Setzen eines Sollwerts gültig (also bei Verwendung des Inhalts eines Speicherworts). (Das bedeutet, dass die Inhalte der adressierten Worte entsprechend der festgelegten Aktualisierungsmethode entweder als BCD- oder Binärdaten verarbeitet werden.)

Detaillierte Informationen über Aktualisierungsmethoden finden Sie im *CS/CJ-Serie Programmierhandbuch*.

Hinweis

1. Bei CS1-H und CJ1-H CPU-Baugruppen, die vor dem 31. Mai 2002 hergestellt wurden, werden die Binärbefehle auf der Programmierkonsole mit dem AWL-Kürzel des äquivalenten Befehls für die BCD-Verarbeitung angezeigt. (z.B. wird TIMX0 &16 als TIM0 &16 angezeigt.) Der Befehl wird dennoch im Binärmodus ausgeführt.
2. Die Aktualisierungsmethode kann nur mit CX-Programmer ab Version 3.0 ausgewählt werden. Sie kann nicht mit Version 2.1 oder früher oder über eine Programmierkonsole ausgewählt werden.
3. Anwenderprogramme, die den binären Aktualisierungsmodus verwenden, können mit CX-Programmer Version 2.1 oder früher nicht gelesen werden. Sie können nur durch Wechsel in den BCD-Modus gelesen werden.

■ Anwendbare Befehle

Klassifizierung	Befehl	AWL	
		BCD	Binär
Zeitgeber- und Zähler-Befehle	TIMER	TIM	TIMX(550)
	HIGH-SPEED TIMER	TIMH(015)	TIMHX(551)
	ONE-MS TIMER	TMHH(540)	TMHHX(552)
	ACCUMULATIVE TIMER	TTIM(087)	TTIMX(555)
	LONG TIMER	TIML(542)	TIMLX(553)
	MULTI-OUTPUT TIMER	MTIM(543)	MTIMX(554)
	COUNTER	CNT	CNTX(546)
	REVERSIBLE COUNTER	CNTR(012)	CNTRX(548)
	RESET TIMER/COUNTER	CNR(545)	CNRX(547)
Blockprogrammierungsbefehle	TIMER WAIT	TIMW(813)	TIMWX(816)
	HIGH-SPEED TIMER WAIT	TMHW(815)	TMHWX(817)
	COUNTER WAIT	CNTW(814)	CNTWX(818)

Grundspezifikationen für Zeitgeber

In der folgenden Tabelle sind die Grundspezifikation für Zeitgeber zusammengestellt.

Eigenschaft	TIM/TIMX(550)	TIMH(015)/ TIMHX(551)	TMHH(540)/ TMHHX(552)	TTIM(087)/ TTIMX(555)	TIML(542)/ TIMLX(553)	MTIM(543)/ MTIMX(554)
Zeitmessmethode	Dekrementierung	Dekrementierung	Dekrementierung	Inkrementierung	Dekrementierung	Inkrementierung
Zeiteinheiten	0,1 s	0,01 s	0,001 s	0,1 s	0,1 s	0,1 s
Max. Sollwert	TIM: 999,9 s TIMX: 6.553,5 s	TIMH: 99,99 s TIMHX: 655,35 s	TMHH: 9,999 s TMHHX: 65,535 s	TTIM: 999,9 s TTIMX: 6.553,5 s	TIML: 115 Tage TIMLX: 49.710 Tage	MTIM: 999,9 s MTIMX: 6.553,5 s
Ausgänge/ Befehl	1	1	1	1	1	8

Eigenschaft		TIM/TIMX(550)	TIMH(015)/ TIMHX(551)	TMHH(540)/ TMHHX(552)	TTIM(087)/ TTIMX(555)	TIML(542)/ TIMLX(553)	MTIM(543)/ MTIMX(554)
Zeitgeber-nummern		Verwendet	Verwendet	Verwendet	Verwendet	Nicht verwendet	Nicht verwendet
Fertig-Bit-Aktualisierung		Bei Ausführung	Bei Ausführung	Durch Interrupt alle 1 ms	Bei Ausführung	Bei Ausführung	Bei Ausführung
Zeitgebersollwert-Aktualisierung		Siehe Hinweis 1	Siehe Hinweis 2	Alle 1 ms	Bei Ausführung	Bei Ausführung	Bei Ausführung
Wert nach Rücksetzung	Fertig-Bits	AUS	AUS	AUS	AUS	AUS	AUS
	Istwerte	Sollwert	Sollwert	Sollwert	0	Sollwert	0

Hinweis

1. Die Aktualisierung der Istwerte von TIM erfolgt bei der Ausführung, am Ende der Programmausführung in jedem Zyklus oder alle 80 ms durch Interrupt, wenn die Zykluszeit 80 ms überschreitet.
2. Die Aktualisierung der Istwerte von TIMH(015)/TIMHX(551) erfolgt bei Ausführung, am Ende der Programmausführung in jedem Zyklus und alle 10 ms durch Interrupt.

Funktionsweise der Zeitgeber

In der folgenden Tabelle sind die Auswirkungen von Betriebs- und Programmierbedingungen auf die Funktion von Zeitgebern aufgeführt.

Eigenschaft		TIM/ TIMX(550)	TIMH(015)/ TIMHX(551)	TMHH(540)/ TMHHX(552)	TTIM(087)/ TTIMX(555)	TIML(542)/ TIMLX(553)	MTIM(543)/ MTIMX(554)
Wechsel der Betriebsart		Istwert = 0 Fertig-Bit = AUS				---	---
Spannungsunterbrechung/Rücksetzung		Istwert = 0 Fertig-Bit = AUS				---	---
Ausführung von CNR(545)/CNRX(547)		Binär: Istwert = FFFF, Fertig-Bit = AUS BCD: Istwert = FFFF oder 9999, Fertig-Bit = AUS				Nicht zutreffend	Nicht zutreffend
Funktion in übersprungenem Programmabschnitt (JMP(004)-JME(005))		Zeitgeber in Betrieb setzen die Zeitmessung fort.			Zeitgeberstatus wird beibehalten.		
Funktion in verriegeltem Programmabschnitt (IL(002)-ILC(003))		Istwert = Sollwert Fertig-Bit = AUS			Zeitgeberstatus wird beibehalten.	Istwert = Sollwert Fertig-Bit = AUS	Zeitgeberstatus wird beibehalten.
Zwangssetzung	Fertig-Bits	EIN				---	---
	Istwerte	Auf 0 gesetzt.				---	---
Zwangsrücksetzung	Fertig-Bits	AUS				---	---
	Istwerte	Auf Sollwert zurückgesetzt.			Auf 0 gesetzt.	---	---

3-6-1 TIMER: TIM/TIMX(550)

Funktion

TIM oder TIMX(550) ist ein dekrementierender Zeitgeber mit Schritten von 0,1 s. Der Einstellbereich für den Sollwert erstreckt sich bei TIM von 0 bis 999,9 s und bei TIMX(550) von 0 bis 6.553,5 s. Die Zeitgebergenauigkeit beträgt 0 bis 0,01 s.

Hinweis Die Zeitgebergenauigkeit bei CS1D CPU-Baugruppen beträgt 10 ms + die Zykluszeit.

Kontaktplansymbol

Istwert-Aktualisierungsmethode	Symbol	Operanden
BCD	TIM N S **N**: Zeitgebernummer **S**: Sollwert	N: 0000 bis 4095 (dezimal) S: #0000 bis #9999 (BCD)
Binär	TIMX(550) N S **N**: Zeitgebernummer **S**: Sollwert	N: 00000 bis 4095 (dezimal) S: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TIM/TIMX(550)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

N: Zeitgebernummer

Die Zeitgebernummer muss zwischen 0000 und 4095 (dezimal) liegen.

S: Sollwert

Der Sollwert muss zwischen #0000 und 9999 (BCD) liegen.
(Bei Einstellung des Sollwerts auf #0000 wird das Fertig-Bit auf EIN gesetzt, wenn TIM/TIMX(550) ausgeführt wird).

Operanden-Spezifikationen

Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A959
Zeitgeberbereich	0000 bis 4095 (dezimal)	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)

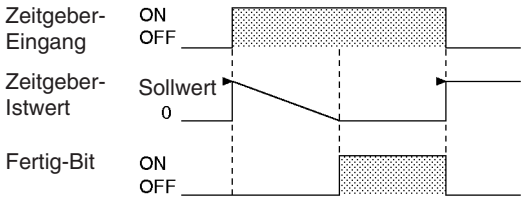
Bereich	N	S
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_032767 (n = 0 bis C)
Konstanten	---	BCD: #0000 bis 9999 (BCD) "&" kann nicht verwendet werden. Binär: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)
Datenregister	---	DR0 bis DR15
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	

Beschreibung

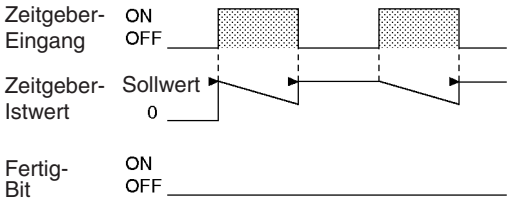
Wenn der Zeitgebereingang auf AUS gesetzt ist, wird der durch N spezifizierte Zeitgeber zurückgesetzt, d.h., der Istwert des Zeitgebers wird auf den Sollwert zurückgesetzt und das Fertig-Bit auf AUS gesetzt.

Wenn der Zeitgebereingang von AUS zu EIN wechselt, startet TIM/TIMX(550) mit der Dekrementierung des Istwerts. Der Istwert wird stetig verringert, solange der Zeitgebereingang auf EIN gesetzt bleibt. Das Zeitgeber-Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 erreicht.

Der Status von Zeitgeberistwert und Fertig-Bit wird nach Ablauf des Zeitgebers beibehalten. Um den Zeitgeber neu zu starten, muss der Zeitgebereingang auf AUS und dann wieder auf EIN gesetzt werden oder der Zeitgeberistwert muss auf einen Wert größer Null geändert werden (z. B. durch MOV(021)).



Das folgende Zeitablaufdiagramm zeigt das Verhalten des Zeitgeberistwerts sowie des Fertig-Bits, wenn der Zeitgebereingang vor Ablauf des Zeitgebers auf AUS gesetzt wird.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N indirekt über ein Indexregister adressiert wird, aber die Adresse im Indexregister nicht die Istwertadresse eines Zeitgebers ist. EIN, falls S im BCD-Modus keine BCD-Daten enthält. In allen anderen Fällen AUS.
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese auf AUS gesetzt.
In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Vorsichtsmaßnahmen

Zeitgebernummern werden von den Befehlen TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TIMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) und TMHWX(817) gemeinsam genutzt. Wenn zwei Zeitgeber dieselbe Zeitgebernummer nutzen, sie aber nicht simultan verwendet werden, wird bei Prüfung des Programms ein Duplizierungsfehler erzeugt, der Zeitgeber arbeitet jedoch normal. Zeitgeber, die dieselbe Zeitgebernummer nutzen, arbeiten nicht einwandfrei, wenn sie simultan verwendet werden.

Mit den Zeitgebernummern 2048 bis 4095 erstellte Zeitgeber arbeiten nicht einwandfrei, wenn die Zykluszeit der CPU-Baugruppe 80 ms überschreitet. Verwenden Sei die Zeitgebernummern 0000 bis 2047, wenn die Zykluszeit länger als 80 ms ist.

Der Istwert von Zeitgebern, die mit Zeitgebernummern von 0000 bis 2047 programmiert wurden, wird auch aktualisiert, wenn der Zeitgeber auf Standby gesetzt ist. Der Istwert von Zeitgebern, die mit Zeitgebernummern von 2048 bis 4095 programmiert wurden, wird gehalten, wenn der Zeitgeber auf Standby gesetzt ist.

In den folgenden Fällen werden Zeitgeber zurückgesetzt oder pausieren. (Bei Rücksetzung eines Zeitgebers wird dessen Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit auf EIN gesetzt.)

Beschreibung	Istwert	Fertig-Bit
Wechsel der Betriebsart von RUN oder MONITOR zu PROGRAM oder umgekehrt.[1]	0000	AUS
Spannungsversorgung unterbrochen und zurückgesetzt[2]	0000	AUS
Ausführung von CNR(545)/CNRX(547), die RESET TIMER/COUNTER-Befehle[3]	BCD: 9999 Binär: FFFF	AUS
Funktion in verriegeltem Programmabschnitt (IL(002)–ILC(003))	Auf Sollwert zurückgesetzt.	AUS
Funktion in übersprungenem Programmabschnitt (JMP(004)–JME(005))	Istwert wird weiter dekrementiert.	Vorheriger Status bleibt erhalten.

Hinweis

1. Wenn das E/A-Haltebit (A50012) auf EIN gesetzt wurde, wird der Status der Zeitgeber-Fertig-Bits und Istwerte bei Wechsel der Betriebsart beibehalten.
2. Wenn das E/A-Haltebit (A50012) auf EIN gesetzt wurde und der Status des E/A-Haltebits im SPS-Setup geschützt ist, wird der Status des Zeitgeber-Fertig-Bits sowie der Istwerte auch bei Unterbrechung der Spannungsversorgung beibehalten.
3. Der Istwert wird bei Ausführung von TIM/TIMX(550) auf den Sollwert gesetzt.

Wenn sich TIM/TIMX(550) in einem Programmabschnitt zwischen IL(002) und ILC(003) befindet und der Programmabschnitt verriegelt ist, wird der Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit wird auf AUS gesetzt.

Wenn sich ein TIM/TIMX(550)-Zeitgeber mit einer Zeitgebernummer zwischen 0000 und 2047 in einem übersprungenen Programmabschnitt (JMP(004), CJMP(510), CJPN(511), JME(005)) befindet, wird die Zeitmessung und Änderung des Zeitgeber-Istwerts fortgesetzt. (Siehe Hinweis.) Der übersprungene TIM/TIMX(550)-Befehl wird nicht ausgeführt, aber der Istwert wird nach Ausführung aller Tasks in jedem Zyklus aktualisiert.

Hinweis Bei CS1D CPU-Baugruppen wird der Istwert im oben genannten Fall nicht aktualisiert.

Wenn ein TIM/TIMX(550)-Zeitgeber zwangsweise gesetzt wird, wird sein Fertig-Bit auf EIN und sein Istwert auf 0000 gesetzt. Wenn ein TIM/TIMX(550)-Zeitgeber zwangsweise zurückgesetzt wird, wird sein Fertig-Bit auf AUS gesetzt und sein Istwert auf den Sollwert zurückgesetzt.

Die Funktion von Gleich-Flag (=) und Negativ-Flag (N) hängt vom Model der CPU-Baugruppe ab. Detaillierte Informationen hierzu finden Sie im Abschnitt *Flags* weiter oben.

Das Zeitgeber-Fertig-Bit wird nur bei Ausführung von TIM/TIMX(550) aktualisiert, so dass für das Fertig-Bit eine Verzögerung von bis zu einem Zyklus auftreten kann, bis es nach Ablauf des Zeitgebers auf EIN gesetzt wird.

Wenn Online-Bearbeitung zur Konvertierung eines Zeitgebers zu einer anderen Art von Zeitgeber mit der selben Zeitgebernummer verwendet wird (wie z.B. TIM/TIMX(550) ↔ TIMH(015)/TIMHX(551) oder TIM/TIMX(550) ↔ TMHH(540)/TMHHX(552)), muss das Fertig-Bit zurückgesetzt werden. Wenn das Fertig-Bit nicht zurückgesetzt wurde, funktioniert der Zeitgeber nicht ordnungsgemäß.

Istwert und Fertig-Bit eines TIM/TIMX(550)-Befehls können in Abhängigkeit von der verwendeten Zeitgebernummer auf die folgende Weise aktualisiert werden.

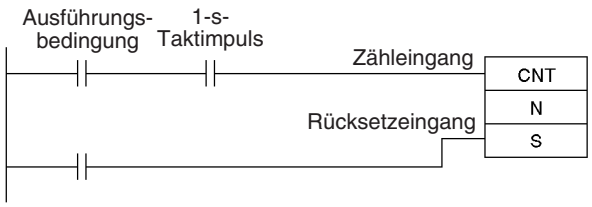
Mit Zeitgebernummern 0000 bis 2047 erstellte Zeitgeber

Ausführung von TIM/TIMX(550)	Der Istwert wird bei jeder Ausführung von TIM/TIMX(550) aktualisiert. Das Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 beträgt. Das Fertig-Bit wird auf AUS gesetzt, wenn der Istwert nicht 0000 beträgt.
Nach Ausführung aller Tasks	Der Istwert wird auch in jedem Zyklus am Ende der Programmausführung aktualisiert.
Aktualisierung in 80-ms-Intervall	Wenn die Zykluszeit 80 ms überschreitet, wird der Zeitgeber-Istwert alle 80 ms aktualisiert.

Mit den Zeitgebernummern 2048 bis 4095 erstellte Zeitgeber

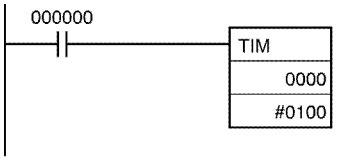
Ausführung von TIM	Der Istwert wird bei jeder Ausführung von TIM aktualisiert. Das Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 beträgt. Das Fertig-Bit wird auf AUS gesetzt, wenn der Istwert nicht 0000 beträgt.

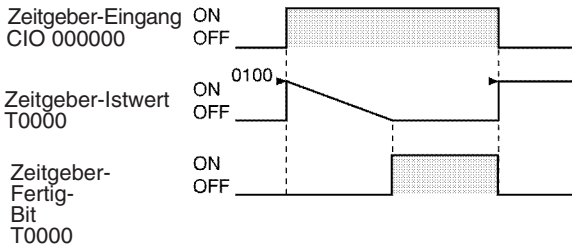
Zeitgeber werden durch eine Unterbrechung der Spannungsversorgung zurückgesetzt (Istwert = Sollwert, Fertig-Bit AUS), es sei denn, das E/A-Haltebit (A50012) ist auf EIN gesetzt und das Bit wird im SPS-Setup geschützt. Es ist auch möglich, ein Taktimpuls-Bit und einen Zählerbefehl zur Programmierung eines Zeitgebers zu verwenden, der seinen Istwert im Falle einer Spannungsunterbrechung beibehält (siehe folgende Abbildung).



Beispiel

Wenn ein Zeitgebereingang CIO 000000 im folgenden Beispiel von AUS zu EIN wechselt, beginnt das Abwärtszählen des Zeitgeber-Istwert vom Sollwert aus. Das Zeitgeber-Fertig-Bit T0000 wird auf EIN gesetzt, wenn der Istwert 0000 erreicht.
Bei Rücksetzung von CIO 000000 auf AUS wird der Zeitgeber-Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit auf AUS gesetzt.





3-6-2 HIGH-SPEED TIMER: TIMH(015)/TIMHX(551)

Funktion

TIMH(015)/TIMHX(551) funktioniert als dekrementierender Zeitgeber mit 10-ms-Schritten. Der Einstellbereich für den Sollwert erstreckt sich von 0 bis 99,99 s bei TIMH(015) bzw. von 0 bis 655,35 s bei TIMHX(551). Die Zeitgebergenauigkeit beträgt 0 bis 0,01 s.

Hinweis Die Zeitgebergenauigkeit bei CS1D CPU-Baugruppen beträgt 10 ms + die Zykluszeit.

Kontaktplansymbol

Istwert-Aktualisierungsmethode	Symbol	Operanden
BCD	TIMH(015) N — **N**: Zeitgebernummer S — **S**: Sollwert	N: 0000 bis 4095 (dezimal) S: #0000 bis #9999 (BCD)
Binär	TIMHX(551) N — **N**: Zeitgebernummer S — **S**: Sollwert	N: 00000 bis 4095 (dezimal) S: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TIMH(015)/ TIMHX(551)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

N: Zeitgebernummer

Die Zeitgebernummer muss zwischen 0000 und 4095 (dezimal) liegen.

S: Sollwert

Der Sollwert muss im BCD-Modus zwischen #0000 und 9999 liegen.

Operanden-Spezifikationen

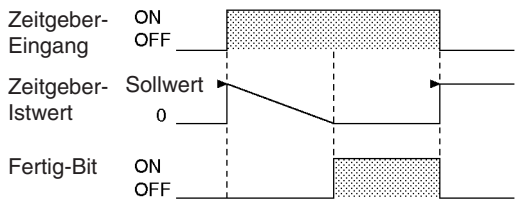
Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A959
Zeitgeberbereich	0000 bis 4095 (dezimal)	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---	BCD: #0000 bis 9999 (BCD) "&" kann nicht verwendet werden. Binär: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)
Datenregister	---	DR0 bis DR15
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	

Beschreibung

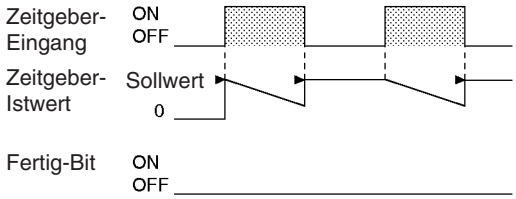
Wenn der Zeitgebereingang auf AUS gesetzt ist, wird der durch N spezifizierte Zeitgeber zurückgesetzt, d.h., der Istwert des Zeitgebers wird auf den Sollwert zurückgesetzt und das Fertig-Bit auf AUS gesetzt.

Wenn der Zeitgebereingang von AUS zu EIN wechselt, startet TIMH(015)/TIMHX(551) mit der Dekrementierung des Istwerts. Der Istwert wird stetig verringert, solange der Zeitgebereingang auf EIN gesetzt bleibt. Das Zeitgeber-Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 erreicht.

Der Status von Zeitgeberistwert und Fertig-Bit wird nach Ablauf des Zeitgebers beibehalten. Um den Zeitgeber neu zu starten, muss der Zeitgebereingang auf AUS und dann wieder auf EIN gesetzt werden oder der Zeitgeberistwert muss auf einen Wert größer Null geändert werden (z. B. durch MOV(021)).



Das folgende Zeitablaufdiagramm zeigt das Verhalten des Zeitgeberistwerts sowie des Fertig-Bits, wenn der Zeitgebereingang vor Ablauf des Zeitgebers auf AUS gesetzt wird.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N indirekt über ein Indexregister adressiert wird, aber die Adresse im Indexregister nicht die Istwertadresse eines Zeitgebers ist. EIN, falls S im BCD-Modus keine BCD-Daten enthält. In allen anderen Fällen AUS.
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese auf AUS gesetzt.
In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Vorsichtsmaßnahmen

Zeitgebernummern werden von den Befehlen TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TIMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) und TMHWX(817) gemeinsam genutzt. Wenn zwei Zeitgeber dieselbe Zeitgebernummer nutzen, sie aber nicht simultan verwendet werden, wird bei Prüfung des Programms ein Duplizierungsfehler erzeugt, der Zeitgeber arbeitet jedoch normal. Zeitgeber, die dieselbe Zeitgebernummer nutzen, arbeiten nicht einwandfrei, wenn sie simultan verwendet werden.

Mit den Zeitgebernummern 2048 bis 4095 erstellte Zeitgeber arbeiten nicht einwandfrei, wenn die Zykluszeit der CPU-Baugruppe 80 ms überschreitet. Verwenden Sei die Zeitgebernummern 0000 bis 2047, wenn die Zykluszeit länger als 80 ms ist.

Mit Zeitgebernummern 0000 bis 0255 erstellte TIMH(015)/TIMHX(551)-Zeitgeber werden alle 10 ms aktualisiert. Verwenden Sie diese Zeitgebernummern, wenn im Anwenderprogramm auf den Istwert Bezug genommen wird.

Der Istwert von Zeitgebern, die mit Zeitgebernummern von 0000 bis 2047 programmiert wurden, wird auch aktualisiert, wenn der Zeitgeber auf Standby gesetzt ist. Der Istwert von Zeitgebern, die mit Zeitgebernummern von 2048 bis 4095 programmiert wurden, wird gehalten, wenn der Zeitgeber auf Standby gesetzt ist.

Die Funktion von Gleich-Flag (=) und Negativ-Flag (N) hängt vom Model der CPU-Baugruppe ab. Detaillierte Informationen hierzu finden Sie im Abschnitt *Flags* weiter oben.

Die Fertig-Bits für TIMH(015)/TIMHX(551)-Zeitgeber werden bei Ausführung des Befehls aktualisiert. (Diese Funktion weicht von der von SPS der CV-Serie und CVM1-SPS ab.)

In den folgenden Fällen werden Zeitgeber zurückgesetzt oder pausieren. (Bei Rücksetzung eines Zeitgebers wird dessen Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit auf EIN gesetzt.)

Beschreibung	Istwert	Fertig-Bit
Wechsel der Betriebsart von RUN oder MONITOR zu PROGRAM oder umgekehrt.[1]	0000	AUS
Spannungsversorgung unterbrochen und zurückgesetzt[2]	0000	AUS
Ausführung von CNR(545)/CNRX(547), die RESET TIMER/COUNTER-Befehle[3]	BCD: 9999 Binär: FFFF	AUS

Beschreibung	Istwert	Fertig-Bit
Funktion in verriegeltem Programmabschnitt (IL(002)–ILC(003))	Auf Sollwert zurückgesetzt.	AUS
Funktion in übersprungenem Programmabschnitt (JMP(004)–JME(005))	Istwert wird weiter dekrementiert.	Vorheriger Status bleibt erhalten.

Hinweis

1. Wenn das E/A-Haltebit (A50012) auf EIN gesetzt wurde, wird der Status der Zeitgeber-Fertig-Bits und Istwerte bei Wechsel der Betriebsart beibehalten.
2. Wenn das E/A-Haltebit (A50012) auf EIN gesetzt wurde und der Status des E/A-Haltebits im SPS-Setup geschützt ist, wird der Status des Zeitgeber-Fertig-Bits sowie der Istwerte auch bei Unterbrechung der Spannungsversorgung beibehalten.
3. Der Istwert wird bei Ausführung von TIMH(015)/TIMHX(551) auf den Sollwert gesetzt.

Wenn sich ein TIMH(015)/TIMHX(551)-Zeitgeber mit einer Zeitgebernummer zwischen 0000 und 2047 in einem übersprungenen Programmabschnitt (JMP(004), CJMP(510), CJPN(511), JME(005)) befindet, wird die Zeitmessung und Änderung des Zeitgeber-Istwerts fortgesetzt. (Siehe Hinweis.) (Der übersprungene TIMH(015)/TIMHX(551)-Befehl wird nicht ausgeführt, aber der Istwert wird alle 10 ms und in jedem Zyklus nach Ausführung aller Tasks aktualisiert.)

Hinweis Bei CS1D CPU-Baugruppen wird der Istwert im oben genannten Fall nicht aktualisiert.

Wenn sich TIMH(015)/TIMHX(551) in einem Programmabschnitt zwischen IL(002) und ILC(003) befindet und der Programmabschnitt verriegelt ist, wird der Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit wird auf AUS gesetzt.

Wenn ein TIMH(015)/TIMHX(551)-Zeitgeber zwangsweise gesetzt wird, wird sein Fertig-Bit auf EIN und sein Istwert auf 0000 gesetzt. Wenn ein TIMH(015)/TIMHX(551)-Zeitgeber zwangsweise zurückgesetzt wird, wird sein Fertig-Bit auf AUS gesetzt und sein Istwert auf den Sollwert zurückgesetzt.

Die Funktion von Gleich-Flag (=) und Negativ-Flag (N) hängt vom Model der CPU-Baugruppe ab. Ausführliche Informationen hierzu finden Sie im Abschnitt *Flags*.

Das Zeitgeber-Fertig-Bit wird nur bei Ausführung von TIMH(015)/TIMHX(551) aktualisiert, so dass für das Fertig-Bit eine Verzögerung von bis zu einem Zyklus auftreten kann, bis es nach Ablauf des Zeitgebers auf EIN gesetzt wird.

Wenn Online-Bearbeitung zur Konvertierung eines Zeitgebers zu einer anderen Art von Zeitgeber mit derselben Zeitgebernummer verwendet wird (wie z.B. TIMH(015)/TIMHX(551) ↔ TIM/TIMX(550) oder TIMH(015)/TIMHX(551) ↔ TMHH(540)/TMHHX(552)), muss das Fertig-Bit zurückgesetzt werden. Wenn das Fertig-Bit nicht zurückgesetzt wurde, funktioniert der Zeitgeber nicht ordnungsgemäß.

Istwert und Fertig-Bit eines TIMH(015)/TIMHX(551)-Befehls können in Abhängigkeit von der verwendeten Zeitgebernummer auf die folgende Weise aktualisiert werden.

Mit den Zeitgebernummern 0000 bis 0255 erstellte Zeitgeber

Ausführung von TIMH(015)/ TIMHX(551)	Das Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 beträgt. Das Fertig-Bit wird auf AUS gesetzt, wenn der Istwert nicht 0000 beträgt.
Aktualisierung in 10-ms-Intervall	Der Istwert des Zeitgebers wird alle 10 ms aktualisiert.

Mit den Zeitgebernummern 0265 bis 2047 erstellte Zeitgeber

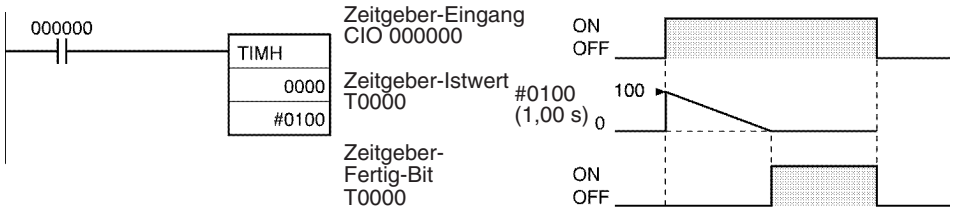
Ausführung von TIMH(015)/ TIMHX(551)	Der Istwert wird bei jeder Ausführung von TIMH(015)/TIMHX(551) aktualisiert. Das Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 beträgt. Das Fertig-Bit wird auf AUS gesetzt, wenn der Istwert nicht 0000 beträgt.
Nach Ausführung aller Tasks	Der Istwert wird auch in jedem Zyklus am Ende der Programmausführung aktualisiert.
Aktualisierung in 80-ms-Intervall	Wenn die Zykluszeit 80 ms überschreitet, wird der Zeitgeber-Istwert alle 80 ms aktualisiert.

Mit den Zeitgebernummern 2048 bis 4095 erstellte Zeitgeber

Ausführung von TIMH(015)/ TIMHX(551)	Der Istwert wird bei jeder Ausführung von TIMH(015) aktualisiert. Das Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 beträgt. Das Fertig-Bit wird auf AUS gesetzt, wenn der Istwert nicht 0000 beträgt.

Beispiel

Wenn ein Zeitgebereingang CIO 000000 im folgenden Beispiel von AUS zu EIN wechselt, beginnt das Abwärtszählen des Zeitgeber-Istwert vom Sollwert (#0064 = 100 = 1,00 s) aus. Das Zeitgeber-Fertig-Bit T0000 wird auf EIN gesetzt, wenn der Istwert 0000 erreicht.
Bei Rücksetzung von CIO 000000 auf AUS wird der Zeitgeber-Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit auf AUS gesetzt.



3-6-3 ONE-MS TIMER: TMHH(540)/TMHHX(552)

Funktion

TMHH(540)/TMHHX(552) funktioniert als dekrementierender Zeitgeber mit 1-ms-Schritten. Der Einstellbereich für den Sollwert erstreckt sich von 0 bis 9,999 s bei TMHH(540) bzw. von 0 bis 65,535 s bei TIMHHX(552). Die Zeitgebergenauigkeit beträgt -0,001 bis 0 s.

Hinweis Die Zeitgebergenauigkeit bei CS1D CPU-Baugruppen beträgt 10 ms + die Zykluszeit.

Kontaktplansymbol

Istwert-Aktualisierungsmethode	Symbol	Operanden
BCD	TMHH(540) N — **N**: Zeitgebernummer S — **S**: Sollwert	N: 0000 bis 15 (dezimal) S: #0000 bis #9999 (BCD)
Binär	TMHHX(552) N — **N**: Zeitgebernummer S — **S**: Sollwert	N: 00000 bis 15 (dezimal) S: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	TMHH(540)/ TMHHX(552)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

N: Zeitgebernummer

Die Zeitgebernummer muss zwischen 0000 und 0015 (dezimal) liegen.

S: Sollwert

Der Sollwert muss zwischen #0000 und 9999 (BCD) liegen.

Operanden-Spezifikationen

Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A959
Zeitgeberbereich	0000 bis 0015 (dezimal)	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binär-adressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---	BCD: #0000 bis 9999 (BCD) "&" kann nicht verwendet werden. Binär: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)
Datenregister	---	DR0 bis DR15
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	

Beschreibung

Wenn der Zeitgebereingang auf AUS gesetzt ist, wird der durch N spezifizierte Zeitgeber zurückgesetzt, d.h., der Istwert des Zeitgebers wird auf den Sollwert zurückgesetzt und das Fertig-Bit auf AUS gesetzt.

Wenn der Zeitgebereingang von AUS zu EIN wechselt, startet TMHH(540)/ TMHHX(552) mit der Dekrementierung des Istwerts. Der Istwert wird stetig verringert, solange der Zeitgebereingang auf EIN gesetzt bleibt. Das Zeitgeber-Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 erreicht.

Der Status von Zeitgeberistwert und Fertig-Bit wird nach Ablauf des Zeitgebers beibehalten. Um den Zeitgeber neu zu starten, muss der Zeitgebereingang auf AUS und dann wieder auf EIN gesetzt werden oder der Zeitgeberistwert muss auf einen Wert größer Null geändert werden (z.B. durch MOV(021)).

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N indirekt über ein Indexregister adressiert wird, aber die Adresse im Indexregister nicht die Istwertadresse eines Zeitgebers ist. EIN, falls S im BCD-Modus keine BCD-Daten enthält. In allen anderen Fällen AUS.
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese auf AUS gesetzt.
In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Vorsichtsmaßnahmen

Zeitgebernummern werden von den Befehlen TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TIMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) und TMHWX(817) gemeinsam genutzt. Wenn zwei Zeitgeber dieselbe Zeitgebernummer nutzen, sie aber nicht simultan verwendet werden, wird bei Prüfung des Programms ein Duplizierungsfehler erzeugt, der Zeitgeber arbeitet jedoch normal. Zeitgeber, die dieselbe Zeitgebernummer nutzen, arbeiten nicht einwandfrei, wenn sie simultan verwendet werden.

Das Fertig-Bit wird nur aktualisiert, wenn TMHH(540)/TMHHX(552) ausgeführt wird. Der Status des Fertig-Bits kann somit um bis zu einer Zykluszeit gegenüber dem aktuellen Sollwert verzögert sein.

Der Istwert von Zeitgebern, die mit Zeitgebernummern von 0000 bis 2047 programmiert wurden, wird auch aktualisiert, wenn der Zeitgeber auf Standby gesetzt ist. Der Istwert von Zeitgebern, die mit Zeitgebernummern von 2048 bis 4095 programmiert wurden, wird gehalten, wenn der Zeitgeber auf Standby gesetzt ist.

In den folgenden Fällen werden Zeitgeber zurückgesetzt oder pausieren. (Bei Rücksetzung eines Zeitgebers wird dessen Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit auf EIN gesetzt.)

Beschreibung	Istwert	Fertig-Bit
Wechsel der Betriebsart von RUN oder MONITOR zu PROGRAM oder umgekehrt.[1]	0000	AUS
Spannungsversorgung unterbrochen und zurückgesetzt[2]	0000	AUS
Ausführung von CNR(545)/CNRX(547), die RESET TIMER/COUNTER-Befehle[3]	BCD: 9999 Binär: FFFF	AUS
Funktion in verriegeltem Programmabschnitt (IL(002)–ILC(003))	Auf Sollwert zurückgesetzt.	AUS
Funktion in übersprungenem Programmabschnitt (JMP(004)–JME(005))	Istwert wird weiter dekrementiert.	Vorheriger Status bleibt erhalten.

Hinweis

1. Wenn das E/A-Haltebit (A50012) auf EIN gesetzt wurde, wird der Status der Zeitgeber-Fertig-Bits und Istwerte bei Wechsel der Betriebsart beibehalten.
2. Wenn das E/A-Haltebit (A50012) auf EIN gesetzt wurde und der Status des E/A-Haltebits im SPS-Setup geschützt ist, wird der Status des Zeitgeber-Fertig-Bits sowie der Istwerte auch bei Unterbrechung der Spannungsversorgung beibehalten.
3. Der Istwert wird bei Ausführung von TMHH(540)/TMHHX(552) auf den Sollwert gesetzt.

Wenn sich ein TMHH(540)/TMHHX(552)-Zeitgeber in einem übersprungenen Programmabschnitt (JMP(004), CJMP(510), CJPN(511), JME(005)) befindet, wird die Zeitmessung und Änderung des Zeitgeber-Istwerts fortgesetzt. (Siehe Hinweis.) (Der übersprungene TMHH(540)/TMHHX(552)-Befehl wird nicht ausgeführt, aber der Istwert wird alle 1 ms aktualisiert.)

Hinweis Bei CS1D CPU-Baugruppen wird der Istwert im oben genannten Fall nicht aktualisiert.

Wenn sich TIMHH(540)/TIMHHX(552) in einem Programmabschnitt zwischen IL(002) und ILC(003) befindet und der Programmabschnitt verriegelt ist, wird der Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit wird auf AUS gesetzt.

Wenn ein TIMHH(540)/TIMHHX(552)-Zeitgeber zwangsweise gesetzt wird, wird sein Fertig-Bit auf EIN und sein Istwert auf 0000 gesetzt. Wenn ein TMHH(540)/TMHHX(552)-Zeitgeber zwangsweise zurückgesetzt wird, wird sein Fertig-Bit auf AUS gesetzt und sein Istwert auf den Sollwert zurückgesetzt.

Die Funktion von Gleich-Flag (=) und Negativ-Flag (N) hängt vom Model der CPU-Baugruppe ab. Detaillierte Informationen hierzu finden Sie im Abschnitt *Flags* weiter oben.

Wenn Online-Bearbeitung zur Konvertierung eines Zeitgebers zu einer anderen Art von Zeitgeber mit der selben Zeitgebernummer verwendet wird (wie z. B. TMHH(540)/TMHHX(552) ↔ TIM/TIMX(550) oder TMHH(540)/TMHHX(552) ↔ TIMH(015)/TIMHX(551)), muss das Fertig-Bit zurückgesetzt werden. Wenn das Fertig-Bit nicht zurückgesetzt wurde, funktioniert der Zeitgeber nicht ordnungsgemäß.

Istwert und Fertig-Bit eines TMHH(540)/TMHHX(552)-Befehls werden wie in der nachfolgenden Tabelle aufgeführt aktualisiert.

Ausführung von TMHH(540)/ TMHHX(552)	Das Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 beträgt. Das Fertig-Bit wird auf AUS gesetzt, wenn der Istwert nicht 0000 beträgt.
Aktualisierung in 1-ms-Intervall	Der Istwert des Zeitgebers wird alle 1 ms aktualisiert.

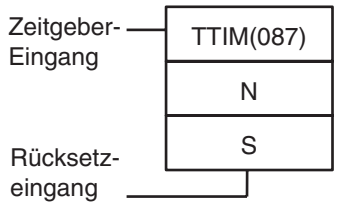
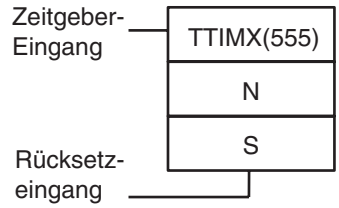
3-6-4 ACCUMULATIVE TIMER: TTIM(087)/TTIMX(555)

Funktion

TTIM(087)/TTIMX(555) ist ein inkrementierender Zeitgeber mit Schritten von 0,1 s. Der Einstellbereich für den Sollwert erstreckt sich von 0 bis 999,9 s bei TTIM(087) bzw. von 0 bis 6.553,5 s bei TTIMX(555). Die Zeitgebergenauigkeit beträgt -0,01 bis 0 s.

Hinweis Die Zeitgebergenauigkeit bei CS1D CPU-Baugruppen beträgt 10 ms + die Zykluszeit.

Kontaktplansymbol

Istwert-Aktualisierungsmethode	Symbol	Operanden
BCD		N: 0000 bis 15 (dezimal) S: #0000 bis #9999 (BCD)
Binär		N: 00000 bis 15 (dezimal) S: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	TTIM(087)/ TTIMX(555)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

N: Zeitgebernummer

Die Zeitgebernummer muss zwischen 0000 und 4095 (dezimal) liegen.

S: Sollwert

Der Sollwert muss zwischen #0000 und 9999 (BCD) liegen.

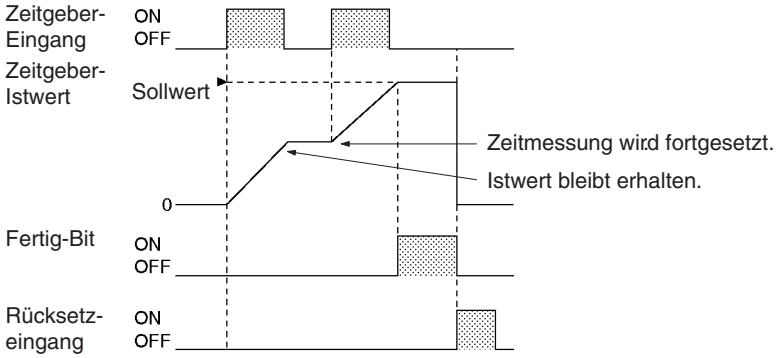
Operanden-Spezifikationen

Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A959
Zeitgeberbereich	0000 bis 4095 (dezimal)	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binär-adressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---	BCD: #0000 bis 9999 (BCD) "&" kann nicht verwendet werden. Binär: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)
Datenregister	---	DR0 bis DR15
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	

Beschreibung

Wenn der Zeitgebereingang auf EIN gesetzt ist, wird der Istwert von TTIM(087)/TTIMX(555) inkrementiert. Wenn der Zeitgeberausgang auf AUS gesetzt wird, stoppt der Zeitgeber die Inkrementierung des Istwerts, der Istwert behält jedoch seinen Wert. Ist der Zeitgebereingang wieder auf EIN gesetzt, wird die Zeitmessung wieder aufgenommen und der Istwert weiter inkrementiert. Das Zeitgeber-Fertig-Bit wird auf EIN gesetzt, wenn der Istwert den Sollwert erreicht.

Der Status von Zeitgeberistwert und Fertig-Bit wird nach Ablauf des Zeitgebers beibehalten. Der Zeitgeber kann auf drei verschiedene Weisen wieder gestartet werden: Der Zeitgeber-Istwert kann auf einen Wert größer Null geändert werden (z.B. mit MOV(021), der Rücksetzeingang kann auf EIN gesetzt oder der Befehl CNR(545)/CNRX(547) kann ausgeführt werden.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N indirekt über ein Indexregister adressiert wird, aber die Adresse im Indexregister nicht die Istwertadresse eines Zeitgebers ist. EIN, falls S im BCD-Modus keine BCD-Daten enthält. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Zeitgebernummern werden von den Befehlen TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TIMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) und TMHWX(817) gemeinsam genutzt. Wenn zwei Zeitgeber dieselbe Zeitgebernummer nutzen, sie aber nicht simultan verwendet werden, wird bei Prüfung des Programms ein Duplizierungsfehler erzeugt, der Zeitgeber arbeitet jedoch normal. Zeitgeber, die dieselbe Zeitgebernummer nutzen, arbeiten nicht einwandfrei, wenn sie simultan verwendet werden.

In den folgenden Fällen werden Zeitgeber zurückgesetzt oder pausieren. (Bei Rücksetzung eines TTIM(087)/TTIMX(555)-Zeitgebers wird dessen Istwert auf 0000 zurückgesetzt und dessen Fertig-Bit auf AUS gesetzt.)

Beschreibung	Istwert	Fertig-Bit
Wechsel der Betriebsart von RUN oder MONITOR zu PROGRAM oder umgekehrt.[1]	0000	AUS
Spannungsversorgung unterbrochen und zurückgesetzt[2]	0000	AUS
Ausführung von CNR(545)/CNRX(547), die RESET TIMER/COUNTER-Befehle[3]	BCD: 9999 Binär: FFFF	AUS
Funktion in verriegeltem Programmabschnitt (IL(002)–ILC(003))	Vorheriger Status bleibt erhalten.	Vorheriger Status bleibt erhalten.
Funktion in übersprungenem Programmabschnitt (JMP(004)–JME(005))	Vorheriger Status bleibt erhalten.	Vorheriger Status bleibt erhalten.

Hinweis

1. Wenn das E/A-Haltebit (A50012) auf EIN gesetzt wurde, wird der Status der Zeitgeber-Fertig-Bits und Istwerte bei Wechsel der Betriebsart beibehalten.
2. Wenn das E/A-Haltebit (A50012) auf EIN gesetzt wurde und der Status des E/A-Haltebits im SPS-Setup geschützt ist, wird der Status des Zeitgeber-Fertig-Bits sowie der Istwerte auch bei Unterbrechung der Spannungsversorgung beibehalten.
3. Der Istwert wird bei Ausführung von TTIM(087)/TTIMX(555) auf den Sollwert gesetzt.

Wenn sich TTIM(087)/TTIMX(555) in einem Programmabschnitt zwischen IL(002) und ILC(003) befindet und der Programmabschnitt verriegelt ist, behält der Istwert seinen vorherigen Wert bei (er wird nicht zurückgesetzt). Die muss bei der Programmierung von TTIM(087)/TTIMX(555) zwischen IL(002) und ILC(003) berücksichtigt werden.

Wenn sich ein TTIM(087)/TTIMX(555)-Zeitgeber in Betrieb in einem Programmabschnitt zwischen JMP(004) und JME(005) befindet und der Programmabschnitt übersprungen wird, behält der Istwert seinen vorherigen Wert bei. Dies muss bei der Programmierung von TTIM(087)/TTIMX(555) zwischen JMP(004) und JME(005) berücksichtigt werden.

Wenn ein TTIM(087)/TTIMX(555)-Zeitgeber zwangsweise gesetzt wird, wird sein Fertig-Bit auf EIN gesetzt und sein Istwert auf 0000 zurückgesetzt. Wenn ein TTIM(087)/TTIMX(555)-Zeitgeber zwangsweise zurückgesetzt wird, wird sein Fertig-Bit auf AUS gesetzt und sein Istwert auf 0000 zurückgesetzt. Zwangssetzung und Zwangsrücksetzung haben Vorrang vor dem Status von Zeitgeber- und Rücksetzeingang.

Der Zeitgeber-Istwert wird nur bei Ausführung von TTIM(087)/TTIMX(555) aktualisiert, sodass der Zeitgeber bei Überschreiten der Zykluszeit von 100 ms nicht ordnungsgemäß arbeitet, da der Zeitgeber in Schritten von 100 ms inkrementiert.

Das Zeitgeber-Fertig-Bit wird nur bei Ausführung von TTIM(087)/TTIMX(555) aktualisiert, so dass für das Fertig-Bit eine Verzögerung von bis zu einem Zyklus auftreten kann, bis es nach Ablauf des Zeitgebers auf EIN gesetzt wird.

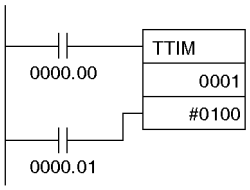
Typische Zeitgeber, wie z.B. TIM/TIMX(550), sind dekrementierende Zähler und der Istwert gibt die verbleibende Zeit bis zum Ablauf des Zeitgebers an. Der Istwert von TTIM(087)/TTIMX(555) gibt die bereits abgelaufene Zeit an, so dass der Istwert für viele Berechnungen und Anzeigeausgaben unverändert verwendet werden kann.

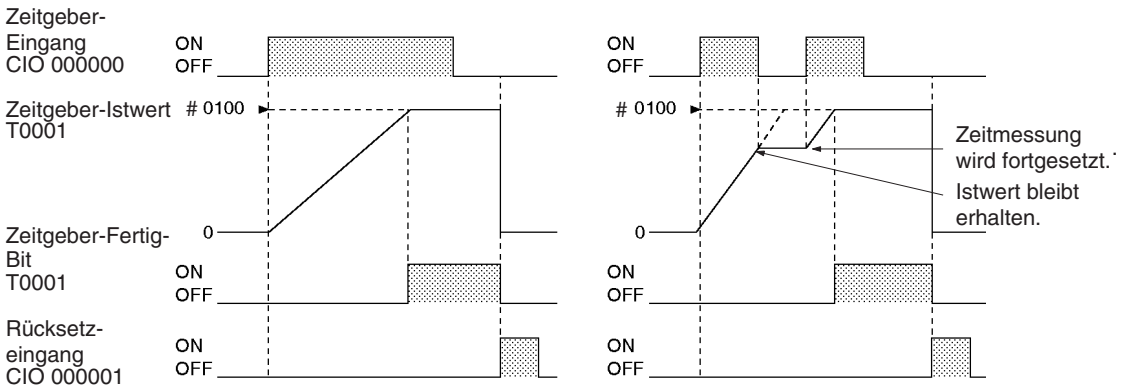
Beispiel

Wenn Zeitgebereingang CIO 000000 im nachfolgenden Beispiel auf EIN gesetzt ist, beginnt der Zeitgeber-Istwert von 0 an aufwärts zu zählen. Das Zeitgeber-Fertig-Bit T0001 wird auf EIN gesetzt, wenn der Istwert den Sollwert erreicht.

Wenn der Rücksetzeingang auf EIN gesetzt wird, wird der Istwert auf 0000 zurückgesetzt und das Fertig-Bit (T0001) auf AUS gesetzt. (Gewöhnlich wird der Rücksetzeingang auf EIN gesetzt, um den Zeitgeber zurückzusetzen. Anschließend wird der Zeitgebereingang zum Starten der Zeitmessung auf EIN gesetzt.)

Wenn der Zeitgebereingang vor Erreichen des Istwerts auf AUS gesetzt wird, stoppt der Zeitgeber die Zeitmessung, der Istwert wird jedoch gehalten. Der Zeitgeber setzt die Zeitmessung mit seinem vorherigen Istwert fort, wenn der Zeitgebereingang wieder auf EIN gesetzt wird.





3-6-5 LONG TIMER: TIML(542)/TIMLX(553)

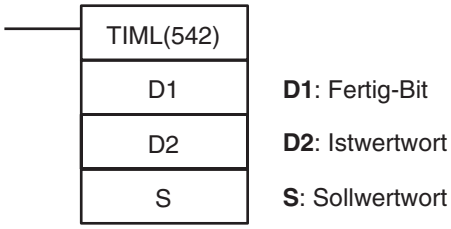
Funktion

TIML(542)/TIMLX(553) ist ein dekrementierender Zeitgeber mit Schritten von 1 s, der eine Zeitmessung von bis zu ca. 115 Tagen bei TIML(542) bzw. 49.710 Tagen bei TIMLX(543) durchführen kann. Die Zeitgebergenauigkeit beträgt 0 bis 0,01 s.

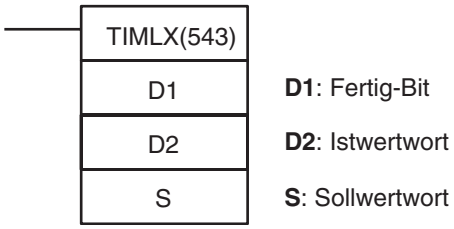
Hinweis Die Zeitgebergenauigkeit bei CS1D CPU-Baugruppen beträgt 10 ms + die Zykluszeit.

Kontaktplansymbol

BCD



Binär



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TIML(542)/ TIMLX(553)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

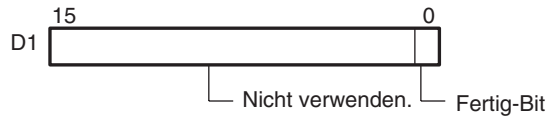
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

D1: Fertig-Bit

Bit 0 von D1 fungiert als Fertig-Bit für TIML(542)/TIMLX(553).



D2: Istwertwort

D2+1 und D2 enthalten den 8-stelligen Binär- oder BCD-Istwert. (D2 und D2+1 müssen im gleichen Datenbereich liegen.) Der Istwert kann bei TIML(542) im Bereich von #00000000 bis #99999999 und bei TIMLX(553) von &00000000 bis &4294967294 (dezimal) oder #00000000 bis #FFFFFFFF (hexadezimal) liegen.



S: Sollwertwort

S+1 und S enthalten den 8-stelligen Binär- oder BCD-Sollwert. (S und S+1 müssen im gleichen Datenbereich liegen.) Der Sollwert muss bei TIML(542) im Bereich von #00000000 bis #99999999 und bei TIMLX(553) von

&00000000 bis &4294967294 (dezimal) oder #00000000 bis #FFFFFFFF (hexadezimal) liegen.



Operanden-Spezifikationen

Bereich	D1	D2	S
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W510	
Arbeitsbereich remanent	H000 bis H511	H000 bis H510	
Systembereich	A448 bis A959	A448 bis A958	A000 bis A958
Zeitgeberbereich	---	---	T0000 bis T4094
Zählerbereich	---	---	C0000 bis C4094
DM-Bereich	D00000 bis D32767	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		BCD: #00000000 bis 99999999 (BCD) "&" kann nicht verwendet werden. Binär: &00000000 bis &4294967294 (dezimal) oder #00000000 bis #FFFFFFFF (hex)
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15		

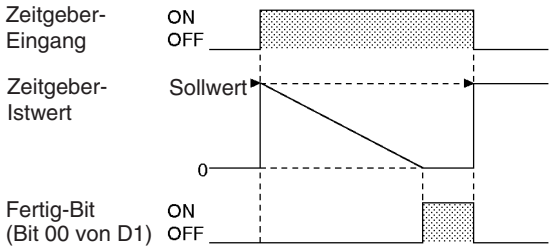
Beschreibung

TIML(542)/TIMLX(553) ist ein dekrementierender Einschaltverzögerungs-Zeitgeber mit Einheiten von 0,1 s, der 8-stellige Ist- und Sollwerte verwendet.

Wenn der Zeitgebereingang auf AUS gesetzt ist, wird der Zeitgeber zurückgesetzt, d.h., der Istwert des Zeitgebers wird auf den Sollwert zurückgesetzt und das Fertig-Bit auf AUS gesetzt.

Wenn der Zeitgebereingang von AUS zu EIN wechselt, startet TIML(542)/TIMLX(553) mit der Dekrementierung des Istwerts in D2+1 und D2. Der Istwert wird stetig verringert, solange der Zeitgebereingang auf EIN gesetzt bleibt. Das Zeitgeber-Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0000 0000 erreicht.

Der Status von Zeitgeberistwert und Fertig-Bit wird nach Ablauf des Zeitgebers beibehalten. Um den Zeitgeber neu zu starten, muss der Zeitgebereingang auf AUS und dann wieder auf EIN gesetzt werden oder der Zeitgeberistwert muss auf einen Wert größer Null geändert werden (z.B. durch MOV(021)).



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der in D2+1 und D2 enthaltene Istwert kein BCD-Wert ist. EIN, wenn der in S+1 und S enthaltene Sollwert kein BCD-Wert ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Anders als die meisten Zeitgeber verwendet TIML(542)/TIMLX(553) keine Zeitgebernummer. (Eine Zeitgeberbereich-Sollwertaktualisierung wird bei TIML(542)/TIMLX(553) nicht ausgeführt.)

Da sich das Fertig-Bit für TIML(542)/TIMLX(553) in einem Datenbereich befindet, kann es wie andere Bits zwangsweise gesetzt oder zurückgesetzt werden, allerdings ändert sich der Istwert dadurch nicht.

Der Zeitgeber-Istwert wird nur bei Ausführung von TIML(542)/TIMLX(553) aktualisiert, sodass der Zeitgeber bei Überschreiten der Zykluszeit von 100 ms nicht ordnungsgemäß arbeitet, da der Zeitgeber in Schritten von 100 ms inkrementiert.

Das Zeitgeber-Fertig-Bit wird nur bei Ausführung von TIML(542)/TIMLX(553) aktualisiert, so dass für das Fertig-Bit eine Verzögerung von bis zu einem Zyklus auftreten kann, bis es nach Ablauf des Zeitgebers auf EIN gesetzt wird.

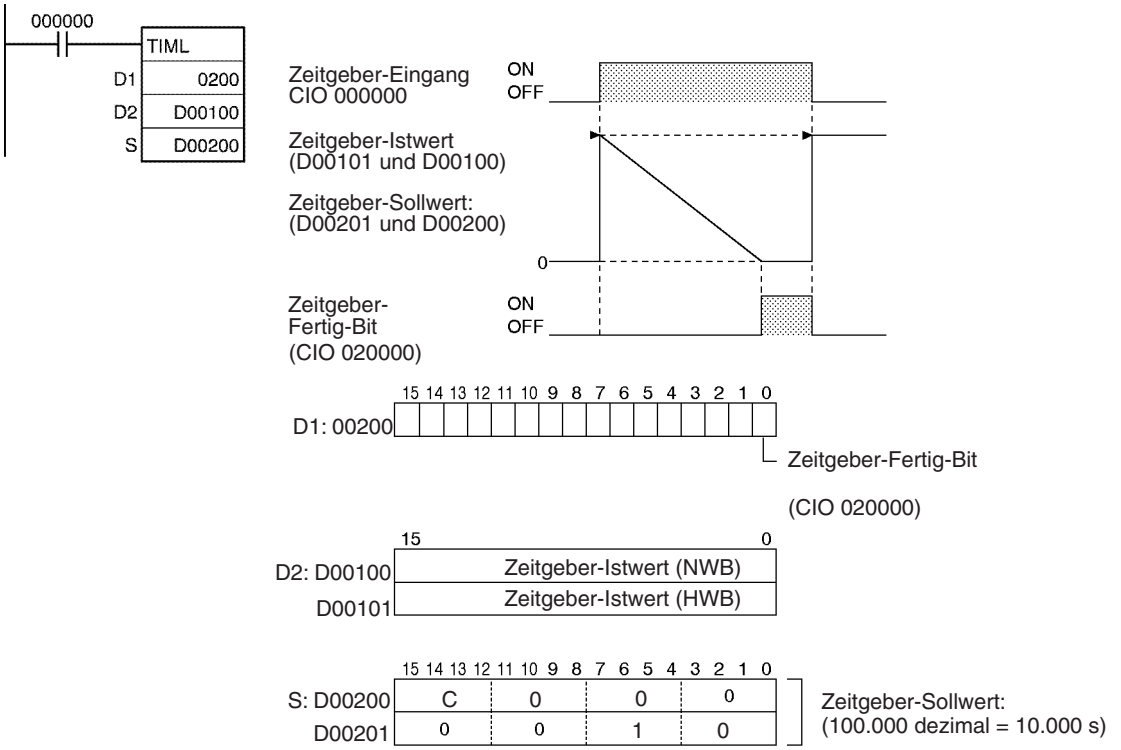
Wenn sich TIML(542)/TIMLX(553) in einem Programmabschnitt zwischen IL(002) und ILC(003) befindet und der Programmabschnitt verriegelt ist, wird der Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit wird auf AUS gesetzt.

Wenn sich ein TIML(542)/TIMLX(553)-Zeitgeber in Betrieb in einem Programmabschnitt zwischen JMP(004) und JME(005) befindet und der Programmabschnitt übersprungen wird, behält der Istwert seinen vorherigen Wert bei. Dies muss bei der Programmierung von TIML(542)/TIMLX(553) zwischen JMP(004) und JME(005) berücksichtigt werden.

Achten Sie darauf, dass die für das Fertig-Bit sowie für den Istwert spezifizierten Worte (D1, D2 und D2+1) nicht von anderen Befehlen verwendet werden. Wenn diese Worte durch andere Befehle beeinflusst werden, kann der Zeitgeber die Zeitmessung evtl. nicht ordnungsgemäß ausführen.

Beispiel

Wenn der Zeitgebereingang CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird der Zeitgeber-Istwert (in D00101 und D00100) auf den Sollwert gesetzt (in D00101 und D00100) und das Herunterzählen des Istwerts beginnt. Das Zeitgeber-Fertig-Bit (CIO 020000) wird auf EIN gesetzt, wenn der Istwert 0000 0000 erreicht.
Bei Rücksetzung von CIO 000000 auf AUS wird der Zeitgeber-Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit auf AUS gesetzt.



3-6-6 MULTI-OUTPUT TIMER: MTIM(543)/MTIMX(554)

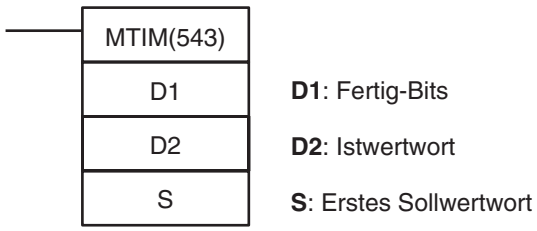
Funktion

MTIM(543)/MTIMX(554) ist ein in 0,1-s-Schritten inkrementierender Zeitgeber mit acht unabhängigen Sollwerten und Fertig-Bits. Der Sollwert bei MTIM(543) hat einen Bereich von 0 bis 999,9 s und bei MTIMX(554) von 0 bis 6553,5 s. Die Zeitgebergenauigkeit beträgt 0 bis 0,01 s.

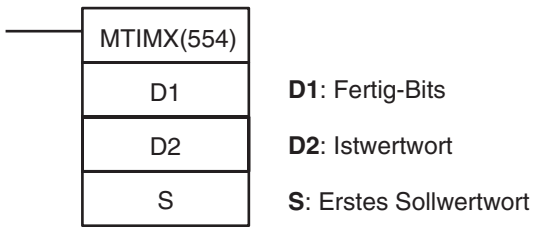
Hinweis Die Zeitgebergenauigkeit bei CS1D CPU-Baugruppen beträgt 10 ms + die Zykluszeit.

Kontaktplansymbol

BCD



Binär



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MTIM(543)/ MTIMX(554)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

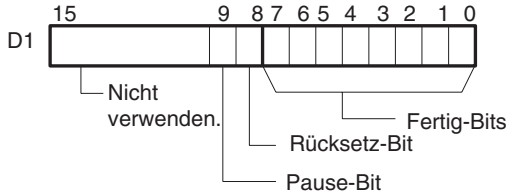
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

D1: Fertig-Bits

D1 enthält die acht Fertig-Bits sowie die Pausen- und Rücksetz-Bits.



D2: Istwertwort

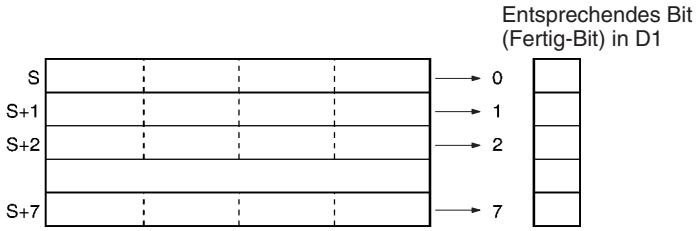
D2 enthält den 4-stelligen Binär- oder BCD-Istwert.

Daten	Bereich
BCD	#0000 bis #9999
Binär	&0 bis &65535 (dezimal) #0000 bis #FFFF (hex)

S: Erstes Sollwertwort

S bis S+7 enthalten die acht unabhängigen Sollwerte. Jeder Sollwert muss folgende Spezifikationen besitzen:

Daten	Bereich
BCD	#0000 bis #9999
Binär	&0 bis &65535 (dezimal) #0000 bis #FFFF (hex)



Daten	Bereich
BCD	Ein Wort für jeden der 8 Zeitgeber-Sollwerte: #0000 bis #9999
Binär	Ein Wort für jeden der 8 Zeitgeber-Sollwerte: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)

Hinweis S bis S+7 müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	D1	D2	S
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6136
Arbeitsbereich nicht remanent	W000 bis W511		W000 bis W504
Arbeitsbereich remanent	H000 bis H511		H000 bis H504
Systembereich	A448 bis A959		A000 bis A952
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4088
Zählerbereich	C0000 bis C4095		C0000 bis C4088

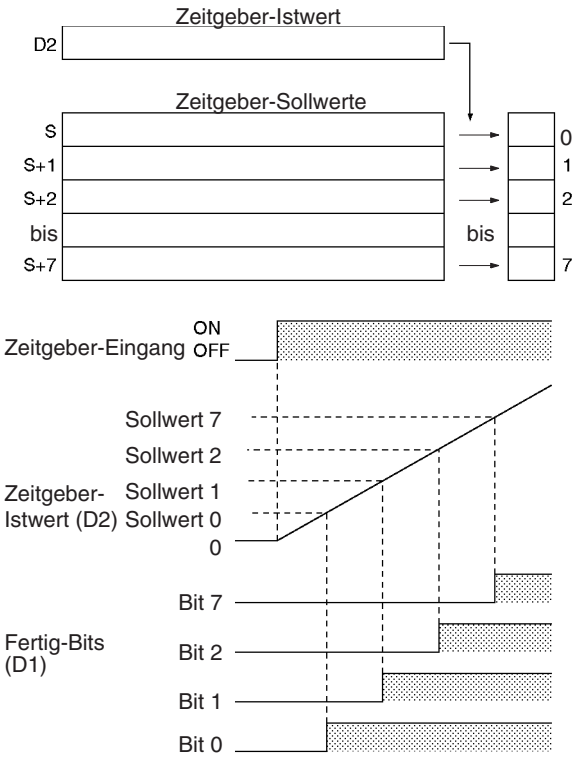
Bereich	D1	D2	S
DM-Bereich	D00000 bis D32767		D00000 bis D32760
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32760
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32760 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---	DR0 bis DR15	---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Wenn die Ausführungsbedingung für MTIM(543)/MTIMX(554) auf EIN gesetzt ist und die Rücksetz- und Zeitgeber-Bits auf AUS gesetzt sind, inkrementiert MTIM(543)/MTIMX(554) den Istwert in D2. Wenn das Pause-Bit auf EIN gesetzt wird, stoppt der Zeitgeber die Inkrementierung des Istwerts, der Istwert behält jedoch seinen Wert. MTIM(543)/MTIMX(554) nimmt die Zeitmessung wieder auf, wenn das Pause-Bit wieder auf AUS gesetzt wird.

Der Istwert (Inhalt von D2) wird bei jeder Ausführung von MTIM(543)/MTIMX(554) mit den acht Sollwerten in S bis S+7 verglichen. Wenn einer der Sollwerte kleiner als oder gleich dem Istwert ist, wird das entsprechende Fertig-Bit (D1, Bits 00 bis 07) auf EIN gesetzt.

Wenn der Istwert 9999 erreicht, wird dieser auf 0000 zurückgesetzt und alle Fertig-Bits werden auf AUS gesetzt. Wird das Rücksetz-Bit bei Betrieb oder Pause des Zeitgebers auf EIN gesetzt, wird der Istwert auf 0000 zurückgesetzt und die Fertig-Bits werden auf AUS gesetzt.



Die nachfolgende Tabelle zeigt die Funktion von MTIM(543)/MTIMX(554) für die vier möglichen Kombinationen der Rücksetz- oder Pause-Bits.

Rücksetz-Bit (Bit 08)	Pause-Bit (Bit 09)	Funktion
AUS	AUS	Der Istwert wird aktualisiert und das entsprechende Fertig-Bit auf EIN gesetzt, wenn Sollwert ≤ Istwert.
	EIN	Der Istwert wird nicht aktualisiert und MTIM(543)/ MTIMX(554) wird als NOP(000) behandelt.
EIN	AUS	Der Istwert wird auf 0000 zurückgesetzt und die Fertig-Bits werden auf AUS gesetzt. Der Istwert wird nicht aktualisiert.
	EIN	

Rücksetz- und Pause-Bits sind nur wirksam, wenn die Ausführungsbedingung für MTIM(543)/MTIMX(554) auf EIN gesetzt ist.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der in D2 enthaltene Istwert kein BCD-Wert ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

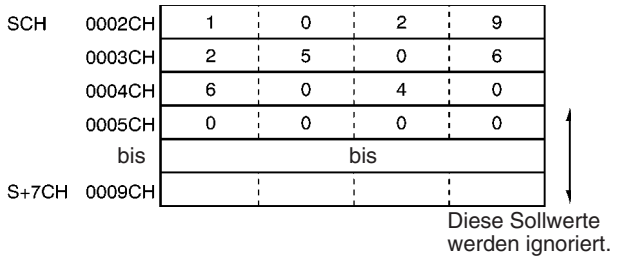
Anders als die meisten Zeitgeber verwendet MTIM(543)/MTIMX(554) keine Zeitgebernummer. (Eine Zeitgeberbereich-Sollwertaktualisierung wird bei MTIM(543)/MTIMX(554)) nicht ausgeführt.

Wenn der Istwert 9999 erreicht, wird dieser auf 0000 zurückgesetzt und alle Fertig-Bits werden auf AUS gesetzt.

Wenn der Befehl im BCD-Modus ausgeführt wird und ein Sollwert in S bis S+7 keine BCD-Daten enthält, wird der Sollwert ignoriert. Es tritt kein Fehler auf und das Fehler-Flag wird nicht auf EIN gesetzt.

Da sich das Fertig-Bit für MTIM(543)/MTIMX(554) in einem Datenbereich befindet, kann es wie andere Bits zwangsweise gesetzt oder zurückgesetzt werden, allerdings ändert sich der Istwert dadurch nicht.

Wenn acht oder weniger Sollwerte erforderlich sind, muss das Wort nach dem letzten Sollwert auf 0000 gesetzt werden. MTIM(543)/MTIMX(554) ignoriert den auf 0000 gesetzten Sollwert sowie alle verbleibenden Sollwerte.



Der Zeitgeber-Istwert wird nur bei Ausführung von MTIM(543)/MTIMX(554) aktualisiert, sodass der Zeitgeber bei Überschreiten der Zykluszeit von 100 ms nicht ordnungsgemäß arbeitet, da der Zeitgeber in Schritten von 100 ms inkrementiert. Um eine präzise Zeitmessung sicherzustellen und Probleme aufgrund langer Zykluszeiten zu vermeiden, muss der MTIM(543)/ MTIMX(554)-Befehl an mehreren Punkten im Programm eingegeben werden.

Das Zeitgeber-Fertig-Bit wird nur bei Ausführung von MTIM(543)/MTIMX(554) aktualisiert, so dass für das Fertig-Bit eine Verzögerung von bis zu einem Zyklus auftreten kann, bis es nach Ablauf des Zeitgebers auf EIN gesetzt wird.

Wenn sich MTIM(543)/MTIMX(554) in einem Programmabschnitt zwischen IL(002) und ILC(003) befindet und der Programmabschnitt verriegelt ist, behält der Istwert seinen vorherigen Wert bei (er wird nicht zurückgesetzt). Dies muss bei der Programmierung von MTIM(543)/MTIMX(554) zwischen IL(002) und ILC(003) berücksichtigt werden.

Wenn sich ein MTIM(543)/MTIMX(554)-Zeitgeber in Betrieb in einem Programmabschnitt zwischen JMP(004) und JME(005) befindet und der Programmabschnitt übersprungen wird, behält der Istwert seinen vorherigen Wert bei. Dies muss bei der Programmierung von MTIM(543)/MTIMX(554) zwischen JMP(004) und JME(005) berücksichtigt werden.

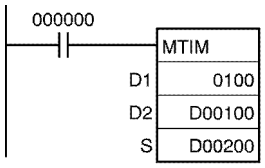
Achten Sie darauf, dass die für die Fertig-Bits sowie für den Istwert spezifizierten Worte (D1 und D2) nicht von anderen Befehlen verwendet werden. Wenn diese Worte durch andere Befehle beeinflusst werden, kann der Zeitgeber die Zeitmessung evtl. nicht ordnungsgemäß ausführen.

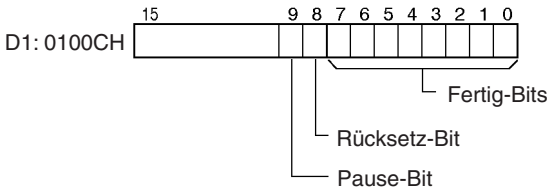
Wenn ein Wort im CIO Bereich für D1 spezifiziert ist, können die Befehle SET und RSET zur Steuerung der Pause- und Rücksetz-Bits verwendet werden.

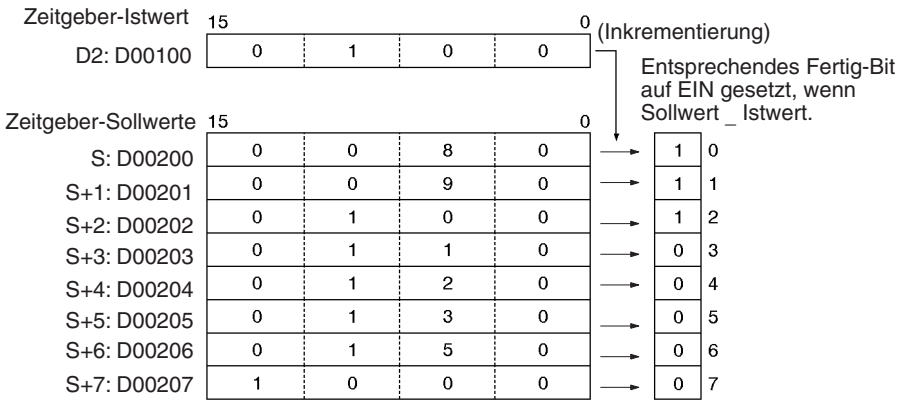
Beispiel

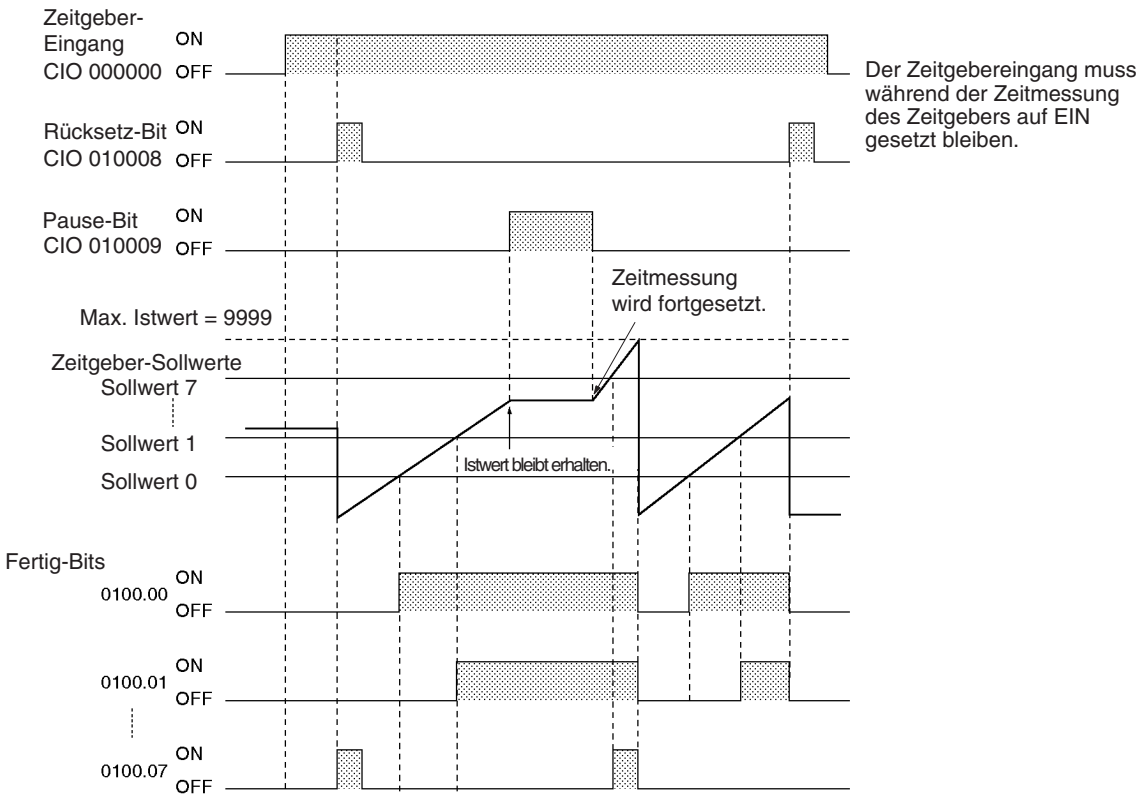
Wenn CIO 000000 im folgenden Beispiel auf EIN und das Pause-Bit (CIO 010009) auf AUS gesetzt ist, startet der Zeitgeber seinen Betrieb, wenn das Rücksetz-Bit (CIO 010008) von EIN auf AUS gesetzt wird. Der Zeitgeber-Istwert beginnt mit dem Hochzählen von 0000.

Die acht Sollwerte in D00200 bis D00207 werden mit dem Istwert verglichen und die entsprechenden Fertig-Bits (CIO 010000 bis CIO 010007) werden auf EIN gesetzt, wenn der jeweilige Sollwert ≤ Istwert.


000000
MTIM
D1 0100
D2 D00100
S D00200


D1: 0100CH
15 9 8 7 6 5 4 3 2 1 0
Fertig-Bits
Rücksetz-Bit
Pause-Bit


Zeitgeber-Istwert
15 0
D2: D00100
0 1 0 0
(Inkrementierung)
Entsprechendes Fertig-Bit auf EIN gesetzt, wenn Sollwert _ Istwert.
Zeitgeber-Sollwerte
15 0
S: D00200 0 0 8 0 → 1 0
S+1: D00201 0 0 9 0 → 1 1
S+2: D00202 0 1 0 0 → 1 2
S+3: D00203 0 1 1 0 → 0 3
S+4: D00204 0 1 2 0 → 0 4
S+5: D00205 0 1 3 0 → 0 5
S+6: D00206 0 1 5 0 → 0 6
S+7: D00207 1 0 0 0 → 0 7


Zeitgeber-Eingang CIO 000000 ON OFF
Der Zeitgebereingang muss während der Zeitmessung des Zeitgebers auf EIN gesetzt bleiben.
Rücksetz-Bit CIO 010008 ON OFF
Pause-Bit CIO 010009 ON OFF
Zeitmessung wird fortgesetzt.
Max. Istwert = 9999
Zeitgeber-Sollwerte
Sollwert 7
Sollwert 1
Sollwert 0
Istwert bleibt erhalten.
Fertig-Bits
0100.00 ON OFF
0100.01 ON OFF
0100.07 ON OFF

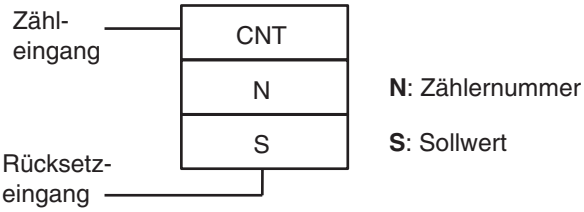
3-6-7 COUNTER: CNT/CNTX(546)

Funktion

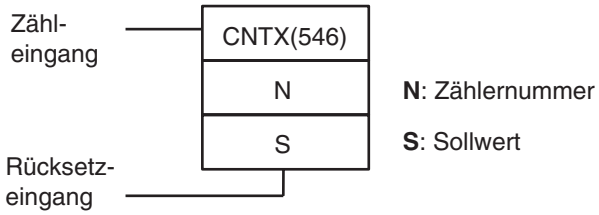
CNT/CNTX(546) ist ein dekrementierender Zähler. Der Zähler hat einen Sollwertbereich von 0 bis 9.999 bei CNT und 0 bis 65.535 bei CNTX(546).

Kontaktplansymbol

BCD



Binär



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CNT/ CNTX(546)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Operanden

N: Zählernummer

Die Zählernummer muss zwischen 0000 und 4095 (dezimal) liegen.

S: Sollwert

Daten	Bereich
BCD	#0000 bis #9999
Binär	&0 bis &65535 (dezimal) #0000 bis #FFFF (hex)

Operanden-Spezifikationen

Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A959
Zeitgeberbereich	---	T0000 bis T4095
Zählerbereich	0000 bis 4095 (dezimal)	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)

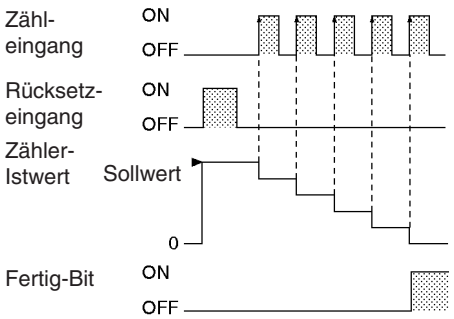
Bereich	N	S
Indirekte DM/EM-Binär-adressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---	BCD: #0000 bis 9999 (BCD) "&" kann nicht verwendet werden. Binär: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)
Datenregister	---	DR0 bis DR15
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	

Beschreibung

Der Zähler-Istwert wird jedes Mal um 1 dekrementiert, wenn der Status des Zähleingang von AUS zu EIN wechselt. Wenn der Istwert 0 erreicht, wird das Fertig-Bit auf EIN gesetzt.

Wenn das Fertig-Bit auf EIN gesetzt wurde, muss der Zähler durch Setzen des Rücksetzeingangs auf EIN oder durch Ausführung des Befehls CNR(545)/CNRX(547) zurückgesetzt werden. Anderenfalls kann der Zähler nicht neu gestartet werden.

Wenn der Rücksetzeingang auf EIN gesetzt ist, wird der Zähler zurückgesetzt und der Zähleingang ignoriert. (Bei Rücksetzung eines Zählers wird dessen Istwert auf den Sollwert zurückgesetzt und das Fertig-Bit auf AUS gesetzt.)



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N indirekt über ein Indexregister adressiert wird, aber die Adresse im Indexregister nicht die Istwertadresse eines Zählers ist. EIN, falls S im BCD-Modus keine BCD-Daten enthält. In allen anderen Fällen AUS.
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese auf AUS gesetzt.
In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

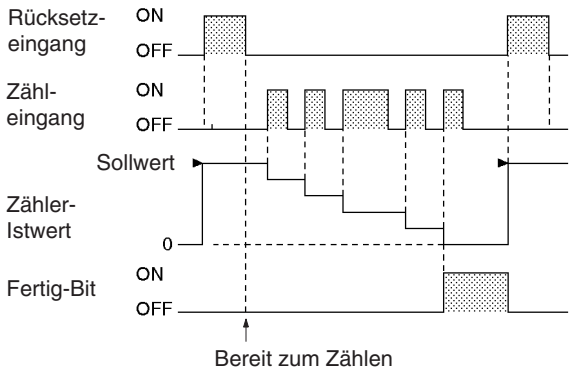
Vorsichtsmaßnahmen

Zählernummern werden von den Befehlen CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814) und CNTWX(818) gemeinsam genutzt. Wenn zwei Zähler dieselbe Zählernummer nutzen, sie aber nicht simultan verwendet werden, wird bei Prüfung des Programms ein Duplizierungsfehler erzeugt, der Zähler arbeitet jedoch normal. Zähler, die dieselbe Zählernummer nutzen, arbeiten nicht ordnungsgemäß, wenn sie simultan verwendet werden.

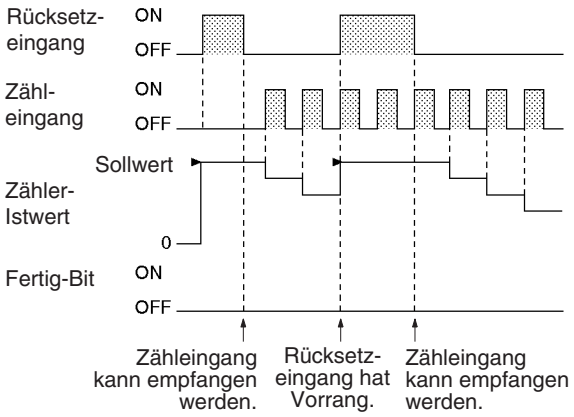
Der Istwert eines Zählers wird aktualisiert, wenn der Zählereingang von AUS auf EIN wechselt und das Fertig-Bit wird bei jeder Ausführung des Befehls CNT/CNTX(546) aktualisiert. Das Fertig-Bit wird auf EIN gesetzt, wenn der Istwert 0 ist und es wird auf AUS gesetzt, wenn der Istwert nicht 0 ist.

Wenn ein CNT/CNTX(546)-Zähler zwangsweise gesetzt wird, wird sein Fertig-Bit auf EIN und sein Istwert auf 0000 gesetzt. Wenn ein CNT/CNTX(546)-Zähler zwangsweise zurückgesetzt wird, wird sein Fertig-Bit auf AUS gesetzt und sein Istwert auf den Sollwert gesetzt.

Stellen Sie sicher, dass der Zähler durch Wechsel des Rücksetzausgangs von AUS → EIN → AUS zurückgesetzt wird, bevor mit dem Zählen über den Zähleingang begonnen wird. Siehe nachfolgende Abbildung. Der Zähleingang wird nicht empfangen, wenn der Rücksetzeingang auf EIN gesetzt ist.



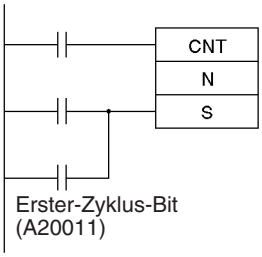
Der Rücksetzeingang hat Vorrang und der Zähler wird zurückgesetzt, wenn Rücksetz- und Zähleingang gleichzeitig auf EIN gesetzt sind. (Der Istwert wird auf den Sollwert zurückgesetzt und das Fertig-Bit auf AUS gesetzt.)



Die Funktion von Gleich-Flag (=) und Negativ-Flag (N) hängt vom Model der CPU-Baugruppe ab. Detaillierte Informationen hierzu finden Sie im Abschnitt *Flags* weiter oben.

Hinweis Wenn das Hinzufügen eines Zählers mit der Online-Bearbeitung erfolgt, muss der Zähler zurückgesetzt werden, bevor er ordnungsgemäß funktioniert. Wird der Zähler nicht zurückgesetzt, wird der vorherige Wert als Zähler-Istwert genommen und der Zähler funktioniert evtl. nicht ordnungsgemäß.

Zähler-Istwerte werden auch bei einer Unterbrechung der Versorgungsspannung beibehalten. Wenn der Neustart der Zählung mit dem Sollwert statt mit dem beibehaltenen Istwert erfolgen soll, muss dem Zähler das Erster-Zyklus-Bit (A20011) als Rücksetzeingang hinzugefügt werden.



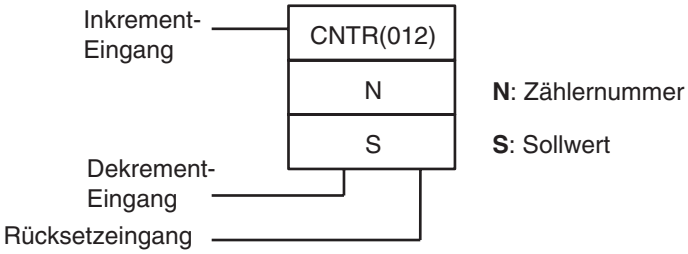
3-6-8 REVERSIBLE COUNTER: CNTR(012)/CNTRX(548)

Funktion

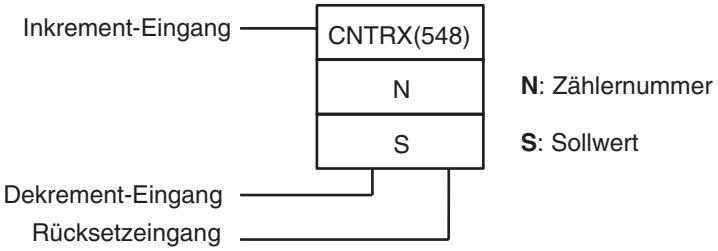
CNTR(012)/CNTRX(548) ist einen reversierbarer Zähler.

Kontaktplansymbol

BCD



Binär



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CNTR(012)/ CNTRX(548)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Operanden

N: Zählernummer

Die Zählernummer muss zwischen 0000 und 4095 (dezimal) liegen.

S: Sollwert

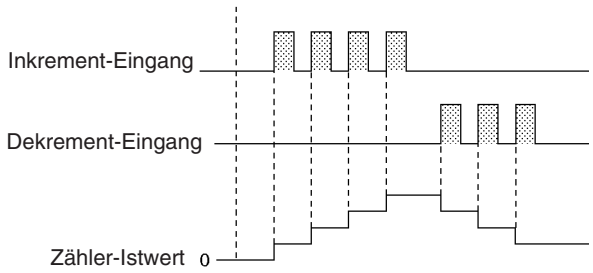
Daten	Bereich
BCD	#0000 bis #9999
Binär	&0 bis &65535 (dezimal) #0000 bis #FFFF (hex)

Operanden-Spezifikationen

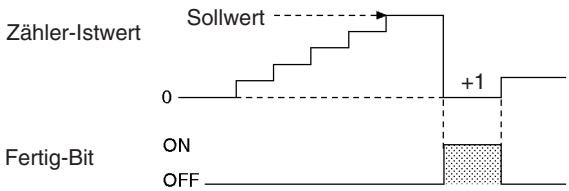
Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A959
Zeitgeberbereich	---	T0000 bis T4095
Zählerbereich	0000 bis 4095 (dezimal)	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/ EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/ EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---	BCD: #0000 bis 9999 (BCD) "&" kann nicht verwendet werden. Binär: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)
Datenregister	---	DR0 bis DR15
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	

Beschreibung

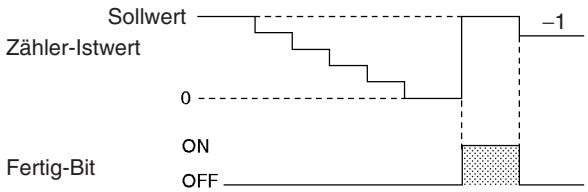
Der Zähler-Istwert wird immer dann um 1 inkrementiert, wenn der Inkrement-Eingang von AUS zu EIN wechselt. Er wird immer dann um 1 dekrementiert, wenn der Dekrement-Eingang von AUS zu EIN wechselt. Der Istwert kann sich im Bereich zwischen 0 und dem Sollwert bewegen.



Beim Inkrementieren wird das Fertig-Bit auf EIN gesetzt, wenn der Istwert vom Sollwert zurück auf 0 inkrementiert ist und es wird wieder auf AUS gesetzt, wenn der Istwert von 0 auf 1 inkrementiert wird.



Beim Dekrementieren wird das Fertig-Bit auf EIN gesetzt, wenn der Istwert von 0 auf den Sollwert dekrementiert wird und es wird wieder auf AUS gesetzt, wenn der Istwert vom Sollwert auf Sollwert–1 dekrementiert wird.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N indirekt über ein Indexregister adressiert wird, aber die Adresse im Indexregister nicht die Istwertadresse eines Zählers ist. EIN, falls S im BCD-Modus keine BCD-Daten enthält. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Zählernummern werden von den Befehlen CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814) und CNTWX(818) gemeinsam genutzt. Wenn zwei Zähler dieselbe Zählernummer nutzen, sie aber nicht simultan verwendet werden, wird bei Prüfung des Programms ein Duplizierungsfehler erzeugt, der Zähler arbeitet jedoch normal. Zähler, die dieselbe Zählernummer nutzen, arbeiten nicht ordnungsgemäß, wenn sie simultan verwendet werden.

Der Istwert wird nicht geändert, wenn die Inkrement- und Dekrement-Eingänge gleichzeitig von AUS auf EIN wechseln. Wenn der Rücksetzeingang auf EIN gesetzt ist, wird der Istwert auf 0 zurückgesetzt und die Zähleingänge werden ignoriert.

Das Fertig-Bit wird nur dann auf EIN gesetzt, wenn der Istwert vom Sollwert auf 0 inkrementiert oder von 0 auf den Sollwert dekrementiert wurde. In allen anderen Fällen ist es auf AUS gesetzt.

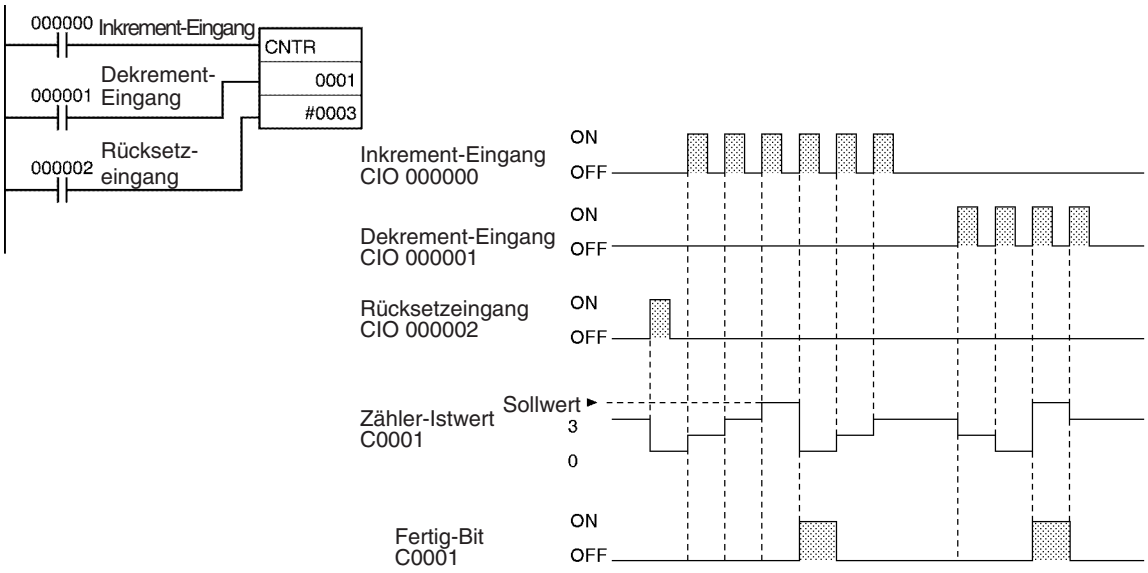
Bei der Eingabe des CNTR(012)/CNTRX(548)-Befehls in der AWL-Form muss zuerst der Inkrement-Eingang (II), dann der Dekrement-Eingang (DI), der Rücksetzeingang (R) und schließlich der CNTR(012)/CNTRX(548)-Befehl eingegeben werden. Bei der Eingabe in Kontaktplanform muss zuerst der Inkrement-Eingang (II), dann der CNTR(012)/CNTRX(548)-Befehl, dann der Dekrement-Eingang (DI) und schließlich der Rücksetzeingang (R) eingegeben werden.

Beispiele

Grundfunktion des CNTR(012)/CNTRX(548)

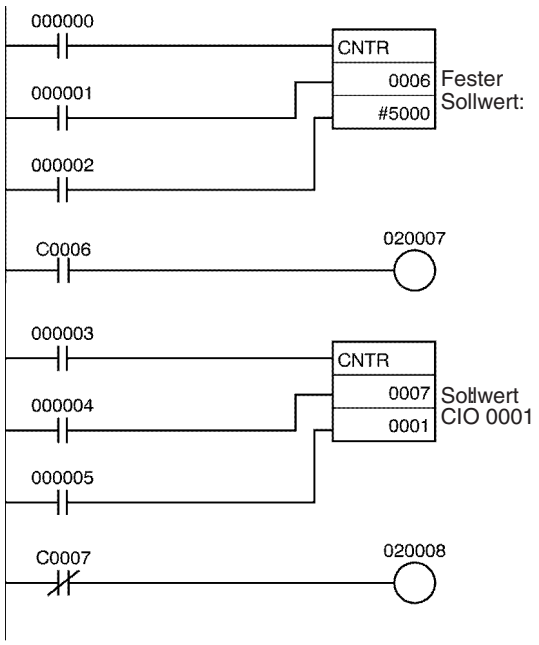
Der Zähler-Istwert wird durch Setzen des Rücksetzeingangs (CIO 000002) auf EIN und AUS auf 0 zurückgesetzt. Der Istwert wird immer um 1 inkrementiert, wenn der Inkrement-Eingang (CIO 000000) von AUS auf EIN wechselt. Wenn der Istwert vom Sollwert (3) inkrementiert wird, wird er automatisch auf 0 zurückgesetzt und das Fertig-Bit auf EIN gesetzt.

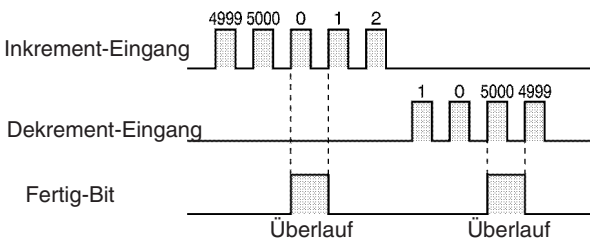
Gleichermaßen wird der Istwert immer um 1 dekrementiert, wenn der Dekrement-Eingang (CIO 000001) von AUS auf EIN wechselt. Wenn der Istwert von 0 dekrementiert wird, wird er automatisch auf den Sollwert (3) gesetzt und das Fertig-Bit auf EIN gesetzt.



Spezifizieren des Sollwerts in einem Wort

Im folgenden Beispiel wird der Sollwert für CNTR(012) 0007 durch den Inhalt von CIO 0001 festgelegt. Wenn der Inhalt von CIO 0001 von einem externen Schalter gesteuert wird, kann der Sollwert über den Schalter manuell geändert werden.





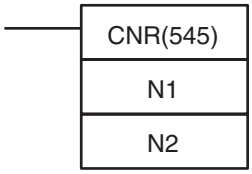
3-6-9 RESET TIMER/COUNTER: CNR(545)/CNRX(547)

Funktion

Setzt Zeitgeber oder Zähler innerhalb des spezifizierten Bereichs von Zeitgeber- bzw. Zählernummern zurück.

Kontaktplansymbol

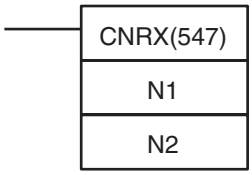
BCD



N_1: Erste Nummer im Bereich

N_2: Letzte Nummer im Bereich

Binär



N_1: Erste Nummer im Bereich

N_2: Letzte Nummer im Bereich

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CNR(545)/ CNRX(547)
	Ausführung einmal bei steigender Flanke	@CNR(545)/ CNRX(547)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N_1: Erste Nummer im Bereich

N_1 muss eine Zeitgebernummer zwischen T0000 und T4095 oder eine Zählernummer zwischen C0000 und C4095 sein.

N_2: Letzte Nummer im Bereich

N_2 muss eine Zeitgebernummer zwischen T0000 und T4095 oder eine Zählernummer zwischen C0000 und C4095 sein.

Hinweis N_1 und N_2 müssen im selben Datenbereich liegen, d.h. N_1 und N_2 müssen Zeitgeber- oder Zählernummern sein.

Operanden-Spezifikationen

Bereich	N_1	N_2
CIO-Bereich	---	---
Arbeitsbereich nicht remanent	---	---
Arbeitsbereich remanent	---	---
Systembereich	---	---
Zeitgeberbereich	C0000 bis C4095	C0000 bis C4095
Zählerbereich	T0000 bis T4095	T0000 bis T4095
DM-Bereich	---	---
EM-Bereich ohne Bank	---	---
EM-Bereich mit Bank	---	---
Indirekte DM/EM-Binäradressen	---	---
Indirekte DM/EM-Adressen in BCD	---	---

Bereich	N_1	N_2
Konstanten	---	---
Datenregister	---	---
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

CNR(545)/CNRX(547) setzt die Fertig-Bits aller Zeitgeber oder Zähler von N_1 bis N_2 zurück. Gleichzeitig werden alle Istwerte auf den Maximalwert (9999 bei BCD bzw. FFFF bei binär) gesetzt. (Der Istwert wird bei der nächsten Ausführung des Zeitgeber- oder Zählerbefehls auf den Sollwert gesetzt.)

Zeitgeber-Rücksetzung durch CNR(545)/CNRX(547)

Die folgenden Zeitgeber werden zurückgesetzt, wenn ihre Zeitgebernummern im spezifizierten Bereich liegen: TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815) und TMHWX(817). Bei Rücksetzung eines Zeitgebers wird dessen Fertig-Bit auf AUS gesetzt und sein Istwert auf den Maximalwert von 9999 gesetzt.

Hinweis Die Zeitgeber TIML(542), TIMLX(553), MTIM(543) und MTIMX(554) werden durch CNR(545)/CNRX(547) nicht zurückgesetzt, da diese Zeitgeber keine Zeitgebernummern verwenden.

Zähler-Rücksetzung durch CNR(545)/CNRX(547)

Die folgenden Zähler werden zurückgesetzt, wenn ihre Zählernummern im spezifizierten Bereich liegen: CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814) und CNTWX(818). Bei Rücksetzung eines Zählers wird dessen Fertig-Bit auf AUS gesetzt und sein Istwert auf den Maximalwert von 9999 gesetzt.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N_1 indirekt über ein Indexregister adressiert wird, aber die Adresse im Indexregister nicht die Istwertadresse eines Zeitgebers oder Zählers ist. EIN, wenn N_2 indirekt über ein Indexregister adressiert wird, aber die Adresse im Indexregister nicht die Istwertadresse eines Zeitgebers oder Zählers ist. EIN, wenn sich N_1 und N_2 nicht im selben Datenbereich befinden. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

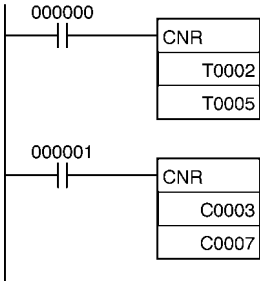
CNR(545)/CNRX(547) setzt die Zeitgeber-/Zählerbefehle selbst nicht zurück, sondern nur die diesen Befehlen zugewiesenen Istwerte und Fertig-Bits. In den meisten Fällen ist die Auswirkung von CNR(545)/CNRX(547) anders als das direkte Rücksetzen der Befehle. Wenn z.B. ein TIM/TIMX(550)-Befehl zurückgesetzt wird, wird dessen Istwert direkt auf den Sollwert gesetzt, wenn dieser Zähler jedoch durch CNR(545)/CNRX(547) zurückgesetzt wird, wird dessen Istwert auf den Maximalwert von 9999 gesetzt.

Bei Spezifizierung von N1 und N2 mit N1>N2 wird nur das Fertig-Bit für den Zähler/Zeitgeber mit dieser Nummer zurückgesetzt.

Beispiel

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden die Fertig-Bits für Zeitgeber T0002 bis T0005 auf AUS gesetzt und die Zeitgeber-Istwerte auf den Maximalwert von 9999 gesetzt.

Wenn CIO 000001 auf EIN gesetzt ist, werden die Fertig-Bits für Zähler C0003 bis C0007 auf AUS gesetzt und die Zeitgeber-Istwerte auf den Maximalwert von 9999 gesetzt.



3-6-10 Beispiele für Zeitgeber- und Zähleranwendungen

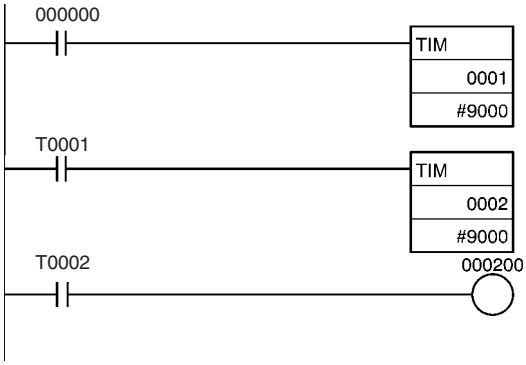
Die folgenden Beispiele zeigen verschiedene Anwendungen von Zeitgeber- und Zählerbefehlen, einschließlich Langzeit-Zeitgeber, zweistufiger Zähler, EIN/AUS-Verzögerung, Einzelimpuls-Bit und Blink-Bit.

Beispiel 1: Langzeit-Zeitgeber

Das folgende Programmbeispiel zeigt drei Wege für die Erstellung von Langzeit-Zeitgebern mit Standard-TIM- und CNT-Befehlen.

Zwei TIM-Befehle

In diesem Beispiel werden zwei TIM-Befehle zu einem 30-Minuten-Zeitgeber kombiniert.

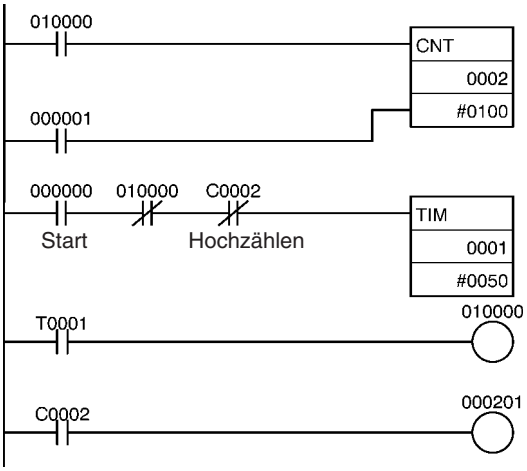


Adresse	Befehl	Operanden
000000	LD	000000
000001	TIM	0001
		#9000
000002	LD	T0001
000003	TIM	0002
		#9000
000004	LD	T0002
000005	OUT	000200

TIM- und CNT-Befehle

In diesem Beispiel werden ein TIM- und ein CNT-Befehl zu einem 500-Sekunden-Zeitgeber kombiniert.

TIM 0001 generiert alle 5 s einen Impuls und CNT 0002 zählt diese Impulse. Der Sollwert für diese Kombination ist Zeitgeberintervall × Zähler-Sollwert. In diesem Fall würde der Zeitgeber-Sollwert 5 s × 100 = 500 s betragen. Bei dieser Kombination ist der Istwert des Langzeit-Zeitgebers tatsächlich der Istwert eines Zählers, der auch bei einer Spannungsunterbrechung gehalten wird.

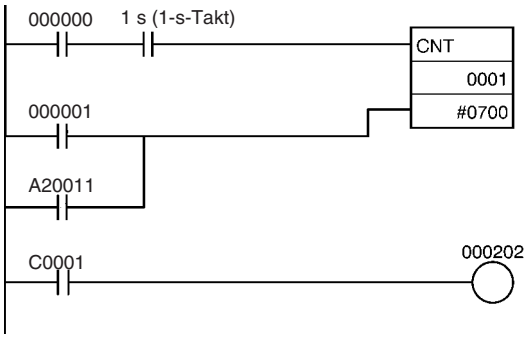


Adresse	Befehl	Operanden
000000	LD	010000
000001	LD	000001
000002	CNT	0002
		#0100
000003	LD	000000
000004	AND NOT	010000
000005	AND NOT	C0002
000006	TIM	0001
		#0050
000007	LD	T0001
000008	OUT	010000
000009	LD	C0002
000010	OUT	000201

Taktimpuls und CNT-Befehl

In diesem Beispiel zählt ein CNT-Befehl die Impulse des 1-s-Taktimpulses, um einen 700-Sekunden-Zeitgeber zu bilden.

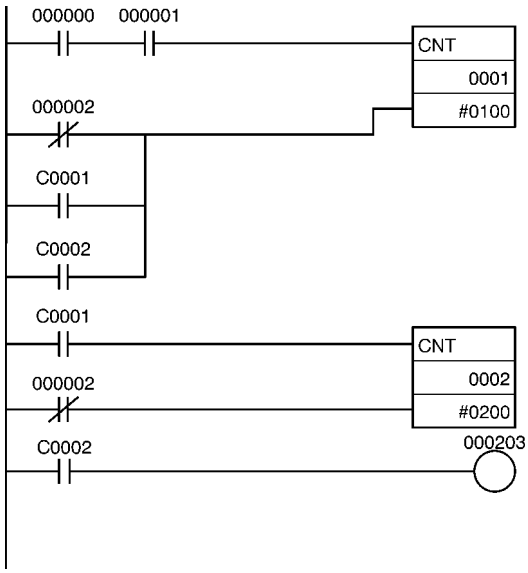
Wenn das Erster-Zyklus-Bit (A20011) eine ODER-Verknüpfung mit dem Zählerrücksetzeingang (CIO 000001) hat, wird der Zähler-Istwert bei Beginn der Programmausführung auf den Sollwert (0700) zurückgesetzt statt die Zählung ab dem vorherigen (gehaltenen) Istwert fortzusetzen.



Adresse	Befehl	Operanden
000000	LD	000000
000001	AND	1 s
000002	LD	000001
000003	OR	A20011
000004	CNT	0001
		#0700
000005	LD	C0001
000006	OUT	000202

Beispiel 2: Zweistufiger Zähler

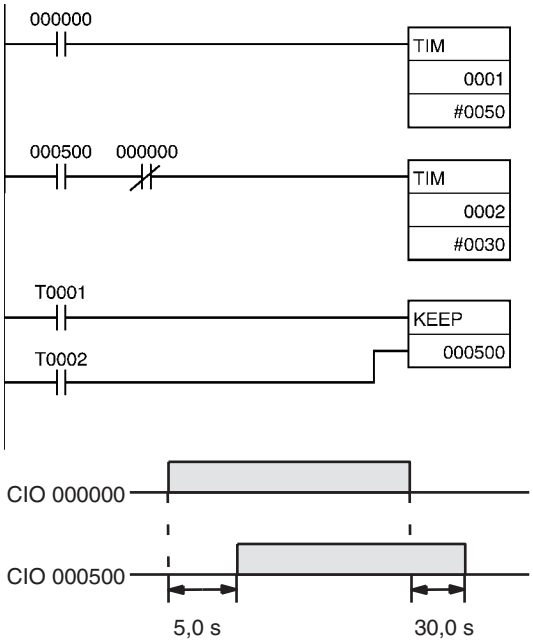
Wenn ein Sollwert größer 9999 erforderlich ist, können zwei Zähler wie im nachstehenden Bespiel gezeigt kombiniert werden. In diesem Fall werden zwei CNT-Befehle kombiniert, um einen BCD-Zähler mit einem Sollwert von 20.000 zu erstellen.



Adresse	Befehl	Operanden
000000	LD	000000
000001	AND	000001
000002	LD NOT	000002
000003	OR	C0001
000004	OR	C0002
000005	CNT	0001
		#0100
000006	LD	C0001
000007	LD NO	000002
000008	CNT	0002
		#0200
000009	LD	C0002
000010	OUT	000203

Beispiel 3: EIN/AUS-Verzögerung

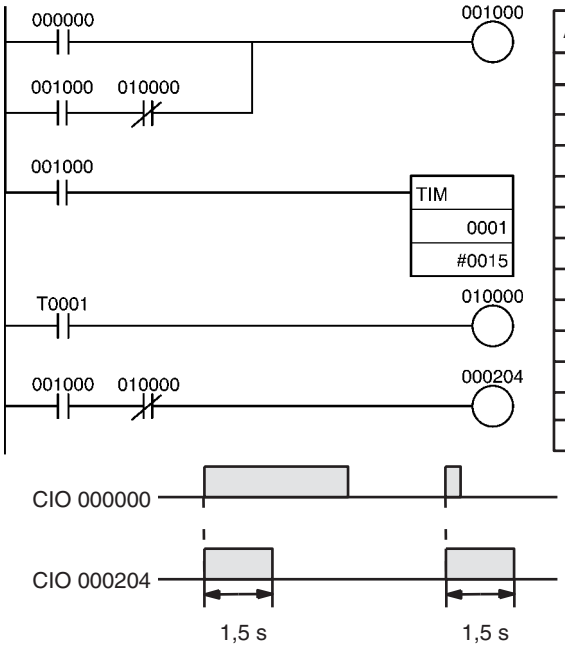
In diesem Beispiel werden zwei TIM-Zeitgeber mit KEEP(011) kombiniert, um eine EIN- und AUS-Verzögerung zu erstellen. CIO 000500 wird 5,0 Sekunden nach CIO 000000 auf EIN gesetzt und 3,0 Sekunden nach CIO 000000 auf AUS gesetzt.



Adresse	Befehl	Operanden
000000	LD	000000
000001	TIM	0001
		#0050
000002	LD	000500
000003	AND NOT	000000
000004	TIM	0002
		#0030
000005	LD	T0001
000006	LD	T0002
000007	KEEP(011)	000500

Beispiel 4: Einzelimpuls-Bit

Ein TIM-Zeitgeber kann mit OUT oder OUT NOT kombiniert werden, um zu steuern, wie lange ein bestimmtes Bit auf EIN oder AUS gesetzt wird. In diesem Beispiel wird CIO 000204 für 1,5 Sekunden (der Sollwert von T0001) auf EIN gesetzt, nachdem CIO 000000 auf EIN gesetzt wurde.



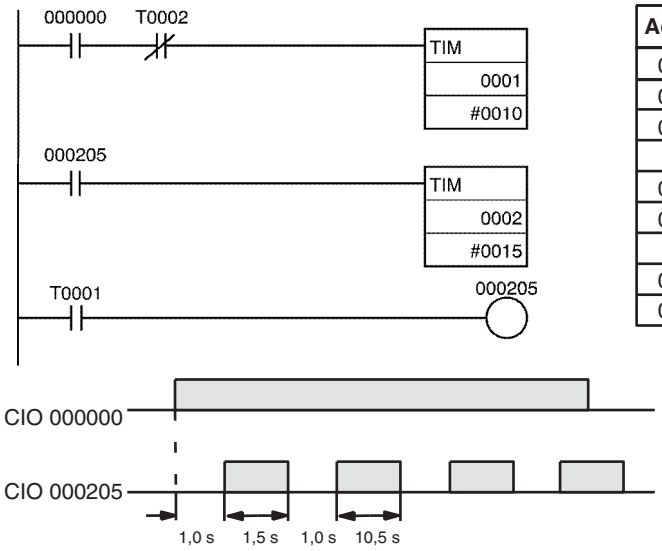
Adresse	Befehl	Operanden
000000	LD	000000
000001	LD	001000
000002	AND NOT	010000
000003	OR	000000
000004	OUT	001000
000005	LD	001000
000006	TIM	0001
		#0015
000007	LD	T0001
000008	OUT	010000
000009	LD	001000
000010	AND NOT	010000
000011	OUT	000204

Beispiel 4: Blink-Bit

Das folgende Programmbeispiel zeigt zwei Wege zur Erstellung von Blink-Bits. Das zweite Beispiel ahmt nur einen Taktimpuls nach.

Zwei TIM-Befehle

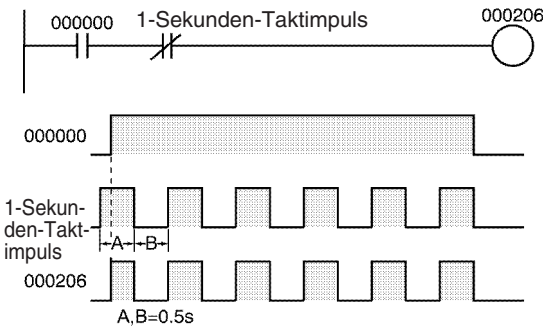
Zwei TIM-Zeitgeber können kombiniert werden, um ein Bit in regelmäßigen Intervallen auf EIN und AUS zu setzen, während die Ausführungsbedingung auf EIN gesetzt ist. In diesem Beispiel wird CIO 000205 eine Sekunde lang auf AUS und dann 1,5 Sekunden lang auf EIN gesetzt, solange CIO 000000 auf EIN gesetzt bleibt.



Adresse	Befehl	Operanden
000000	LD	000000
000001	AND	T0002
000002	TIM	0001
		#0010
000003	LD	000205
000004	TIM	0002
		#0015
000005	LD	T0001
000006	OUT	000205

Taktimpuls

Die gewünschte Ausführungsbedingung kann mit einem Taktimpuls kombiniert werden, um den Taktimpuls (0,1 s, 0,2 s oder 1,0 s) nachzuahmen.



Adresse	Befehl	Operanden
000000	LD	000000
000001	AND	1 s
000002	OUT	000206

3-6-11 Indirekte Adressierung von Zeitgeber-/Zählernummern

Zeitgeber- und Zählernummern können durch Verwendung von Indexregistern indirekt adressiert werden. Wenn Indexregister zur indirekten Adressierung verwendet werden, verwenden Sie MOVRW(561) (MOVE TIMER/COUNTER PV TO REGISTER) zum Speichern der SPS-Speicheradresse des gewünschten Zeitgeber- oder Zähler-Istwerts in das gewünschte Indexregister.

Folgende Zeitgeber und Zähler können durch Verwendung von Indexregistern indirekt adressiert werden: TIM, TIMX(550), TIMH(015), TIMHX(551), TTIM(087), TTIMX(555), TMHH(540), TMHHX(552), TIMW(813), TIMWX(816), TMHW(815), TMHWX(817), CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814) und CNTWX(818). (Es handelt sich um Zeitgeber und Zähler, die Zeitgeber- und Zählernummern verwenden.)

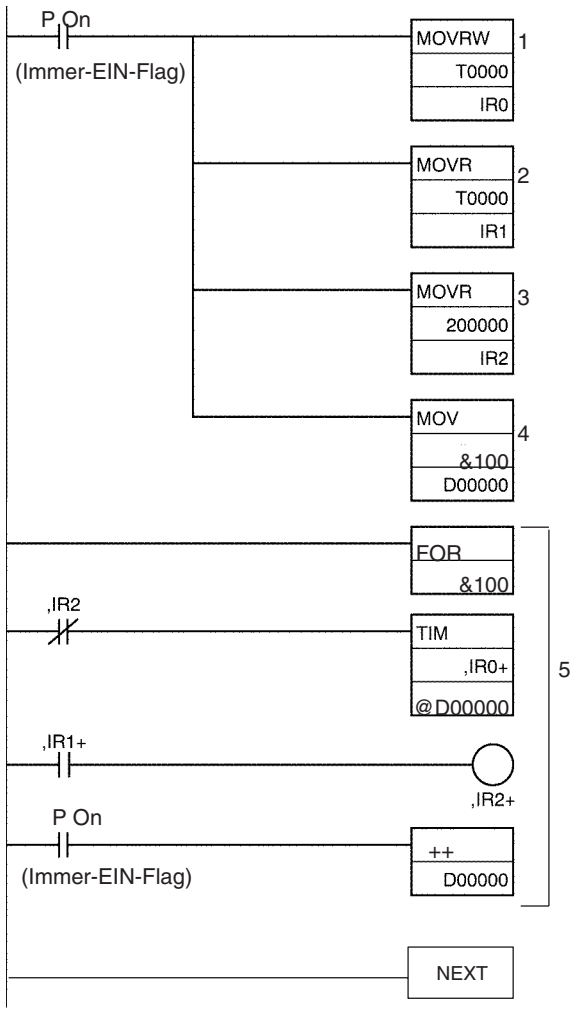
Der Zeitgeber- bzw. Zählerbefehl wird nicht ausgeführt, wenn die SPS-Speicheradresse im spezifizierten Indexregister nicht die Adresse eines Zeitgeber- oder Zähler-Istwerts ist.

Die Verwendung von Indexregistern zur indirekten Adressierung von Zeitgebern oder Zählern kann die Programmgröße reduzieren und die Flexibilität steigern. Es können z.B. gemeinsame Unterprogramme erstellt werden.

Beispiel

Das folgende Beispiel zeigt einen Programmabschnitt, der die indirekte Adressierung zur Definition sowie zum Starten von 100 Zeitgebern mit Sollwerten, die in D00100 bis D00199 enthalten sind, verwendet. IR0 enthält die SPS-Speicheradresse des Zeitgeber-Istwerts und IR1 enthält die SPS-Speicheradresse des Zeitgeber-Fertig-Bits.

DM-Adresse	Inhalt	Funktion
D00100	0010	Sollwert für T0000
D00101	0100	Sollwert für T0001
D00102	0050	Sollwert für T0002
. . .	. . .	. . .
D00199	0999	Sollwert für T0099



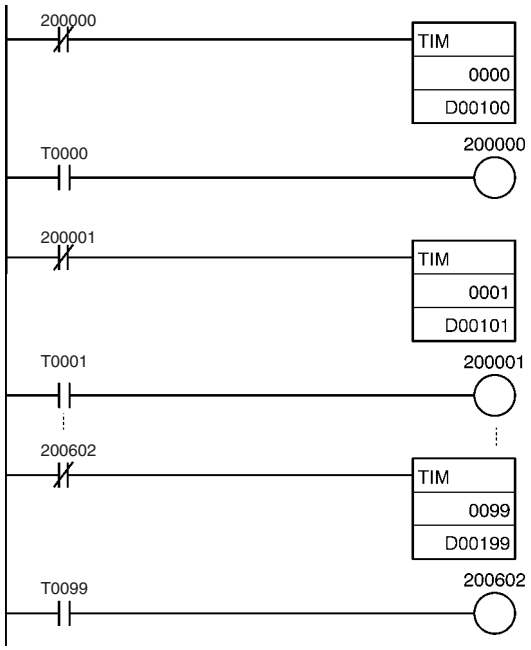
1,2,3...

1. MOVRW(561) überträgt die SPS-Speicheradresse des Istwerts für Zeitgeber T0000 in IR0. Anschließend kann IR0 anstelle der Zeitgebernummer verwendet werden.
2. MOVR(560) überträgt die SPS-Speicheradresse des Fertig-Bits für Zeitgeber T0000 in IR1.
3. MOVR(560) überträgt die SPS-Speicheradresse von CIO 200000 in IR2.
4. MOV(021) überträgt &100 in D00000 zur indirekten Adressierung der Zeitgeber-Istwerte.
5. Wenn der Inhalt von IR0, IR1, IR2 und D00000 bei jeder 100. Ausführung dieser Schleife um 1 inkrementiert wird, werden die Zeitgeber T0000 bis T0099 gestartet.

Die Schleife im obigen Programm besitzt 4 Eingangsparameter, welche zum Starten aller 100 Zeitgeber mit diesem gemeinsamen Unterprogramm verwendet werden.

IR0 Die SPS-Speicheradresse des Zeitgeber-Istwerts
IR1 Die SPS-Speicheradresse des Zeitgeber-Fertig-Bits
IR2 Die SPS-Speicheradresse des Zeitgeber-Ausführungsbedingung
D00000 Die DM-Adresse des Worts, das den Zeitgeber-Sollwert enthält

Das oben dargestellte Unterprogramm entspricht den 400 nachfolgenden Befehlen.



Adresse	Befehl	Operanden
000000	LD NOT	200000
000001	TIM	0000
		D00100
000002	LD	T0000
000003	OUT	200000
000004	LD NOT	200001
000005	TIM	0001
		D00101
000006	LD	T0001
000007	OUT	200001
000008	LD NOT	200002
000009	TIM	0002
		D00102
000010	LD	T0002
000011	OUT	200002
≈	≈	≈
000396	LD NOT	200602
000397	TIM	0099
		D00199
000398	LD	T0000
000399	OUT	200602

3-7 Vergleichsbefehle

In diesem Abschnitt werden die Befehle beschrieben, die zum Vergleichen von Daten verschiedener Längen und auf verschiedene Weisen verwendet werden.

Befehl	AWL	Funktions-code	Seite
Symbolvergleichsbefehle	=, <>, <, <=, >, >= (S, L) (LD, AND, OR)	300 bis 328	275
Zeitvergleichsbefehle	=DT, <>DT, <DT, <=DT, >DT, >=DT (LD, AND, OR)	341 bis 346	281
COMPARE	CMP	020	287
DOUBLE COMPARE	CMPL	060	290
SIGNED BINARY COMPARE	CPS	114	293
DOUBLE SIGNED BINARY COMPARE	CPSL	115	296
MULTIPLE COMPARE	MCMP	019	299
TABLE COMPARE	TCMP	085	301
BLOCK COMPARE	BCMP	068	304
EXPANDED BLOCK COMPARE	BCMP2	502	306

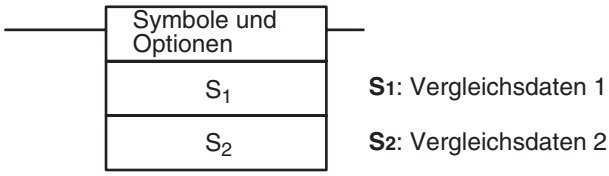
3-7-1 Symbolvergleichsbefehle (300 bis 328)

Funktion

Symbolvergleichsbefehle vergleichen zwei Werte (Konstanten und/oder den Inhalt spezifizierter Worte) und erzeugen eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Es sind Symbolvergleichsbefehle für den Vergleich von Daten mit oder ohne Vorzeichen und einer Länge von einem oder zwei Worten verfügbar.

Hinweis Detaillierte Angaben zu Symbolvergleichsbefehlen für Gleitkommadaten einfacher Genauigkeit finden Sie unter *3-15-21 Vergleichsbefehle für Gleitkommadaten einfacher Genauigkeit*, detaillierte Angaben zu Symbolvergleichsbefehlen für Gleitkommadaten doppelter Genauigkeit unter *3-16-21 Symbolvergleichsbefehle für Gleitkommadaten doppelter Genauigkeit*.

Kontaktplansymbol



Variationen

Variationen	Erzeugt in jedem Zyklus ein EIN, in dem die Vergleichsbedingung wahr ist	Symbolvergleichsbefehl
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen von Befehlen für Einzelwortdaten

Bereich	S_1	S_2
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	

Bereich	S_1	S_2
Systembereich	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_ 32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #FFFF (binär)	
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Operanden-Spezifikationen von Befehlen für Doppelwortdaten

Bereich	S_1	S_2
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	
Datenregister	---	

Bereich	S_1	S_2
Indexregister	IR0 bis IR15 (nur bei Daten ohne Vorzeichen)	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

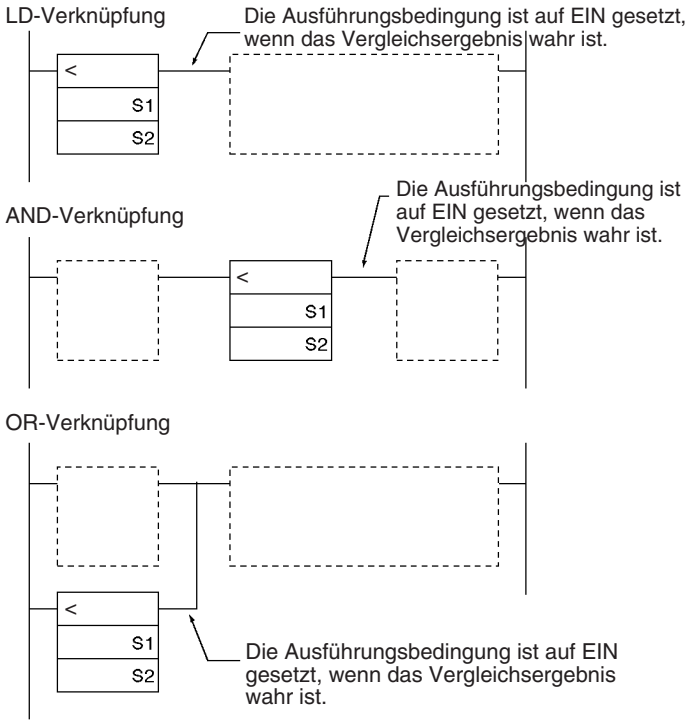
Beschreibung

Der Symbolvergleichsbefehl vergleicht S_1 und S_2 als Werte mit oder ohne Vorzeichen miteinander und erzeugt eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Im Gegensatz zu den Befehlen CMP(020) und CMPL(060) wird das Ergebnis eines Symbolvergleichsbefehls direkt als Ausführungsbedingung wiedergegeben, sodass es nicht erforderlich ist, auf das Ergebnis des Vergleichs über ein arithmetisches Flag zuzugreifen, wodurch das Programm einfacher und schneller wird.

Eingabe der Befehle

Die Symbolvergleichsbefehle werden genau wie die Befehl LD, AND und OR behandelt, um die Ausführung nachfolgender Befehle zu steuern.

Eingangs-typ	Funktion
LD	Der Befehl kann direkt mit der linken Stromschiene verbunden werden.
AND	Der Befehl kann nicht direkt mit der linken Stromschiene verbunden werden.
OR	Der Befehl kann direkt mit der linken Stromschiene verbunden werden.



Optionen

Die Symbolvergleichsbefehle können Daten mit oder ohne Vorzeichen miteinander vergleichen. Ebenso können Sie Einzel- oder Doppelwort-Werte miteinander vergleichen. Wenn keine Optionen spezifiziert werden, erfolgt der

Vergleich von vorzeichenlosen Einzelwortdaten. Durch die drei Eingangsarten und die zwei Optionen ergeben sich 72 verschiedene Symbolvergleichsbefehle.

Symbol		Option (Datenformat)	Option (Datenlänge)
=	(Gleich)	Ohne: vorzeichenlose Daten S: vorzeichenbehaftete Daten	Ohne: Einzelwortdaten L: Daten doppelter Länge
< >	(Ungleich)		
<	(Kleiner als)		
<=	(Kleiner als oder gleich)		
>	(Größer als)		
>=	(Größer als oder gleich)		

Vorzeichenlose Symbolvergleichsbefehle (also Befehle ohne die Option S) können vorzeichenlose Binär- oder BCD-Daten verarbeiten. Vorzeichenbehaftete Symbolvergleichsbefehle (also befehle mit der Option S) verarbeiten Binärdaten mit Vorzeichen.

Übersicht über Symbolvergleichsbefehle

In der folgenden Tabelle sind Funktionscodes, AWL-Kürzel, Namen und Funktionen der 72 Symbolvergleichsbefehle aufgeführt. (Für Einzelwortvergleiche C1 = S_1 und C2 = S_2; für Doppelwortvergleiche C1 = S_1+1, S_1 und C2 = S_2+1, S_2.)

Code	AWL	Name	Funktion
300	LD=	LOAD EQUAL	Wahr, wenn C1 = C2
	AND=	AND EQUAL	
	OR=	OR EQUAL	
301	LD=L	LOAD DOUBLE EQUAL	
	AND=L	AND DOUBLE EQUAL	
	OR=L	OR DOUBLE EQUAL	
302	LD=S	LOAD SIGNED EQUAL	
	AND=S	AND SIGNED EQUAL	
	OR=S	OR SIGNED EQUAL	
303	LD=SL	LOAD DOUBLE SIGNED EQUAL	
	AND=SL	AND DOUBLE SIGNED EQUAL	
	OR=SL	OR DOUBLE SIGNED EQUAL	
305	LD<>	LOAD NOT EQUAL	Wahr, wenn C1 ≠ C2
	AND<>	AND NOT EQUAL	
	OR<>	OR NOT EQUAL	
306	LD<>L	LOAD DOUBLE NOT EQUAL	
	AND<>L	AND DOUBLE NOT EQUAL	
	OR<>L	OR DOUBLE NOT EQUAL	
307	LD<>S	LOAD SIGNED NOT EQUAL	
	AND<>S	AND SIGNED NOT EQUAL	
	OR<>S	OR SIGNED NOT EQUAL	
308	LD<>SL	LOAD DOUBLE SIGNED NOT EQUAL	
	AND<>SL	AND DOUBLE SIGNED NOT EQUAL	
	OR<>SL	OR DOUBLE SIGNED NOT EQUAL	

Code	AWL	Name	Funktion
310	LD<	LOAD LESS THAN	Wahr, wenn C1 < C2
	AND<	AND LESS THAN	
	OR<	OR LESS THAN	
311	LD<L	LOAD DOUBLE LESS THAN	
	AND<L	AND DOUBLE LESS THAN	
	OR<L	OR DOUBLE LESS THAN	
312	LD<S	LOAD SIGNED LESS THAN	
	AND<S	AND SIGNED LESS THAN	
	OR<S	OR SIGNED LESS THAN	
313	LD<SL	LOAD DOUBLE SIGNED LESS THAN	
	AND<SL	AND DOUBLE SIGNED LESS THAN	
	OR<SL	OR DOUBLE SIGNED LESS THAN	
315	LD<=	LOAD LESS THAN OR EQUAL	Wahr, wenn C1 ≤ C2
	AND<=	AND LESS THAN OR EQUAL	
	OR<=	OR LESS THAN OR EQUAL	
316	LD<=L	LOAD DOUBLE LESS THAN OR EQUAL	
	AND<=L	AND DOUBLE LESS THAN OR EQUAL	
	OR<=L	OR DOUBLE LESS THAN OR EQUAL	
317	LD<=S	LOAD SIGNED LESS THAN OR EQUAL	
	AND<=S	AND SIGNED LESS THAN OR EQUAL	
	OR<=S	OR SIGNED LESS THAN OR EQUAL	
318	LD<=SL	LOAD DOUBLE SIGNED LESS THAN OR EQUAL	Wahr, wenn C1 ≤ C2
	AND<=SL	AND DOUBLE SIGNED LESS THAN OR EQUAL	
	OR<=SL	OR DOUBLE SIGNED LESS THAN OR EQUAL	
320	LD>	LOAD GREATER THAN	Wahr, wenn C1 > C2
	AND>	AND GREATER THAN	
	OR>	OR GREATER THAN	
321	LD>L	LOAD DOUBLE GREATER THAN	
	AND>L	AND DOUBLE GREATER THAN	
	OR>L	OR DOUBLE GREATER THAN	
322	LD>S	LOAD SIGNED GREATER THAN	
	AND>S	AND SIGNED GREATER THAN	
	OR>S	OR SIGNED GREATER THAN	
323	LD>SL	LOAD DOUBLE SIGNED GREATER THAN	
	AND>SL	AND DOUBLE SIGNED GREATER THAN	
	OR>SL	OR DOUBLE SIGNED GREATER THAN	
325	LD>=	LOAD GREATER THAN OR EQUAL	Wahr, wenn C1 ≥ C2
	AND>=	AND GREATER THAN OR EQUAL	
	OR>=	OR GREATER THAN OR EQUAL	
326	LD>=L	LOAD DOUBLE GREATER THAN OR EQUAL	
	AND>=L	AND DOUBLE GREATER THAN OR EQUAL	
	OR>=L	OR DOUBLE GREATER THAN OR EQUAL	
327	LD>=S	LOAD SIGNED GREATER THAN OR EQUAL	
	AND>=S	AND SIGNED GREATER THAN OR EQUAL	
	OR>=S	OR SIGNED GREATER THAN OR EQUAL	
328	LD>=SL	LOAD DBL SIGNED GREATER THAN OR EQUAL	
	AND>=SL	AND DBL SIGNED GREATER THAN OR EQUAL	
	OR>=SL	OR DBL SIGNED GREATER THAN OR EQUAL	

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Größer-als-Flag	>	EIN, wenn $S_1 > S_2$ bei Einzelwortdaten. EIN, wenn $S_1+1, S_1 > S_2+1, S_2$ bei Daten doppelter Länge. In allen anderen Fällen AUS.
Größer-oder-gleich-Flag	> =	EIN, wenn $S_1 \geq S_2$ bei Einzelwortdaten. EIN, wenn $S_1+1, S_1 \geq S_2+1, S_2$ bei Daten doppelter Länge. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn $S_1 = S_2$ bei Einzelwortdaten. EIN, wenn $S_1+1, S_1 = S_2+1, S_2$ bei Daten doppelter Länge. In allen anderen Fällen AUS.
Ungleich-Flag	=	EIN, wenn $S_1 \neq S_2$ bei Einzelwortdaten. EIN, wenn $S_1+1, S_1 \neq S_2+1, S_2$ bei Daten doppelter Länge. In allen anderen Fällen AUS.
Kleiner-als-Flag	<	EIN, wenn $S_1 < S_2$ bei Einzelwortdaten. EIN, wenn $S_1+1, S_1 < S_2+1, S_2$ bei Daten doppelter Länge. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Flag	< =	EIN, wenn $S_1 \leq S_2$ bei Einzelwortdaten. EIN, wenn $S_1+1, S_1 \leq S_2+1, S_2$ bei Daten doppelter Länge. In allen anderen Fällen AUS.
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

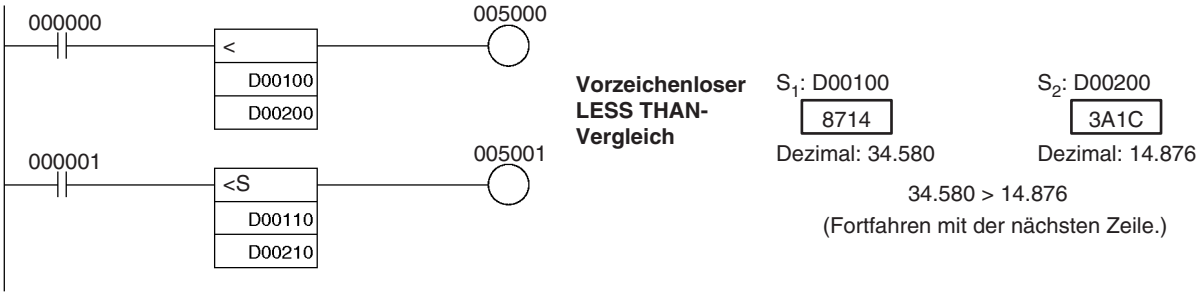
Vorsichtsmaßnahmen

Symbolvergleichsbefehle können nicht mit Verbindung zur rechten Stromschiene verwendet werden, d. h., zwischen ihnen und der rechten Stromschiene muss ein weiterer Befehl eingesetzt werden.

Beispiele

AND LESS THAN: AND<(310)

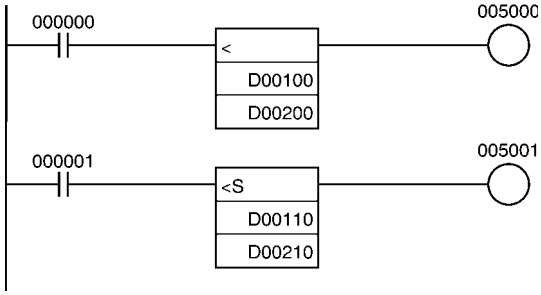
Wenn CIO 000000 in dem folgenden Beispiel EIN ist, werden die Inhalte von D00100 und D00200 als vorzeichenlose Binärdaten miteinander verglichen. Wenn der Inhalt von D00100 kleiner ist als der von D00200, wird CIO 005000 auf EIN gesetzt und die Programmausführung mit der nächsten Zeile fortgesetzt. Wenn der Inhalt von D00100 nicht kleiner ist als der von D00200 ist, wird der Rest der Befehlszeile übersprungen, und die Programmausführung fährt mit der nächsten Befehlszeile fort.



AND SIGNED LESS THAN: AND<S(312)

Wenn CIO 000001 in dem folgenden Beispiel EIN ist, werden die Inhalte von D00110 und D00210 als Binärdaten mit Vorzeichen miteinander verglichen.

Wenn der Inhalt von D00110 kleiner ist als der von D00210, wird CIO 005001 auf EIN gesetzt und die Programmausführung mit der nächsten Zeile fortgesetzt. Wenn der Inhalt von D00110 nicht kleiner ist als der von D00210 ist, wird der Rest der Befehlszeile übersprungen, und die Programmausführung fährt mit der nächsten Befehlszeile fort.



Vorzeichenbehafteter LESS THAN-Vergleich

S_1: D00110 — 8714 — Dezimal: –30.956

S_2: D00210 — 3A1C — Dezimal: 14.876

–30.956 < 14.876

(Fortfahren mit der nächsten Zeile.)

3-7-2 Zeitvergleichsbefehle (341 bis 346)

Funktion

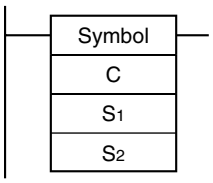
Zeitvergleichsbefehle vergleichen zwei BCD-Zeitwerte und erzeugen eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist.

Die Zeitvergleichsbefehle werden genau wie die Befehl LD, AND und OR behandelt, um die Ausführung nachfolgender Befehle zu steuern.

Diese Befehle werden nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol

LD

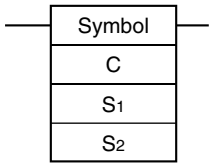


C: Steuerwort

S_1: Erstes Wort der Ist-Zeit

S_2: Erstes Wort der Vergleichszeit

AND

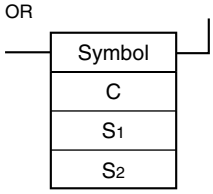


C: Steuerwort

S_1: Erstes Wort der Ist-Zeit

S_2: Erstes Wort der Vergleichszeit

OR



C: Steuerwort

S_1: Erstes Wort der Ist-Zeit

S_2: Erstes Wort der Vergleichszeit

Variationen

Variationen	Erzeugt in jedem Zyklus ein EIN, in dem die Vergleichsbedingung wahr ist	Zeitvergleichsbefehl
Spezifikation der direkten Aktualisierung		Nicht unterstützt

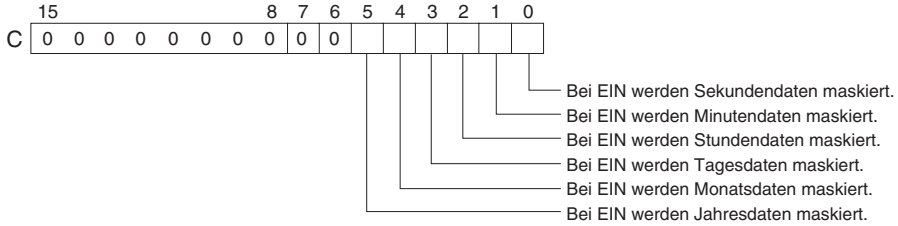
Anwendbarkeit in Programmbereichen

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

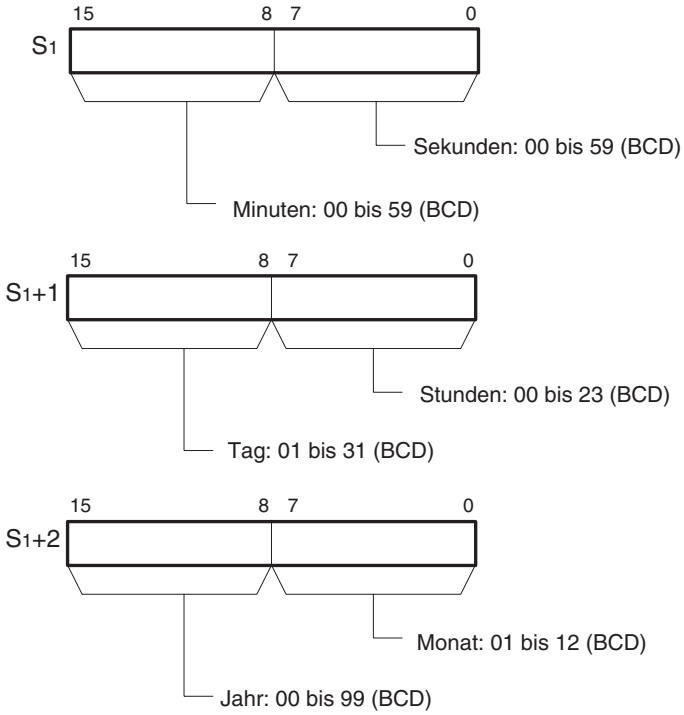
C: Steuerwort

Die Bits 00 bis 05 von C spezifizieren, ob die Zeitdaten für den Vergleich maskiert werden. Die Bits 00 bis 05 maskieren die Sekunden, Minuten, Stunden, Monate bzw. Jahre. Wenn alle 6 Werte maskiert sind, wird der Befehl nicht ausgeführt, die Ausführungsbedingung bleibt auf AUS gesetzt und das Fehler-Flag wird auf EIN gesetzt.



S_1 bis S_1+2: Istzeit-Daten

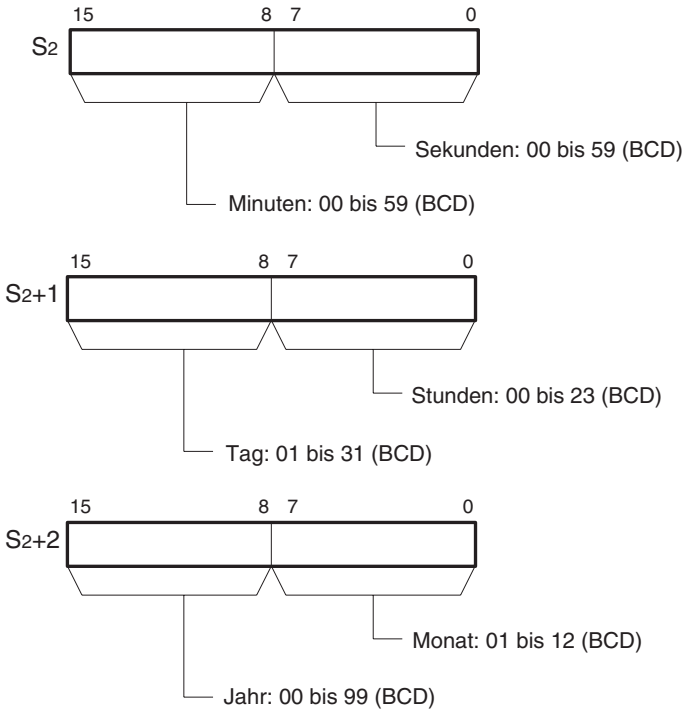
S_1 bis S_1+2 enthalten die Istzeit-Daten. S_1 bis S_1+2 müssen im gleichen Datenbereich liegen.



Hinweis Bei Verwendung der internen Uhr der CPU-Baugruppe für den Vergleich müssen Sie S_1 auf A351 setzen, um die Daten der internen Uhr der CPU-Baugruppe (A351 bis A353) zu spezifizieren.

S_2 bis S_2+2: Vergleichszeit-Daten

S_2 bis S_2+2 enthalten die Vergleichszeit-Daten. S_2 bis S_2+2 müssen im gleichen Datenbereich liegen.



Hinweis Als Jahreswert werden die beiden letzten Stellen der Jahreszahl angegeben. Die Werte 00 bis 97 werden als 2000 bis 2097 interpretiert. Die Werte 98 und 99 werden als 1998 und 1999 interpretiert.

Operanden-Spezifikationen

Bereich	C	S_1	S_2
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6141	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W509	W000 bis W510
Arbeitsbereich remanent	H000 bis H511	H000 bis H509	H000 bis H510
Systembereich	A448 bis A959	A000 bis A957	A000 bis A958
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4093	T0000 bis T4094
Zählerbereich	C0000 bis C4095	C0000 bis C4093	C0000 bis C4094
DM-Bereich	D00000 bis D32767	D00000 bis D32765	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32765	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32765 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	

Bereich	C	S_1	S_2
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	Siehe vorherige Seite.	Siehe vorherige Seite.	---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

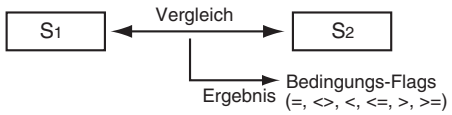
Der Zeitvergleichsbefehl vergleicht die unmaskierten Werte (das entsprechende Bit von C ist auf 0 gesetzt) der Istzeit-Daten in S_1 bis S_1+2 mit den Vergleichszeitdaten in S_2 bis S_2+2 und erzeugt eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Gleichzeitig wird das Ergebnis des Zeitvergleichsbefehls in den arithmetischen Flags (=, <>, <, <=, >, >=) wiedergegeben.

Es gibt 18 mögliche Kombinationen von Zeitvergleichsbefehlen.

Alle Werte, die im Steuerwort (C) maskiert sind, werden nicht in den Vergleich einbezogen.

Der folgenden Tabelle ist der EIN/AUS-Status der einzelnen Flags bei den verschiedenen Vergleichsergebnissen zu entnehmen.

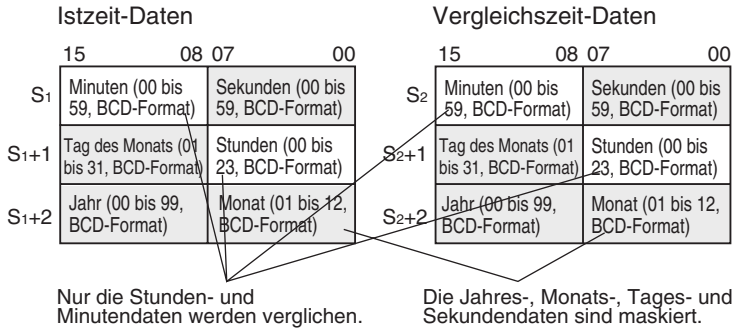
Ergebnis	Flag-Status					
	=	<>	<	<=	>	>=
$S_1 = S_2$	EIN	AUS	AUS	EIN	AUS	EIN
$S_1 > S_2$	AUS	EIN	AUS	AUS	EIN	EIN
$S_1 < S_2$	AUS	EIN	EIN	EIN	AUS	AUS



Maskieren von Zeitwerten

Die einzelnen Komponenten von Zeitwerten können maskiert und dadurch vom Vergleichsvorgang ausgeschlossen werden. Zum Maskieren eines Zeitwerts setzen Sie das entsprechende Bit im Steuerwort (C) auf 1. Die Bits 00 bis 05 von C maskieren die Sekunden, Minuten, Stunden, Monate bzw. Jahre.

Beispiel:
Wenn C = 39 hex ist, haben die niederwertigsten 6 Bits von C den Status 111001 (Jahr = 1, Monat = 1, Tag = 1, Stunden = 0, Minuten = 0 und Sekunden = 1), also werden nur die Stunden- und Minutenwerte verglichen. Diese Maskierungseinstellungen können verwendet werden, um einen bestimmten Vorgang jeden Tag zu einer bestimmten Uhrzeit (Stunden und Minuten) auszuführen.



Vorherige Datenvergleichsbefehle haben Daten in 16-Bit-Einheiten verglichen. Die Zeitvergleichsbefehle sind auf den Vergleich von 8-Bit-Zeitwerten beschränkt.

In der folgenden Tabelle ist die Struktur des internen Kalender-/Uhrbereichs der CPU-Baugruppe dargestellt.

Adressen	Inhalt
A35100 bis A35107	Sekunden (00 bis 59, BCD-Format)
A35108 bis A35115	Minuten (00 bis 59, BCD-Format)
A35200 bis A35207	Stunden (00 bis 23, BCD-Format)
A35208 bis A35215	Tag des Monats (01 bis 31, BCD-Format)
A35300 bis A35307	Monat (01 bis 12, BCD-Format)
A35308 bis A35315	Jahr (00 bis 99, BCD-Format)

Werte im Kalender-/Uhrbereich können mit einem Programmiergerät (einschließlich einer Programmierkonsole) und dem Befehl DATE(735) oder dem FINS-Befehl “CLOCK WRITE” (0702 hex) gesetzt werden.

Übersicht über Zeitvergleichsbefehle

In der folgenden Tabelle sind Funktionscodes, AWL-Kürzel, Namen und Funktionen der 18 Zeitvergleichsbefehle aufgeführt.

Code	AWL	Name	Funktion
341	LD=DT	LOAD EQUAL	Wahr, wenn S1 = S2
	AND=DT	AND EQUAL	
	OR=DT	OR EQUAL	
342	LD<>DT	LOAD NOT EQUAL	Wahr, wenn S1 ≠ S2
	AND<>DT	AND NOT EQUAL	
	OR<>DT	OR NOT EQUAL	
343	LD<DT	LOAD LESS THAN	Wahr, wenn S1 < S2
	AND<DT	AND LESS THAN	
	OR<DT	OR LESS THAN	
344	LD<=DT	LOAD LESS THAN OR EQUAL	Wahr, wenn S1 ≤ S2
	AND<=DT	AND LESS THAN OR EQUAL	
	OR<=DT	OR LESS THAN OR EQUAL	
345	LD>DT	LOAD GREATER THAN	Wahr, wenn S1 > S2
	AND>DT	AND GREATER THAN	
	OR>DT	OR GREATER THAN	
346	LD>=DT	LOAD GREATER THAN OR EQUAL	Wahr, wenn S1 ≥ S2
	AND>=DT	AND GREATER THAN OR EQUAL	
	OR>=DT	OR GREATER THAN OR EQUAL	

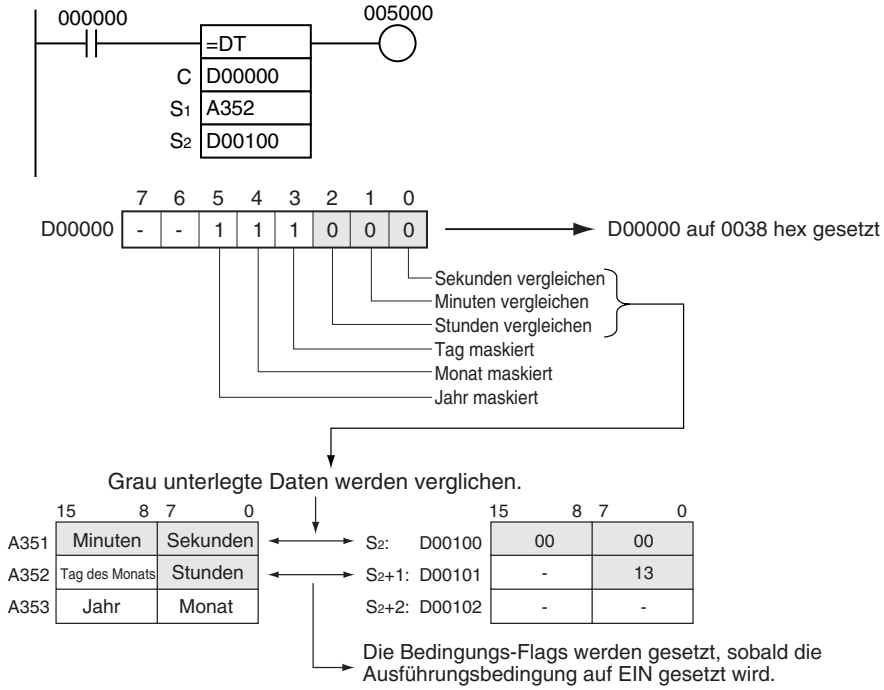
Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn alle 6 Maskierungsbits (Bits 00 bis 05 von C) auf EIN gesetzt sind. In allen anderen Fällen AUS.
Größer-als-Flag	>	EIN, wenn $S_1 > S_2$. In allen anderen Fällen AUS.
Größer-oder-gleich-Flag	> =	EIN, wenn $S_1 \geq S_2$. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn $S_1 = S_2$. In allen anderen Fällen AUS.
Ungleich-Flag	=	EIN, wenn $S_1 \neq S_2$. In allen anderen Fällen AUS.
Kleiner-als-Flag	<	EIN, wenn $S_1 < S_2$. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Flag	< =	EIN, wenn $S_1 \leq S_2$. In allen anderen Fällen AUS.
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Vorsichtsmaßnahmen Zeitvergleichsbefehle können nicht mit Verbindung zur rechten Stromschiene verwendet werden, d. h., zwischen ihnen und der rechten Stromschiene muss ein weiterer Befehl eingesetzt werden.

Beispiel Wenn CIO 000000 auf EIN gesetzt und die Zeit 13:00:00 ist, wird CIO 005000 auf EIN gesetzt. Die Inhalte von A351 bis A353 (die internen Kalender-/Uhrdaten der CPU-Baugruppe) werden als Istzeit-Daten und die Inhalte von D00100 bis D00102 als Vergleichszeit-Daten verwendet. Die Jahres-, Monats- und Tageswerte sind maskiert, sodann nur die Stunden-, Minuten- und Sekundendaten verglichen werden.

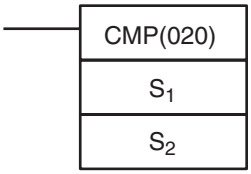


3-7-3 COMPARE: CMP(020)

Funktion

Vergleicht zwei vorzeichenlose Binärwerte (Konstanten und/oder den Inhalt spezifizierter Worte) und gibt das Ergebnis an die arithmetischen Flags im Systembereich aus.

Kontaktplansymbol



S1: Vergleichsdaten 1

S2: Vergleichsdaten 2

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	CMP(020)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!CMP(020)

Hinweis Direkte Aktualisierung wird von CS1D CPU-Baugruppen für Duplex-CPU-Systeme nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

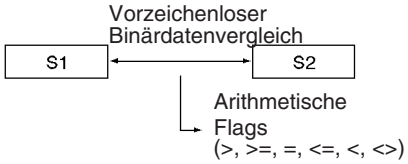
Operanden-Spezifikationen

Bereich	S_1	S_2
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #FFFF (binär)	
Datenregister	DR0 bis DR15	

Bereich	S_1	S_2
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl CMP(020) vergleicht die vorzeichenlosen Binärdaten in S_1 und S_2 miteinander und gibt das Ergebnis an die arithmetischen Flags (Größer-als-, Größer-oder-gleich-, Gleich-, Kleiner-oder-gleich-, Kleiner-als- und Ungleich-Flag) im Systembereich aus.



Bedingungs-Flag-Status

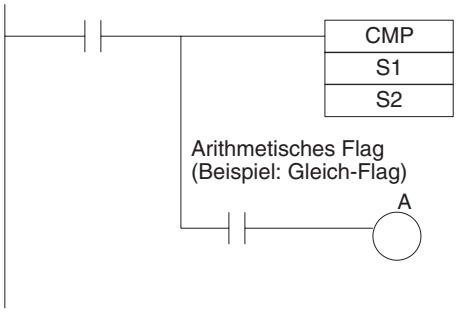
In der folgenden Tabelle ist der Status der einzelnen arithmetischen Flags nach der Ausführung von CMP(020) zu entnehmen. (Die Statusangabe "---" bedeutet, dass das Flag auf EIN oder AUS gesetzt sein kann.)

Ergebnis von CMP(020)	Flag-Status					
	>	> =	=	< =	<	< >
$S_1 > S_2$	EIN	EIN	AUS	AUS	AUS	EIN
$S_1 = S_2$	AUS	EIN	EIN	EIN	AUS	AUS
$S_1 < S_2$	AUS	AUS	AUS	EIN	EIN	EIN

Verwendung der Ergebnisse von CMP(020) im Programm

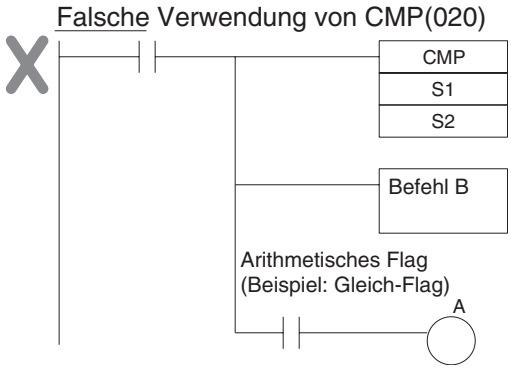
Nach der Ausführung von CMP(020) wird das Ergebnis durch die arithmetischen Flags wiedergegeben. Steuern Sie den gewünschten Ausgang oder Befehl auf der rechten Seite mit einer Verzweigung von der gleichen Eingangsbedingung, die den Befehl CMP(020) steuert (siehe folgende Abbildung). In diesem Fall werden das Gleich-Flag und der Ausgang A auf EIN gesetzt, wenn $S_1 = S_2$.

Korrekte Verwendung von CMP(020)



Verwendung der Ergebnisse von CMP(020) im Programm

Programmieren Sie keinen anderen Befehl zwischen CMP(020) und dem durch das arithmetische Flag gesteuerten Befehl, da der andere Befehl des Status des arithmetischen Flags ändern könnte. In diesem Fall könnte das Ergebnis von Befehl B die Ergebnisse von CMP(020) ändern.



Die Befehlsvariante mit direkter Aktualisierung (!CMP(020)) kann mit Worten verwendet werden, die externen Eingängen zugewiesen und in S_1 und/oder S_2 spezifiziert sind. Bei Ausführung von !CMP(020) wird die Eingangsaktualisierung für das in S_1 und/oder S_2 spezifizierte, dem externen Eingang zugewiesene Wort durchgeführt, und der aktualisierte Wert wird für den Vergleich verwendet. (Die direkte Aktualisierung ist nicht bei Eingängen möglich, die Multipoint-E/A-Baugruppen oder in einem Remote-Slave-Block installierten Baugruppen zugewiesen sind.)

Flags

Name	Bezeichnung bei CX-Programmer	Bezeichnung bei Programmierkonsole	Funktion
Fehler-Flag	P_ER	ER	AUS oder unverändert (siehe Hinweis)
Größer-als-Flag	P_GT	>	EIN, wenn $S_1 > S_2$. In allen anderen Fällen AUS.
Größer-oder-gleich-Flag	P_GE	> =	EIN, wenn $S_1 \geq S_2$. In allen anderen Fällen AUS.
Gleich-Flag	P_EQ	=	EIN, wenn $S_1 = S_2$. In allen anderen Fällen AUS.
Ungleich-Flag	P_NE	=	EIN, wenn $S_1 \neq S_2$. In allen anderen Fällen AUS.
Kleiner-als-Flag	P_LT	<	EIN, wenn $S_1 < S_2$. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Flag	P_LE	< =	EIN, wenn $S_1 \leq S_2$. In allen anderen Fällen AUS.
Negativ-Flag	P_N	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

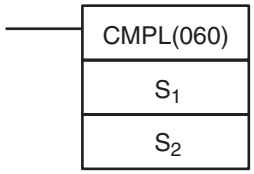
Vorsichtsmaßnahmen Programmieren Sie keine anderen Befehle zwischen CMP(020) und einer Eingangsbedingung, die auf das Ergebnis von CMP(020) zugreift, da der andere Befehl den Status der arithmetischen Flags ändern könnte.

3-7-4 DOUBLE COMPARE: CMPL(060)

Funktion

Vergleicht zwei vorzeichenlose 2-Wort-Binärwerte (Konstanten und/oder den Inhalt spezifizierter Worte) und gibt das Ergebnis an die arithmetischen Flags im Systembereich aus.

Kontaktplansymbol



S1: Vergleichsdaten 1

S2: Vergleichsdaten 2

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	CMPL(060)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

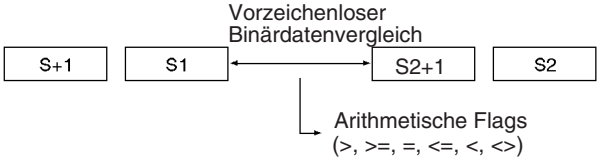
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S_1	S_2
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	
Datenregister	---	
Indexregister	IR0 bis IR15	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl CMPL(060) vergleicht die vorzeichenlosen Binärdaten in S_1+1, S_1 und S_2+1, S_2 miteinander und gibt das Ergebnis an die arithmetischen Flags (Größer-als-, Größer-oder-gleich-, Gleich-, Kleiner-oder-gleich-, Kleiner-als- und Ungleich-Flag) im Systembereich aus.



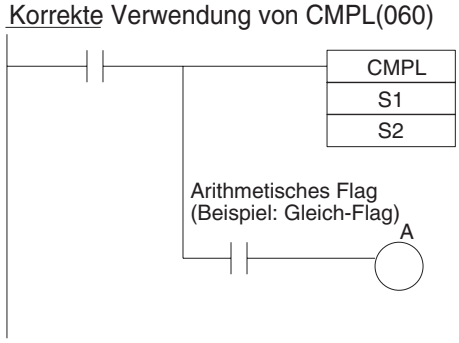
Status der arithmetischen Flags

In der folgenden Tabelle ist der Status der einzelnen arithmetischen Flags nach der Ausführung von CMPL(060) zu entnehmen. (Die Statusangabe "---" bedeutet, dass das Flag auf EIN oder AUS gesetzt sein kann.)

Ergebnis von CMPL(060)	Flag-Status					
	>	> =	=	< =	<	< >
S_1+1, S_1 > S_2+1, S_2	EIN	EIN	AUS	AUS	AUS	EIN
S_1+1, S_1 = S_2+1, S_2	AUS	EIN	EIN	EIN	AUS	AUS
S_1+1, S_1 < S_2+1, S_2	AUS	AUS	AUS	EIN	EIN	EIN

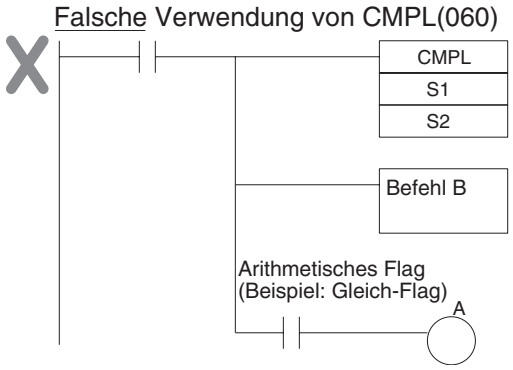
Verwendung der Ergebnisse von CMPL(060) im Programm

Nach der Ausführung von CMPL(060) wird das Ergebnis durch die arithmetischen Flags wiedergegeben. Steuern Sie den gewünschten Ausgang oder Befehl auf der rechten Seite mit einer Verzweigung von der gleichen Eingangsbedingung, die den Befehl CMPL(060) steuert (siehe folgende Abbildung). In diesem Fall werden das Gleich-Flag und Ausgang A auf EIN gesetzt, wenn S_1+1, S_1 = S_2+1, S_2.



Verwendung der Ergebnisse von CMPL(060) im Programm

Programmieren Sie keinen anderen Befehl zwischen CMPL(060) und dem durch das arithmetische Flag gesteuerten Befehl, da der andere Befehl des Status des arithmetischen Flags ändern könnte. In diesem Fall könnte das Ergebnis von Befehl B die Ergebnisse von CMPL(060) ändern.



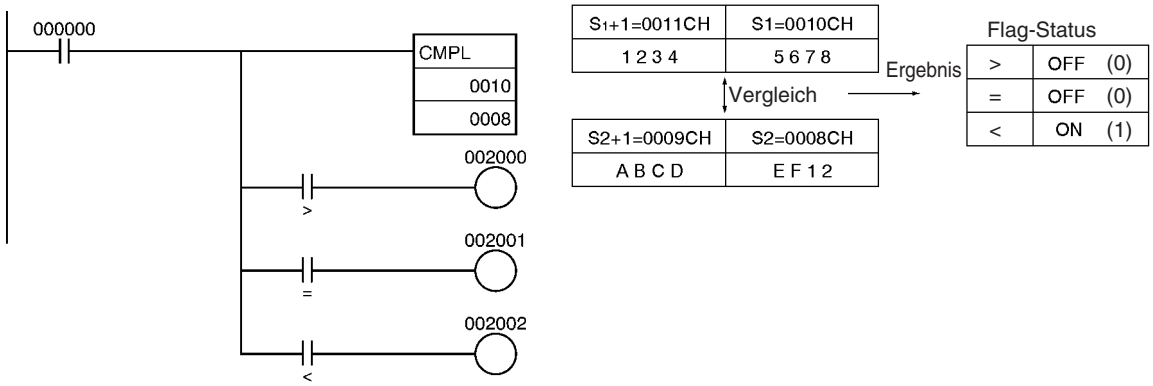
Flags

Name	Bezeichnung bei CX-Programmer	Bezeichnung bei Programmierkonsole	Funktion
Fehler-Flag	P_ER	ER	AUS oder unverändert (siehe Hinweis)
Größer-als-Flag	P_GT	>	EIN, wenn S_1+1, $S_1 > S_2+1$, S_2. In allen anderen Fällen AUS.
Größer-oder-gleich-Flag	P_GE	> =	EIN, wenn S_1+1, $S_1 \geq S_2+1$, S_2. In allen anderen Fällen AUS.
Gleich-Flag	P_EQ	=	EIN, wenn S_1+1, $S_1 = S_2+1$, S_2. In allen anderen Fällen AUS.
Ungleich-Flag	P_NE	<>	EIN, wenn S_1+1, $S_1 \neq S_2+1$, S_2. In allen anderen Fällen AUS.
Kleiner-als-Flag	P_LT	<	EIN, wenn S_1+1, $S_1 < S_2+1$, S_2. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Flag	P_LE	< =	EIN, wenn S_1+1, $S_1 \leq S_2+1$, S_2. In allen anderen Fällen AUS.
Negativ-Flag	P_N	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Vorsichtsmaßnahmen Programmieren Sie keine anderen Befehle zwischen CMPL(060) und einer Eingangsbedingung, die auf das Ergebnis von CMPL(060) zugreift, da der andere Befehl den Status der arithmetischen Flags ändern könnte.

Beispiel Wenn CIO 000000 in dem folgenden Beispiel EIN ist, werden die 8-stelligen vorzeichenlosen Binärdaten in CIO 0011 und CIO 0010 mit den 8-stelligen vorzeichenlosen Binärdaten in CIO 0009 und CIO 0008 verglichen, und das Ergebnis wird an die arithmetischen Flags ausgegeben. Die in den Größer-als-, Gleich- und Kleiner-als-Flags aufgezeichneten Ergebnisse werden unmittelbar in CIO 000200 (Größer als), CIO 000201 (Gleich) und CIO 000202 (Kleiner als) gespeichert.

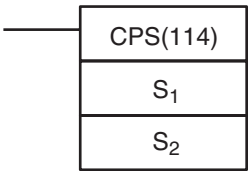


3-7-5 SIGNED BINARY COMPARE: CPS(114)

Funktion

Vergleicht zwei Binärwerte mit Vorzeichen (Konstanten und/oder den Inhalt spezifizierter Worte) miteinander und gibt das Ergebnis an die arithmetischen Flags im Systembereich aus.

Kontaktplansymbol



S1: Vergleichsdaten 1

S2: Vergleichsdaten 2

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	CPS(114)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!CPS(114)

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

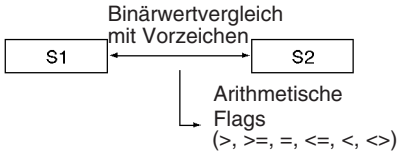
Operanden-Spezifikationen

Bereich	S_1	S_2
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	

Bereich	S_1	S_2
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #FFFF (binär)	
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl CPS(114) vergleicht die vorzeichenbehafteten Binärdaten in S_1 und S_2 miteinander und gibt das Ergebnis an die arithmetischen Flags (Größer-als-, Größer-oder-gleich-, Gleich-, Kleiner-oder-gleich-, Kleiner-als- und Ungleich-Flag) im Systembereich aus.



Hinweis

Der Befehl CPS(114) behandelt die Daten in S_1 und S_2 als Binärdaten mit Vorzeichen in Bereich zwischen 8000 und 7FFF hex (–32.768 bis 32.767 dezimal).

Status der arithmetischen Flags

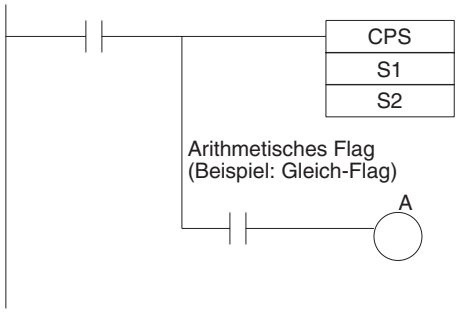
In der folgenden Tabelle ist der Status der einzelnen arithmetischen Flags nach der Ausführung von CPS(114) zu entnehmen. (Die Statusangabe „---“ bedeutet, dass das Flag auf EIN oder AUS gesetzt sein kann.)

Ergebnis von CPS(114)	Flag-Status					
	>	> =	=	< =	<	< >
$S_1 > S_2$	EIN	EIN	AUS	AUS	AUS	EIN
$S_1 = S_2$	AUS	EIN	EIN	EIN	AUS	AUS
$S_1 < S_2$	AUS	AUS	AUS	EIN	EIN	EIN

Verwendung der Ergebnisse von CPS(114) im Programm

Nach der Ausführung von CPS(114) wird das Ergebnis durch die arithmetischen Flags wiedergegeben. Steuern Sie den gewünschten Ausgang oder Befehl auf der rechten Seite mit einer Verzweigung von der gleichen Eingangsbedingung, die den Befehl CPS(114) steuert (siehe folgende Abbildung). In diesem Fall werden das Gleich-Flag und Ausgang A auf EIN gesetzt, wenn $S_1 = S_2$.

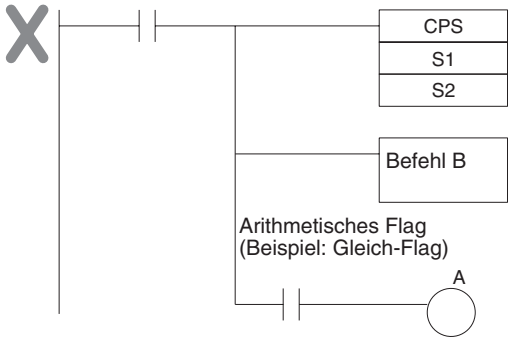
Korrekte Verwendung von CPS(114)



Verwendung der Ergebnisse von CPS(114) im Programm

Programmieren Sie keinen anderen Befehl zwischen CPS(114) und dem durch das arithmetische Flag gesteuerten Befehl, da der andere Befehl des Status des arithmetischen Flags ändern könnte. In diesem Fall könnte das Ergebnis von Befehl B die Ergebnisse von CPS(114) ändern.

Falsche Verwendung von CPS(114)



Die Befehlsvariante mit direkter Aktualisierung (!CPS(114)) kann mit Worten verwendet werden, die externen Eingängen zugewiesen und in S_1 und/oder S_2 spezifiziert sind. Bei Ausführung von !CPS(114) wird die Eingangsaktualisierung für das in S_1 und/oder S_2 spezifizierte, dem externen Eingang zugewiesene Wort durchgeführt, und der aktualisierte Wert wird für den Vergleich verwendet. (Die direkte Aktualisierung ist nicht bei Eingängen möglich, die Multipoint-E/A-Baugruppen oder in einem Slave-Block installierten Baugruppen zugewiesen sind.)

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Größer-als-Flag	>	EIN, wenn $S_1 > S_2$. In allen anderen Fällen AUS.
Größer-oder-gleich-Flag	> =	EIN, wenn $S_1 \geq S_2$. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn $S_1 = S_2$. In allen anderen Fällen AUS.
Ungleich-Flag	<>	EIN, wenn $S_1 \neq S_2$. In allen anderen Fällen AUS.
Kleiner-als-Flag	<	EIN, wenn $S_1 < S_2$. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Flag	< =	EIN, wenn $S_1 \leq S_2$. In allen anderen Fällen AUS.
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

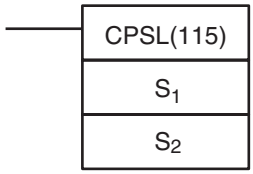
Vorsichtsmaßnahmen Programmieren Sie keine anderen Befehle zwischen CPS(114) und einer Eingangsbedingung, die auf das Ergebnis von CPS(114) zugreift, da der andere Befehl den Status der arithmetischen Flags ändern könnte.

3-7-6 DOUBLE SIGNED BINARY COMPARE: CPSL(115)

Funktion

Vergleicht zwei Doppelwort-Binärwerte mit Vorzeichen (Konstanten und/oder den Inhalt spezifizierter Worte) miteinander und gibt das Ergebnis an die arithmetischen Flags im Systembereich aus.

Kontaktplansymbol



S1: Vergleichsdaten 1

S2: Vergleichsdaten 2

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CPSL(115)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

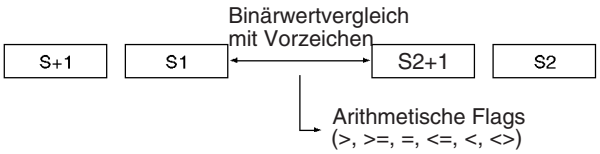
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S_1	S_2
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl CPSL(115) vergleicht die Doppelwort-Binärdaten mit Vorzeichen in S_1+1, S_1 und S_2+1, S_2 miteinander und gibt das Ergebnis an die arithmetischen Flags (Größer-als-, Größer-oder-gleich-, Gleich-, Kleiner-oder-gleich-, Kleiner-als- und Ungleich-Flag) im Systembereich aus.



Hinweis

Der Befehl CPSL(115) behandelt die Daten in S_1 und S_2 als Doppelwort-Binärdaten mit Vorzeichen in Bereich zwischen 8000 0000 und 7FFF FFFF hex (–2.147.483.648 bis 2.147.483.647 dezimal).

Status der arithmetischen Flags

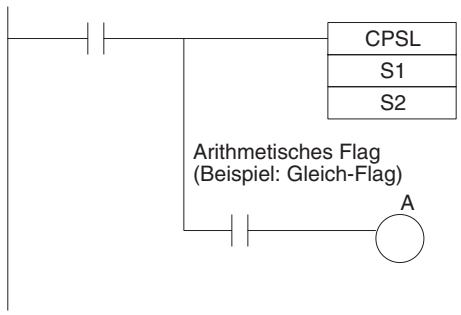
In der folgenden Tabelle ist der Status der einzelnen arithmetischen Flags nach der Ausführung von CPSL(115) zu entnehmen. (Die Statusangabe „---“ bedeutet, dass das Flag auf EIN oder AUS gesetzt sein kann.)

Ergebnis von CPSL(115)	Flag-Status					
	>	> =	=	< =	<	< >
S_1+1, S_1 > S_2+1, S_2	EIN	EIN	AUS	AUS	AUS	EIN
S_1+1, S_1 = S_2+1, S_2	AUS	EIN	EIN	EIN	AUS	AUS
S_1+1, S_1 < S_2+1, S_2	AUS	AUS	AUS	EIN	EIN	EIN

Verwendung der Ergebnisse von CPSL(115) im Programm

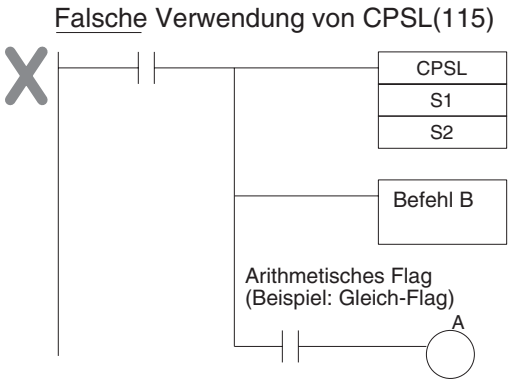
Nach der Ausführung von CPSL(115) wird das Ergebnis durch die arithmetischen Flags wiedergegeben. Steuern Sie den gewünschten Ausgang oder Befehl auf der rechten Seite mit einer Verzweigung von der gleichen Eingangsbedingung, die den Befehl CPSL(115) steuert (siehe folgende Abbildung). In diesem Fall werden das Gleich-Flag und Ausgang A auf EIN gesetzt, wenn S_1+1, S_1 = S_2+1, S_2.

Korrekte Verwendung von CPSL(115)



Verwendung der Ergebnisse von CPSL(115) im Programm

Programmieren Sie keinen anderen Befehl zwischen CPSL(115) und dem durch das arithmetische Flag gesteuerten Befehl, da der andere Befehl des Status des arithmetischen Flags ändern könnte. In diesem Fall könnte das Ergebnis von Befehl B die Ergebnisse von CPSL(115) ändern.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Größer-als-Flag	>	EIN, wenn $S_1+1, S_1 > S_2+1, S_2$. In allen anderen Fällen AUS.
Größer-oder-gleich-Flag	> =	EIN, wenn $S_1+1, S_1 \geq S_2+1, S_2$. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn $S_1+1, S_1 = S_2+1, S_2$. In allen anderen Fällen AUS.
Ungleich-Flag	=	EIN, wenn $S_1+1, S_1 \neq S_2+1, S_2$. In allen anderen Fällen AUS.
Kleiner-als-Flag	<	EIN, wenn $S_1+1, S_1 < S_2+1, S_2$. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Flag	< =	EIN, wenn $S_1+1, S_1 \leq S_2+1, S_2$. In allen anderen Fällen AUS.
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

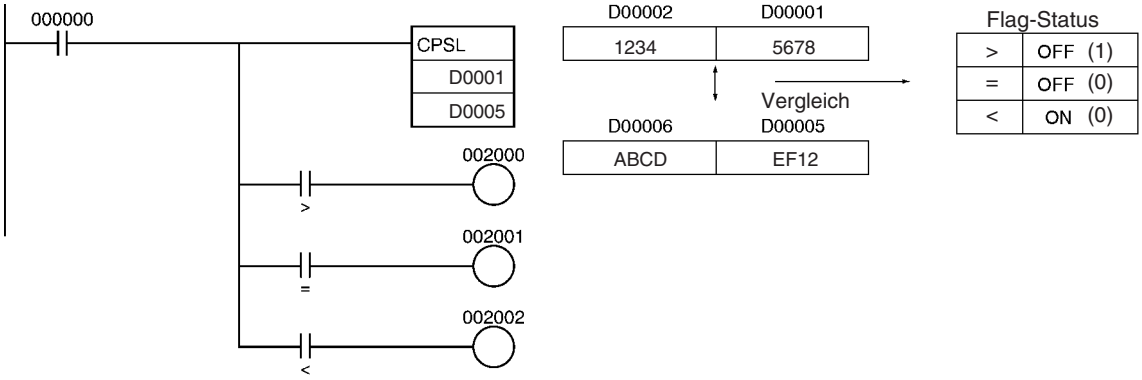
Vorsichtsmaßnahmen

Programmieren Sie keine anderen Befehle zwischen CPSL(115) und einer Eingangsbedingung, die auf das Ergebnis von CPSL(115) zugreift, da der andere Befehl den Status der arithmetischen Flags ändern könnte.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, werden die 8-stelligen Binärdaten mit Vorzeichen in D00002 und D00001 mit den 8-stelligen Binärdaten mit Vorzeichen in D00006 und D00005 verglichen, und das Ergebnis wird in die arithmetischen Flags ausgegeben.

- Wenn der Inhalt von D00002 und D00001 größer ist als der von D00006 und D00005, wird das Größer-als-Flag auf EIN gesetzt, wodurch wiederum CIO 002000 auf EIN gesetzt wird.
- Wenn der Inhalt von D00002 und D00001 gleich dem von D00006 und D00005 ist, wird das Gleich-Flag auf EIN gesetzt, wodurch wiederum CIO 002001 auf EIN gesetzt wird.
- Wenn der Inhalt von D00002 und D00001 kleiner ist als der von D00006 und D00005, wird das Kleiner-als-Flag auf EIN gesetzt, wodurch wiederum CIO 002002 auf EIN gesetzt wird.

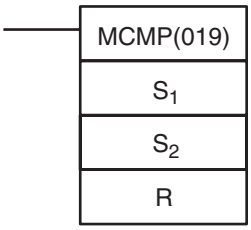


3-7-7 MULTIPLE COMPARE: MCMP(019)

Funktion

Vergleicht 16 aufeinander folgende Worte mit 16 anderen aufeinander folgenden Worten und setzt das entsprechende Bit im Ergebniswort auf EIN, wenn die Inhalte zweier verglichener Worte **nicht** gleich sind.

Kontaktplansymbol



S1: Erstes Wort von Satz 1
S2: Erstes Wort von Satz 2
R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MCMP(019)
	Ausführung einmal bei steigender Flanke	@MCMP(019)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S_1: Erstes Wort von Satz 1

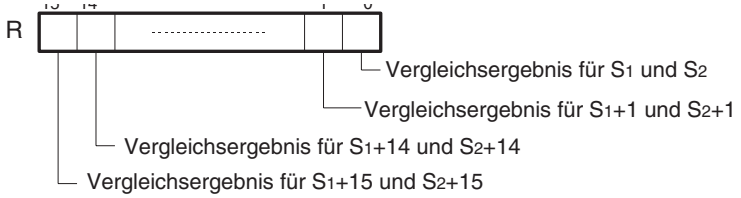
Spezifiziert das Anfangswort des ersten 16-Worte-Bereichs. S_1 bis S_1+15 müssen im gleichen Datenbereich liegen.

S_2: Erstes Wort von Satz 2

Spezifiziert das Anfangswort des zweiten 16-Worte-Bereichs. S_2 bis S_2+15 müssen im gleichen Datenbereich liegen.

R: Ergebniswort

Die einzelnen Bits von R enthalten des Ergebnis eines Vergleichs zwischen zwei einander entsprechenden Worten in den 16-Worte-Sätzen. Bit n von R (n = 00 bis 15) enthält das Ergebnis des Vergleichs zwischen den Worten S_1+n und S_2+n.



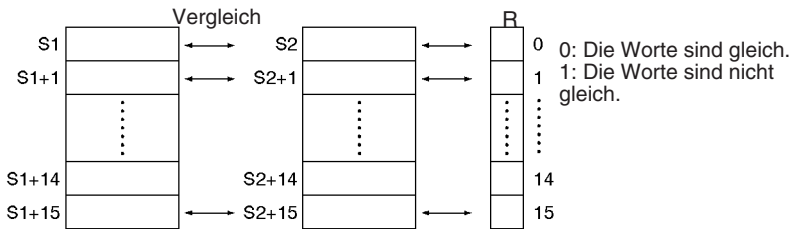
Operanden-Spezifikationen

Bereich	S_1	S_2	R
CIO-Bereich	CIO 0000 bis CIO 6128		CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W496		W000 bis W511
Arbeitsbereich remanent	H000 bis H496		H000 bis H511
Systembereich	A000 bis A944		A448 bis A959
Zeitgeberbereich	T0000 bis T4080		T0000 bis T4095
Zählerbereich	C0000 bis C4080		C0000 bis C4095
DM-Bereich	D00000 bis D32752		D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32752		E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis 32752 (n = 0 bis C)		En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl MCMP(019) vergleicht die Inhalte der 16 Worte S_1 bis S_1+15 mit den Inhalten der 16 Worte S_2 bis S_2+15, und setzt das entsprechende Bit im Ergebniswort R auf EIN, wenn die Inhalte **nicht** gleich sind.

Der Inhalt von S_1 wird mit dem Inhalt von S_2 verglichen, der Inhalt von S_1+1 mit dem Inhalt von S_2+1, ... und der Inhalt von S_1+15 mit dem Inhalt von S_2+15. Bit n von R wird auf AUS gesetzt, wenn der Inhalt von S_1+n gleich dem Inhalt von S_2+n ist. Bit n von R wird auf EIN gesetzt, wenn die Inhalte nicht gleich sind. Wenn die Inhalte aller 16 Wortpaare identisch sind, wird das Gleich-Flag im Anschluss an die Befehlsausführung auf EIN gesetzt.

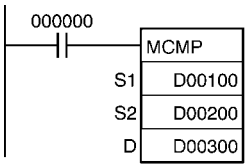


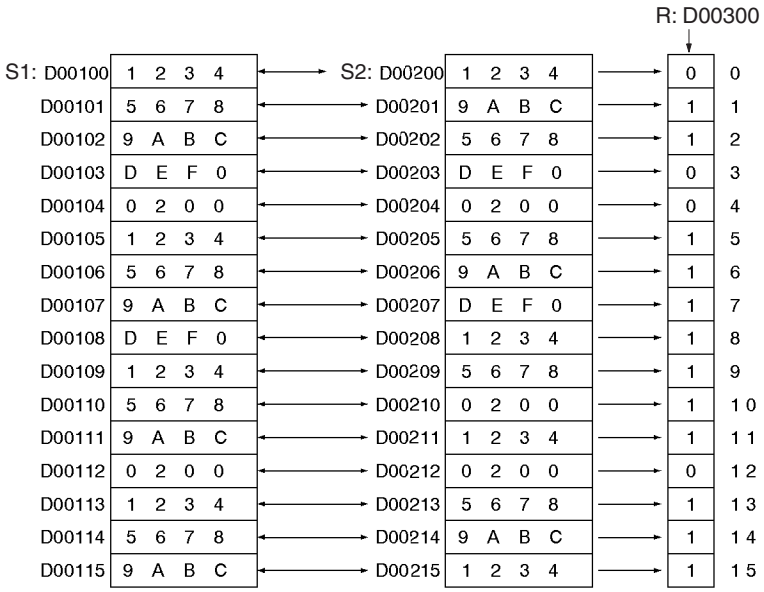
Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebniswort den Wert 0000 hat. (Die zwei 16-Worte-Sätze enthalten die gleichen Daten.) In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, vergleicht der Befehl MCMP(019) die Worte D00100 bis D00115 der Reihenfolge nach mit den Worten D00200 bis D00215 und setzt die entsprechenden Bits in D00300 auf EIN, wenn zwei Worte **nicht** gleich sind.



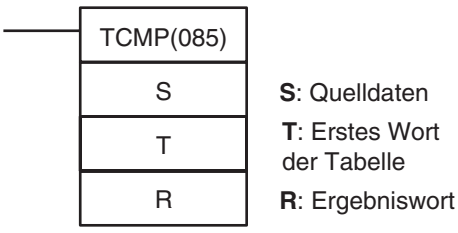


3-7-8 TABLE COMPARE: TCMP(085)

Funktion

Vergleicht die Quelldaten mit den Inhalten von 16 aufeinander folgenden Worten und setzt das entsprechende Bit im Ergebniswort auf EIN, wenn die Inhalte **gleich** sind.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TCMP(085)
	Ausführung einmal bei steigender Flanke	@TCMP(085)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

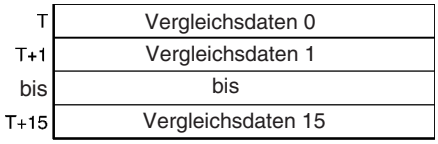
Operanden

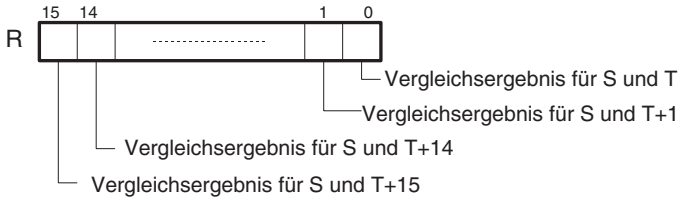
T: Erstes Wort der Tabelle

Spezifiziert das Anfangswort der 16-Worte-Tabelle. T und T+15 müssen im gleichen Datenbereich liegen.

R: Ergebniswort

Die einzelnen Bits von R enthalten des Ergebnis eines Vergleichs zwischen S und einem Wort in der 16-Worte-Tabelle. Bit n von R (n = 00 bis 15) enthält das Ergebnis des Vergleichs zwischen S und T+n.





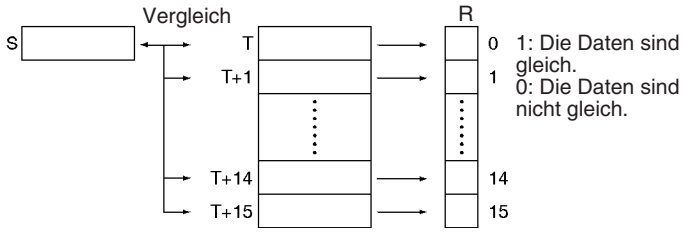
Operanden-Spezifikationen

Bereich	S	T	R
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6128	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W496	W000 bis W511
Arbeitsbereich remanent	H000 bis H511	H000 bis H496	H000 bis H511
Systembereich	A000 bis A959	A000 bis A944	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4080	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4080	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32752	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32752	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32752 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl TCMP(085) vergleicht die Quelldaten (S) mit jedem der 16 Worte T bis T+15 und setzt das entsprechende Bit im Wort R auf EIN, wenn die Daten **gleich** sind. Bit n von R wird auf EIN gesetzt, wenn der Inhalt von T+n gleich dem von S ist. Es wird auf AUS gesetzt, wenn die Inhalte nicht gleich sind.

S wird mit dem Inhalt von T verglichen und Bit 00 von R wird auf EIN gesetzt, wenn die Inhalte gleich sind, bzw. auf AUS gesetzt, wenn die Inhalte nicht gleich sind; S wird mit dem Inhalt von T+1 verglichen und Bit 01 von R wird auf EIN gesetzt, wenn die Inhalte gleich sind, bzw. auf AUS gesetzt, wenn die Inhalte nicht gleich sind, ... und S wird mit dem Inhalt von T+15 verglichen und Bit 15 von R wird auf EIN gesetzt, wenn die Inhalte gleich sind, bzw. auf AUS gesetzt, wenn die Inhalte nicht gleich sind.

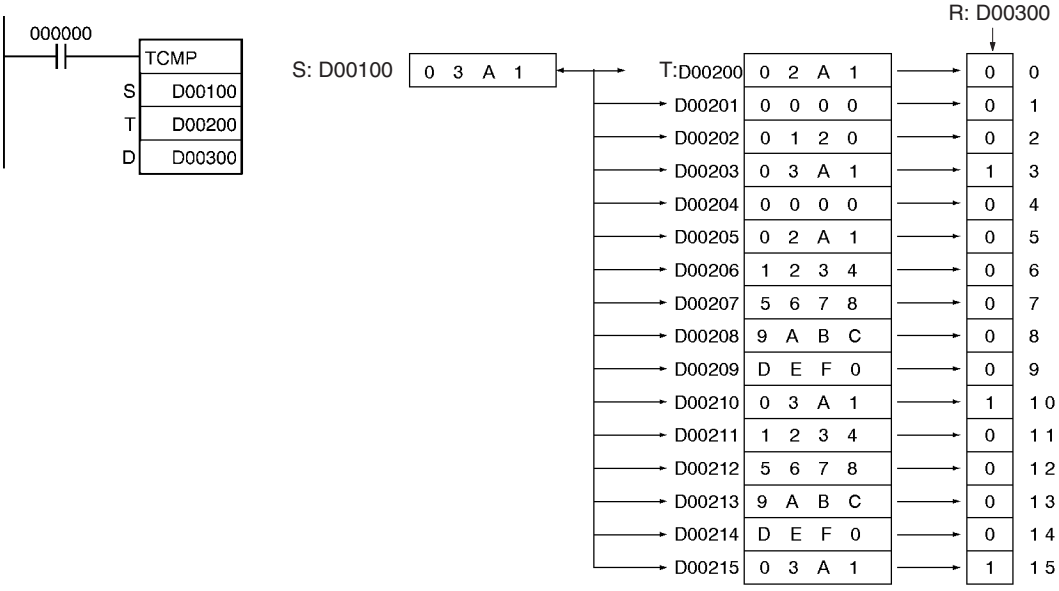


Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebniswort den Wert 0000 hat. (Keines der 16 Worte in der Tabelle ist gleich S.) In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, vergleicht der Befehl TCMP(085) den Inhalt von D00100 mit den Inhalten der Worte D00200 bis D00215 und setzt die entsprechenden Bits in D00300 auf EIN, wenn die Inhalte gleich sind, bzw. auf AUS, wenn sie nicht gleich sind.

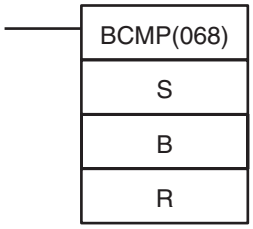


3-7-9 BLOCK COMPARE: BCMP(068)

Funktion

Vergleicht die Quelldaten mit 16 Bereichen (durch 16 untere und 16 obere Grenzwerte definiert), und setzt das entsprechende Bit im Ergebniswort auf EIN, wenn sich die Quelldaten innerhalb eines Bereichs befinden.

Kontaktplansymbol



S: Quelldaten

B: Erstes Wort des Blocks

R: Ergebniswort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	BCMP(068)
	Ausführung einmal bei steigender Flanke	@BCMP(068)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

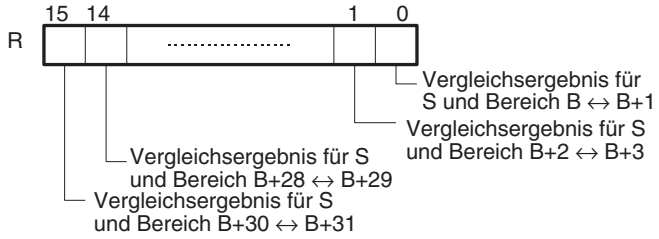
Operanden

B: Erstes Wort des Blocks

Spezifiziert das Anfangswort eines 32-Worte-Blocks (16 Paare unterer/oberer Grenzwerte). B und B+31 müssen im gleichen Datenbereich liegen.

R: Ergebniswort

Jedes Bit von R enthält das Ergebnis des Vergleichs zwischen S und einem der 16 im 32-Worte-Block definierten Bereiche. Bit n von R (n = 00 bis 15) enthält das Ergebnis des Vergleichs zwischen S und dem n-ten Wortepaar.



Operanden-Spezifikationen

Bereich	S	B	R
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6112	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W480	W000 bis W511
Arbeitsbereich remanent	H000 bis H511	H000 bis H480	H000 bis H511
Systembereich	A000 bis A959	A000 bis A928	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4064	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4064	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32736	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32736	E00000 bis E32767

Bereich	S	B	R
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32736 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl BCMP(068) vergleicht die Quelldaten (S) mit den 16 Bereichen, die durch die Paare aus unteren und oberen Grenzwerten in B bis B+31 definiert sind. Das erste Wort jedes Paars (B+2n) enthält den unteren Grenzwert, das zweite Wort (B+2n+1) den oberen Grenzwert des Bereichs n (n = 0 bis 15). Wenn S innerhalb eines dieser Bereiche (einschließlich unterem und oberem Grenzwert) liegt, wird das entsprechende Bit in R auf EIN gesetzt. Die restlichen Bits in R werden auf AUS gesetzt.

B	≤ S ≤	B+1	Bit 00 von R
B+2	≤ S ≤	B+3	Bit 01 von R
B+4	≤ S ≤	B+5	Bit 02 von R
B+6	≤ S ≤	B+7	Bit 03 von R
B+8	≤ S ≤	B+9	Bit 04 von R
B+10	≤ S ≤	B+11	Bit 05 von R
B+12	≤ S ≤	B+13	Bit 06 von R
B+14	≤ S ≤	B+15	Bit 07 von R
B+16	≤ S ≤	B+17	Bit 08 von R
B+18	≤ S ≤	B+19	Bit 09 von R
B+20	≤ S ≤	B+21	Bit 10 von R
B+22	≤ S ≤	B+23	Bit 11 von R
B+24	≤ S ≤	B+25	Bit 12 von R
B+26	≤ S ≤	B+27	Bit 13 von R
B+28	≤ S ≤	B+29	Bit 14 von R
B+30	≤ S ≤	B+31	Bit 15 von R

Ein Beispiel: Bit 00 von R wird auf EIN gesetzt, wenn S innerhalb des ersten Bereichs (B ≤ S ≤ B+1) liegt, Bit 01 von R wird auf EIN gesetzt, wenn S innerhalb des zweiten Bereichs (B+2 ≤ S ≤ B+3) liegt, ... und Bit 15 von R wird auf EIN gesetzt, wenn S innerhalb des fünfzehnten Bereichs (B+30 ≤ S ≤ B+31) liegt. Alle anderen Bits von R werden auf AUS gesetzt.

Flags

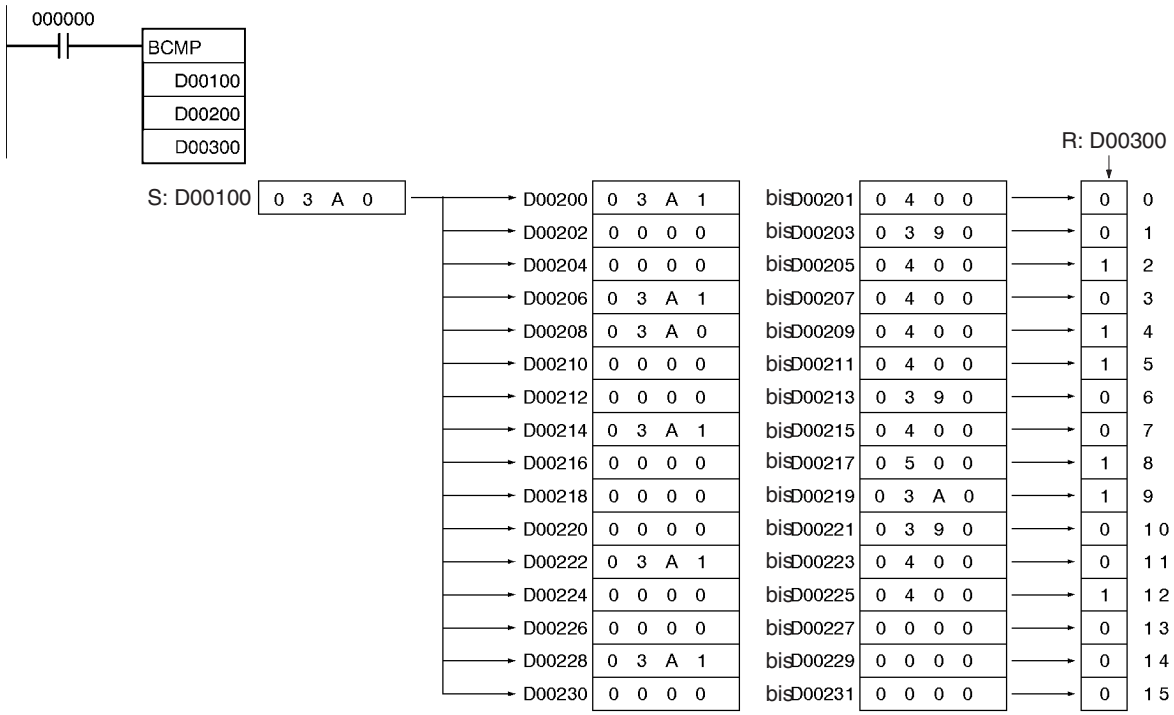
Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebniswort den Wert 0000 hat. (S liegt in keinem der 16 Bereiche.) In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn der untere Grenzwert eines Bereichs größer ist als sein oberer Grenzwert, tritt zwar kein Fehler auf, jedoch wird in jedem Fall 0 (nicht innerhalb des Bereichs) in das entsprechende Bit von R ausgegeben.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, vergleicht der Befehl BCMP(068) den Inhalt von D00100 mit den 16 Bereichen, die in D00200 bis D00231 definiert sind, und setzt die entsprechenden Bits in D00300 auf EIN, wenn S innerhalb des Bereichs liegt, bzw. auf AUS, wenn S nicht innerhalb des Bereichs liegt.

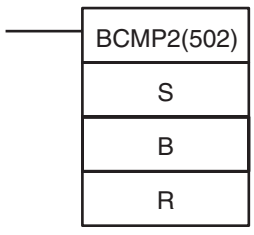


3-7-10 EXPANDED BLOCK COMPARE: BCMP2(502)

Funktion

Vergleicht die Quelldaten mit bis zu 256 Bereichen (durch 256 untere und 256 obere Grenzwerte definiert), und setzt das entsprechende Bit im Ergebniswort auf EIN, wenn sich die Quelldaten innerhalb eines Bereichs befinden. BCMP2(502) wird nur durch CS1-H, CJ1-H und CS1D CPU-Baugruppen ab Baugruppenversion 2.0 und CJ1M CPU-Baugruppen (Baugruppenversion vor 2.0 und ab 2.0) unterstützt.

Kontaktplansymbol



S: Quelldaten

B: Erstes Wort des Blocks

R: Erstes Ergebniswort

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	BCMP2(502)
	Ausführung einmal bei steigender Flanke	@BCMP2(502)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

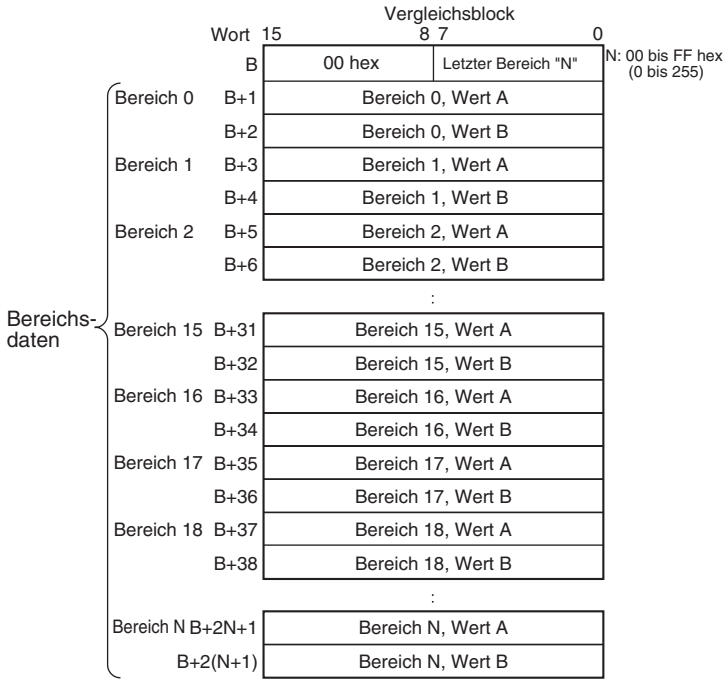
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

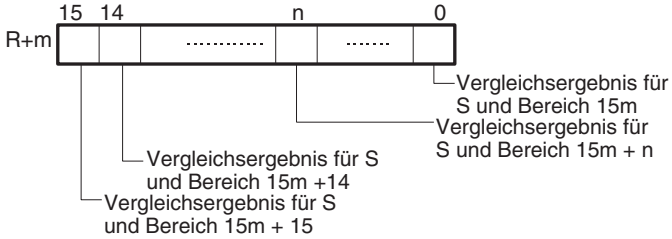
B: Erstes Wort des Blocks

Spezifiziert das Anfangswort eines Vergleichsblocks, der bis zu 513 Worte umfasst, die bis zu 256 Paare aus unterem/oberem Grenzwert enthalten. Alle Worte müssen im gleichen Datenbereich liegen.



R: Erstes Ergebniswort

Jedes Bit von jedem R-Wort enthält das Ergebnis des Vergleichs zwischen S und einem der im Vergleichsblock definierten Bereiche. Die maximale Anzahl der Ergebnisworte beträgt 16, d. h. „m“ ist gleich 0 bis 15.



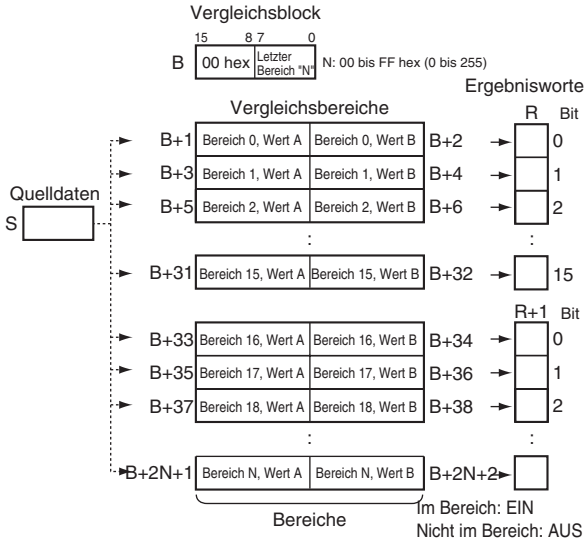
Operanden-Spezifikationen

Bereich	S	B	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	---		
EM-Bereich mit Bank	---		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767		
Konstanten	#0000 bis #FFFF (binär)	---	
Datenregister	DR0 bis DR15	---	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl BCMP2(502) vergleicht die Quelldaten (S) mit den Bereichen, die durch die Paare aus unteren und oberen Grenzwerten im Vergleichsblock definiert sind. Wenn S innerhalb eines dieser Bereiche (einschließlich unterem und oberem Grenzwert) liegt, werden die entsprechenden Bits in den Ergebnisworten (R bis max. R+15) auf EIN gesetzt. Die restlichen Bits in den Ergebnisworten werden auf AUS gesetzt.

Die Anzahl der Bereiche wird durch den Wert von N bestimmt, der durch das niederwertige Byte von B spezifiziert wird. N kann eine Zahl zwischen 0 und 255 sein. Das höherwertige Byte von B muss auf 00 hex gesetzt sein.

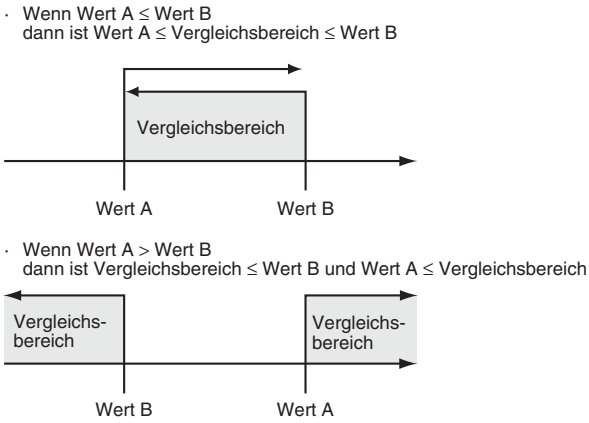


Anzahl der Bereiche

Die Anzahl der Bereiche im Vergleichsblock wird im ersten Wort des Blocks eingestellt. Es können bis zu 256 Bereiche eingestellt werden.

Einstellung der Bereiche

Die Werte A und B jedes Bereichs bestimmen die Funktionsweise des Vergleichs je nachdem, welcher der Werte größer ist (siehe folgende Abbildung).



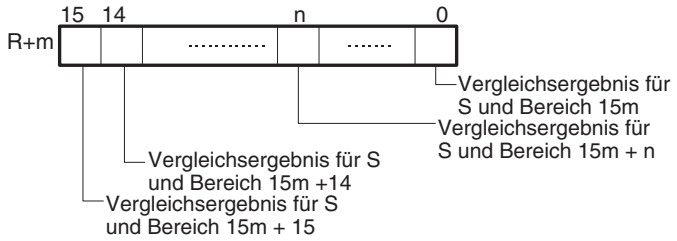
Beispiel

Wenn B+1 ≤ B+2
Wenn B+1 ≤ S ≤ B+2, dann wird Bit 0 von R auf EIN gesetzt,
Wenn B+3 ≤ S ≤ B+4, dann wird Bit 1 von R auf EIN gesetzt,
Wenn S < B+5 und B+6 < S, dann wird Bit 2 von R auf AUS gesetzt, und
wenn S < B+7 und B+8 < S, dann wird Bit 3 von R auf AUS gesetzt.

Wenn B+1 > B+2
Wenn S ≤ B+2 und B+1 ≤ S, dann wird Bit 0 von R auf EIN gesetzt.
Wenn S ≤ B+4 und B+3 ≤ S, dann wird Bit 1 von R auf EIN gesetzt.
Wenn B+6 < S < B+5, dann wird Bit 2 von R auf AUS gesetzt, und
wenn B+8 < S < B+7, dann wird Bit 3 von R auf AUS gesetzt.

Ergebnisspeicherort

Die Ergebnisse werden in die entsprechenden Bits des Worts R ausgegeben. Wenn mehr als 16 Vergleichsbereiche vorhanden sind, werden auf R folgenden Worte für weitere Ergebnisse verwendet. Die maximale Anzahl der Ergebnisworte beträgt 16, d. h. „m“ ist gleich 0 bis 15.



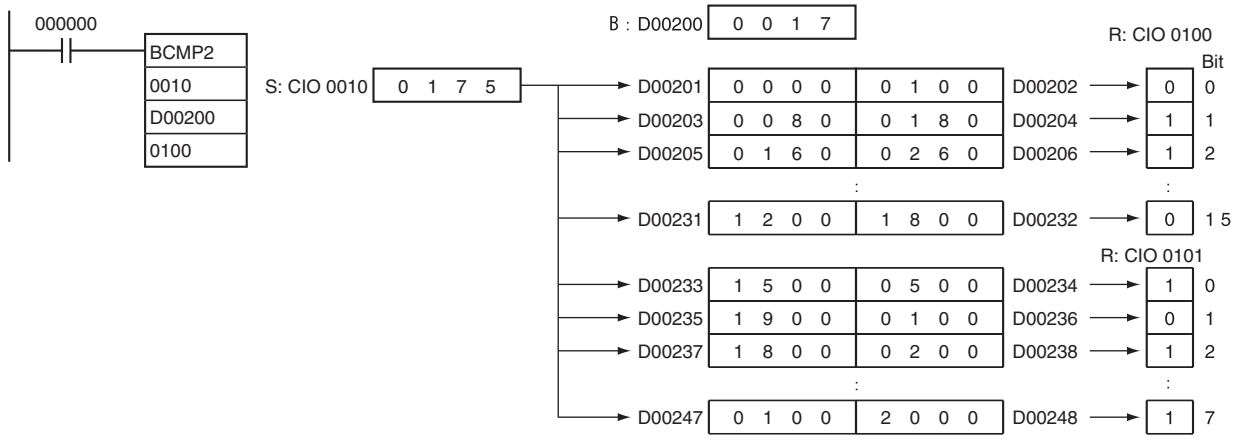
Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, vergleicht der Befehl BCMP2(502) den Inhalt von CIO 0010 mit den 24 Bereichen, die in D00200 bis D00247 (N = 17 hex = 23 dezimal, d. h. 24 Bereiche) definiert sind, und setzt die entsprechenden Bits in CIO 0100 und CIO 0101 auf EIN, wenn S innerhalb des Bereichs liegt, bzw. auf AUS, wenn S nicht innerhalb des Bereichs liegt. Ein Beispiel: Wenn die Quelldaten in CIO 0010 innerhalb des durch D00201 und D00202 definierten Bereichs liegen, dann wird Bit 00 von CIO 0100 auf EIN gesetzt. Liegen sie nicht innerhalb des Bereichs, wird Bit 00

von CIO 0100 auf AUS gesetzt. Auf gleiche Weise werden die Quelldaten in CIO 0010 mit den durch D00203 und D00204, D00247 und D00248 und den anderen Worten im Vergleichsblock definierten Bereichen verglichen, und Bit 1 in CIO 0100, Bit 7 in CIO 1010 und die anderen Bits in den Ergebnisworten werden entsprechend des Vergleichsergebnisses gesetzt.



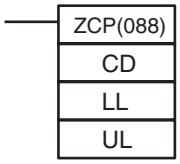
3-7-11 AREA RANGE COMPARE: ZCP(088)

Funktion

Vergleicht einen vorzeichenlosen 16-Bit-Binärwert (CD) mit dem Bereich, der durch den unteren Grenzwert LL und den oberen Grenzwert UL definiert ist. Die Ergebnisse werden an die arithmetischen Flags ausgegeben.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



CD: Vergleichsdaten
LL: Untere Grenze des Bereichs
UL: Obere Grenze des Bereichs

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ZCP(088)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	CD	LL	UL
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		

Bereich	CD	LL	UL
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl ZCP(088) vergleicht die vorzeichenlosen 16-Bit-Binärdaten in CD mit dem durch LL und UL definierten Bereich und gibt das Ergebnis an die Größer-als-, Gleich- und Kleiner-als-Flags im Systembereich aus. (Die Kleiner-oder-gleich-, Größer-oder-gleich- und Ungleich-Flags bleiben unverändert.)

Status der arithmetischen Flags

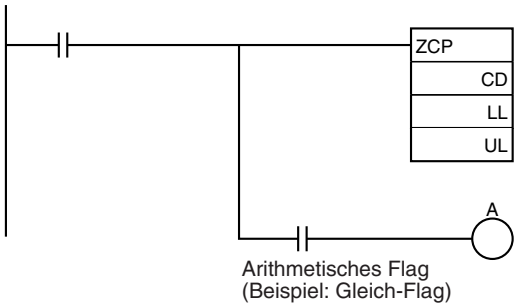
In der folgenden Tabelle ist der Status der einzelnen arithmetischen Flags nach der Ausführung von ZCP(088) zu entnehmen.

Ergebnis von ZCP(088)	Flag-Status		
	>	=	<
CD > UL	EIN	AUS	AUS
CD = UL	AUS	EIN	
LL < CD < UL			
CD = LL			
CD < LL		AUS	EIN

Verwendung der Ergebnisse von ZCP(088) im Programm

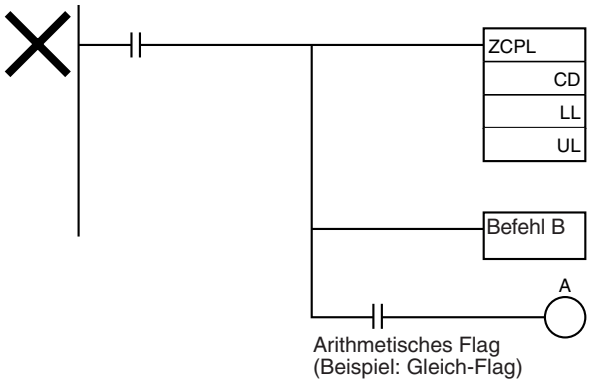
Nach der Ausführung von ZCP(088) wird das Ergebnis durch die arithmetischen Flags wiedergegeben. Steuern Sie den gewünschten Ausgang oder Befehl auf der rechten Seite mit einer Verzweigung von der gleichen Eingangsbedingung, die den Befehl ZCP(088) steuert (siehe folgende Abbildung). In diesem Fall werden das Gleich-Flag und Ausgang A auf EIN gesetzt, wenn LL ≤ CD ≤ UL.

Korrekte Verwendung von ZCP(088)



Programmieren Sie keinen anderen Befehl zwischen ZCP(088) und dem durch das arithmetische Flag gesteuerten Befehl, da der andere Befehl des Status des arithmetischen Flags ändern könnte. In diesem Fall könnte das Ergebnis von Befehl B die Ergebnisse von ZCP(088) ändern.

Falsche Verwendung von ZCP(088)



Flags

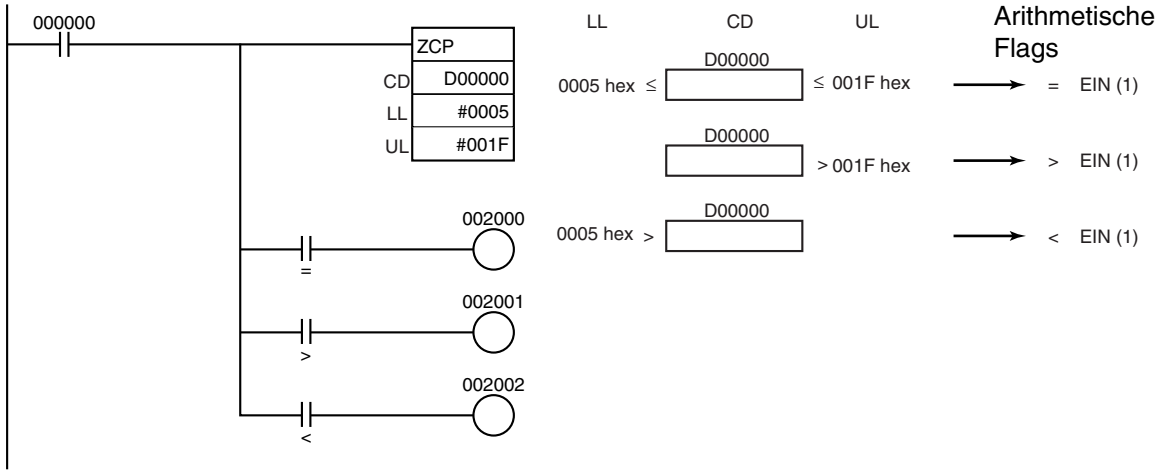
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn LL > UL.
Größer-als-Flag	>	EIN, wenn CD > UL. In allen anderen Fällen AUS.
Größer-oder-gleich-Flag	> =	Bleibt unverändert.
Gleich-Flag	=	EIN, wenn LL ≤ CD ≤ UL. In allen anderen Fällen AUS.
Ungleich-Flag	<>	Bleibt unverändert.
Kleiner-als-Flag	<	EIN, wenn CD < LL. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Flag	< =	Bleibt unverändert.
Negativ-Flag	N	Bleibt unverändert.

Vorsichtsmaßnahmen

Programmieren Sie keine anderen Befehle zwischen ZCP(088) und einer Eingangsbedingung, die auf das Ergebnis von ZCP(088) zugreift, da der andere Befehl den Status der arithmetischen Flags ändern könnte.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, werden die vorzeichenlosen 16-Bit-Binärdaten in D00000 mit dem Bereich von 0005 bis 001F hex (5 bis 31 dezimal) verglichen, und das Ergebnis wird an die arithmetischen Flags ausgegeben.
CIO 000200 wird auf EIN gesetzt, wenn 0005 hex ≤ Inhalt von D00000 ≤ 001F hex.
CIO 000201 wird auf EIN gesetzt, wenn der Inhalt von D00000 > 001F hex ist.
CIO 000202 wird auf EIN gesetzt, wenn der Inhalt von D00000 < 0005 hex ist.



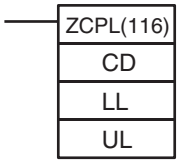
3-7-12 DOUBLE AREA RANGE COMPARE: ZCPL(116)

Funktion

Vergleicht einen vorzeichenlosen 32-Bit-Binärwert (CD+1, CD) mit dem Bereich, der durch den unteren Grenzwert (LL+1, LL) und den oberen Grenzwert (UL+1, UL) definiert ist. Die Ergebnisse werden an die arithmetischen Flags ausgegeben.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



CD: Erstes Wort der Vergleichsdaten
LL: Erstes Wort des unteren Grenzwerts
UL: Erstes Wort des oberen Grenzwerts

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ZCP(088)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	CD	LL	UL
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich nicht remanent	W000 bis W510		
Arbeitsbereich remanent	H000 bis H510		
Systembereich	A000 bis A958		
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		

Bereich	CD	LL	UL
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 0000 bis #FFFF FFFF (binär)		
Datenregister	---		
Indexregister	IR0 bis IR15		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl ZCPL(116) vergleicht die vorzeichenlosen 32-Bit-Binärdaten in CD+1, CD mit dem durch LL+1, LL und UL+1, UL definierten Bereich und gibt das Ergebnis an die Größer-als-, Gleich- und Kleiner-als-Flags im Systembereich aus. (Die Kleiner-oder-gleich-, Größer-oder-gleich- und Ungleich-Flags bleiben unverändert.)

Status der arithmetischen Flags

In der folgenden Tabelle ist der Status der einzelnen arithmetischen Flags nach der Ausführung von ZCPL(116) zu entnehmen.

Ergebnis von ZCPL(116)	Flag-Status		
	>	=	<
CD+1, CD > UL+1, UL	EIN	AUS	AUS
CD+1, CD = UL+1, UL	AUS	EIN	
LL+1, LL < CD+1, CD < UL+1, UL			
CD+1, CD = LL+1, LL			
CD+1, CD < LL+1, LL		AUS	EIN

Verwendung der Ergebnisse von ZCPL(116) im Programm

Nach der Ausführung von ZCPL(116) wird das Ergebnis durch die arithmetischen Flags wiedergegeben. Steuern Sie den gewünschten Ausgang oder Befehl auf der rechten Seite mit einer Verzweigung von der gleichen Eingangsbedingung, die den Befehl ZCPL(116) steuert.

Programmieren Sie keinen anderen Befehl zwischen ZCPL(116) und dem durch das arithmetische Flag gesteuerten Befehl, da der andere Befehl des Status des arithmetischen Flags ändern könnte.

Die Funktion des Befehls ZCPL(116) ist weitgehend mit der des Befehls ZCP(088) identisch, abgesehen davon, dass ZCPL(116) 32-Bit-Werte statt 16-Bit-Werte vergleicht. Abbildungen, in denen dargestellt ist, wie die Ergebnisse im Programm verwendet werden, sowie einen Beispiel-Programmabschnitt finden Sie unter *3-7-11 AREA RANGE COMPARE: ZCP(088)*.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn LL+1, LL > UL+1, UL.
Größer-als-Flag	>	EIN. wenn CD > UL+1, UL. In allen anderen Fällen AUS.

Name	Bezeich-nung	Funktion
Größer-oder-gleich-Flag	> =	Bleibt unverändert.
Gleich-Flag	=	EIN, wenn LL+1, LL ≤ CD+1, CD ≤ UL+1, UL. In allen anderen Fällen AUS.
Ungleich-Flag	<>	Bleibt unverändert.
Kleiner-als-Flag	<	EIN. wenn CD+1, CD < LL+1, LL. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Flag	< =	Bleibt unverändert.
Negativ-Flag	N	Bleibt unverändert.

Vorsichtsmaßnahmen

Programmieren Sie keine anderen Befehle zwischen ZCPL(116) und einer Eingangsbedingung, die auf das Ergebnis von ZCPL(116) zugreift, da der andere Befehl den Status der arithmetischen Flags ändern könnte.

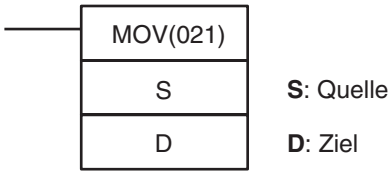
3-8 Datenübertragungsbefehle

3-8-1 MOVE: MOV(021)

Funktion

Überträgt ein Datenwort in das angegebene Wort.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	MOV(021)
	Ausführung einmal bei steigender Flanke	@MOV(021)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung (siehe Hinweis)		!MOV(021)
Kombinierte Variationen	**Einmalige Ausführung und unmittelbare Zielaktualisierung bei steigender Flanke (siehe Hinweis).**	!@MOV(021)

Hinweis Die direkte Aktualisierung wird von CS1D CPU-Baugruppen nicht unterstützt.

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	

Bereich	S	D
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #FFFF (binär)	---
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Überträgt S in D. Wenn S eine Konstante ist, kann der Befehl zur Dateneinstellung verwendet werden.



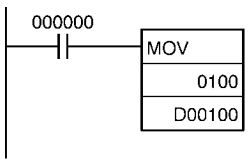
Für den Befehl MOV(021) gibt es eine Vatiante mit direkter Aktualisierung (!MOV(021)). Für S können Bits externer Eingänge spezifiziert werden, ebenso können für D Bits externer Ausgänge spezifiziert werden. Die für S verwendeten Eingangsbits werden unmittelbar vor, die für D verwendeten Ausgangsbits unmittelbar nach Ausführung des Befehls aktualisiert, sofern diese Bits nicht einer Multipoint-E/A-Baugruppe der Gruppe 2, einer Multipoint-Spezial-E/A-Baugruppe oder einer Baugruppe zugewiesen sind, die in einem dezentralen Slave-Block montiert ist,

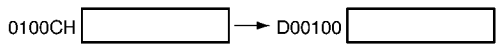
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn die übertragenen Daten den Wert 0000 haben. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit der übertragenen Daten 1 ist. In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird der Inhalt von CIO 0100 in D00100 kopiert.



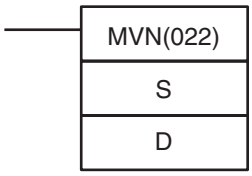


3-8-2 MOVE NOT: MVN(022)

Funktion

Überträgt das Komplement eines Datenworts an das angegebene Wort.

Kontaktplansymbol



S: Quelle

D: Ziel

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MVN(022)
	Ausführung einmal bei steigender Flanke	@MVN(022)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #FFFF (binär)	---
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl MVN(022) invertiert die Bits in S und überträgt das Ergebnis in D. Der Inhalt von S bleibt unverändert erhalten.

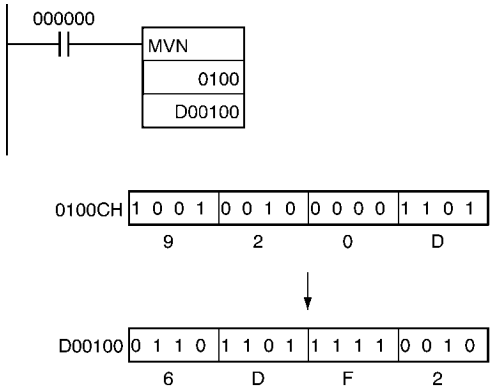


Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn der Inhalt von D nach der Ausführung den Wert 0000 hat. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit von D nach der Ausführung 1 ist. In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird der Status der Bits in CIO 0100 invertiert und das Ergebnis in D00100 kopiert.

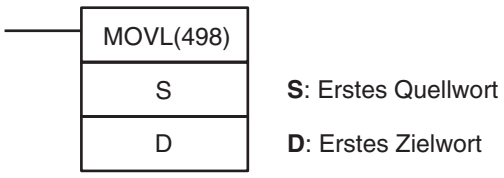


3-8-3 DOUBLE MOVE: MOVL(498)

Funktion

Überträgt zwei Datenworte an die angegebenen Worte.

Kontaktplansymbol



S: Erstes Quellwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MOVL(498)
	Ausführung einmal bei steigender Flanke	@MOVL(498)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

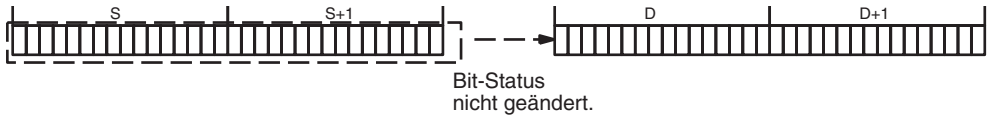
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	IR0 bis IR15	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl MOVL(498) überträgt S+1 und S in D+1 und D. Wenn S+1 und S Konstanten sind, kann der Befehl zur Dateneinstellung verwendet werden.

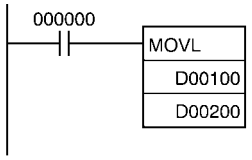


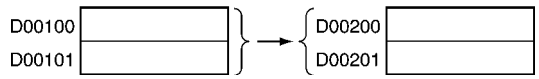
Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn die Inhalte von D+1 und D nach Ausführung des Befehls 0000 0000 sind. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit von D+1 nach der Ausführung 1 ist. In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird der Inhalt von D00101 und D00100 in D00201 und D00200 kopiert.



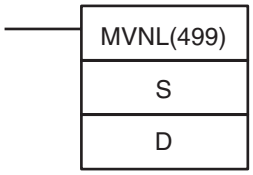


3-8-4 DOUBLE MOVE NOT: MVNL(499)

Funktion

Überträgt das Komplement von zwei Datenworten an die angegebenen Worte.

Kontaktplansymbol



S: Erstes Quellwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MVNL(499)
	Ausführung einmal bei steigender Flanke	@MVNL(499)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

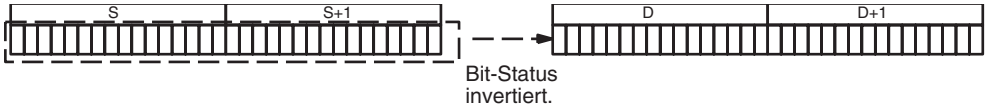
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---

Bereich	S	D
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl MVNL(499) invertiert die Bits in S+1 und S und überträgt das Ergebnis in D+1 und D. Die Inhalte von S+1 und S bleiben unverändert erhalten.

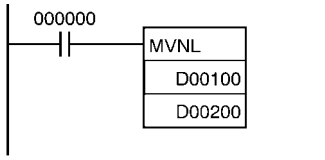


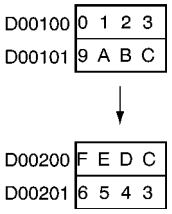
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn die Inhalte von D+1 und D nach Ausführung des Befehls 0000 0000 sind. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit von D+1 nach der Ausführung 1 ist. In allen anderen Fällen AUS.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird der Status der Bits in D00101 und D00100 invertiert und das Ergebnis in D00201 und D00200 kopiert. (Die ursprünglichen Inhalte von D00101 und D00100 bleiben unverändert erhalten.)



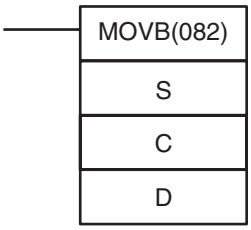


3-8-5 MOVE BIT: MOVB(082)

Funktion

Überträgt das spezifizierte Bit.

Kontaktplansymbol



S: Quellwort oder -daten

C: Steuerwort

D: Zielwort

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	MOVB(082)
	Ausführung einmal bei steigender Flanke	@MOVB(082)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

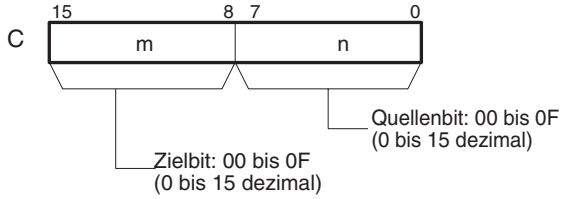
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

Die niederwertigen zwei Stellen von C spezifizieren, welches Bit von S das Quellenbit ist, und die höherwertigen zwei Stellen von C spezifizieren, welches Bit von D das Zielbit ist.

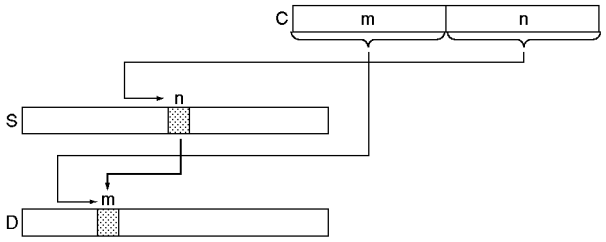


Operanden-Spezifikationen

Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	Nur spezifizierte Werte	---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl MOVB(082) kopiert das spezifizierte Bit (n) von S in das spezifizierte Bit (m) in D. Die anderen Bits im Zielwort bleiben unverändert.



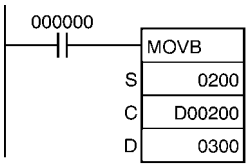
Hinweis Für S und D kann das gleiche Wort spezifiziert werden, um ein Bit innerhalb eines Worts zu kopieren.

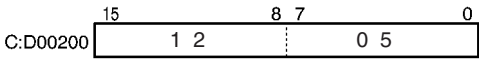
Flags

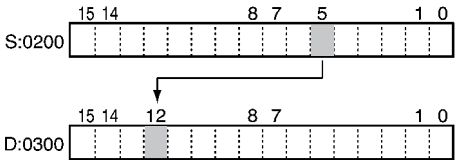
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn die niederwertigen zwei Stellen von C nicht innerhalb des spezifizierten Bereichs von 00 bis 0F hex liegen. In allen anderen Fällen AUS.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird entsprechend des Steuerwort-Werts von 0C05 das fünfte Bit des Quellworts (CIO 0200) in das zwölfte Bit des Zielworts (CIO 0300) kopiert.





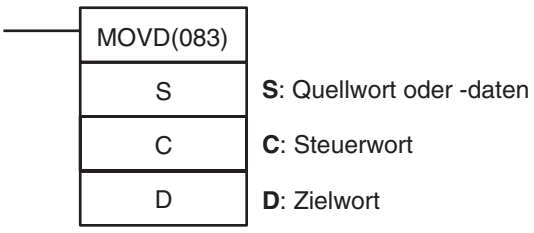


3-8-6 MOVE DIGIT: MOVD(083)

Funktion

Überträgt die spezifizierte(n) Stelle oder Stellen. (Jede Stelle besteht aus 4 Bits.)

Kontaktplansymbol



S: Quellwort oder -daten

C: Steuerwort

D: Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MOVD(083)
	Ausführung einmal bei steigender Flanke	@MOVD(083)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

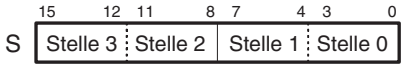
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

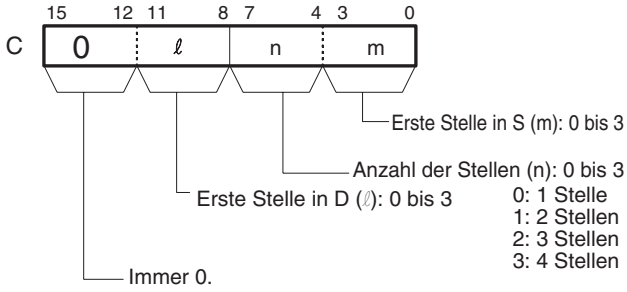
S: Quellwort

Die Quellenstellen werden von rechts nach links gelesen, und ggf. wird wieder mit der rechten Stelle (Stelle 0) fortgefahren.



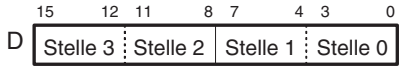
C: Steuerwort

Die ersten drei Stellen von C spezifizieren das erste Quellwort (m), die Anzahl der zu übertragenden Stellen (n) und das erste Zielwort (ℓ), wie in der folgenden Abbildung gezeigt.



D: Zielwort

Die Zielstellen werden von rechts nach links geschrieben, und ggf. wird wieder mit der rechten Stelle (Stelle 0) fortgefahren.



Operanden-Spezifikationen

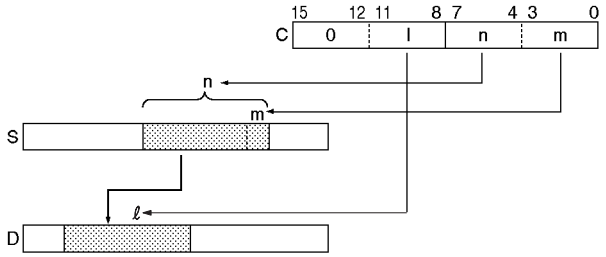
Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	Nur spezifizierte Werte	---
Datenregister	DR0 bis DR15		

Bereich	S	C	D
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl MOVD(083) kopiert den Inhalt von n Stellen aus S (mit Stelle m beginnend) in D (mit Stelle ℓ beginnend). Es werden nur die spezifizierten Stellen geändert, die restlichen Stellen bleiben unverändert erhalten.

Wenn die spezifizierte Anzahl der zu lesenden oder zu schreibenden Stellen über die höchstwertige Stelle von S bzw. D hinausgeht, fährt der Befehl MOVD(083) mit dem Lesen bzw. Schreiben an der niederwertigsten Stelle des gleichen Worts fort.



Hinweis Für S und D kann das gleiche Wort spezifiziert werden, um Stellen innerhalb eines Worts zu kopieren.

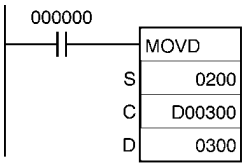
Flags

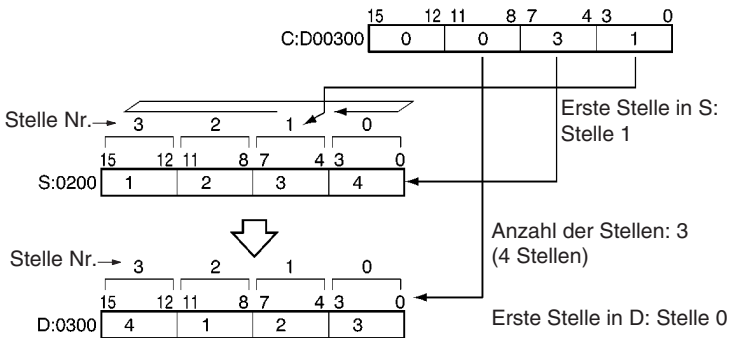
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn eine der ersten drei Stellen von C nicht innerhalb des spezifizierten Bereichs von 0 bis 3 liegt. In allen anderen Fällen AUS.

Beispiele

Übertragung von vier Stellen

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, werden vier Stellen Daten von CIO 0200 in CIO 0300 kopiert. Die Übertragung beginnt entsprechend des Steuerwort-Werts von 0031 mit Stelle 1 von CIO 0200 und Stelle 0 von CIO 0300.

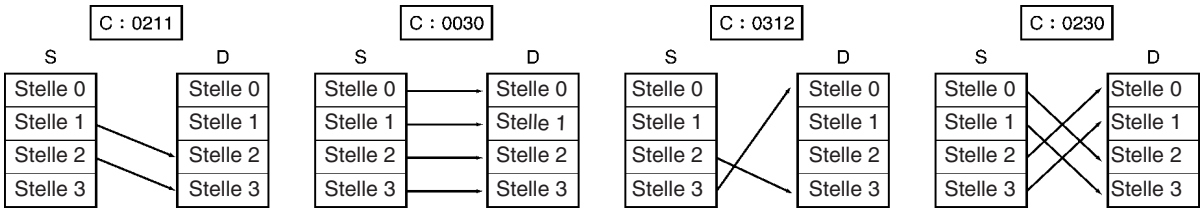




Hinweis Nach dem Lesen der höchstwertigen Stelle von S (Stelle 3) fährt der Befehl MOVD(083) mit der niederwertigsten Stelle (Stelle 0) fort.

Beispiele für C

In der folgenden Abbildung sind Beispiele für Datenübertragungen bei verschiedenen Werten für C dargestellt.

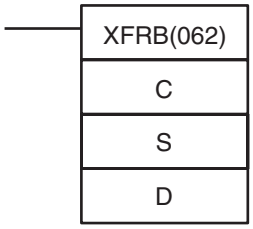


3-8-7 MULTIPLE BIT TRANSFER: XFRB(062)

Funktion

Überträgt die spezifizierte Anzahl von aufeinander folgenden Bits.

Kontaktplansymbol



C: Steuerwort

S: Erstes Quellwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	XFRB(062)
	Ausführung einmal bei steigender Flanke	@XFRB(062)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

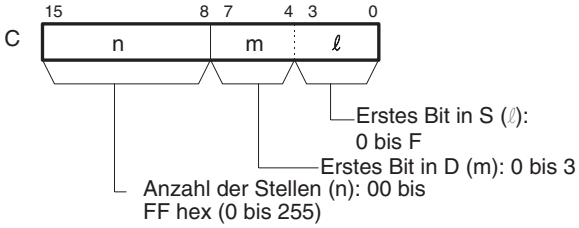
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

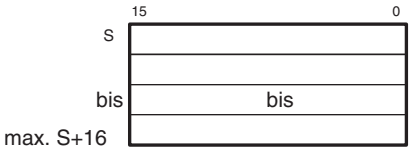
C: Steuerwort

Die ersten drei Stellen von C spezifizieren das erste Quellwort (m), die Anzahl der zu übertragenden Stellen (n) und das erste Zielwort (ℓ), wie in der folgenden Abbildung gezeigt.



S: Erstes Quellwort

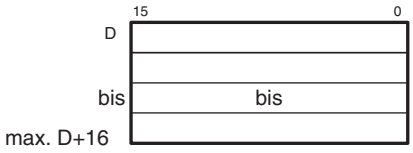
Spezifiziert das erste Quellwort. Die Bits werden von rechts nach links gelesen und ggf. wird mit dem Lesen der folgenden Worte (bis max. S+16) fortgefahren.



Hinweis Die Quellworte müssen im gleichen Datenbereich liegen.

D: Erstes Zielwort

Spezifiziert das erste Zielwort. Die Bits werden von rechts nach links geschrieben und ggf. wird mit dem Lesen der folgenden Worte (bis max. D+16) fortgefahren.



Hinweis Die Zielworte müssen im gleichen Datenbereich liegen.

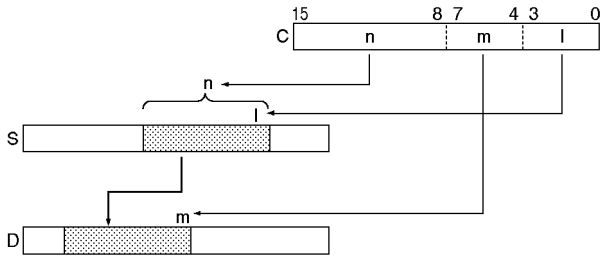
Operanden-Spezifikationen

Bereich	C	S	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	Nur spezifizierte Werte	---	---
Datenregister	DR0 bis DR15	---	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,+(++)IR0 bis ,+(++)IR15 ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl XFRB(062) überträgt bis zu 255 aufeinander folgende Bits aus den Quellworten (mit Bit ℓ von S beginnend) in die Zielworte (mit Bit m von D beginnend). Bits in den Zielworten, die nicht durch die Quellenbits überschrieben werden, bleiben unverändert erhalten.

Die Anfangsbits sowie die Anzahl der zu übertragenden Bits werden in C spezifiziert (siehe nachfolgende Abbildung).



Es ist möglich und zulässig, dass sich Quellen- und Zielworte überlappen. Durch die Übertragung von in mehreren Worten überlappenden Daten können die Daten effizienter in dem Datenbereich gepackt werden. (Dies ist insbesondere bei der Verarbeitung von Positionsdaten bei der Positioniersteuerung hilfreich.)

Da sich Quellen- und Zielworte überlappen können, kann der Befehl XFRB(062) mit dem Befehl ANDW(034) kombiniert werden, um m Bits um n Stellen zu verschieben.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS

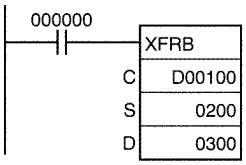
Vorsichtsmaßnahmen

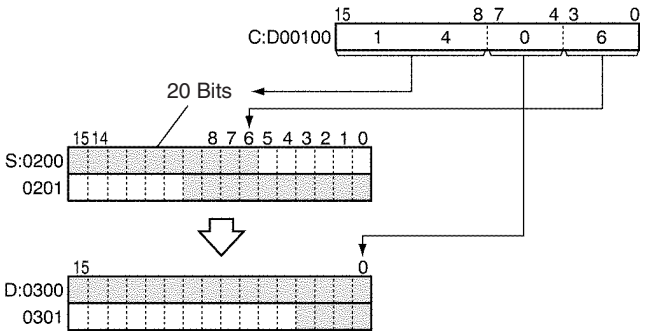
Bei jeder Ausführung von XFRB(062) können bis zu 255 Bits Daten übertragen werden.

Achten Sie darauf, dass sich die Quellen- und Zielworte nicht über das Ende des Datenbereichs hinaus erstrecken.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, werden die mit CIO 020006 beginnenden 20 Bits in die mit CIO 030000 beginnenden 20 Bits kopiert.



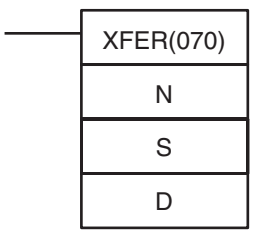


3-8-8 BLOCK TRANSFER: XFER(070)

Funktion

Überträgt die spezifizierte Anzahl von aufeinander folgenden Worten.

Kontaktplansymbol



N: Anzahl der Worte

S: Erstes Quellwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	XFER(070)
	Ausführung einmal bei steigender Flanke	@XFER(070)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

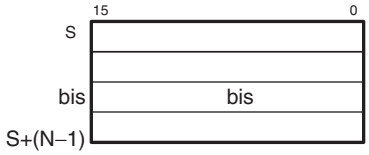
Operanden

N: Anzahl der Worte

Spezifiziert die Anzahl der zu übertragenden Worte. Der gültige Bereich für N erstreckt sich von 0000 bis FFFF (0 bis 65.535 dezimal).

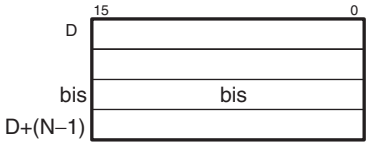
S: Erstes Quellwort

Spezifiziert das erste Quellwort.



D: Erstes Zielwort

Spezifiziert das erste Zielwort.



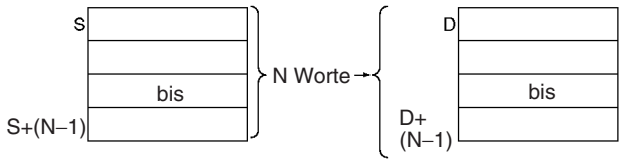
Operanden-Spezifikationen

Bereich	N	S	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär) oder &0 bis &65535	---	---
Datenregister	DR0 bis DR15	---	

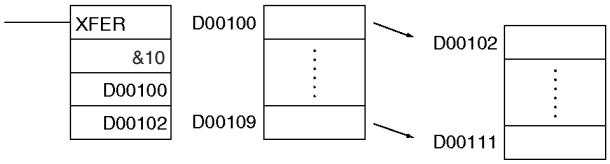
Bereich	N	S	D
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl XFER(070) kopiert N mit S beginnende Worte (S bis S+(N–1)) in die N mit D beginnenden Worte (D bis D+(N–1)).



Es ist möglich, dass sich Quellen- und Zielworte überpappen, sodass XFER(070) für Wort-Verschiebevorgänge genutzt werden kann.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS

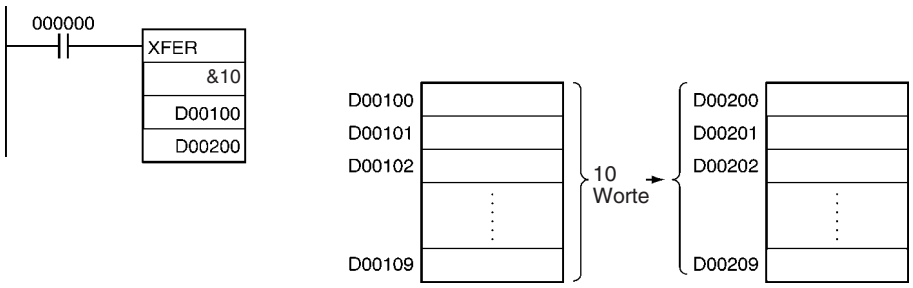
Vorsichtsmaßnahmen

Achten Sie darauf, dass sich die Quellworte (S bis S+N–1) und die Zielworte (D bis D+N–1) nicht über das Ende des Datenbereichs hinaus erstrecken.

Für die Ausführung des Befehls XFER(070) ist einige Zeit erforderlich, wenn eine große Anzahl von Worten übertragen wird. In diesem Fall ist die Übertragung durch XFER(070) möglicherweise noch nicht abgeschlossen, wenn es während der Ausführung des Befehls zu einer Unterbrechung der Versorgungsspannung kommt.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, werden die 10 Worte D00100 bis D00109 in D00200 bis D00209 kopiert.

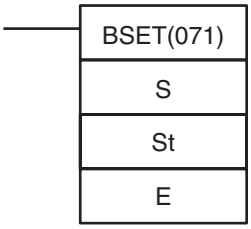


3-8-9 BLOCK SET: BSET(071)

Funktion

Kopiert dasselbe Wort in einen Bereich aufeinander folgender Worte.

Kontaktplansymbol



S: Quellwort

St: Anfangswort

E: Endwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	BSET(071)
	Ausführung einmal bei steigender Flanke	@BSET(071)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Quellwort

Spezifiziert die Quelldaten oder das Wort, das die Quelldaten enthält.

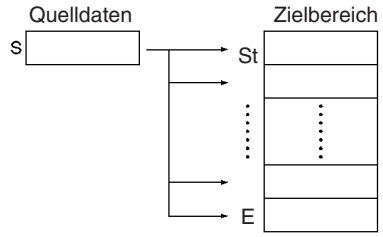
St: Anfangswort

Spezifiziert das erste Wort des Zielbereichs.

E: Endwort

Spezifiziert das letzte Wort des Zielbereichs.





Hinweis St und E müssen im gleichen Datenbereich liegen.

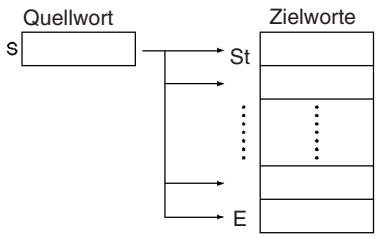
Operanden-Spezifikationen

Bereich	S	St	E
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		

Bereich	S	St	E
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	
Datenregister	DR0 bis DR15	---	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl BSET(071) kopiert dasselbe Quellwort (S) in alle Worte des Zielbereichs von St bis E.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn St größer als E ist. In allen anderen Fällen AUS.

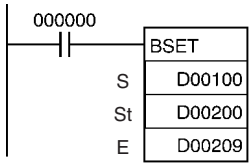
Vorsichtsmaßnahmen

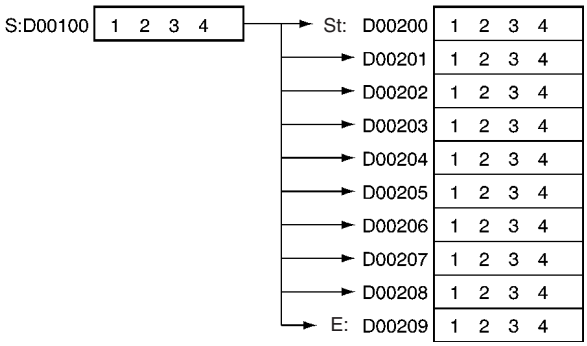
Achten Sie darauf, dass das Anfangswort (St) und das Endwort (E) im gleichen Datenbereich liegen und dass die Bedingung St ≤ E erfüllt ist.

Für die Ausführung des Befehls BSET(071) ist einige Zeit erforderlich, wenn die Quelldaten in eine große Anzahl von Worten übertragen werden. In diesem Fall ist die Übertragung durch BSET(071) möglicherweise noch nicht abgeschlossen, wenn es während der Ausführung des Befehls zu einer Unterbrechung der Versorgungsspannung kommt.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, werden die Quelldaten in D00100 in die Worte D00200 bis D00209 kopiert.



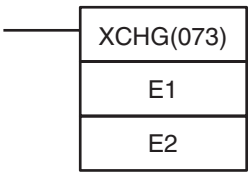


3-8-10 DATA EXCHANGE: XCHG(073)

Funktion

Tauscht den Inhalt der zwei spezifizierten Worte aus.

Kontaktplansymbol



E1: Erstes Austauschwort

E2: Zweites Austauschwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	XCHG(073)
	Ausführung einmal bei steigender Flanke	@XCHG(073)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

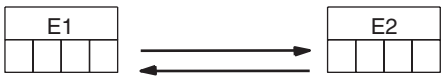
Operanden-Spezifikationen

Bereich	E1	E2
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	

Bereich	E1	E2
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl tauscht die Inhalte von E1 und E2 gegeneinander aus.



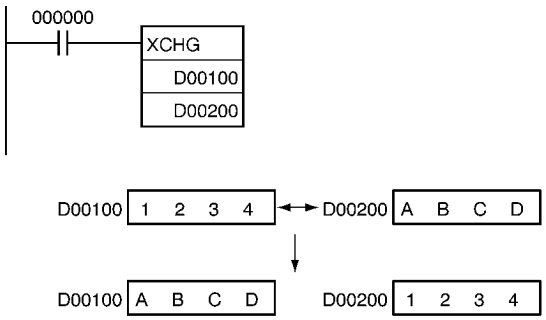
Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird der Inhalt D00100 mit dem Inhalt von D00200 vertauscht.

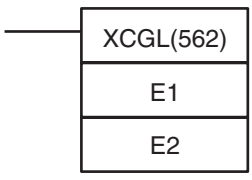


3-8-11 DOUBLE DATA EXCHANGE: XCGL(562)

Funktion

Tauscht den Inhalt von zwei aufeinander folgenden Worten mit dem von zwei anderen aufeinander folgenden Worten aus.

Kontaktplansymbol



E1: Erstes Austauschwort

E2: Zweites Austauschwort

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	XCGL(562)
	Ausführung einmal bei steigender Flanke	@XCGL(562)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

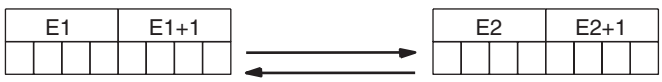
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

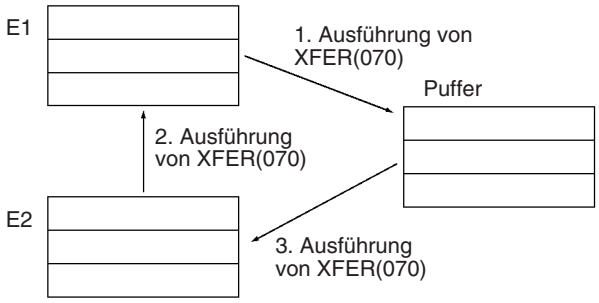
Bereich	E1	E2
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A448 bis A958	
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	---
Datenregister	---	
Indexregister	IR0 bis IR15	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl XCHG(073) tauscht die Inhalte von E1+1 und E1 und die Inhalte von E2+1 und E2 gegeneinander aus.



Verwenden Sie zum Austausch von 3 oder mehr Worten des Befehl XFER(070), um den Inhalt der Worte in eine dritte Gruppe von Worten (einen Puffer) zu übertragen, wie in der folgenden Abbildung gezeigt.

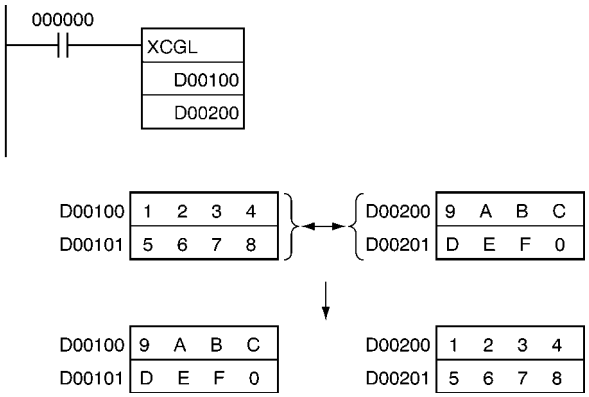


Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

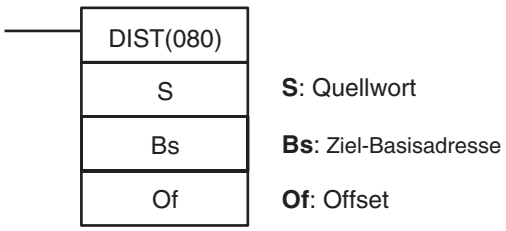
Beispiel Wenn CIO 000000 in dem folgenden Beispiel EIN ist, werden die Inhalte von D00100 und D00101 mit den Inhalten von D00200 und D00201 vertauscht.



3-8-12 SINGLE WORD DISTRIBUTE: DIST(080)

Funktion Überträgt das Quellwort an das durch Addition eines Offsetwerts zur Basisadresse errechnete Zielwort.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	DIST(080)
	Ausführung einmal bei steigender Flanke	@DIST(080)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

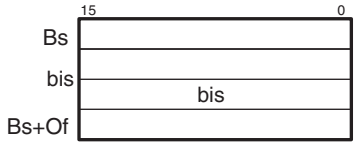
Operanden

Bs: Ziel-Basisadresse

Spezifiziert die Ziel-Basisadresse. Zu dieser Adresse wird der Offset addiert, um die Adresse des Zielworts zu berechnen.

Of: Offset

Dieser Wert wird zu der Basisadresse addiert, um die Adresse des Zielworts zu berechnen. Der Offset kann ein beliebiger Wert zwischen 0000 und FFFF hex (0 bis 65.535 dezimal) sein, allerdings müssen Bs und Bs+Of im gleichen Datenbereich liegen.

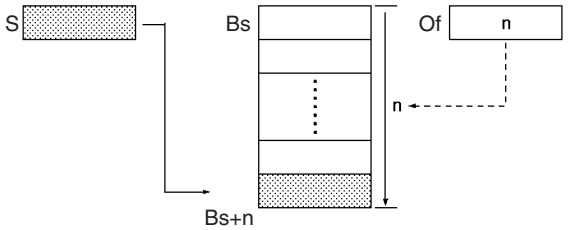


Operanden-Spezifikationen

Bereich	S	Bs	Of
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959	A448 bis A959	A000 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	#0000 bis #FFFF (binär) oder &0 bis &65535
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl DIST(080) kopiert S in das Zielwort, dessen Adresse durch die Addition von Of zu Bs berechnet wird. Es kann der gleiche DIST(080)-Befehl verwendet werden, um den Inhalt des Quellworts an verschiedene Worte im gleichen Datenbereich zu verteilen, indem der Wert von Of geändert wird.



Flags

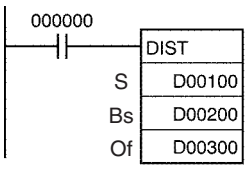
Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn die Quellendaten 0000 sind. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit der Quelldaten 1 ist. In allen anderen Fällen AUS.

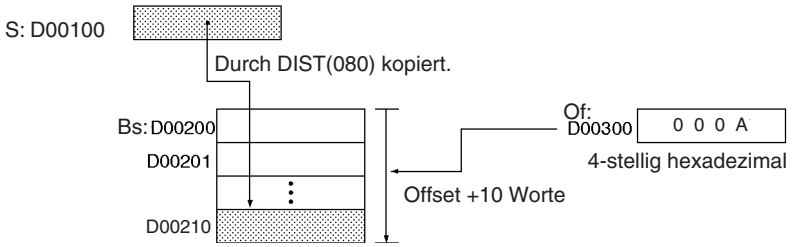
Vorsichtsmaßnahmen

Achten Sie darauf, dass der Offset nicht zu einer Überschreitung des Datenbereichs führt, also dass Bs und Bs+Of im gleichen Datenbereich liegen.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird der Inhalt von D00100 in D00210 (D00200 + 10) kopiert, wenn der Inhalt von D00300 den Wert 10 (0A hexadezimal) hat. Der Inhalt von D00100 kann auch in andere Worte kopiert werden, indem der Offset in D00300 geändert wird.



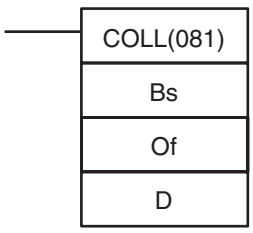


3-8-13 DATA COLLECT: COLL(081)

Funktion

Überträgt das Quellwort (errechnet durch Addition eines Offsetwerts zur Basisadresse) an das Zielwort.

Kontaktplansymbol



Bs: Quell-Basisadresse

Of: Offset

D: Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	COLL(081)
	Ausführung einmal bei steigender Flanke	@COLL(081)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

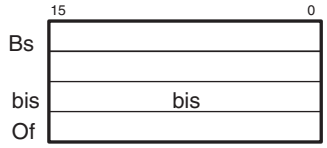
Operanden

Bs: Quellen-Basisadresse

Spezifiziert die Quellen-Basisadresse. Zu dieser Adresse wird der Offset addiert, um die Adresse des Quellworts zu berechnen.

Of: Offset

Dieser Wert wird zu der Basisadresse addiert, um die Adresse des Quellworts zu berechnen. Der Offset kann ein beliebiger Wert zwischen 0000 und FFFF hex (0 bis 65.535 dezimal) sein, allerdings müssen Bs und Bs+Of im gleichen Datenbereich liegen.

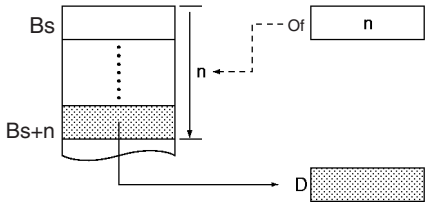


Operanden-Spezifikationen

Bereich	Bs	Of	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0000 bis #FFFF (binär) oder &0 bis &65535	---
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl COLL(081) kopiert den Inhalt des Quellworts (dessen Adresse durch die Addition von Of zu Bs berechnet wird) in das Zielwort. Der gleiche COLL(080)-Befehl kann verwendet werden, um Daten aus verschiedenen Quellworten innerhalb des Datenbereichs zu sammeln, indem der Wert von Of geändert wird.



Flags

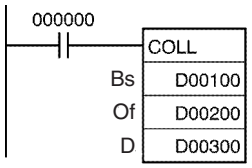
Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn die Quellendaten 0000 sind. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit der Quelldaten 1 ist. In allen anderen Fällen AUS.

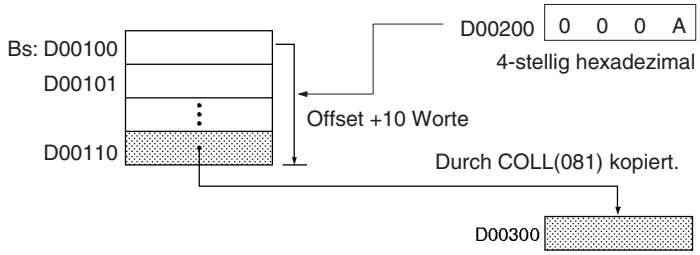
Vorsichtsmaßnahmen

Achten Sie darauf, dass der Offset nicht zu einer Überschreitung des Datenbereichs führt, also dass Bs und Bs+Of im gleichen Datenbereich liegen.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird der Inhalt von D00110 (D00100 + 10) in D00300 kopiert, wenn der Inhalt von D00300 den Wert 10 (0A hexadezimal) hat. Der Inhalt anderer Worte kann in D00300 kopiert werden, indem der Offset in D00200 geändert wird.



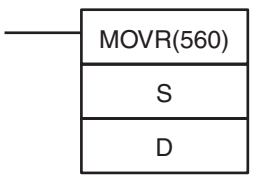


3-8-14 MOVE TO REGISTER: MOVR(560)

Funktion

Schreibt die SPS-Speicheradresse des spezifizierten Worts, Bits oder Zeitgeber-/Zähler-Fertig-Bits in das spezifizierte Indexregister. (Verwenden Sie den Befehl MOVRW(561) zum Festlegen der SPS-Speicheradresse des jeweiligen Istwerts für einen Zeitgeber/Zähler im Indexregister.)

Kontaktplansymbol



S: Quelle (gewünschtes Wort oder Bit)
D: Ziel (Indexregister)

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MOVR(560)
	Ausführung einmal bei steigender Flanke	@MOVR(560)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

D: Ziel

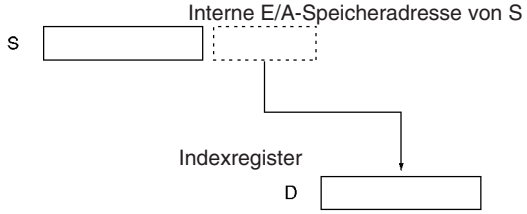
Das Ziel muss ein Indexregister (IR0 bis IR15) sein.

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6143 CIO 000000 bis CIO 614315	---
Arbeitsbereich nicht remanent	W000 bis W511 W00000 bis W51115	---
Arbeitsbereich remanent	H000 bis H511 H00000 bis H51115	---
Systembereich	A000 bis A447 A448 bis A959 A00000 bis A44715 A44800 bis A95915	---
Zeitgeberbereich	T0000 bis T4095 (Fertig-Bit)	---
Zählerbereich	C0000 bis C4095 (Fertig-Bit)	---
Task-Flag	TK0000 bis TK0031	---
DM-Bereich	D00000 bis D32767	---
EM-Bereich ohne Bank	E00000 bis E32767	---
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	---
Indirekte DM/EM-Binäradressen	---	
Indirekte DM/EM-Adressen in BCD	---	
Konstanten	---	
Datenregister	---	
Indexregister	---	IR0 bis IR15
Indirektes Adressieren mittels Indexregistern	---	

Beschreibung

Der Befehl MOVR(560) ermittelt die SPS-Speicheradresse (absolute Adresse) von S und schreibt diese Adresse in D (ein Indexregister).



Wenn in S ein Zeitgeber oder Zähler spezifiziert ist, schreibt MOVR(560) die SPS-Speicheradresse des Zeitgeber-/Zähler-Fertig-Bits in D. Verwenden Sie den Befehl MOVRW(561), um die SPS-Speicheradresse des Zeitgeber-/Zähler-Istwerts in D zu schreiben.

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Vorsichtsmaßnahmen

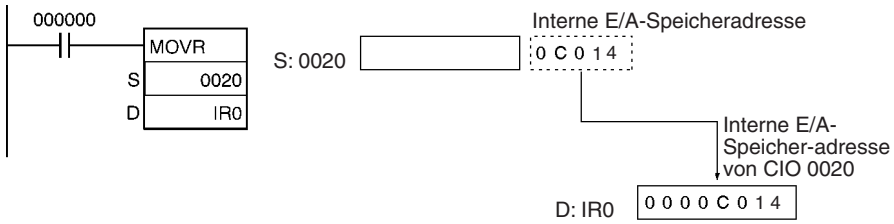
Der Befehl MOVR(560) kann nicht die SPS-Speicheradressen von Zeitgeber-/Zähler-Istwerten schreiben. Verwenden Sie den Befehl MOVRW(561), um die SPS-Speicheradressen von Zeitgeber-/Zähler-Istwerten zu schreiben.

Der Inhalt eines Indexregisters ist in einer Interrupt-Task nicht vorhersehbar, so lange er nicht geschrieben wird. Achten Sie darauf, den Inhalt eines Indexregister mit MOVR(560) in einer Interrupt-Task zu schreiben, bevor Sie das Indexregister verwenden.

Alle in einer Interrupt-Task an den Inhalten von IR und DR erfolgten Änderungen wirken sich nicht auf den Inhalt des Registers in einer zyklischen Task aus.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, schreibt MOVR(560) die SPS-Speicheradresse von CIO 0020 in IR0.

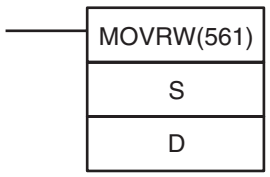


3-8-15 MOVE TIMER/COUNTER PV TO REGISTER: MOVRW(561)

Funktion

Schreibt die SPS-Speicheradresse des spezifizierten Zeitgeber- oder Zähler-Istwerts in das spezifizierte Indexregister. (Verwenden Sie den Befehl MOVR(560), um die SPS-Speicheradresse eines Worts, Bits oder Zeitgeber-/Zähler-Fertig-Bits in ein Indexregister zu schreiben.)

Kontaktplansymbol



S: Quelle (gewünschte Zeitgeber/Zähler-Nummer)

D: Ziel (Indexregister)

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MOVR(561)
	Ausführung einmal bei steigender Flanke	@MOVR(561)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

D: Ziel

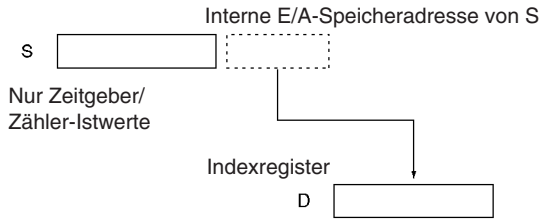
Das Ziel muss ein Indexregister (IR0 bis IR15) sein.

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	---	
Arbeitsbereich nicht remanent	---	
Arbeitsbereich remanent	---	
Systembereich	---	
Zeitgeberbereich	T0000 bis T4095 (Istwert)	---
Zählerbereich	C0000 bis C4095 (Istwert)	---
DM-Bereich	---	
EM-Bereich ohne Bank	---	
EM-Bereich mit Bank	---	
Indirekte DM/EM-Binäradressen	---	
Indirekte DM/EM-Adressen in BCD	---	
Konstanten	---	
Datenregister	---	
Indexregister	---	IR0 bis IR15
Indirektes Adressieren mittels Indexregistern	---	

Beschreibung

Der Befehl MOVRW(561) ermittelt die SPS-Speicheradresse des in S spezifizierten Zeitgebers bzw. Zählers und schreibt diese Adresse in D (ein Indexregister).



MOVRW(561) schreibt die SPS-Speicheradresse des Zeitgeber- oder Zähler-Istwerts in D. Verwenden Sie den Befehl MOVR(560), um die SPS-Speicheradresse des Zeitgeber- oder Zähler-Fertig-Bits zu schreiben.

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

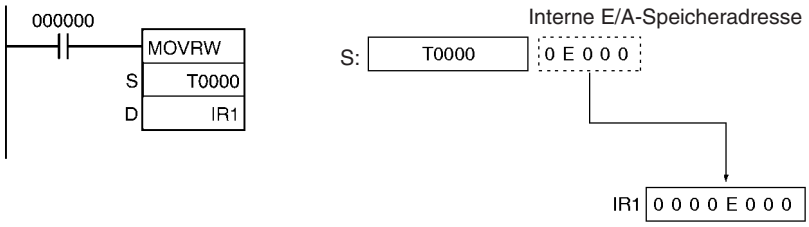
Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Vorsichtsmaßnahmen

Der Befehl MOVRW(561) kann nicht die SPS-Speicherdressen von Datenbereichsworten, Bits oder Zeitgeber-/Zähler-Fertig-Bits schreiben. Verwenden Sie zum schreiben dieser Adressen den Befehl MOVR(560).

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, schreibt MOVRW(561) die SPS-Speicheradresse des Istwerts von Zeitgeber T0000 in IR1.



3-9 Datenverschiebungsbefehle

In diesem Abschnitt werden die Befehle beschrieben, die zum Verschieben von Daten innerhalb von oder zwischen Worten, jedoch um verschiedene Beträge und in verschiedene Richtungen verwendet werden.

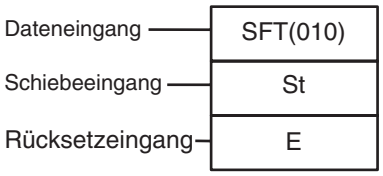
Befehl	AWL	Funktionscode	Seite
SHIFT REGISTER	SFT	010	345
REVERSIBLE SHIFT REGISTER	SFTR	084	346
ASYNCHRONOUS SHIFT REGISTER	ASFT	017	349
WORD SHIFT	WSFT	016	352
ARITHMETIC SHIFT LEFT	ASL	025	354
DOUBLE SHIFT LEFT	ASLL	570	355
ARITHMETIC SHIFT RIGHT	ASR	026	357
DOUBLE SHIFT RIGHT	ASRL	571	358
ROTATE LEFT	ROL	027	360
DOUBLE ROTATE LEFT	ROLL	572	362
ROTATE LEFT WITHOUT CARRY	RLNC	574	367
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL	576	369
ROTATE RIGHT	ROR	028	364
DOUBLE ROTATE RIGHT	RORL	573	365
ROTATE RIGHT WITHOUT CARRY	RRNC	575	371
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL	577	372
ONE DIGIT SHIFT LEFT	SLD	074	374
ONE DIGIT SHIFT RIGHT	SRD	075	376
SHIFT N-BIT DATA LEFT	NSFL	578	377
SHIFT N-BIT DATA RIGHT	NSFR	579	379
SHIFT N-BITS LEFT	NASL	580	381
DOUBLE SHIFT N-BITS LEFT	NSLL	582	384
SHIFT N-BITS RIGHT	NASR	581	387
DOUBLE SHIFT N-BITS RIGHT	NSRL	583	389

3-9-1 SHIFT REGISTER: SFT(010)

Funktion

Steuert ein Schieberegister.

Kontaktplansymbol



St: Anfangswort

E: Endwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SFT(010)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

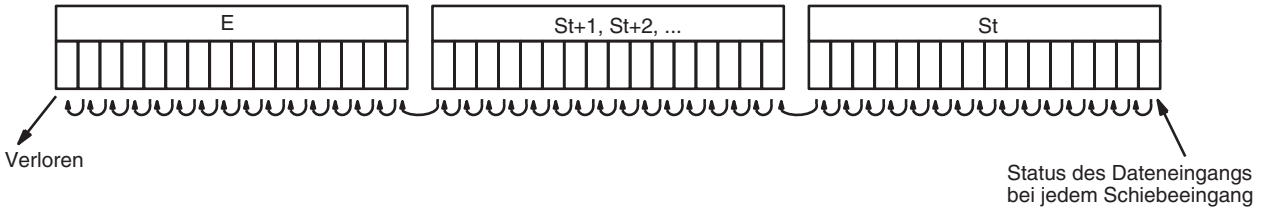
Hinweis St und E müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	St	E
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A448 bis A959	
Zeitgeberbereich	---	
Zählerbereich	---	
DM-Bereich	---	
EM-Bereich ohne Bank	---	
EM-Bereich mit Bank	---	
Indirekte DM/EM-Binäradressen	---	
Indirekte DM/EM-Adressen in BCD	---	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	

Beschreibung

Wenn die Ausführungsbedingung am Schiebeeingang von AUS auf EIN wechselt, werden alle Daten zwischen St und E um ein Bit nach links verschoben (vom niederwertigsten zum höchstwertigen Bit), und der EIN/AUS-Status des Dateneingangs wird in das niederwertigste Bit geschrieben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn die indirekte IR-Adresse von St und E nicht im CIO-, AR-, HR- oder WR-Datenbereich liegt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Bitdaten, die aus dem Schieberegister heraus geschoben werden, werden verworfen.

Wenn der Rücksetzeingang auf EIN wechselt, werden alle Bits zwischen dem niederwertigsten spezifizierten Wort (St) und dem höchstwertigen spezifizierten Wort (E) zurückgesetzt (d. h. auf 0 gesetzt). Der Rücksetzeingang besitzt Priorität gegenüber allen anderen Eingängen.

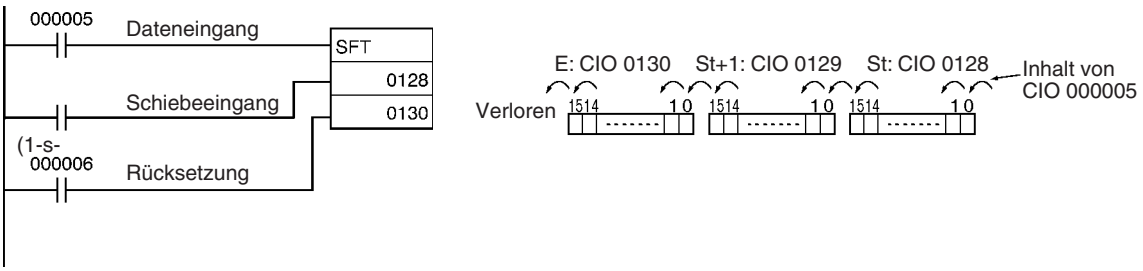
St muss kleiner oder gleich E sein. Doch selbst wenn St auf einen Wert größer als E gesetzt wird, tritt kein Fehler auf, und ein Wort Daten in ST wird verschoben.

Werden St und E indirekt mittels Indexregistern spezifiziert und die tatsächlichen Adressen im E/A-Speicher liegen nicht innerhalb der Speicherbereiche für Daten, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Beispiele

Ein 16 Bits überschreitendes Schieberegister

Im folgenden Beispiel ist ein 48-Bit-Schieberegister dargestellt, das die Worte CIO 0128 bis CIO 0130 nutzt. Es wird ein 1-Sekunden-Taktimpuls verwendet, sodass die von CIO 000005 erzeugte Ausführungsbedingung jede Sekunde in ein 3-Worte-Register zwischen CIO 012800 und CIO 013015 geschoben wird.

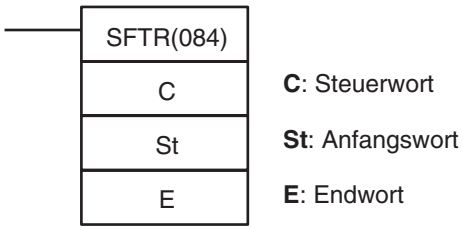


3-9-2 REVERSIBLE SHIFT REGISTER: SFTR(084)

Funktion

Erstellt ein Schieberegister, in dem Daten nach rechts oder nach links verschoben werden.

Kontaktplansymbol



C: Steuerwort

St: Anfangswort

E: Endwort

Variationen

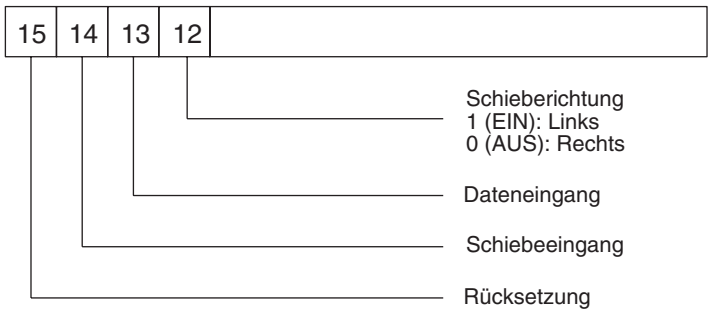
Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	SFTR(084)
	Ausführung einmal bei steigender Flanke	@SFTR(084)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort



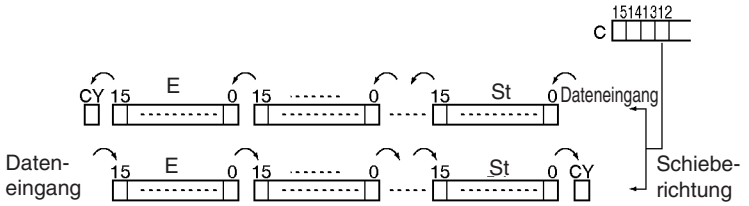
Hinweis St und E müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	C	St	E
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	DR0 bis DR15	---	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Wenn die Ausführungsbedingung des Schiebeeingangs-Bits (Bit 14 von C) auf EIN wechselt, werden alle Daten zwischen St und E um ein Bit in die spezifizierte Schieberichtung (spezifiziert durch Bit 12 von C) verschoben, und der EIN/AUS-Status des Dateneingangs wird im niederwertigsten bzw. höchstwertigen Bit gespeichert. Die Bitdaten, die aus dem Schieberegister heraus verschoben werden, werden im Übertragsflag (CY) gespeichert.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn St größer als E ist. In allen anderen Fällen AUS.
Übertragsflag	CY	EIN, wenn ein 1 in das Übertragsflag geschoben wird. AUS, wenn eine 0 in das Übertragsflag geschoben wird. AUS, wenn die Rücksetzung auf 1 gesetzt ist.

Vorsichtsmaßnahmen

Die oben beschriebenen Schiebevorgänge werden ausgeführt, wenn das Rücksetzbit (Bit 15 von C) auf AUS gesetzt ist.

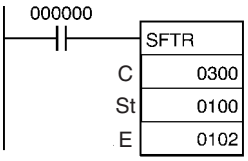
Wenn die Rücksetzung (Bit 15 von C) auf EIN gesetzt wird, werden alle Bits im Schieberegister, von St bis E zurückgesetzt (d. h., auf 0 gesetzt).

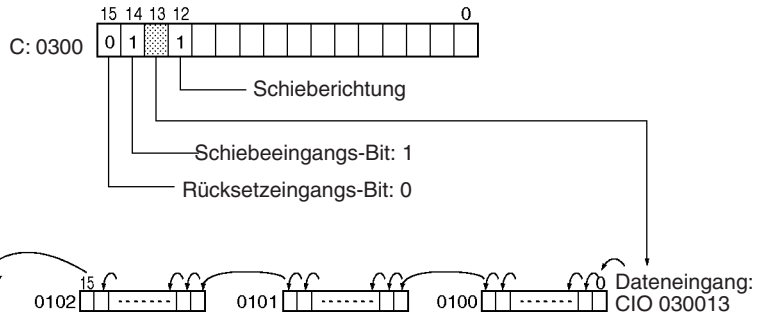
Wenn St größer als E ist, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Beispiele

Verschieben von Daten

Wenn der Schiebeeingang CIO 030014 auf EIN wechselt, während CIO 000000 EIN ist und das Rücksetzbit CIO 030015 auf AUS gesetzt ist, werden die Worte CIO 0100 bis CIO 0102 um ein Bit in die durch CIO 030012 spezifizierte Richtung (in diesem Beispiel 1: nach rechts) verschoben, und der Inhalt des Eingangsbits CIO 030013 wird in das niederwertigste Bit, CIO 010000, verschoben. Der Inhalt von CIO 010215 wird in das Übertragsflag (CY) verschoben.





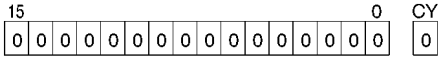
Rücksetzen der Daten

Wenn CIO 030014 EIN ist, während CIO 000000 und das Rücksetzbit CIO 030015 auf EIN gesetzt sind, werden die Worte CIO 0100 bis CIO 0102 und das Übertragsflag auf 0 zurückgesetzt.

Steuerung von Daten

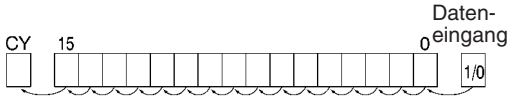
Rücksetzen der Daten

Alle Bits von St bis E und das Übertragsflag werden auf 0 gesetzt und es können 0 and es können keine anderen Daten empfangen werden, solange das Rücksetzeingangs-Bit (Bit 15 von C) EIN ist.



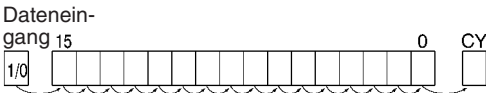
Verschieben von Daten nach links (vom niederwertigsten zum höchstwertigen Bit)

Wenn das Schiebeeingangs-Bit (Bit 14 von C) EIN ist, wird der Inhalt des Eingangsbits (Bit 13 von C) in Bit 00 des Anfangsworts verschoben, und alle folgenden Bits werden um ein Bit nach links verschoben. Der Status von Bit 15 des Endworts wird in das Übertragsflag verschoben.



Verschieben von Daten nach rechts (vom höchstwertigen zum niederwertigsten Bit)

Wenn das Schiebeeingangs-Bit (Bit 14 von C) EIN ist, wird der Inhalt des Eingangsbits (Bit 13 von C) (E/A) in Bit 15 des Endworts verschoben, und alle folgenden Bits werden um ein Bit nach rechts verschoben. Der Status von Bit 00 des Anfangsworts wird in das Übertragsflag verschoben.

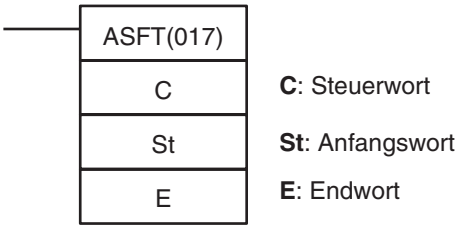


3-9-3 ASYNCHRONOUS SHIFT REGISTER: ASFT(017)

Funktion

Verschiebt alle Wortdaten ungleich Null innerhalb des spezifizierten Wortbereichs entweder in Richtung St oder in Richtung E, wobei die 0000-hex-Wortdaten ersetzt werden.

Kontaktplansymbol



Variationen

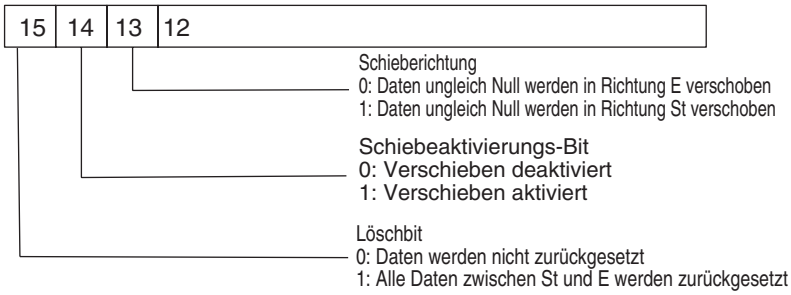
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ASFT(017)
	Ausführung einmal bei steigender Flanke	@ASFT(017)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort



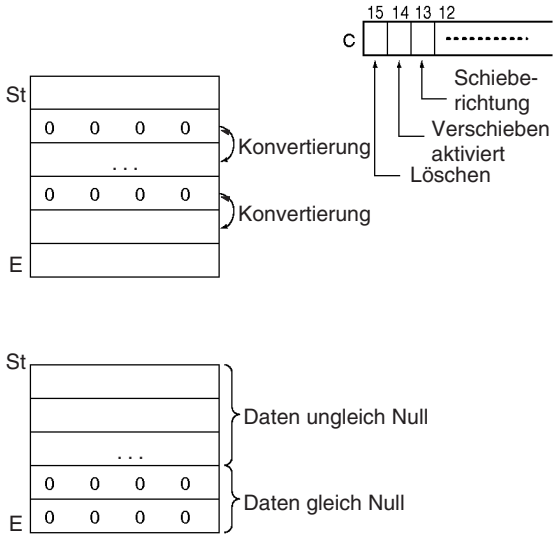
Hinweis St und E müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	C	St	E
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	DR0 bis DR15	---	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Wenn das Schiebeaktivierungs-Bit (Bit 14 von C) EIN ist, werden alle Worte innerhalb des Wortebereichs zwischen St und E, die nicht den Inhalt 0 haben, um ein Wort in die durch das Schieberichtungs-Bit (Bit 13 von C) bestimmte Richtung verschoben, wenn das Wort in Schieberichtung nur Nullen enthält. Wenn der Befehl ASFT(017) oft genug wiederholt ausgeführt wird, werden alle nur Nullen enthaltenden Worte durch Worte mit einem Inhalt ungleich Null ersetzt. Dies führt zu dem Ergebnis, dass alle Daten zwischen St und E in Nulldaten und Daten ungleich Null aufgeteilt werden.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn St größer als E ist. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstellen-Frei-Flag der als *Komm.-Port-Nummer* für *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

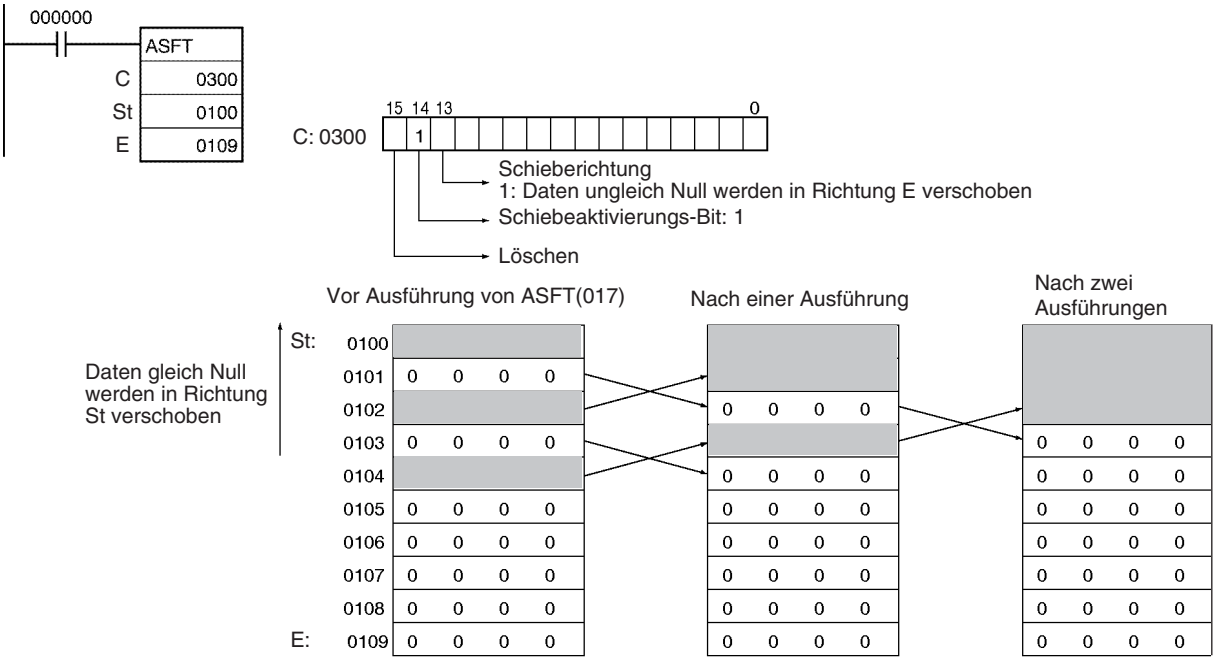
Wenn das Löschbit (Bit 15 von C) auf EIN gesetzt wird, werden alle Bits im Schieberegister von St bis E zurückgesetzt (d. h., auf 0 gesetzt). Das Löschbit besitzt Priorität gegenüber dem Schiebeaktivierungs-Bit (Bit 14 von C).

Wenn St größer als E ist, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Beispiele

Verschieben von Daten:

Wenn das Schiebeaktivierungs-Bit CIO 030014 auf EIN wechselt, während CIO 000000 EIN ist, werden alle Worte mit einem Inhalt ungleich Null zwischen CIO 0100 und CIO 0109 in die durch das Schieberichtungs-Bit CIO 030013 bestimmte Richtung (in diesem Beispiel 1: in Richtung St) verschoben, sofern das Wort links des Worts mit Daten ungleich Null nur Nullen enthält.

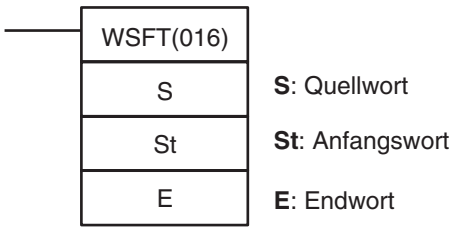


3-9-4 WORD SHIFT: WSFT(016)

Funktion

Verschiebt Daten zwischen St und E in Worteinheiten.

Kontaktplansymbol



Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	WSFT(016)
	Ausführung einmal bei steigender Flanke	@WSFT(016)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Hinweis St und E müssen im gleichen Datenbereich liegen.

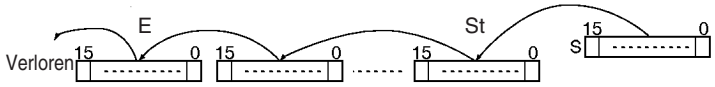
Operanden-Spezifikationen

Bereich	S	St	E
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		

Bereich	S	St	E
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	
Datenregister	DR0 bis DR15	---	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl WSFT(016) verschiebt Daten zwischen St und E in Worteinheiten, und schreibt die Daten aus dem Quellwort S in St. Der Inhalt von E geht verloren.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn St größer als E ist. In allen anderen Fällen AUS.

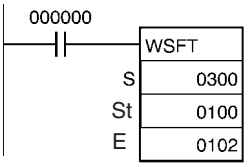
Vorsichtsmaßnahmen

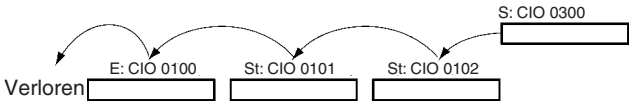
Wenn St größer als E ist, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Hinweis Wenn große Datenmengen verschoben werden, ist die Befehlsausführungszeit relativ lang. Achten Sie darauf, dass die Spannungsversorgung nicht während der Ausführung von WSFT(016) unterbrochen wird, wodurch der Verschiebevorgang unvollendet abgebrochen würde.

Beispiele

Wenn CIO 000000 EIN ist, werden die Daten zwischen CIO 0100 und CIO 0102 um ein Wort in Richtung E verschoben. Der Inhalt von CIO 0300 wird in CIO 0100 gespeichert und der Inhalt von CIO 0102 geht verloren.





3-9-5 ARITHMETIC SHIFT LEFT: ASL(025)

Funktion

Verschiebt den Inhalt von Wd ein Bit nach links.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ASL(025)
	Ausführung einmal bei steigender Flanke	@ASL(025)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

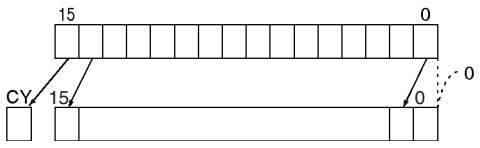
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ASL(025) verschiebt den Inhalt von Wd um ein Bit nach links (von niederwertigsten zum höchstwertigen Bit). Im niederwertigsten Bit wird der Wert 0 gespeichert und die Daten des höchstwertigen Bits werden in das Übertrags-Flag (CY) verschoben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

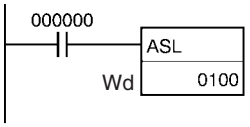
Nach der Ausführung von ASL(025) wird das Fehler-Flag auf AUS gesetzt.

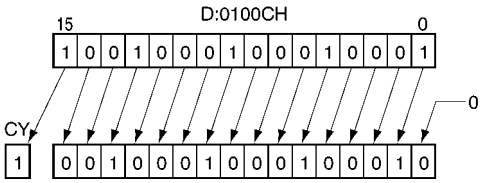
Wenn der Inhalt von Wd als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von Wd als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, wird CIO 0100 um ein Bit nach links verschoben. In CIO 010000 wird eine 0 geschrieben und der Inhalt von CIO 010115 wird in das Übertrags-Flag (CY) verschoben.



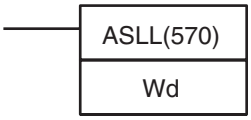


3-9-6 DOUBLE SHIFT LEFT: ASLL(570)

Funktion

Verschiebt den Inhalt von Wd und Wd+1 ein Bit nach links.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ASLL(570)
	Ausführung einmal bei steigender Flanke	@ASLL(570)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

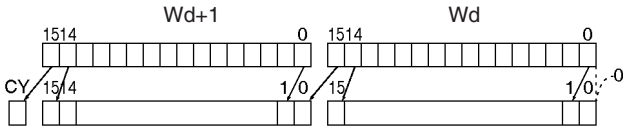
Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510

Bereich	Wd
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ASLL(570) verschiebt den Inhalt von Wd und Wd+1 um ein Bit nach links (von niederwertigsten zum höchstwertigen Bit). In das niederwertigste Bit von Wd wird eine 0 geschrieben und das höchstwertige Bit von Wd und Wd+1 wird in das Übertrags-Flag (CY) verschoben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

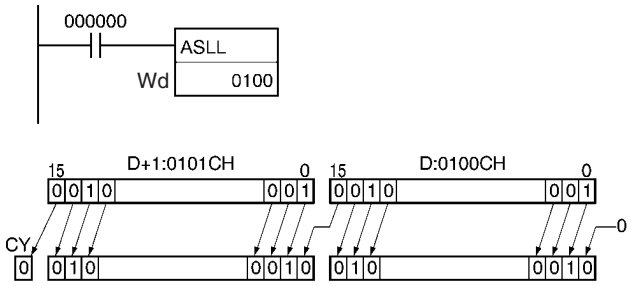
Nach der Ausführung von ASLL(570) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von Wd und Wd+1 als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von Wd und Wd+1 als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, werden CIO 0100 und CIO 0101 um ein Bit nach links verschoben. In CIO 010000 wird eine 0 geschrieben und der Inhalt von CIO 010015 wird in das Übertrags-Flag (CY) verschoben.

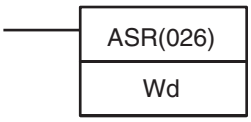


3-9-7 ARITHMETIC SHIFT RIGHT: ASR(026)

Funktion

Verschiebt den Inhalt von Wd ein Bit nach rechts.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	ASR(026)
	Ausführung einmal bei steigender Flanke	@ASR(026)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

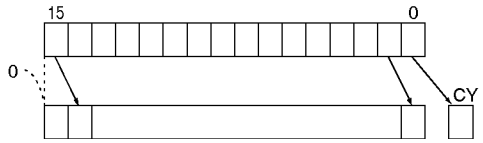
Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	DR0 bis DR15

Bereich	Wd
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ASR(026) verschiebt den Inhalt von Wd um ein Bit nach rechts (von höchstwertigen zum niederwertigsten Bit). In das höchstwertige Bit wird eine 0 geschrieben und der Inhalt des niederwertigsten Bits wird in das Übertrags-Flag (CY) verschoben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	AUS

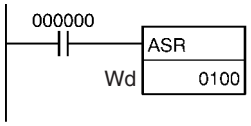
Vorsichtsmaßnahmen

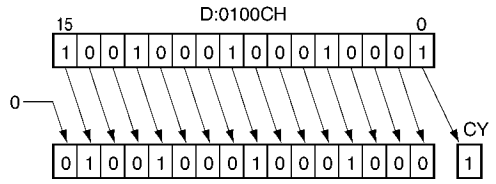
Nach der Ausführung von ASR(026) werden das Fehler-Flag und das Negativ-Flag auf AUS gesetzt.

Wenn der Inhalt von Wd als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, wird CIO 0100 um ein Bit nach rechts verschoben. In CIO 010015 wird eine 0 geschrieben und der Inhalt von CIO 010000 wird in das Übertrags-Flag (CY) verschoben.



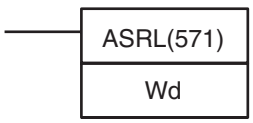


3-9-8 DOUBLE SHIFT RIGHT: ASRL(571)

Funktion

Verschiebt den Inhalt von Wd und Wd+1 ein Bit nach rechts.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	ASRL(571)
	Ausführung einmal bei steigender Flanke	@ASRL(571)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

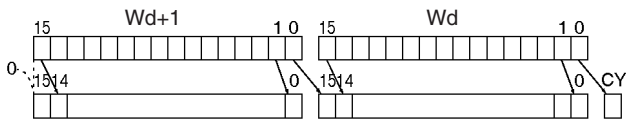
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ASRL(571) verschiebt den Inhalt von Wd und Wd+1 um ein Bit nach rechts (von höchstwertigen zum niederwertigsten Bit). In das höchstwertige Bit von WD+1 wird eine 0 geschrieben und der Inhalt des niederwertigsten Bits von Wd wird in das Übertrags-Flag (CY) verschoben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.

Name	Bezeich-nung	Funktion
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	AUS

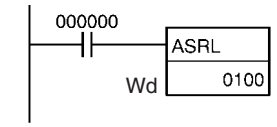
Vorsichtsmaßnahmen

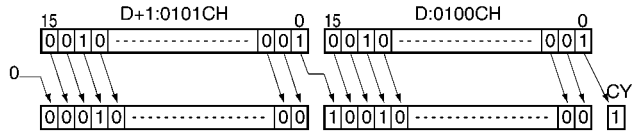
Nach der Ausführung von ASRL(571) werden das Fehler-Flag und das Negativ-Flag auf AUS gesetzt.

Wenn der Inhalt von Wd und Wd+1 als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, werden CIO 0100 und CIO 0101 um ein Bit nach rechts verschoben. In CIO 010115 wird eine 0 geschrieben und der Inhalt von CIO 010000 wird in das Übertrags-Flag (CY) verschoben.



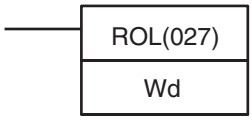


3-9-9 ROTATE LEFT: ROL(027)

Funktion

Verschiebt alle Bits von Wd einschließlich des Übertrags-Flags (CY) ein Bit nach links.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ROL(027)
	Ausführung einmal bei steigender Flanke	@ROL(027)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

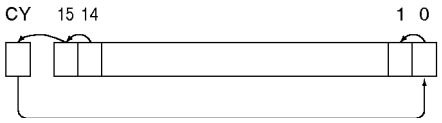
Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095

Bereich	Wd
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ROL(027) verschiebt alle Bits von Wd einschließlich des Übertrags-Flags (CY) nach links (vom niederwertigsten zum höchstwertigen Bit).



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von ROL(027) wird das Fehler-Flag auf AUS gesetzt.

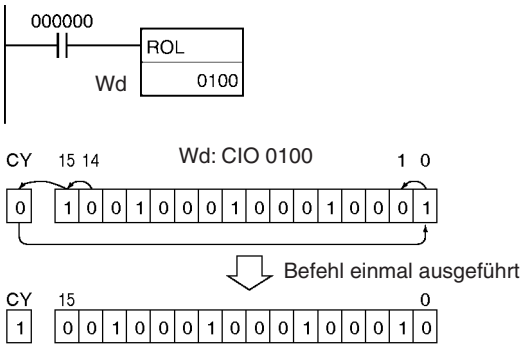
Wenn der Inhalt von Wd als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von Wd als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Hinweis Durch Verwendung der Befehle SET CARRY (STC(040)) bzw. CLEAR CARRY (CLC(041)) ist es möglich, den Inhalt des Übertrags-Flags unmittelbar vor Ausführung des Befehls auf 1 bzw. 0 zu setzen.

Beispiele

Wenn CIO 000000 EIN ist, werden CIO 0100 und das Übertrags-Flag (CY) um ein Bit nach links verschoben. Der Inhalt von CIO 010015 wird in das Übertrags-Flag (CY) verschoben, und der Inhalt des Übertrags-Flags wird in CIO 010000 verschoben.

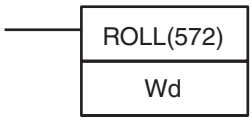


3-9-10 DOUBLE ROTATE LEFT: ROLL(572)

Funktion

Verschiebt alle Bits von Wd und Wd+1 einschließlich des Übertrags-Flags (CY) um ein Bit nach links.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ROLL(572)
	Ausführung einmal bei steigender Flanke	@ROLL(572)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

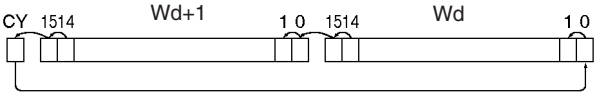
Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---

Bereich	Wd
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ROLL(572) verschiebt alle Bits von Wd und Wd+1 einschließlich des Übertrags-Flags (CY) nach links (vom niederwertigsten zum höchstwertigen Bit).



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von ROLL(572) wird das Fehler-Flag auf AUS gesetzt.

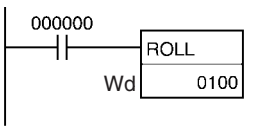
Wenn der Inhalt von Wd und Wd+1 als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

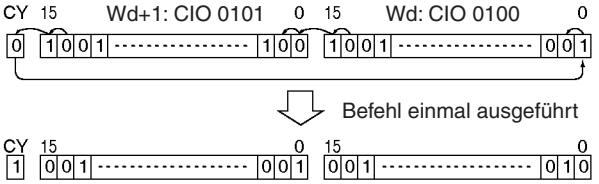
Wenn der Inhalt des höchstwertigen Bits von Wd und Wd+1 als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Hinweis Durch Verwendung der Befehle SET CARRY (STC(040)) bzw. CLEAR CARRY (CLC(041)) ist es möglich, den Inhalt des Übertrags-Flags unmittelbar vor Ausführung des Befehls auf 1 bzw. 0 zu setzen.

Beispiele

Wenn CIO 000000 EIN ist, werden die Worte CIO 0100, CIO 0101 und das Übertrags-Flag (CY) um ein Bit nach links verschoben. Der Inhalt von CIO 010015 wird in das Übertrags-Flag (CY) verschoben, und der Inhalt des Übertrags-Flags wird in CIO 010000 verschoben.



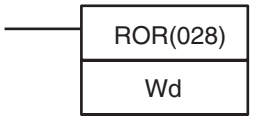


3-9-11 ROTATE RIGHT: ROR(028)

Funktion

Verschiebt alle Bits von Wd einschließlich des Übertrags-Flags (CY) um ein Bit nach rechts.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	ROR(028)
	Ausführung einmal bei steigender Flanke	@ROR(028)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

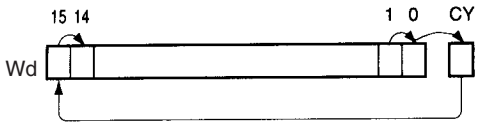
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ROR(027) verschiebt alle Bits von Wd einschließlich des Übertrags-Flags (CY) nach rechts (vom höchstwertigen zum niederwertigsten Bit).



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von ROR(028) wird das Fehler-Flag auf AUS gesetzt.

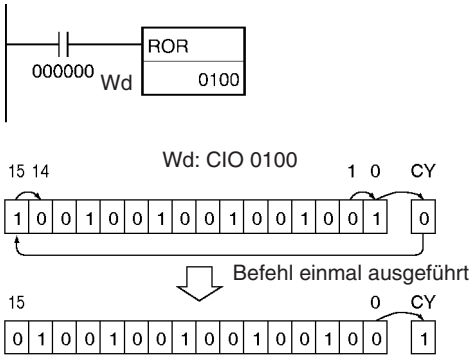
Wenn der Inhalt von Wd als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von Wd als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Hinweis Durch Verwendung der Befehle SET CARRY (STC(040)) bzw. CLEAR CARRY (CLC(041)) ist es möglich, den Inhalt des Übertrags-Flags unmittelbar vor Ausführung des Befehls auf 1 bzw. 0 zu setzen.

Beispiele

Wenn CIO 000000 EIN ist, werden CIO 0100 und das Übertrags-Flag (CY) um ein Bit nach rechts verschoben. Der Inhalt von CIO 010000 wird in das Übertrags-Flag (CY) verschoben, und der Inhalt des Übertrags-Flags wird in CIO 010015 verschoben.

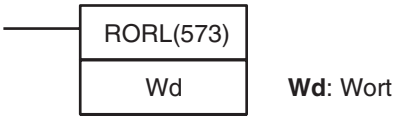


3-9-12 DOUBLE ROTATE RIGHT: RORL(573)

Funktion

Verschiebt alle Bits von Wd und Wd+1 einschließlich des Übertrags-Flags (CY) um ein Bit nach rechts.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	RORL(573)
	Ausführung einmal bei steigender Flanke	@RORL(573)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

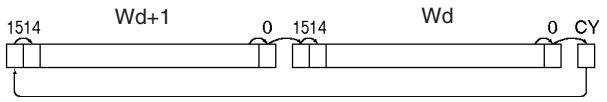
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl RORL(573) verschiebt alle Bits von Wd und Wd+1 einschließlich des Übertrags-Flags (CY) nach rechts (vom höchstwertigen zum niederwertigsten Bit).



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von RORL(573) wird das Fehler-Flag auf AUS gesetzt.

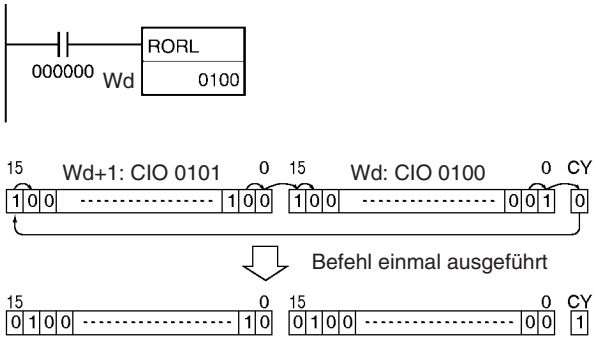
Wenn der Inhalt von Wd und Wd+1 als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von Wd und Wd+1 als Ergebnis der Verschiebung +1 ist, wird das Negativ-Flag auf EIN gesetzt.

Hinweis Durch Verwendung der Befehle SET CARRY (STC(040)) bzw. CLEAR CARRY (CLC(041)) ist es möglich, den Inhalt des Übertrags-Flags unmittelbar vor Ausführung des Befehls auf 1 bzw. 0 zu setzen.

Beispiele

Wenn CIO 000000 EIN ist, werden die Worte CIO 0100, CIO 0101 und das Übertrags-Flag (CY) um ein Bit nach rechts verschoben. Der Inhalt von CIO 010000 wird in das Übertrags-Flag (CY) verschoben, und der Inhalt des Übertrags-Flags wird in CIO 010115 verschoben.

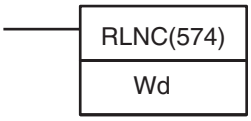


3-9-13 ROTATE LEFT WITHOUT CARRY: RLNC(574)

Funktion

Verschiebt alle Bits von Wd ohne Übertrags-Flag (CY) ein Bit nach links.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	RLNC(574)
	Ausführung einmal bei steigender Flanke	@RLNC(574)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

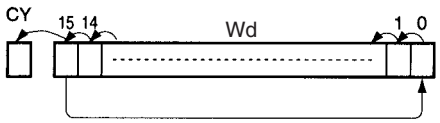
Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)

Bereich	Wd
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl RLNC(574) verschiebt alle Bits von Wd nach links (von niederwertigsten zum höchstwertigen Bit). Der Inhalt des höchstwertigen Bits (links) von Wd wird zum niederwertigsten Bit (rechts) und zum Übertrags-Flag (CY) verschoben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

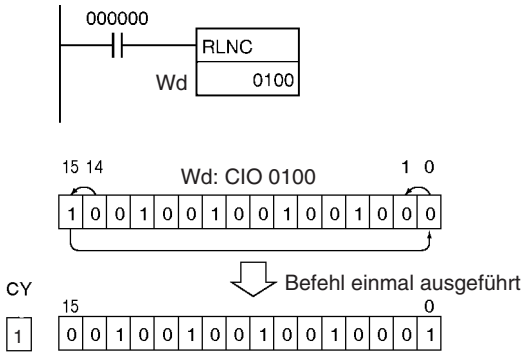
Nach der Ausführung von RLNC(574) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von Wd als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von Wd als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, wird das Wort CIO 0100 (ohne das Übertrags-Flag (CY)) um ein Bit nach links verschoben. Der Inhalt von CIO 010015 wird in CIO 010000 verschoben.

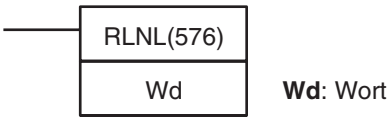


3-9-14 DOUBLE ROTATE LEFT WITHOUT CARRY: RLNL(576)

Funktion

Verschiebt alle Bits von Wd und Wd+1 ohne das Übertrags-Flag (CY) um ein Bit nach links.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	RLNL(576)
	Ausführung einmal bei steigender Flanke	@RLNL(576)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

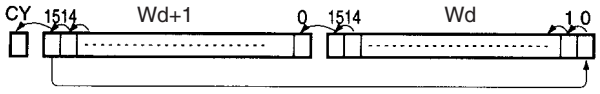
Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---

Bereich	Wd
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl RLNL(576) verschiebt alle Bits von Wd und Wd+1 nach links (von niederwertigsten zum höchstwertigen Bit). Der Inhalt des höchstwertigen Bits von Wd+1 wird zum niederwertigsten Bit von Wd und zum Übertrags-Flag (CY) verschoben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

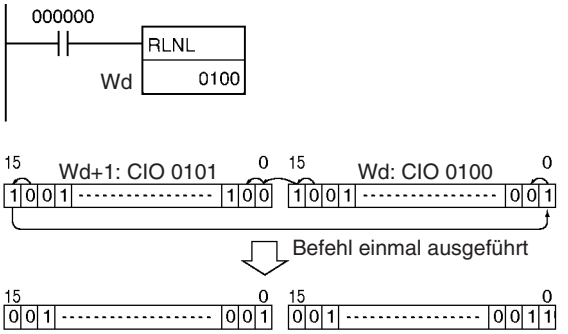
Nach der Ausführung von RLNL(576) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von Wd und Wd+1 als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von Wd und Wd+1 als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, werden CIO 0100 und CIO 0101 (ohne das Übertrags-Flag (CY)) um ein Bit nach links verschoben. Der Inhalt von CIO 010115 wird in CIO 010000 verschoben.

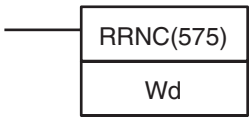


3-9-15 ROTATE RIGHT WITHOUT CARRY: RRNC(575)

Funktion

Verschiebt alle Bits von Wd ohne das Übertrags-Flag (CY) ein Bit nach rechts. Der Inhalt des niederwertigsten Bits (rechts) von Wd wird zum höchstwertigen Bit (links) und dem Übertrags-Flag (CY) verschoben.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	RRNC(575)
	Ausführung einmal bei steigender Flanke	@RRNC(575)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

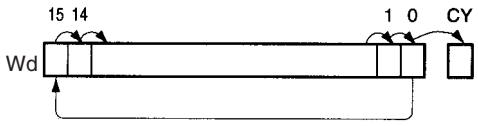
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl RRNC(575) verschiebt alle Bits von Wd ohne das Übertrags-Flag (CY) nach rechts (vom höchstwertigen zum niederwertigsten Bit).



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

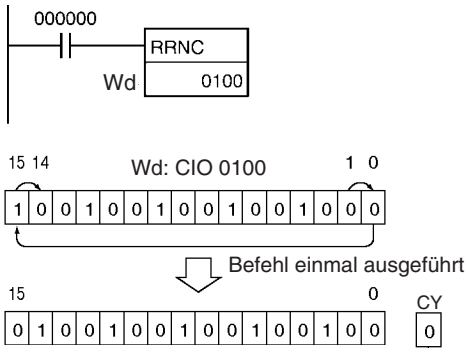
Nach der Ausführung von RRNC(575) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von Wd als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von Wd als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, wird das Wort CIO 0100 (ohne das Übertrags-Flag (CY)) um ein Bit nach rechts verschoben. Der Inhalt von CIO 010000 wird in CIO 010015 verschoben.

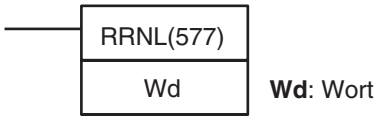


3-9-16 DOUBLE ROTATE RIGHT WITHOUT CARRY: RRNL(577)

Funktion

Verschiebt alle Bits von Wd und Wd+1 ohne das Übertrags-Flag (CY) um ein Bit nach rechts. Der Inhalt des niederwertigsten Bits von Wd+1 wird zum höchstwertigen Bit von Wd und zum Übertrags-Flag (CY) verschoben.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	RRNL(577)
	Ausführung einmal bei steigender Flanke	@RRNL(577)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

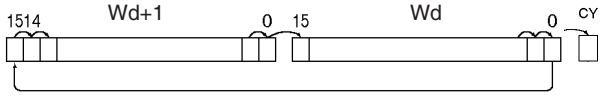
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl RRNL(577) verschiebt alle Bits von Wd und Wd+1 ohne das Übertrags-Flag (CY) nach rechts (vom höchstwertigen zum niederwertigsten Bit).



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

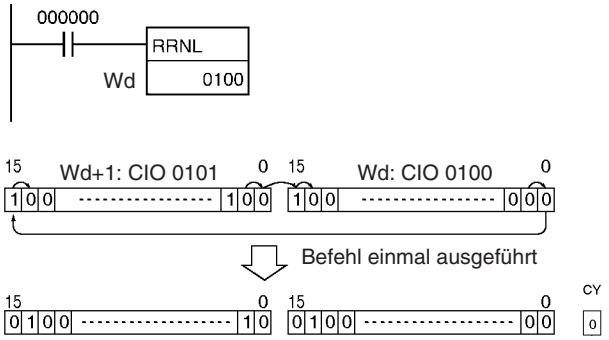
Nach der Ausführung von RRNL(577) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von Wd und Wd+1 als Ergebnis der Verschiebung Null ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von Wd und Wd+1 als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Hinweis Durch Verwendung der Befehle SET CARRY (STC(040)) bzw. CLEAR CARRY (CLC(041)) ist es möglich, den Inhalt des Übertrags-Flags unmittelbar vor Ausführung des Befehls auf 1 bzw. 0 zu setzen.

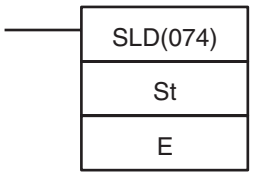
Beispiele Wenn CIO 000000 EIN ist, werden die Worte CIO 0100 und CIO 0101 (ohne das Übertrags-Flag (CY)) um ein Bit nach rechts verschoben. Der Inhalt von CIO 010000 wird in CIO 010115 verschoben.



3-9-17 ONE DIGIT SHIFT LEFT: SLD(074)

Funktion Verschiebt Daten um eine Stelle (4 Bits) nach links.

Kontaktplansymbol



St: Anfangswort

E: Endwort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SLD(074)
	Ausführung einmal bei steigender Flanke	@SLD(074)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Hinweis St und E müssen im gleichen Datenbereich liegen.

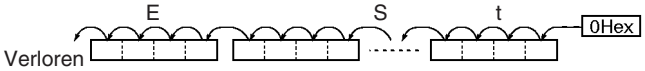
Operanden-Spezifikationen

Bereich	St	E
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	

Bereich	St	E
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl SLD(074) verschiebt die Daten zwischen St und E um eine Stelle (4 Bits) nach links. In die niederwertigste Stelle (Bits 3 bis 0 von St) wird 0 geschrieben und der Inhalt der höchstwertigen Stelle (Bits 15 bis 12 von E) geht verloren.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn St größer als E ist. In allen anderen Fällen AUS.

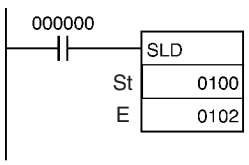
Vorsichtsmaßnahmen

Wenn St größer als E ist, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Hinweis Wenn große Datenmengen verschoben werden, ist die Befehlsausführungszeit relativ lang. Achten Sie darauf, dass die Spannungsversorgung nicht während der Ausführung von SLD(074) unterbrochen wird, wodurch der Verschiebevorgang unvollendet abgebrochen würde.

Beispiele

Wenn CIO 000000 EIN ist, werden die Worte CIO 0100 bis CIO 0102 um eine Stelle (4 Bits) nach links verschoben. In die Bits 0 bis 3 des Worts CIO 0100 wird eine 0 geschrieben und der Inhalt der Bits 12 bis 15 von CIO 0102 geht verloren.



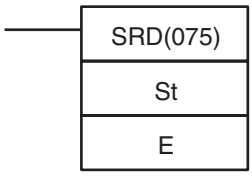


3-9-18 ONE DIGIT SHIFT RIGHT: SRD(075)

Funktion

Verschiebt Daten um eine Stelle (4 Bits) nach rechts.

Kontaktplansymbol



St: Anfangswort

E: Endwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SRD(075)
	Ausführung einmal bei steigender Flanke	@SRD(075)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

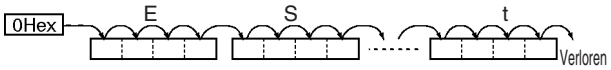
Hinweis St und E müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	St	E
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl SRD(075) verschiebt die Daten zwischen St und E um eine Stelle (4 Bits) nach rechts. In die höchstwertige Stelle (Bits 15 bis 12 von E) wird 0 geschrieben und der Inhalt der niederwertigsten Stelle (Bits 3 bis 0 von St) geht verloren.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn St größer als E ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

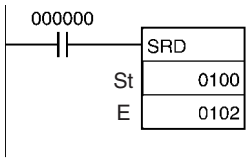
Wenn St größer als E ist, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

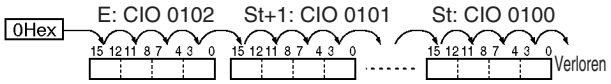
Nach der Ausführung von SRD(075) werden Gleich-Flag und Negativ-Flag auf AUS gesetzt.

Hinweis Wenn große Datenmengen verschoben werden, ist die Befehlsausführungszeit relativ lang. Achten Sie darauf, dass die Spannungsversorgung nicht während der Ausführung von SRD(075) unterbrochen wird, wodurch der Verschiebevorgang unvollendet abgebrochen würde.

Beispiele

Wenn CIO 000000 EIN ist, werden die Worte CIO 0100 bis CIO 0102 um eine Stelle (4 Bits) nach rechts verschoben. In die Bits 12 bis 15 des Worts CIO 0102 wird eine 0 geschrieben und der Inhalt der Bits 0 bis 3 von CIO 0100 geht verloren.



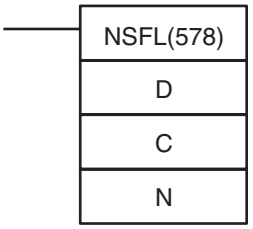


3-9-19 SHIFT N-BIT DATA LEFT: NSFL(578)

Funktion

Verschiebt die spezifizierte Anzahl von Bits nach links.

Kontaktplansymbol



D: Anfangswort für das Verschieben

C: Anfangsbit

N: Verschiebedatenlänge

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	NSFL(578)
	Ausführung einmal bei steigender Flanke	@NSFL(578)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: 0000 bis 000F hex (0 bis 15)
N: 0000 bis FFFF hex (0 bis 65.535)

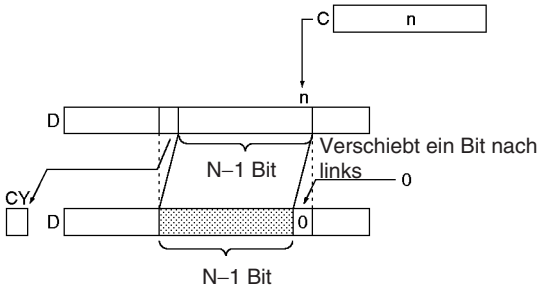
Hinweis Alle Worte im Schieberegister müssen im gleichen Bereich liegen.

Operanden-Spezifikationen

Bereich	D	C	N
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A448 bis A959	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0000 bis #000F (binär) oder &0 bis &15	#0000 bis #FFFF (binär) oder &0 bis &65535
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl NSFL(578) verschiebt die durch die Verschiebedatenlänge (N) spezifizierte Anzahl von Bits mit dem Anfangsbit (C) im durch D spezifizierten niederwertigsten Wort beginnend um ein Bit nach links (zum höchstwertigen Wort und höchstwertigen Bit). Im Anfangsbit wird der Wert 0 gespeichert und der Inhalt des höchstwertigen Bits im Verschiebebereich wird in das Übertrags-Flag (CY) verschoben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn der Wert für C nicht zwischen 0000 und 000F hex liegt. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.

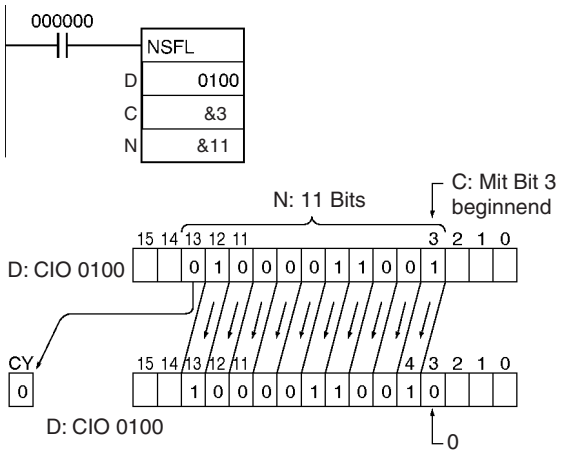
Vorsichtsmaßnahmen

Wenn die Verschiebedatenlänge (N) gleich 0 ist, wird der Inhalt des Anfangsbits in das Übertrags-Flag (CY) kopiert. Der Inhalt des Bits bleibt unverändert erhalten.

Es werden nur die Bits verändert, die in das niederwertigste Wort im Verschiebebereich verschoben werden (also die Daten des höchstwertigen Worts).

Beispiele

Wenn CIO 000000 EIN ist, werden die Bits ab dem Anfangsbit 3 bis zur Verschiebedatenlänge (B hex) um ein Bit nach links verschoben (von niederwertigsten zum höchstwertigen Bit). In Bit 3 von CIO 0100 wird eine 0 geschrieben. Der Inhalt des höchstwertigen Bits im Verschiebebereich (Bit 13 von CIO 0100) wird in das Übertrags-Flag (CY) kopiert.

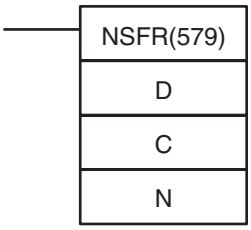


3-9-20 SHIFT N-BIT DATA RIGHT: NSFR(579)

Funktion

Verschiebt die spezifizierte Anzahl von Bits nach rechts.

Kontaktplansymbol



D: Anfangswort für das Verschieben

C: Anfangsbit

N: Verschiebedatenlänge

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	NSFR(579)
	Ausführung einmal bei steigender Flanke	@NSFR(579)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: 0000 bis 000F hex (0 bis 15)
N: 0000 bis FFFF hex (0 bis 65.535)

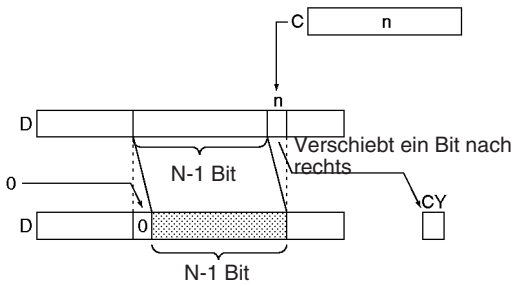
Hinweis Alle Worte im Schieberegister müssen im gleichen Bereich liegen.

Operanden-Spezifikationen

Bereich	D	C	N
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A448 bis A959	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0000 bis #000F (binär) oder &0 bis &15	#0000 bis #FFFF (binär) oder &0 bis &65535
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl NSFR(579) verschiebt die durch die Verschiebedatenlänge (N) spezifizierte Anzahl von Bits mit dem Anfangsbit (C) im durch D spezifizierten niederwertigsten Wort beginnend um ein Bit nach rechts (zum niederwertigsten Wort und niederwertigsten Bit). In das Anfangsbit wird eine 0 geschrieben und der Inhalt des niederwertigsten Bits im Verschiebebereich wird in das Übertrags-Flag (CY) verschoben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn der Wert für C nicht zwischen 0000 und 000F hex liegt. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.

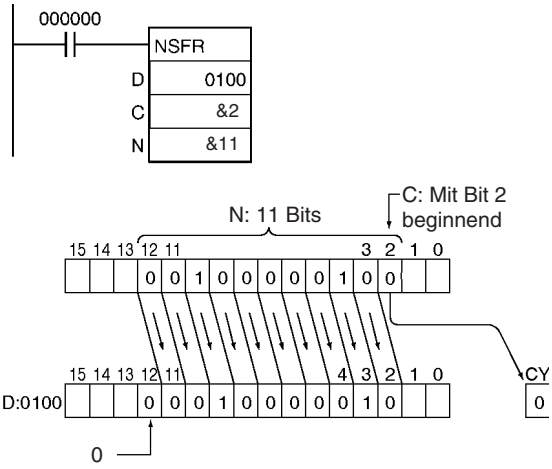
Vorsichtsmaßnahmen

Wenn die Verschiebedatenlänge (N) gleich 0 ist, wird der Inhalt des Anfangsbits in das Übertrags-Flag (CY) kopiert. Der Inhalt des Bits bleibt unverändert erhalten.

Es werden nur die Bits verändert, die in das niederwertigste Wort im Verschiebebereich verschoben werden (also die Daten des höchstwertigen Worts).

Beispiele

Wenn CIO 000000 EIN ist, werden die 11 Bits ab dem Anfangsbit 2 bis zum Ende der Verschiebedatenlänge (B hex) um ein Bit nach rechts verschoben (von höchstwertigen zum niederwertigsten Bit). In Bit 12 von CIO 0100 wird eine 0 verschoben. Der Inhalt des niederwertigsten Bits im Verschiebebereich (Bit 2 von CIO 0100) wird in das Übertrags-Flag (CY) kopiert.

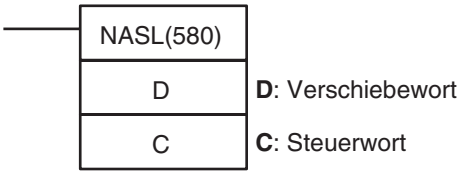


3-9-21 SHIFT N-BITS LEFT: NASL(580)

Funktion

Verschiebt die spezifizierten 16 Bits Wortdaten um die spezifizierte Anzahl von Bits nach links.

Kontaktplansymbol



Variationen

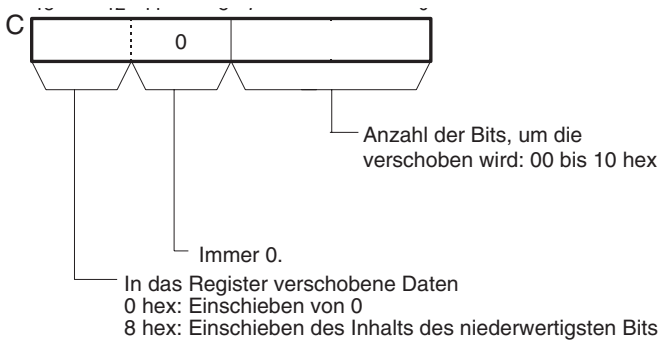
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	NASL(580)
	Ausführung einmal bei steigender Flanke	@NASL(580)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

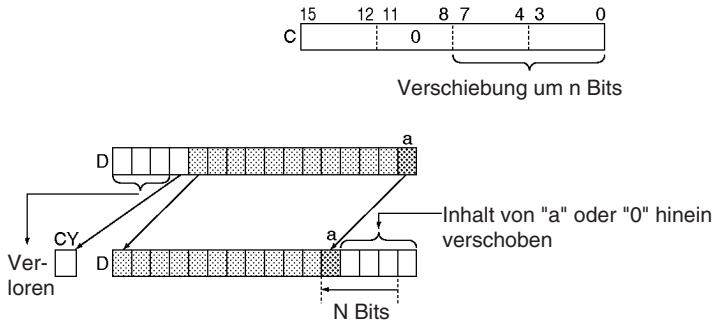


Operanden-Spezifikationen

Bereich	D	C
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A448 bis A959	A000 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	Nur spezifizierte Werte
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl NASL(580) verschiebt das Verschiebewort D um die in C spezifizierte Anzahl von Binärbits nach links (vom niederwertigsten zum höchstwertigen Bit). In die spezifizierte Anzahl von Bits des Verschiebeworts werden mit dem niederwertigsten Bit beginnend entweder Nullen oder der Wert des niederwertigsten Bits geschrieben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn das Steuerwort C (die Anzahl der Bits, um die verschoben werden soll) außerhalb des gültigen Bereichs liegt. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Von allen Bits, die aus den spezifizierten Wort heraus verschoben werden, wird nur der Inhalt des letzten Bits in das Übertrags-Flag (CY) verschoben. Die anderen Daten gehen verloren.

Wenn die in C spezifizierte Anzahl der Bits, um die verschoben werden soll, gleich 0 ist, werden die Daten nicht verschoben. Dennoch werden die entsprechenden Flags gemäß der Daten im spezifizierten Wort auf EIN bzw. AUS gesetzt.

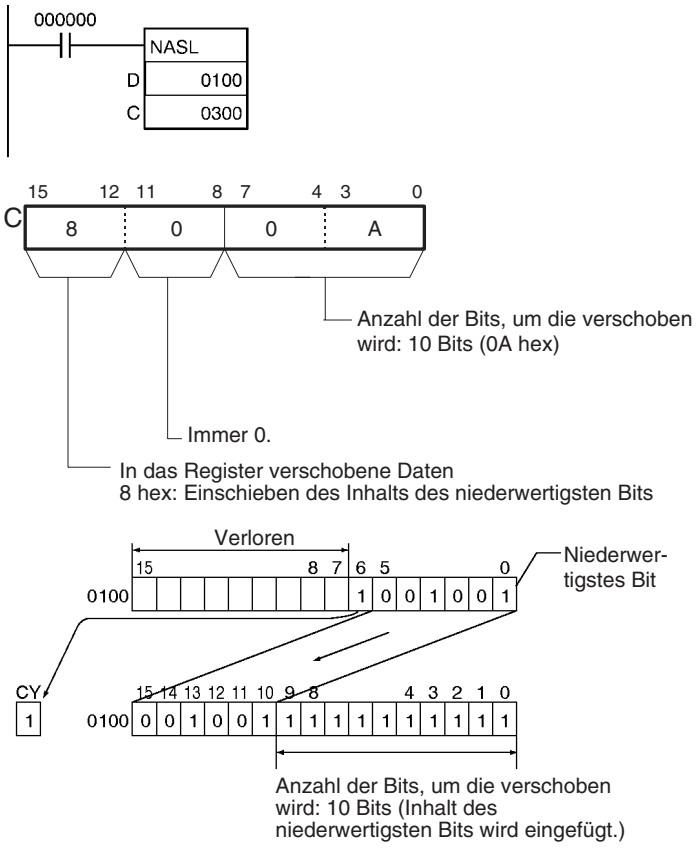
Wenn der Inhalt des Steuerworts C außerhalb des gültigen Bereichs liegt, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von D als Ergebnis der Verschiebung den Wert 0000 hex hat, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von D als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, wird der Inhalt von CIO 0100 um 10 Bits nach links (vom niederwertigsten zum höchstwertigen Bit) verschoben. Die Anzahl von Bits, um die die Verschiebung erfolgen soll, ist in den Bits 0 bis 7 des Worts CIO 0300 (Steuerdaten) spezifiziert. In die Bits, aus denen die Daten verschoben wurden, wird der Inhalt von Bit 0 von CIO 0100 kopiert. Der Inhalt des niederwertigsten Bits, das aus dem Verschiebebereich heraus verschoben wurde, wird in das Übertrags-Flag verschoben. Alle übrigen Daten gehen verloren.

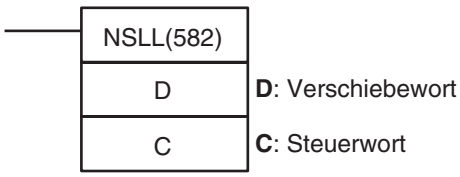


3-9-22 DOUBLE SHIFT N-BITS LEFT: NSLL(582)

Funktion

Verschiebt die spezifizierten 32 Bits Wortdaten um die spezifizierte Anzahl von Bits nach links.

Kontaktplansymbol



Variationen

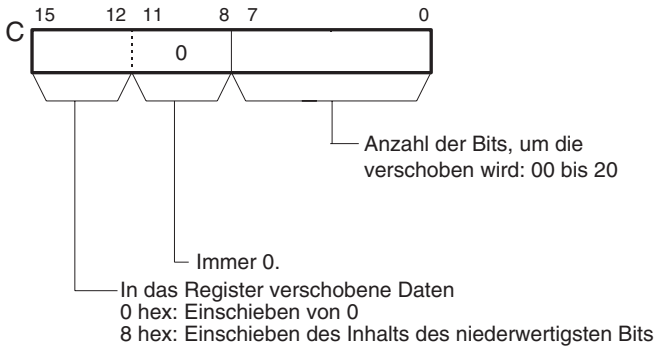
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	NSLL(582)
	Ausführung einmal bei steigender Flanke	@NSLL(582)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

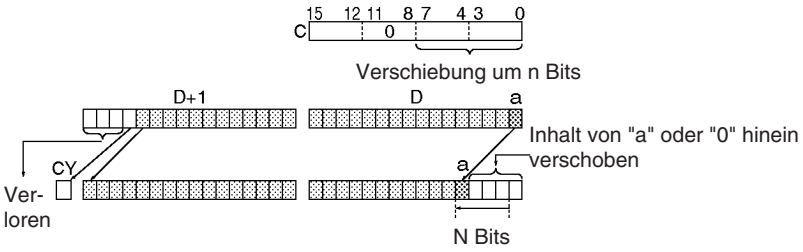


Operanden-Spezifikationen

Bereich	D	C
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W510	W000 bis W511
Arbeitsbereich remanent	H000 bis H510	H000 bis H511
Systembereich	A448 bis A958	A000 bis A959
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	Nur spezifizierte Werte
Datenregister	---	DR0 bis DR15
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl NALL(582) verschiebt die Verschiebeworte D und D+1 um die in C spezifizierte Anzahl von Binärbits nach links (vom niederwertigsten zum höchstwertigen Bit). In die spezifizierte Anzahl von Bits des Verschiebeworts werden mit dem niederwertigsten Bit beginnend entweder Nullen oder der Wert des niederwertigsten Bits geschrieben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn das Steuerwort C (die Anzahl der Bits, um die verschoben werden soll) außerhalb des gültigen Bereichs liegt. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Von allen Bits, die aus den spezifizierten Wort heraus verschoben werden, wird nur der Inhalt des letzten Bits in das Übertrags-Flag (CY) verschoben. Die anderen Daten gehen verloren.

Wenn die in C spezifizierte Anzahl der Bits, um die verschoben werden soll, gleich 0 ist, werden die Daten nicht verschoben. Dennoch werden die entsprechenden Flags gemäß der Daten im spezifizierten Wort auf EIN bzw. AUS gesetzt.

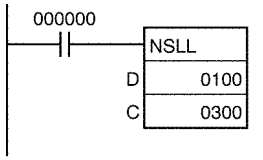
Wenn die Inhalte des Steuerworts C außerhalb der gültigen Bereiche liegen, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

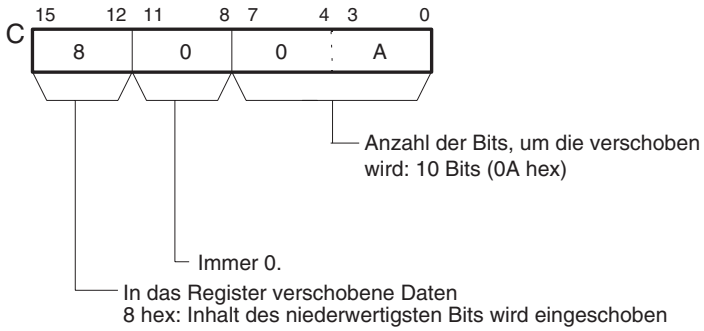
Wenn der Inhalt von D als Ergebnis der Verschiebung den Wert 0000 hat, wird das Gleich-Flag auf EIN gesetzt.

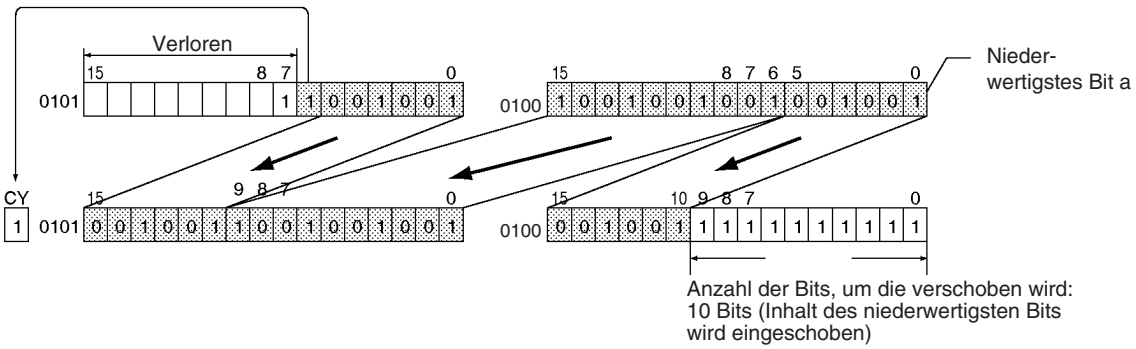
Wenn der Inhalt des höchstwertigen Bits von D und D+1 als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, werden die Worte CIO 0100 und CIO 0101 um 10 Bits nach links (vom niederwertigsten zum höchstwertigen Bit) verschoben. Die Anzahl von Bits, um die die Verschiebung erfolgen soll, ist in den Bits 0 bis 7 des Worts CIO 0300 (Steuerdaten) spezifiziert. In die Bits, aus denen die Daten verschoben wurden, wird der Inhalt von Bit 0 von CIO 0100 kopiert. Der Inhalt des niederwertigsten Bits, das aus dem Verschiebebereich heraus verschoben wurde, wird in das Übertrags-Flag verschoben. Alle übrigen Daten gehen verloren.





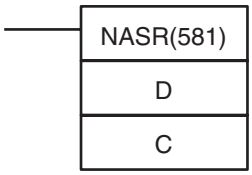


3-9-23 SHIFT N-BITS RIGHT: NASR(581)

Funktion

Verschiebt die spezifizierten 16 Bits Wortdaten um die spezifizierte Anzahl von Bits nach rechts.

Kontaktplansymbol



D: Verschiebewort

C: Steuerwort

Variationen

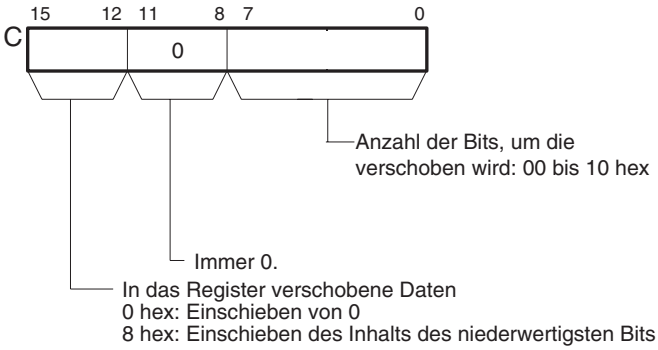
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	NASR(581)
	Ausführung einmal bei steigender Flanke	@NASR(581)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort



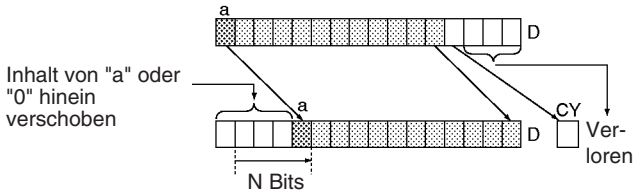
Operanden-Spezifikationen

Bereich	D	C
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A448 bis A959	A000 bis A447 A448 bis A959
Zeitgeberbereich	T0000 bis T4095	

Bereich	D	C
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	Nur spezifizierte Werte
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl NASR(581) verschiebt das Verschiebewort D um die in C spezifizierte Anzahl von Binärbits nach rechts (vom höchstwertigen zum niederwertigsten Bit). In die spezifizierte Anzahl von Bits des Verschiebeworts werden mit dem niederwertigsten Bit beginnend entweder Nullen oder der Wert des niederwertigsten Bits geschrieben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn das Steuerwort C (die Anzahl der Bits, um die verschoben werden soll) außerhalb des gültigen Bereichs liegt. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Von allen Bits, die aus den spezifizierten Wort heraus verschoben werden, wird nur der Inhalt des letzten Bits in das Übertrags-Flag (CY) verschoben. Die anderen Daten gehen verloren.

Wenn die in C spezifizierte Anzahl der Bits, um die verschoben werden soll, gleich 0 ist, werden die Daten nicht verschoben. Dennoch werden die entsprechenden Flags gemäß der Daten im spezifizierten Wort auf EIN bzw. AUS gesetzt.

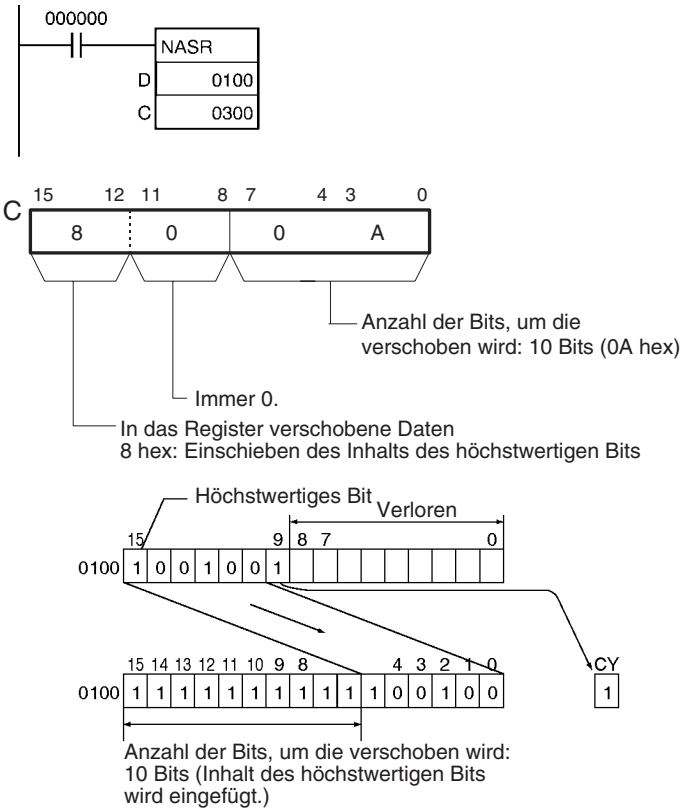
Wenn die Inhalte des Steuerworts C außerhalb der gültigen Bereiche liegen, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von D als Ergebnis der Verschiebung den Wert 0000 hex hat, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von D als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, wird CIO 0100 um 10 Bits nach rechts (vom höchstwertigen zum niederwertigsten Bit) verschoben. Die Anzahl von Bits, um die die Verschiebung erfolgen soll, ist in den Bits 0 bis 7 des Worts CIO 0300 spezifiziert. In die Bits, aus denen die Daten verschoben wurden, wird der Inhalt von Bit 15 von CIO 0100 kopiert. Der Inhalt des höchstwertigen Datenbits, das aus dem Verschiebebereich heraus verschoben wurde, wird in das Übertrags-Flag verschoben. Alle übrigen Daten gehen verloren.

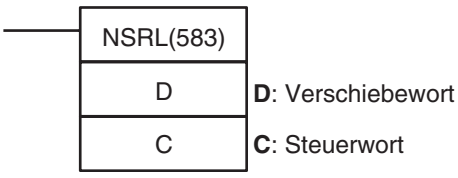


3-9-24 DOUBLE SHIFT N-BITS RIGHT: NSRL(583)

Funktion

Verschiebt die spezifizierten 32 Bits Wortdaten um die spezifizierte Anzahl von Bits nach rechts.

Kontaktplansymbol



Variationen

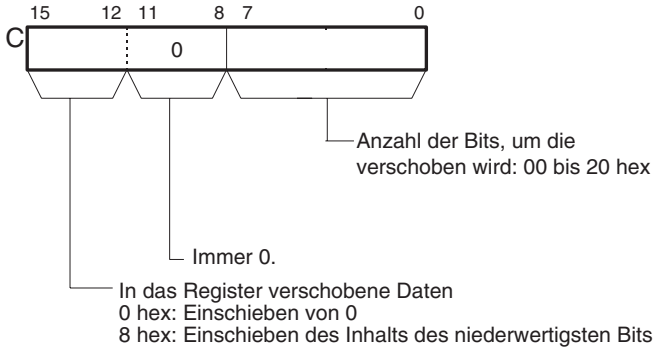
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	NSRL(583)
	Ausführung einmal bei steigender Flanke	@NSRL(583)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

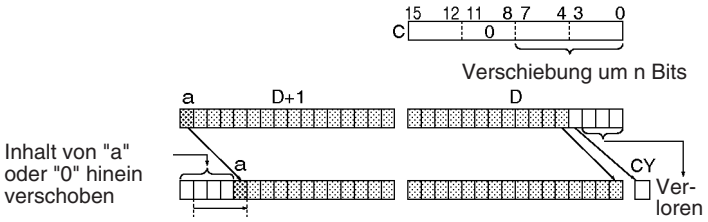


Operanden-Spezifikationen

Bereich	D	C
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W510	W000 bis W511
Arbeitsbereich remanent	H000 bis H510	H000 bis H511
Systembereich	A448 bis A958	A000 bis A959
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	Nur spezifizierte Werte
Datenregister	---	DR0 bis DR15
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl NSRL(583) verschiebt die Verschiebeworte D und D+1 um die in C spezifizierte Anzahl von Binärbits nach rechts (vom höchstwertigen zum niederwertigsten Bit). In die spezifizierte Anzahl von Bits des Verschiebeworts werden mit dem niederwertigsten Bit beginnend entweder Nullen oder der Wert des niederwertigsten Bits geschrieben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das Steuerwort C (die Anzahl der Bits, um die verschoben werden soll) außerhalb des gültigen Bereichs liegt. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis der Verschiebung 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine 1 in das Übertrags-Flag (CY) verschoben wird. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit als Ergebnis der Verschiebung den Wert 1 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Von allen Bits, die aus den spezifizierten Wort heraus verschoben werden, wird nur der Inhalt des letzten Bits in das Übertrags-Flag (CY) verschoben. Die anderen Daten gehen verloren.

Wenn die in C spezifizierte Anzahl der Bits, um die verschoben werden soll, gleich 0 ist, werden die Daten nicht verschoben. Dennoch werden die entsprechenden Flags gemäß der Daten im spezifizierten Wort auf EIN bzw. AUS gesetzt.

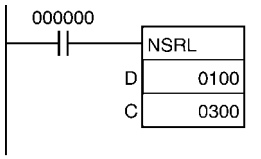
Wenn die Inhalte des Steuerworts C außerhalb der gültigen Bereiche liegen, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

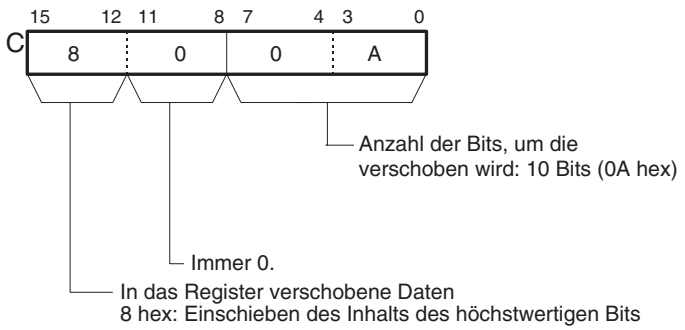
Wenn der Inhalt von D+1 als Ergebnis der Verschiebung den Wert 00000000 hex hat, wird das Gleich-Flag auf EIN gesetzt.

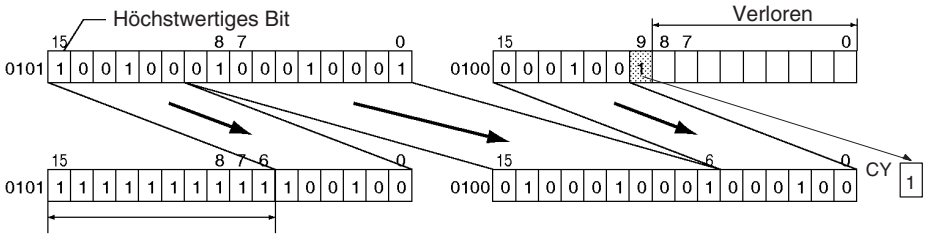
Wenn der Inhalt des höchstwertigen Bits von D+1 als Ergebnis der Verschiebung 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 EIN ist, werden CIO 0100 und CIO 0101 um 10 Bits nach rechts (vom höchstwertigen zum niederwertigsten Bit) verschoben. Die Anzahl von Bits, um die die Verschiebung erfolgen soll, ist in den Bits 0 bis 7 des Worts CIO 0300 (Steuerdaten) spezifiziert. In die Bits, aus denen die Daten verschoben wurden, wird der Inhalt von Bit 15 von CIO 0101 kopiert. Der Inhalt des höchstwertigen Datenbits, das aus dem Verschiebebereich heraus verschoben wurde, wird in das Übertrags-Flag verschoben. Alle übrigen Daten gehen verloren.







Anzahl der Bits, um die verschoben wird: 10 Bits (Inhalt des höchstwertigen Bits wird eingefügt.)

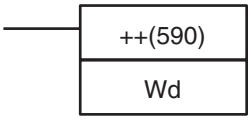
3-10 Inkrementier-/Dekrementierbefehle

3-10-1 INCREMENT BINARY: ++(590)

Funktion

Inkrementiert den 4-stelligen Hexadezimal-Inhalt des spezifizierten Wortes um 1.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	++(590)
	Ausführung einmal bei steigender Flanke	@++(590)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

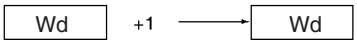
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ++(590) addiert den Wert 1 zum Binärinhalt des Wortes Wd. Das spezifizierte Wort wird in jedem Zyklus um 1 inkrementiert, so lange die Ausführungsbedingung von ++(590) auf EIN gesetzt ist. Wenn die Variante mit Ausführung bei steigender Flanke dieses Befehls (@++(590)) verwendet wird, wird das spezifizierte Wort nur in einem Zyklus inkrementiert, nachdem die Ausführungsbedingung von AUS zu EIN gewechselt ist.



Das Gleich-Flag wird auf EIN gesetzt, wenn das Ergebnis 0000 lautet. Das Übertrags-Flag wird auf EIN gesetzt, wenn eine Stelle von F zu 0 wechselt. Das Negativ-Flag wird auf EIN gesetzt, wenn Bit 15 von Wd im Ergebnis den Wert EIN hat.

Das Gleich-Flag und das Übertrags-Flag werden auf EIN gesetzt, wenn sich der Inhalt von Wd von FFFF zu 0000 ändert.

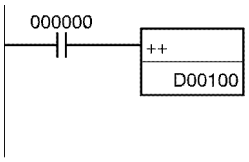
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn der Inhalt von Wd nach der Ausführung den Wert 0000 hat. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine Stelle in Wd während der Ausführung von F zu 0 wechselt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn Bit 15 von Wd nach der Ausführung auf EIN gesetzt ist. In allen anderen Fällen AUS.

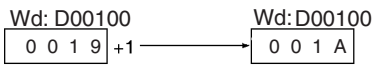
Beispiele

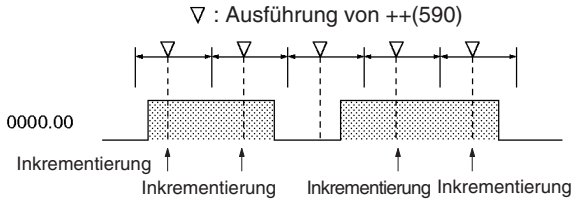
Funktion von ++(590)

Im folgenden Beispiel wird der Inhalt von D00100 in jedem Zyklus um 1 inkrementiert, so lange CIO 000000 auf EIN gesetzt ist.



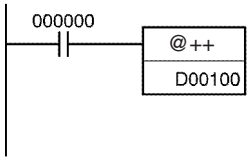
Inkrementierung erfolgt in jedem Zyklus, solange CIO 000000 auf EIN gesetzt ist.





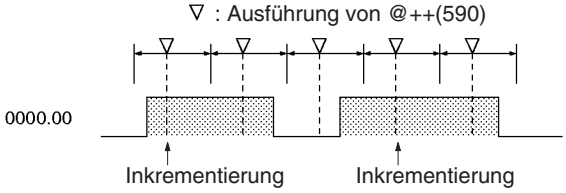
Funktion von @++(590)

Im folgenden Beispiel wird die Variante zur Ausführung bei steigender Flanke verwendet, wobei der Inhalt von D00100 nur in einem Zyklus um 1 inkrementiert wird, nachdem CIO 000000 von AUS zu EIN gewechselt ist.



Inkrementierung erfolgt nur bei steigender Flanke.





3-10-2 DOUBLE INCREMENT BINARY: ++L(591)

Funktion

Inkrementiert den 8-stelligen Hexadezimal-Inhalt der spezifizierten Worte um 1.

Kontaktplansymbol



Wd: Erstes Wort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	++L(591)
	Ausführung einmal bei steigender Flanke	@++L(591)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

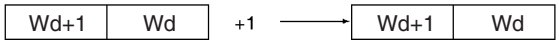
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	IR0 bis IR15
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ++L(591) addiert den Wert 1 zum 8-stelligen Hexadezimal-Inhalt von Wd+1 und Wd. Der Inhalt der spezifizierten Worte wird in jedem Zyklus um 1 inkrementiert, so lange die Ausführungsbedingung von ++L(591) auf EIN gesetzt ist. Wenn die Variante zur Ausführung bei steigender Flanke dieses Befehls (@++L(591)) verwendet wird, wird der Inhalt der spezifizierten Worte nur in einem Zyklus inkrementiert, nachdem die Ausführungsbedingung von AUS zu EIN gewechselt ist.



Das Gleich-Flag wird auf EIN gesetzt, wenn das Ergebnis 0000 0000 lautet. Das Übertrags-Flag wird auf EIN gesetzt, wenn eine Stelle von F zu 0 wechselt. Das Negativ-Flag wird auf EIN gesetzt, wenn Bit 15 von Wd+1 im Ergebnis den Wert EIN hat.

Das Gleich-Flag und das Übertrags-Flag werden auf EIN gesetzt, wenn sich der Inhalt von FFFF FFFF zu 0000 0000 ändert.

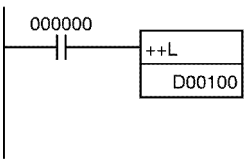
Flags

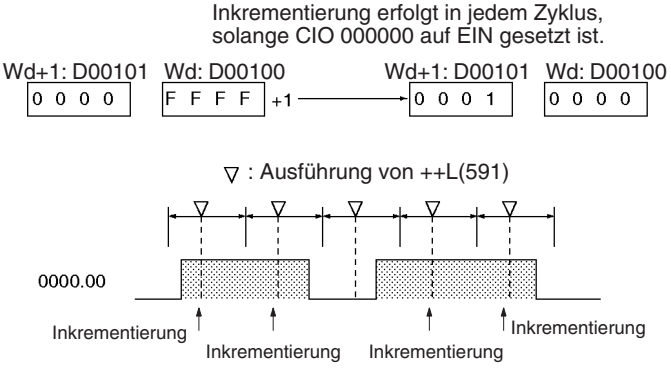
Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis nach der Ausführung 0000 0000 lautet. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine Stelle in Wd+1 oder Wd während der Ausführung von F zu 0 wechselt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn Bit 15 von Wd+1 nach der Ausführung auf EIN gesetzt ist. In allen anderen Fällen AUS.

Beispiele

Funktion von ++L(591)

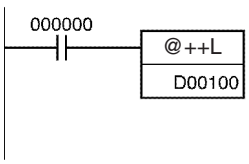
Im folgenden Beispiel wird der 8-stellige Hexadezimal-Inhalt von D00101 und D00100 in jedem Zyklus um 1 inkrementiert, so lange CIO 000000 auf EIN gesetzt ist.

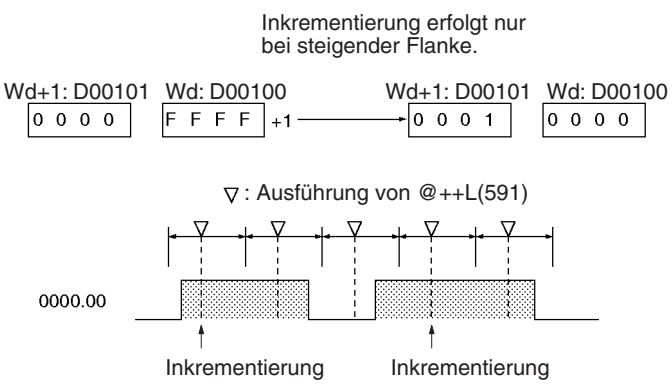




Funktion von @++L(591)

Im folgenden Beispiel wird die Variante zur Ausführung bei steigender Flanke verwendet, wobei der Inhalt von D00101 und D00100 nur in einem Zyklus um 1 inkrementiert wird, nachdem CIO 000000 von AUS zu EIN gewechselt ist.



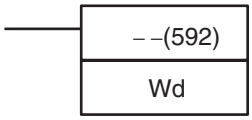


3-10-3 DECREMENT BINARY: – –(592)

Funktion

Dekrementiert den 4-stelligen Hexadezimal-Inhalt des spezifizierten Wortes um 1.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	– – (592)
	Ausführung einmal bei steigender Flanke	@– – (592)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl – –(592) subtrahiert 1 vom Binärinhalt von Wd. Das spezifizierte Wort wird in jedem Zyklus um 1 dekrementiert, so lange die Ausführungsbedingung von – –(592) auf EIN gesetzt ist. Wenn die Variante zur Ausführung bei steigender Flanke dieses Befehls (@– –(592)) verwendet wird, wird das spezifizierte Wort nur in einem Zyklus dekrementiert, nachdem die Ausführungsbedingung von AUS zu EIN gewechselt ist.



Das Gleich-Flag wird auf EIN gesetzt, wenn das Ergebnis 0000 lautet. Das Übertrags-Flag wird auf EIN gesetzt, wenn eine Stelle von F zu 0 wechselt. Das Negativ-Flag wird auf EIN gesetzt, wenn Bit 15 von Wd im Ergebnis den Wert EIN hat.

Das Übertrags-Flag und das Negativ-Flag werden auf EIN gesetzt, wenn sich der Inhalt von Wd von 0000 zu FFFF ändert.

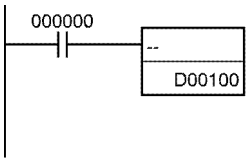
Flags

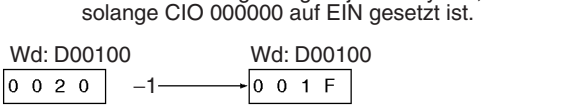
Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn der Inhalt von Wd nach der Ausführung den Wert 0000 hat. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine Stelle in Wd während der Ausführung von 0 zu F wechselt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn Bit 15 von Wd nach der Ausführung auf EIN gesetzt ist. In allen anderen Fällen AUS.

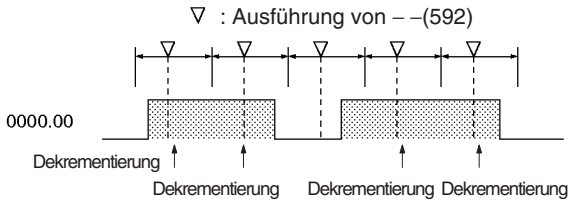
Beispiele

Funktion von – –(592)

Im folgenden Beispiel wird der Inhalt von D00100 in jedem Zyklus um 1 dekrementiert, so lange CIO 000000 auf EIN gesetzt ist.

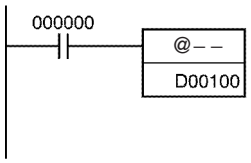


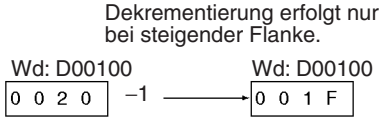


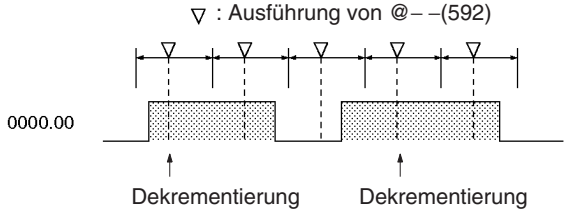


Funktion von @– –(592)

Im folgenden Beispiel wird die Variante zur Ausführung bei steigender Flanke verwendet, wobei der Inhalt von D00100 nur in einem Zyklus um 1 dekrementiert wird, nachdem CIO 000000 von AUS zu EIN gewechselt ist.





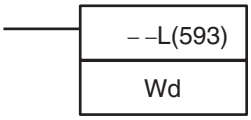


3-10-4 DOUBLE DECREMENT BINARY: – –L(593)

Funktion

Dekrementiert den 8-stelligen Hexadezimal-Inhalt der spezifizierten Worte um 1.

Kontaktplansymbol



Wd: Erstes Wort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	– –L(593)
	Ausführung einmal bei steigender Flanke	@– –L(593)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

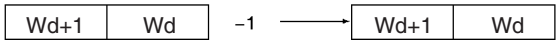
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	IR0 bis IR15
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl – –L(593) subtrahiert 1 vom 8-stelligen Hexadezimal-Inhalt von Wd+1 und Wd. Der Inhalt der spezifizierten Worte wird in jedem Zyklus um 1 dekrementiert, so lange die Ausführungsbedingung von – –L(593) auf EIN gesetzt ist. Wenn die Variante zur Ausführung bei steigender Flanke dieses Befehls (@– –L(593)) verwendet wird, wird der Inhalt der spezifizierten Worte nur in einem Zyklus dekrementiert, nachdem die Ausführungsbedingung von AUS zu EIN gewechselt ist.



Das Gleich-Flag wird auf EIN gesetzt, wenn das Ergebnis 0000 0000 lautet. Das Übertrags-Flag wird auf EIN gesetzt, wenn eine Stelle von 0 zu F wechselt. Das Negativ-Flag wird auf EIN gesetzt, wenn Bit 15 von Wd+1 im Ergebnis den Wert EIN hat.

Das Übertrags-Flag und das Negativ-Flag werden auf EIN gesetzt, wenn sich der Inhalt von 0000 0000 zu FFFF FFFF ändert.

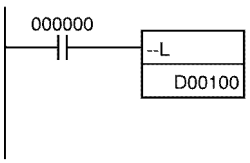
Flags

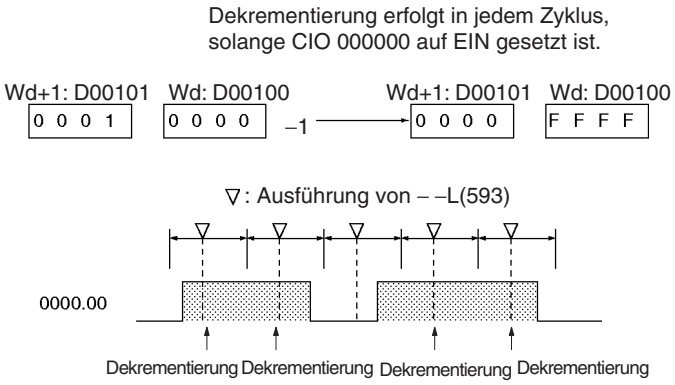
Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis nach der Ausführung 0000 0000 lautet. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine Stelle in Wd+1 oder Wd während der Ausführung von 0 zu F wechselt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn Bit 15 von Wd+1 nach der Ausführung auf EIN gesetzt ist. In allen anderen Fällen AUS.

Beispiele

Funktion von – –L(593)

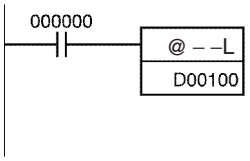
Im folgenden Beispiel wird der 8-stellige Hexadezimal-Inhalt von D00101 und D00100 in jedem Zyklus um 1 dekrementiert, so lange CIO 000000 auf EIN gesetzt ist.

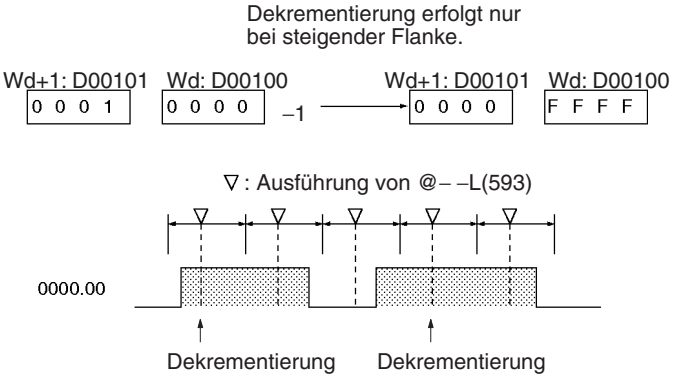




Funktion von @– –L(593)

Im folgenden Beispiel wird die Variante zur Ausführung bei steigender Flanke verwendet, wobei der Inhalt von D00101 und D00100 nur in einem Zyklus um 1 dekrementiert wird, nachdem CIO 000000 von AUS zu EIN gewechselt ist.



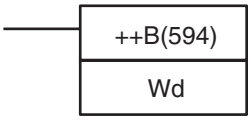


3-10-5 INCREMENT BCD: ++B(594)

Funktion

Inkrementiert den 4-stelligen BCD-Inhalt des spezifizierten Wortes um 1.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	++B(594)
	Ausführung einmal bei steigender Flanke	@++B(594)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

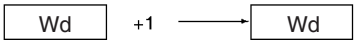
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ++B(594) addiert den Wert 1 zum Binärinhalt des Wortes Wd. Das spezifizierte Wort wird in jedem Zyklus um 1 inkrementiert, so lange die Ausführungsbedingung von ++B(594) auf EIN gesetzt ist. Wenn die Variante zur Ausführung bei steigender Flanke dieses Befehls (@++B(594)) verwendet wird, wird das spezifizierte Wort nur in einem Zyklus inkrementiert, nachdem die Ausführungsbedingung von AUS zu EIN gewechselt ist.



Das Gleich-Flag wird auf EIN gesetzt, wenn das Ergebnis den Wert 0000 hat. Das Übertrags-Flag wird auf EIN gesetzt, wenn eine Stelle von 9 zu 0 wechselt.

Das Gleich-Flag und das Übertrags-Flag werden auf EIN gesetzt, wenn sich der Inhalt von Wd von 9999 zu 0000 ändert.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von Wd kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn der Inhalt von Wd nach der Ausführung den Wert 0000 hat. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine Stelle in Wd während der Ausführung von 9 zu 0 wechselt. In allen anderen Fällen AUS.

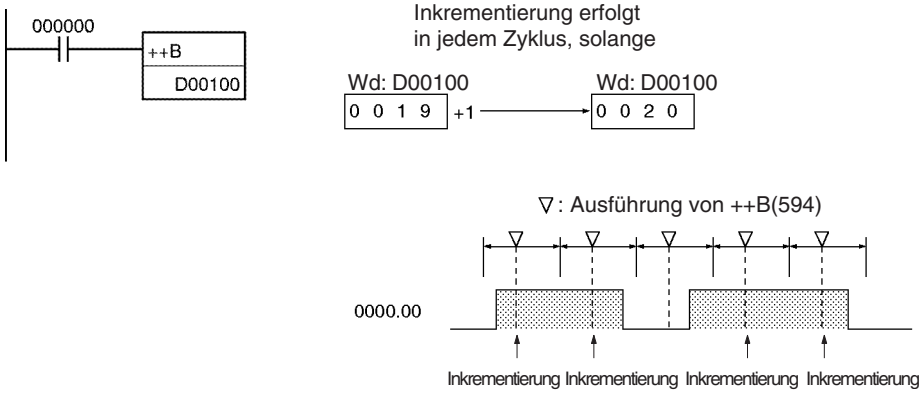
Vorsichtsmaßnahmen

Der Inhalt von Wd muss ein BCD-Wert sein. Wenn der Inhalt kein BCD-Wert ist, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Beispiele

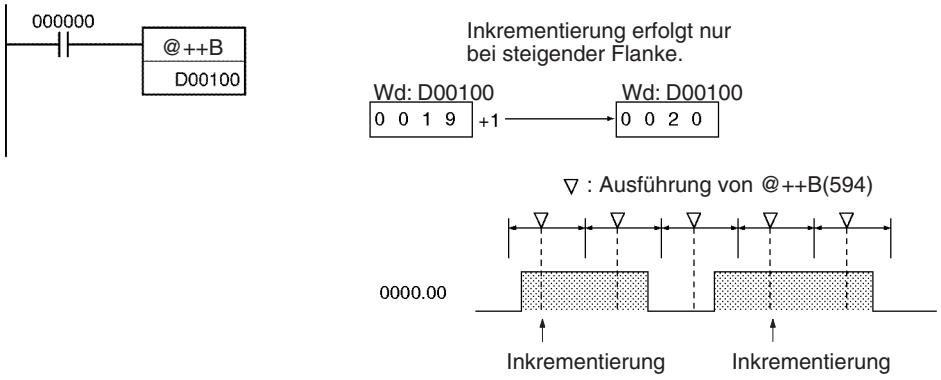
Funktion von ++B(594)

Im folgenden Beispiel wird der BCD-Inhalt von D00100 in jedem Zyklus um 1 inkrementiert, so lange CIO 000000 auf EIN gesetzt ist.



Funktion von @++B(594)

Im folgenden Beispiel wird die Variante zur Ausführung bei steigender Flanke verwendet, wobei der Inhalt von D00100 nur in einem Zyklus um 1 inkrementiert wird, nachdem CIO 000000 von AUS zu EIN gewechselt ist.



3-10-6 DOUBLE INCREMENT BCD: ++BL(595)

Funktion

Inkrementiert den 8-stelligen BCD-Inhalt der spezifizierten Worte um 1.

Kontaktplansymbol



Wd: Erstes Wort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	++BL(595)
	Ausführung einmal bei steigender Flanke	@++BL(595)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

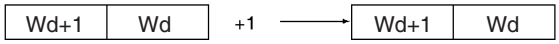
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl ++BL(595) addiert den Wert 1 zum 8-stelligen BCD-Inhalt von Wd+1 und Wd. Der Inhalt der spezifizierten Worte wird in jedem Zyklus um 1 inkrementiert, so lange die Ausführungsbedingung von ++BL(595) auf EIN gesetzt ist. Wenn die Variante zur Ausführung bei steigender Flanke dieses Befehls (@++BL(595)) verwendet wird, wird der Inhalt der spezifizierten Worte nur in einem Zyklus inkrementiert, nachdem die Ausführungsbedingung von AUS zu EIN gewechselt ist.



Das Gleich-Flag wird auf EIN gesetzt, wenn das Ergebnis den Wert 0000 0000 hat. Das Übertrags-Flag wird auf EIN gesetzt, wenn eine Stelle von 9 zu 0 wechselt.

Das Gleich-Flag und das Übertrags-Flag werden auf EIN gesetzt, wenn sich der Inhalt von 9999 9999 zu 0000 0000 ändert.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von Wd+1 und Wd kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis nach der Ausführung 0000 0000 lautet. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine Stelle in Wd+1 oder Wd während der Ausführung von 9 zu 0 wechselt. In allen anderen Fällen AUS.

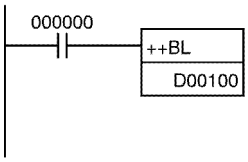
Vorsichtsmaßnahmen

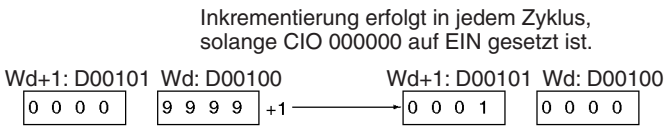
Der Inhalt von Wd+1 und Wd muss ein BCD-Wert sein. Wenn der Inhalt kein BCD-Wert ist, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

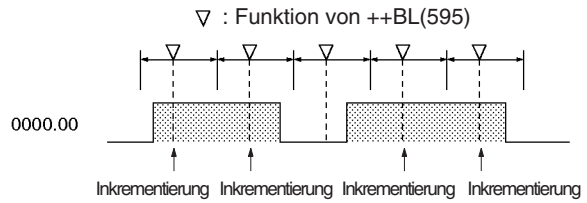
Beispiele

Funktion von ++BL(595)

Im folgenden Beispiel wird der 8-stellige BCD-Inhalt von D00101 und D00100 in jedem Zyklus um 1 inkrementiert, so lange CIO 000000 auf EIN gesetzt ist.

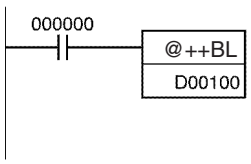


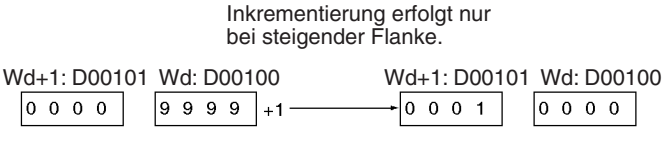


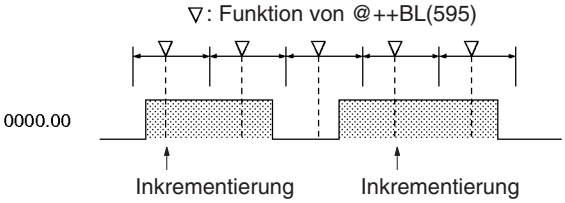


Funktion von @++BL(595)

Im folgenden Beispiel wird die Variante zur Ausführung bei steigender Flanke verwendet, wobei der BCD-Inhalt von D00101 und D00100 nur in einem Zyklus um 1 inkrementiert wird, nachdem CIO 000000 von AUS zu EIN gewechselt ist.





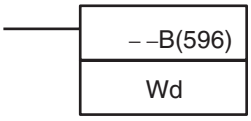


3-10-7 DECREMENT BCD: – –B(596)

Funktion

Dekrementiert den 4-stelligen BCD-Inhalt des spezifizierten Worts um 1.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	– –B(596)
	Ausführung einmal bei steigender Flanke	@– –B(596)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

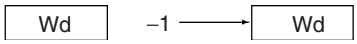
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl – –B(596) subtrahiert 1 vom BCD-Inhalt von Wd. Das spezifizierte Wort wird in jedem Zyklus um 1 dekrementiert, so lange die Ausführungsbedingung von – –B(596) auf EIN gesetzt ist. Wenn die Variante zur Ausführung bei steigender Flanke dieses Befehls (@– –B(596)) verwendet wird, wird das spezifizierte Wort nur in einem Zyklus dekrementiert, nachdem die Ausführungsbedingung von AUS zu EIN gewechselt ist.



Das Gleich-Flag wird auf EIN gesetzt, wenn das Ergebnis den Wert 000 hat. Das Übertrags-Flag wird auf EIN gesetzt, wenn eine Stelle von 0 zu 9 wechselt.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von Wd kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn der Inhalt von Wd nach der Ausführung den Wert 0000 hat. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine Stelle in Wd während der Ausführung von 0 zu 9 wechselt. In allen anderen Fällen AUS.

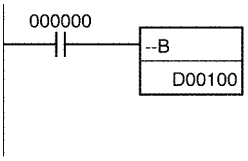
Vorsichtsmaßnahmen

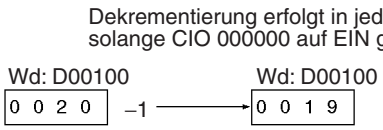
Der Inhalt von Wd muss ein BCD-Wert sein. Wenn der Inhalt kein BCD-Wert ist, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

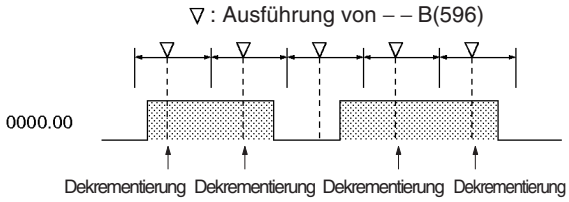
Beispiele

Funktion von – –B(596)

Im folgenden Beispiel wird der BCD-Inhalt von D00100 in jedem Zyklus um 1 dekrementiert, so lange CIO 000000 auf EIN gesetzt ist.

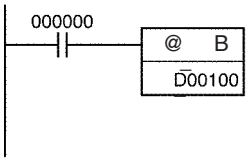


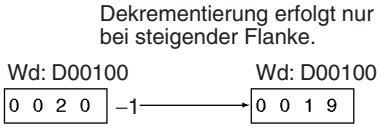


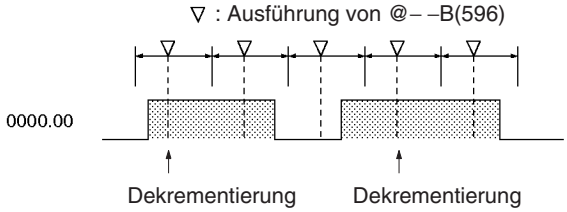


Funktion von @– –B(596)

Im folgenden Beispiel wird die Variante zur Ausführung bei steigender Flanke verwendet, wobei der BCD-Inhalt von D00100 nur in einem Zyklus um 1 dekrementiert wird, nachdem CIO 000000 von AUS zu EIN gewechselt ist.





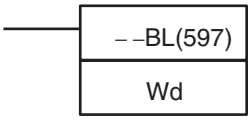


3-10-8 DOUBLE DECREMENT BCD: – –BL(597)

Funktion

Dekrementiert den 8-stelligen BCD-Inhalt der spezifizierten Worte um 1.

Kontaktplansymbol



Wd: Erstes Wort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	– –BL(597)
	Ausführung einmal bei steigender Flanke	@– –BL(597)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

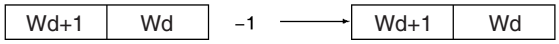
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510
Arbeitsbereich remanent	H000 bis H510
Systembereich	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl – –BL(597) subtrahiert 1 vom 8-stelligen BCD-Inhalt von Wd+1 und Wd. Der Inhalt der spezifizierten Worte wird in jedem Zyklus um 1 dekrementiert, so lange die Ausführungsbedingung von – –BL(597) auf EIN gesetzt ist. Wenn die Variante zur Ausführung bei steigender Flanke dieses Befehls (@– –BL(597)) verwendet wird, wird der Inhalt der spezifizierten Worte nur in einem Zyklus dekrementiert, nachdem die Ausführungsbedingung von AUS zu EIN gewechselt ist.



Das Gleich-Flag wird auf EIN gesetzt, wenn das Ergebnis den Wert 0000 0000 hat. Das Übertrags-Flag wird auf EIN gesetzt, wenn eine Stelle von 0 zu 9 wechselt.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von Wd+1 und Wd kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis nach der Ausführung 0000 0000 lautet. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn eine Stelle in Wd+1 oder Wd während der Ausführung von 0 zu 9 wechselt. In allen anderen Fällen AUS.

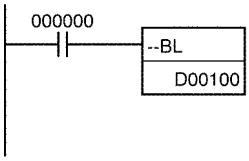
Vorsichtsmaßnahmen

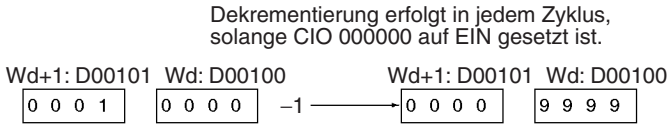
Der Inhalt von Wd+1 und Wd muss ein BCD-Wert sein. Wenn der Inhalt kein BCD-Wert ist, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

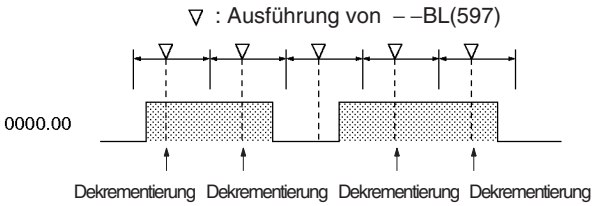
Beispiele

Funktion von – –BL(597)

Im folgenden Beispiel wird der 8-stellige BCD-Inhalt von D00101 und D00100 in jedem Zyklus um 1 dekrementiert, so lange CIO 000000 auf EIN gesetzt ist.

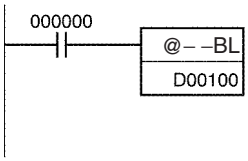


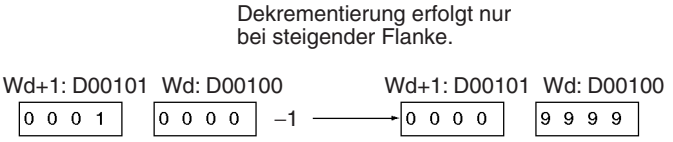


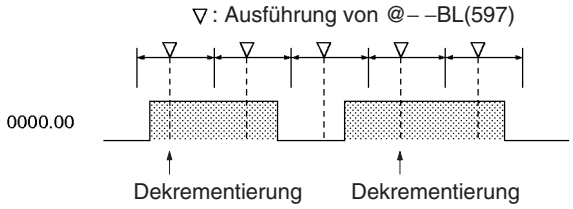


Funktion von @– –BL(597)

Im folgenden Beispiel wird die Variante zur Ausführung bei steigender Flanke verwendet, wobei der BCD-Inhalt von D00101 und D00100 nur in einem Zyklus um 1 dekrementiert wird, nachdem CIO 000000 von AUS zu EIN gewechselt ist.







3-11 Symbolarithmetische Befehle

In diesem Abschnitt werden die symbolarithmetischen Befehle beschrieben, die arithmetische Operationen mit BCD-Daten oder Binärdaten durchführen.

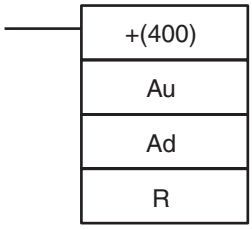
Befehl	AWL	Funktionscode	Seite
SIGNED BINARY ADD WITHOUT CARRY	+	400	410
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	401	412
SIGNED BINARY ADD WITH CARRY	+C	402	414
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	403	416
BCD ADD WITHOUT CARRY	+B	404	418
DOUBLE BCD ADD WITHOUT CARRY	+BL	405	419
BCD ADD WITH CARRY	+BC	406	421
DOUBLE BCD ADD WITH CARRY	+BCL	407	423
SIGNED BINARY SUBTRACT WITHOUT CARRY	–	410	424
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	–L	411	426
SIGNED BINARY SUBTRACT WITH CARRY	–C	412	430
DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	–CL	413	432
BCD SUBTRACT WITHOUT CARRY	–B	414	435
DOUBLE BCD SUBTRACT WITHOUT CARRY	–BL	415	436
BCD SUBTRACT WITH CARRY	–BC	416	440
DOUBLE BCD SUBTRACT WITH CARRY	–BCL	417	441
SIGNED BINARY MULTIPLY	*	420	443
DOUBLE SIGNED BINARY MULTIPLY	*L	421	445
UNSIGNED BINARY MULTIPLY	*U	422	447
DOUBLE UNSIGNED BINARY MULTIPLY	*UL	423	449
BCD MULTIPLY	*B	424	450
DOUBLE BCD MULTIPLY	*BL	425	452
SIGNED BINARY DIVIDE	/	430	454
DOUBLE SIGNED BINARY DIVIDE	/L	431	456
UNSIGNED BINARY DIVIDE	/U	432	458
DOUBLE UNSIGNED BINARY DIVIDE	/UL	433	460
BCD DIVIDE	/B	434	462
DOUBLE BCD DIVIDE	/BL	435	464

3-11-1 SIGNED BINARY ADD WITHOUT CARRY: +(400)

Funktion

Addiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten.

Kontaktplansymbol



Au: 1. Summand

Ad: 2. Summand

R: Ergebniswort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	+(400)
	Ausführung einmal bei steigender Flanke	@+(400)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

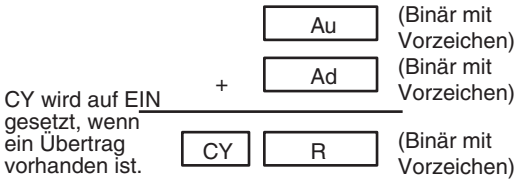
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Au	Ad	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

+(400) addiert die Binärwerte in Au und Ad und gibt das Ergebnis in R aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Addition zu einem Übertrag führt. In allen anderen Fällen AUS.
Überlauf-Flag	OF	EIN, wenn das Ergebnis der Addition von zwei positiven Zahlen im Bereich von 8000 bis FFFF hex liegt. In allen anderen Fällen AUS.
Unterlauf-Flag	UF	EIN, wenn das Ergebnis der Addition von zwei negativen Zahlen im Bereich von 0000 to 7FFF hex liegt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von +(400) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R als Ergebnis der Addition 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Addition zu einem Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

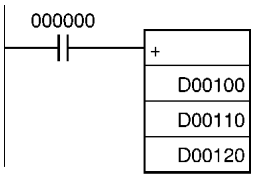
Wenn das Ergebnis der Addition von zwei positiven Zahlen negativ (im Bereich von 8000 bis FFFF hex) ist, wird das Überlauf-Flag auf EIN gesetzt.

Wenn das Ergebnis der Addition von zwei negativen Zahlen positiv (im Bereich von 0000 bis 7FFF hex) ist, wird das Unterlauf-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R als Ergebnis der Addition 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00100 und D00110 als 4-stellige Binärwerte mit Vorzeichen addiert und das Ergebnis wird in D00120 ausgegeben.

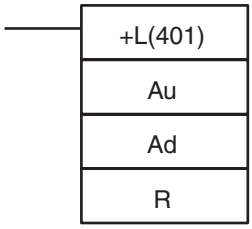


3-11-2 DOUBLE SIGNED BINARY ADD WITHOUT CARRY: +L(401)

Funktion

Addiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten.

Kontaktplansymbol



Au: 1. Summand, erstes Wort
Ad: 2. Summand, erstes Wort
R: Erstes Ergebniswort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	+L(401)
	Ausführung einmal bei steigender Flanke	@+L(401)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

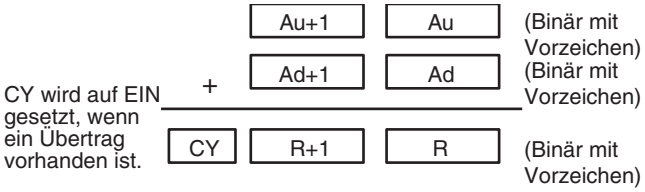
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Au	Ad	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich nicht remanent	W000 bis W510		
Arbeitsbereich remanent	H000 bis H510		
Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	IR0 bis IR15		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

+L(401) addiert die Binärwerte in Ad und Ad+1 zu Au und Au+1 und gibt das Ergebnis in R und R+1 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Addition zu einem Übertrag führt. In allen anderen Fällen AUS.
Überlauf-Flag	OF	EIN, wenn das Ergebnis der Addition von zwei positiven Zahlen im Bereich von 80000000 bis FFFFFFFF hex liegt. In allen anderen Fällen AUS.
Unterlauf-Flag	UF	EIN, wenn das Ergebnis der Addition von zwei negativen Zahlen im Bereich von 00000000 bis 7FFFFFFF hex liegt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von +L(401) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R, R+1 als Ergebnis der Addition 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Addition zu einem Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

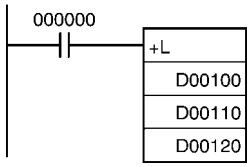
Wenn das Ergebnis der Addition von zwei positiven Zahlen negativ (im Bereich von 80000000 bis FFFFFFFF hex) ist, wird das Überlauf-Flag auf EIN gesetzt.

Wenn das Ergebnis der Addition von zwei negativen Zahlen positiv (im Bereich von 00000000 to 7FFFFFFF hex) ist, wird das Unterlauf-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+1 als Ergebnis der Addition 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 auf EIN gesetzt ist, werden die 8-stellige Binärwerte mit Vorzeichen in D00111 und D00110 zu denen in D00100 und D00110 addiert und das Ergebnis wird in D00120 und D00120 ausgegeben.

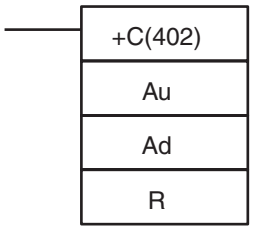


3-11-3 SIGNED BINARY ADD WITH CARRY: +C(402)

Funktion

Addiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten mit Übertrags-Flag (CY).

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	+C(402)
	Ausführung einmal bei steigender Flanke	@+C(402)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

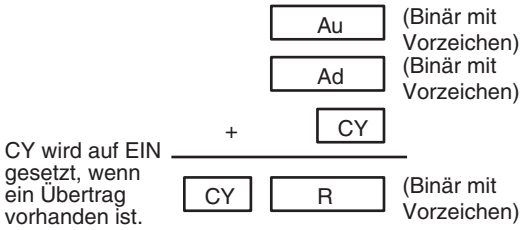
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Au	Ad	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

+C(402) addiert die Binärwerte in Au, Ad und CY und gibt das Ergebnis in R aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Additionsergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Addition zu einem Übertrag führt. In allen anderen Fällen AUS.
Überlauf-Flag	OF	EIN, wenn das Ergebnis der Addition von zwei positiven Zahlen und CY im Bereich von 8000 bis FFFF hex liegt. In allen anderen Fällen AUS.
Unterlauf-Flag	UF	EIN, wenn das Ergebnis der Addition von zwei negativen Zahlen und CY im Bereich von 0000 to 7FFF hex liegt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von +C(402) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R als Ergebnis der Addition 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Addition zu einem Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Wenn das Ergebnis der Addition von zwei positiven Zahlen und CY negativ (im Bereich von 8000 bis FFFF hex) ist, wird das Überlauf-Flag auf EIN gesetzt.

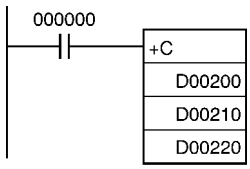
Wenn das Ergebnis der Addition von zwei negativen Zahlen und CY positiv (im Bereich von 0000 bis 7FFF hex) ist, wird das Unterlauf-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R als Ergebnis der Addition 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Hinweis Zum Löschen des Übertrags-Flags (CY) führen Sie den Befehl CLEAR CARRY (CLC(041)) aus.

Beispiele

Wenn CIO 000000 auf EIN gesetzt ist, werden die Inhalte von D00100, D00110 und CY als 4-stellige Binärdaten mit Vorzeichen addiert und das Ergebnis wird in D00220 ausgegeben.

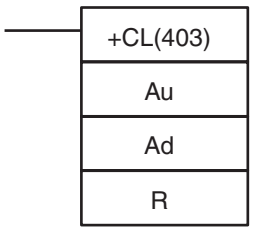


3-11-4 DOUBLE SIGNED BINARY ADD WITH CARRY: +CL(403)

Funktion

Addiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten mit Übertrags-Flag (CY).

Kontaktplansymbol



Au: 1. Summand, erstes Wort
Ad: 2. Summand, erstes Wort
R: Erstes Ergebniswort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	+CL(403)
	Ausführung einmal bei steigender Flanke	@+CL(403)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

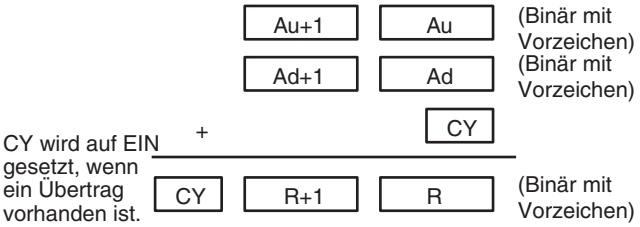
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Au	Ad	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich nicht remanent	W000 bis W510		
Arbeitsbereich remanent	H000 bis H510		
Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

+CL(403) addiert die Binärwerte in Au und Au+1, Ad und Ad+1 und CY und gibt das Ergebnis in R und R+1 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn das Ergebnis zu einem Übertrag führt. In allen anderen Fällen AUS.
Überlauf-Flag	OF	EIN, wenn das Ergebnis der Addition von zwei positiven Zahlen und CY im Bereich von 80000000 bis FFFFFFFF hex liegt. In allen anderen Fällen AUS.
Unterlauf-Flag	UF	EIN, wenn das Ergebnis der Addition von zwei negativen Zahlen und CY im Bereich von 00000000 bis 7FFFFFFF hex liegt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von +CL(403) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R, R+1 als Ergebnis der Addition 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Addition zu einem Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Wenn das Ergebnis der Addition von zwei positiven Zahlen und CY negativ (im Bereich von 80000000 bis FFFFFFFF hex) ist, wird das Überlauf-Flag auf EIN gesetzt.

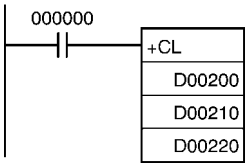
Wenn das Ergebnis der Addition von zwei negativen Zahlen und CY positiv (im Bereich von 00000000 bis 7FFFFFFF hex) ist, wird das Unterlauf-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+1 als Ergebnis der Addition 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Hinweis Zum Löschen des Übertrags-Flags (CY) führen Sie den Befehl CLEAR CARRY (CLC(041)) aus.

Beispiele

Wenn CIO 000000 auf EIN gesetzt ist, werden die Inhalte von D00201, D00200, D00211, D00210 und CY als 8-stellige Binärdaten mit Vorzeichen addiert und das Ergebnis wird in D00221 und D00220 ausgegeben.

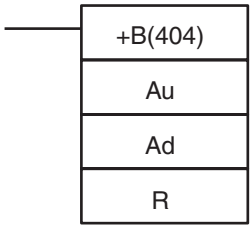


3-11-5 BCD ADD WITHOUT CARRY: +B(404)

Funktion

Addiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten.

Kontaktplansymbol



Au: 1. Summand

Ad: 2. Summand

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	+B(404)
	Ausführung einmal bei steigender Flanke	@+B(404)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

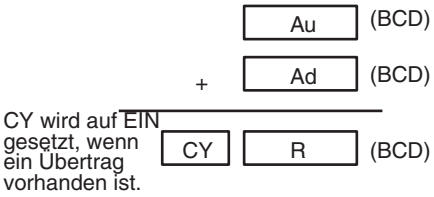
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Au	Ad	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	0000 bis 9999 (BCD)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

+B(404) addiert die BCD-Werte in Au und Ad und gibt das Ergebnis an R aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von Au kein BCD-Wert ist. EIN, wenn der Inhalt von Ad kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Addition zu einem Übertrag führt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

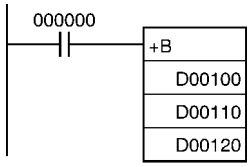
Wenn Au oder Ad keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von R als Ergebnis der Addition 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Addition zu einem Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00100 und D00110 als 4-stellige BCD-Werte addiert und das Ergebnis wird in D00120 ausgegeben.

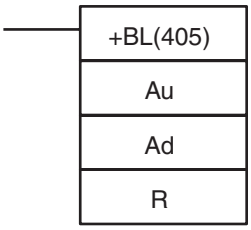


3-11-6 DOUBLE BCD ADD WITHOUT CARRY: +BL(405)

Funktion

Addiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten.

Kontaktplansymbol



Au: 1. Summand, erstes Wort
Ad: 2. Summand, erstes Wort
R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	+BL(405)
	Ausführung einmal bei steigender Flanke	@+BL(405)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

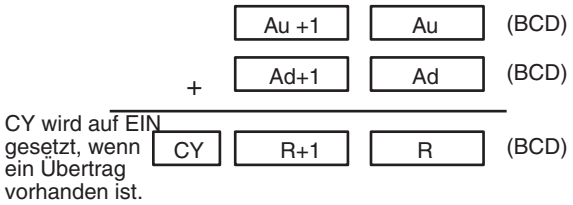
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Au	Ad	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich nicht remanent	W000 bis W510		
Arbeitsbereich remanent	H000 bis H510		
Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #99999999 (BCD)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

+BL(405) addiert die BCD-Werte in Ad und Ad+1 zu denen in Au und Au+1 und gibt das Ergebnis in R, R+1 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Au und Au +1 keine BCD-Werte sind. EIN, wenn Ad und Ad +1 keine BCD-Werte sind. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Addition zu einem Übertrag führt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

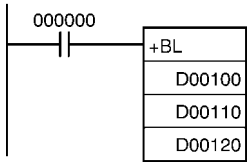
Wenn Au, Au +1 oder Ad, Ad +1 keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von R, R +1 als Ergebnis der Addition 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Addition zu einem Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden die Inhalte von D00101 und D00100 zu D00111 und D00110 als 8-stellige BCD-Werte addiert und das Ergebnis wird in D00121 und D00120 ausgegeben.

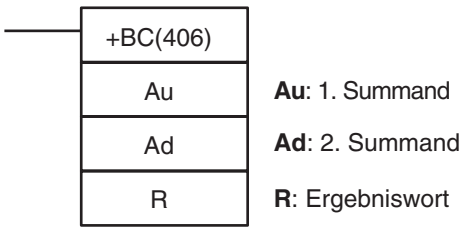


3-11-7 BCD ADD WITH CARRY: +BC(406)

Funktion

Addiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten mit Übertrags-Flag (CY).

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	+BC(406)
	Ausführung einmal bei steigender Flanke	@+BC(406)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

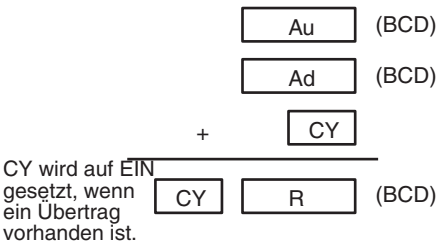
Operanden-Spezifikationen

Bereich	Au	Ad	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		

Bereich	Au	Ad	R
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 to 9999 (BCD)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

+BC(406) addiert die BCD-Werte in Au, Ad und CY und gibt das Ergebnis in R aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von Au kein BCD-Wert ist. EIN, wenn der Inhalt von Ad kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Addition zu einem Übertrag führt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn Au oder Ad keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

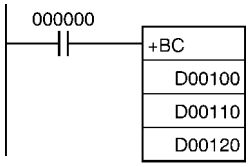
Wenn der Inhalt von R als Ergebnis der Addition 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Addition zu einem Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Hinweis Zum Löschen des Übertrags-Flags (CY) führen Sie den Befehl CLEAR CARRY (CLC(041)) aus.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden die Inhalte von D00100, D00110 und CY als 4-stellige BCD-Werte addiert und das Ergebnis wird in D00120 ausgegeben.

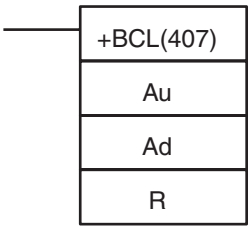


3-11-8 DOUBLE BCD ADD WITH CARRY: +BCL(407)

Funktion

Addiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten mit Übertrags-Flag (CY).

Kontaktplansymbol



Au: 1. Summand, erstes Wort
Ad: 2. Summand, erstes Wort
R: Erstes Ergebniswort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	+BCL(407)
	Ausführung einmal bei steigender Flanke	@+BCL(407)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

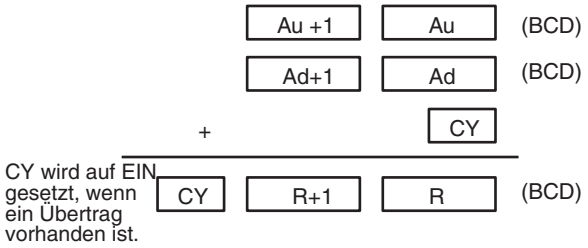
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Au	Ad	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich nicht remanent	W000 bis W510		
Arbeitsbereich remanent	H000 bis H510		
Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #99999999 (BCD)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

+BCL(407) addiert die BCD-Werte in Au und Au+1, Ad und Ad+1 und CY und gibt das Ergebnis in R, R+1 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Au und Au +1 keine BCD-Werte sind. EIN, wenn Ad und Ad +1 keine BCD-Werte sind. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Addition zu einem Übertrag führt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn Au, Au +1 oder Ad, Ad +1 keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

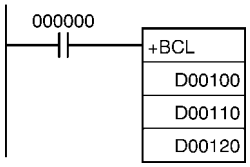
Wenn der Inhalt von R, R +1 als Ergebnis der Addition 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Addition zu einem Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Hinweis Zum Löschen des Übertrags-Flags (CY) führen Sie den Befehl CLEAR CARRY (CLC(041)) aus.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00101, D00100, D00111, D00110 und CY als 8-stellige BCD-Werte addiert und das Ergebnis wird in D00121 und D00120 ausgegeben.

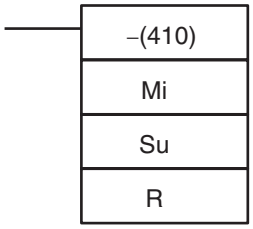


3-11-9 SIGNED BINARY SUBTRACT WITHOUT CARRY: –(410)

Funktion

Subtrahiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten.

Kontaktplansymbol



Mi: Minuend-Wort

Su: Subtrahend-Wort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	–(410)
	Ausführung einmal bei steigender Flanke	@–(410)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

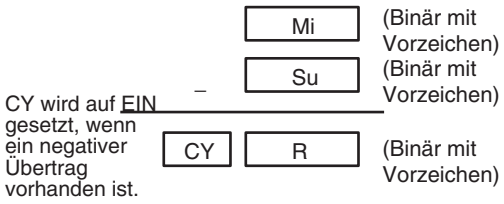
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Mi	Su	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D0000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl –(400) subtrahiert die Binärwerte in Su von dem in Mi und gibt das Ergebnis in R aus. Wenn das Ergebnis negativ ist, wird es als Zweierkomplement in R ausgegeben. (Ein Beispiel für die Verarbeitung von Zweierkomplementen finden Sie unter *3-11-10 DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: –L(411)*.)



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Subtraktion zu einem negativen Übertrag führt. In allen anderen Fällen AUS.
Überlauf-Flag	OF	EIN, wenn das Ergebnis der Subtraktion einer negativen Zahl von einer positiven Zahl im Bereich von 8000 bis FFFF hex liegt. In allen anderen Fällen AUS.

Name	Bezeichnung	Funktion
Unterlauf-Flag	UF	EIN, wenn das Ergebnis der Subtraktion einer negativen Zahl von einer positiven Zahl im Bereich von 0000 bis 7FFF hex liegt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von –(410) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R als Ergebnis der Subtraktion 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Subtraktion zu einem negativen Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

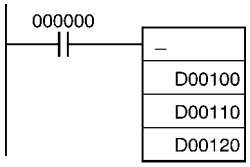
Wenn das Ergebnis der Subtraktion einer negativen Zahl von einer positiven Zahl negativ (im Bereich von 8000 bis FFFF hex) ist, wird das Überlauf-Flag auf EIN gesetzt.

Wenn das Ergebnis der Subtraktion einer positiven Zahl von einer negativen Zahl positiv (im Bereich von 8000 bis FFFF hex) ist, wird das Unterlauf-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R als Ergebnis der Subtraktion 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird D00110 von D00100 als 4-stellige Binärwerte mit Vorzeichen subtrahiert und das Ergebnis wird in D00120 ausgegeben.

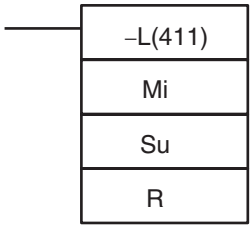


3-11-10 DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: –L(411)

Funktion

Subtrahiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten.

Kontaktplansymbol



Mi: Minuend-Wort

Su: Subtrahend-Wort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	–L(411)
	Ausführung einmal bei steigender Flanke	@–L(411)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Mi	Su	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich nicht remanent	W000 bis W510		
Arbeitsbereich remanent	H000 bis H510		
Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	IR0 bis IR15		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl –(411) subtrahiert den Binärwert in Su und Su+1 von dem in Mi und Mi+1 und gibt das Ergebnis in R, R+1 aus. Wenn das Ergebnis negativ ist, wird es als Zweierkomplement an R und R+1 ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Subtraktion zu einem negativen Übertrag führt. In allen anderen Fällen AUS.
Überlauf-Flag	OF	EIN, wenn das Ergebnis der Subtraktion einer negativen Zahl von einer positiven Zahl im Bereich von 80000000 bis FFFFFFFF hex liegt. In allen anderen Fällen AUS.

Name	Bezeichnung	Funktion
Unterlauf-Flag	UF	EIN, wenn das Ergebnis der Subtraktion einer positiven Zahl von einer negativen Zahl im Bereich von 00000000 bis 7FFFFFFF hex liegt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von –L(411) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R, R+1 als Ergebnis der Subtraktion 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Subtraktion zu einem negativen Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

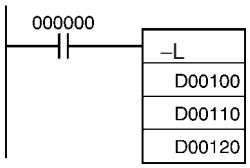
Wenn das Ergebnis der Subtraktion einer negativen Zahl von einer positiven Zahl negativ (im Bereich von 80000000 bis FFFFFFFF hex) ist, wird das Überlauf-Flag auf EIN gesetzt.

Wenn das Ergebnis der Subtraktion einer positiven Zahl von einer negativen Zahl positiv (im Bereich von 00000000 bis 7FFFFFFF hex) ist, wird das Unterlauf-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+1 als Ergebnis der Subtraktion 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00111 und D00110 von D00101 und D00100 als 8-stellige Binärwerte mit Vorzeichen subtrahiert und das Ergebnis wird in D00121 und D00120 ausgegeben.



Beispiele

Wenn das Ergebnis der Subtraktion eine negative Zahl ist (Mi<Su oder Mi+1, Mi <Su+1, Su), wird das Ergebnis als Zweierkomplement ausgegeben und das Übertrags-Flag (CY) wird auf EIN gesetzt, um darauf hinzuweisen, dass das Ergebnis der Subtraktion negativ ist. Um das Zweierkomplement in die tatsächliche Zahl zu konvertieren, ist zum Subtrahieren des Ergebnisses von 0 ein Befehl erforderlich, der das Übertrags-Flag (CY) als Ausführungsbedingung verwendet.

Hinweis Zweierkomplement

Ein Zweierkomplement ist der Wert, der durch Subtrahieren der einzelnen Binärstellen von 1 und das Addieren von 1 zum Ergebnis erzielt wird. Beispielsweise wird das Zweierkomplement von 1101 folgendermaßen berechnet: 1111 (F hexadezimal) – 1101 (D hexadezimal) + 1 (1 hexadezimal) = 0011 (3 hexadezimal). Das Zweierkomplement von 3039 (hexadezimal) wird folgendermaßen berechnet: FFFF (hexadezimal) – 3039 (hexadezimal) + 0001 (hexadezimal) – CFC7 (hexadezimal). Demzufolge kann das Zweierkomplement im Fall eines 4-stelligen Hexadezimalwerts wie folgt berechnet werden: FFFF (hexadezimal) – a (hexadezimal) + 0001 (hexadezimal) = b (hexadezimal). So wird die tatsächliche Zahl aus dem Zweierkomplement b (hexadezimal) erhalten: a (hexadezimal) = 10000 (hexadezimal) – b (hexadezimal). Beispielsweise wird die tatsächliche Zahl aus dem Zweierkomplement CFC7 (hexadezimal) wie folgt erhalten: 10000 (hexadezimal) – CFC7 = 3039.

Beispiel 1 Vorzeichenbehaftete Daten Vorzeichenlose Daten

FFFF hex	–1	65535
–) 0001 hex	–) +1	–) 1
FFFE hex	–2 Hinweis 1	65534 Hinweis 2

Negativ-Flag EIN
Übertrags-Flag AUS

Hinweis
1. Da das Negativ-Flag auf EIN gesetzt ist, ist das Ergebnis (FFFE hex) ein negativer Wert (Zweierkomplement) und lautet demzufolge –2.
2. Da das Übertrags-Flag auf AUS gesetzt ist, hat das Ergebnis (FFFE hex) den positiven Wert 65534 ohne Vorzeichen.

Beispiel 2 Vorzeichenbehaftete Daten Vorzeichenlose Daten

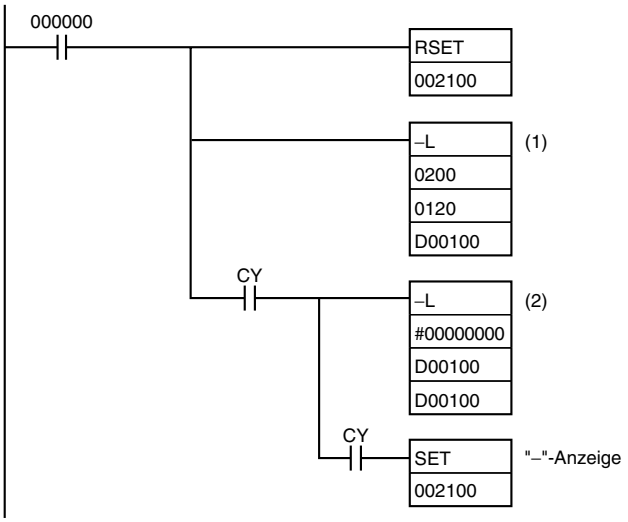
FFFD hex	–3	65533
–)FFFF hex	–) –1	–) 65535
FFFE hex	–2 Hinweis 3	65534 Hinweis 4

Negativ-Flag EIN
Übertrags-Flag AUS

3. Da das Negativ-Flag auf EIN gesetzt ist, ist das Ergebnis (FFFE hex) ein negativer Wert (Zweierkomplement) und lautet demzufolge
4. Da der Übertragsmarker auf EIN gesetzt ist, ist das Ergebnis (FFFE hex) ein negativer Wert (Zweierkomplement) und lautet demzufolge –2, wenn es in den tatsächlichen Wert konvertiert wird.

Programmbeispiel

20F55A10 – B8A360E3 = –97AE06D3.
In diesem Beispiel wird der 8-stellige Binärwert in CIO 0121 und CIO 0120 vom Wert in CIO 0201 und CIO 0200 subtrahiert und das Ergebnis wird als 8-stelliger Binärwert in D00101 und D00100 ausgegeben. Wenn das Ergebnis negativ ist, wird der Befehl bei (2) ausgeführt und das tatsächliche Ergebnis anschließend in D00101 und D00100 ausgegeben.

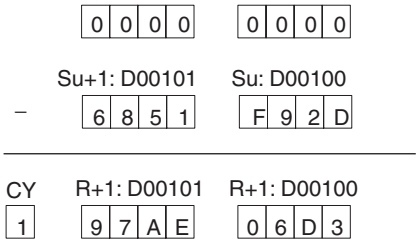


Subtraktion bei 1

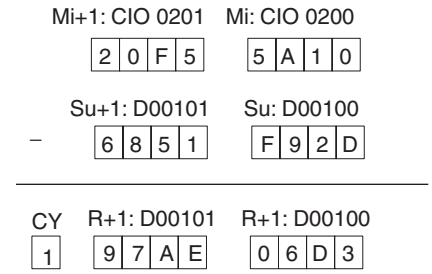
	Mi+1: CIO 0201	Mi: CIO 0200
	2 0 F 5	5 A 1 0
	Su+1: CIO 0121	Su: CIO 0120
–	B 8 A 3	6 0 E 3
CY	R+1: D00101	R+1: D00100
1	6 8 5 1	F 9 2 D

Da das Übertrags-Flag (CY) auf EIN gesetzt ist, wird das Ergebnis von 0000 0000 subtrahiert, um den tatsächlichen Wert zu erhalten.

Subtraktion bei 2



Endergebnis der Subtraktion



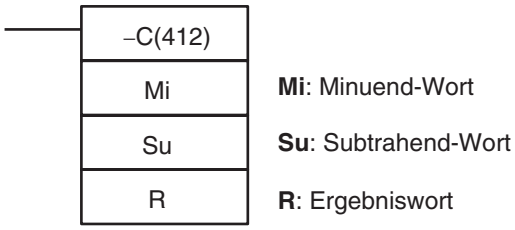
Wenn das Übertrags-Flag (CY) auf EIN gesetzt ist, lautet der tatsächliche Wert –97AE06D3. Da der Inhalt von D00101 und D00100 negativ ist, wird CIO 002100 durch CY auf EIN gesetzt, um dies anzugeben.

3-11-11 SIGNED BINARY SUBTRACT WITH CARRY: –C(412)

Funktion

Subtrahiert 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten mit Übertrags-Flag (CY).

Kontaktplansymbol



Mi: Minuend-Wort

Su: Subtrahend-Wort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	–C(412)
	Ausführung einmal bei steigender Flanke	@–C(412)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

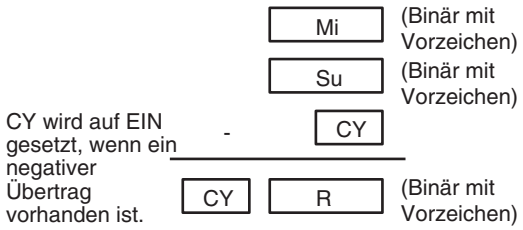
Operanden-Spezifikationen

Bereich	Mi	Su	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959

Bereich	Mi	Su	R
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl –C(412) subtrahiert die Binärwerte in Su und CY von Mi und gibt das Ergebnis in R aus. Wenn das Ergebnis negativ ist, wird es als Zweierkomplement in R ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Subtraktionsergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Subtraktion zu einem negativen Übertrag führt. In allen anderen Fällen AUS.
Überlauf-Flag	OF	EIN, wenn das Ergebnis der Subtraktion einer negativen Zahl und CY von einer positiven Zahl im Bereich von 8000 bis FFFF hex liegt. In allen anderen Fällen AUS.
Unterlauf-Flag	UF	EIN, wenn das Ergebnis der Subtraktion einer positiven Zahl und CY von einer negativen Zahl im Bereich von 0000 bis 7FFF hex liegt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von –C(412) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R als Ergebnis der Subtraktion 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Subtraktion zu einem negativen Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Wenn das Ergebnis der Subtraktion einer negativen Zahl und CY von einer positiven Zahl negativ (im Bereich von 8000 bis FFFF hex) ist, wird das Überlauf-Flag auf EIN gesetzt.

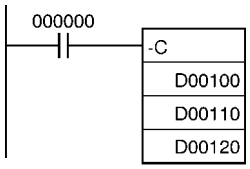
Wenn das Ergebnis der Subtraktion einer positiven Zahl und CY von einer negativen Zahl positiv (im Bereich von 0000 bis 7FFF hex) ist, wird das Unterlauf-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R als Ergebnis der Subtraktion 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Hinweis Zum Löschen des Übertrags-Flags (CY) führen Sie den Befehl CLEAR CARRY (CLC(041)) aus.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00110 und CY als 4-stellige Binärwerte mit Vorzeichen von D00100 subtrahiert und das Ergebnis wird in D00120 ausgegeben.

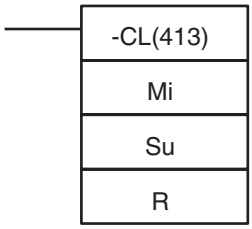


3-11-12 DOUBLE SIGNED BINARY SUBTRACT WITH CARRY: –CL(413)

Funktion

Subtrahiert 8-stellige (Doppelwort-) Hexadezimaldaten und/oder -konstanten mit Übertrags-Flag (CY).

Kontaktplansymbol



Mi: Minuend-Wort

Su: Subtrahend-Wort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	–CL(413)
	Ausführung einmal bei steigender Flanke	@–CL(413)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

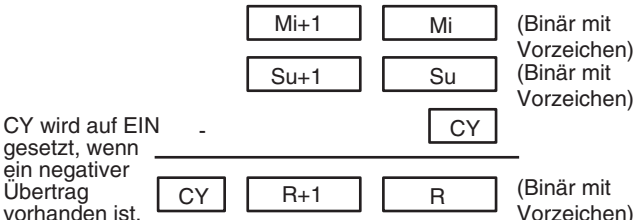
Operanden-Spezifikationen

Bereich	Mi	Su	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich nicht remanent	W000 bis W510		
Arbeitsbereich remanent	H000 bis H510		
Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		

Bereich	Mi	Su	R
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

–CL(413) subtrahiert die Binärwerte in Su, Su+1 und CY von dem in Mi und Mi+1 und gibt das Ergebnis in R, R+1 aus. Wenn das Ergebnis negativ ist, wird es als Zweierkomplement in R, R+1 ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn das Ergebnis zu einem negativen Übertrag führt. In allen anderen Fällen AUS.
Überlauf-Flag	OF	EIN, wenn das Ergebnis der Subtraktion einer negativen Zahl und CY von einer positiven Zahl im Bereich von 80000000 bis FFFFFFFF hex liegt. In allen anderen Fällen AUS.
Unterlauf-Flag	UF	EIN, wenn das Ergebnis der Subtraktion einer positiven Zahl und CY von einer negativen Zahl im Bereich von 00000000 bis 7FFFFFFF hex liegt. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von –CL(413) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R, R+1 als Ergebnis der Subtraktion 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn die Subtraktion zu einem negativen Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Wenn das Ergebnis der Subtraktion einer negativen Zahl und CY von einer positiven Zahl negativ (im Bereich von 80000000 bis FFFFFFFF hex) ist, wird das Überlauf-Flag auf EIN gesetzt.

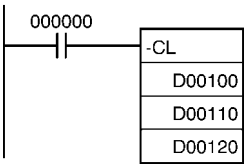
Wenn das Ergebnis der Subtraktion einer positiven Zahl und CY von einer negativen Zahl positiv (im Bereich von 00000000 bis 7FFFFFFF hex) ist, wird das Unterlauf-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+1 als Ergebnis der Subtraktion 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Hinweis Zum Löschen des Übertrags-Flags (CY) führen Sie den Befehl CLEAR CARRY (CLC(041)) aus.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00111, D00110 und CY von D00101 und D00100 als 8-stellige Binärwerte mit Vorzeichen subtrahiert und das Ergebnis wird in D00121 und D00120 ausgegeben.



Wenn das Ergebnis der Subtraktion eine negative Zahl (Mi<Su oder Mi+1, Mi <Su+1, Su) ist, wird das Ergebnis als Zweierkomplement ausgegeben. Das Übertrags-Flag (CY) wird auf EIN gesetzt. Um das Zweierkomplement in die tatsächliche Zahl zu konvertieren, ist zum Subtrahieren des Ergebnisses von 0 ein Befehl erforderlich, der das Übertrags-Flag (CY) als Eingangsbedingung verwendet. Das auf EIN gesetzte Übertrags-Flag weist somit darauf hin, dass das Ergebnis der Subtraktion negativ ist.

Hinweis Zweierkomplement

Ein Zweierkomplement ist der Wert, der durch Subtrahieren der einzelnen Binärstellen von 1 und das Addieren von 1 zum Ergebnis erzielt wird.

Beispiel: Das Zweierkomplement der Binärzahl 1101 wird wie folgt gebildet:
1111 (F hex) – 1101 (D hex) + 1 (1 hex) = 0011 (3 hex).

Beispiel: Das Zweierkomplement der 4-stelligen Hexadezimalzahl 3039 wird wie folgt gebildet:
FFFF hex – 3039 hex + 0001 hex = CFC7 hex.

Dementsprechend lautet das Zweierkomplement des 4-stelligen Hexadezimalwerts „a“ wie folgt:
FFFF hex – a hex + 0001 hex = b hex.

So wird die tatsächliche Zahl „a“ hex aus dem Zweierkomplement „b“ hex erhalten:
a hex + 10000 hex – b hex.

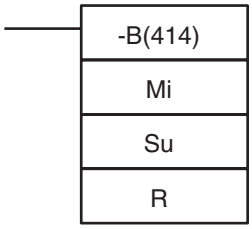
Beispiel: So wird die tatsächliche Zahl aus dem Zweierkomplement CFC7 hex erhalten:
10000 hex – CFC7 hex = 3039 hex.

3-11-13 BCD SUBTRACT WITHOUT CARRY: –B(414)

Funktion

Subtrahiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten.

Kontaktplansymbol



Mi: Minuend-Wort

Su: Subtrahend-Wort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	–B(414)
	Ausführung einmal bei steigender Flanke	@–B(414)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

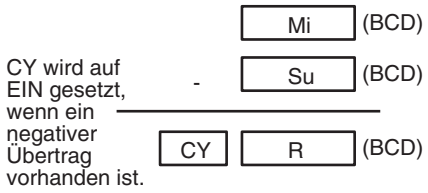
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Mi	Su	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	0000 bis 9999 (BCD)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl –B(414) subtrahiert den BCD-Wert in Su von dem in Mi und gibt das Ergebnis in R aus. Wenn das Ergebnis der Subtraktion negativ ist, wird das Ergebnis als Zehnerkomplement ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Mi kein BCD-Wert ist. EIN, wenn Su kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Subtraktion zu einem negativen Übertrag führt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

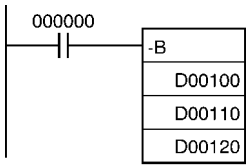
Wenn Mi und/oder Su keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von R als Ergebnis der Subtraktion 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn das Ergebnis zu einem negativen Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird D00110 von D00100 als 4-stelliger BCD-Wert subtrahiert und das Ergebnis wird in D00120 ausgegeben.

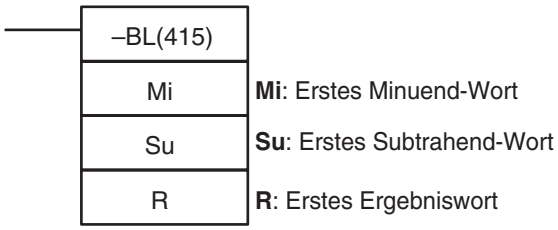


3-11-14 DOUBLE BCD SUBTRACT WITHOUT CARRY: –BL(415)

Funktion

Subtrahiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	–BL(415)
	Ausführung einmal bei steigender Flanke	@–BL(415)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

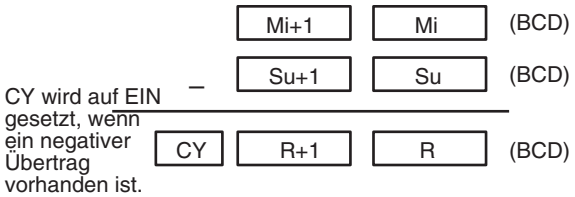
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Mi	Su	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich nicht remanent	W000 bis W510		
Arbeitsbereich remanent	H000 bis H510		
Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #99999999 (BCD)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl –BL(415) subtrahiert den BCD-Wert in Su und Su+1 von dem in Mi und Mi+1 und gibt das Ergebnis in R, R+1 aus. Wenn das Ergebnis negativ ist, wird es als Zehnerkomplement in R, R+1 ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Mi und/oder Mi +1 keine BCD-Werte sind. EIN, wenn Su und/oder Su +1 keine BCD-Werte sind. In allen anderen Fällen AUS.

Name	Bezeichnung	Funktion
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Subtraktion zu einem negativen Übertrag führt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

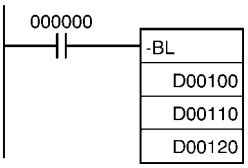
Wenn Mi, Mi +1 und/oder Su, Su +1 keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von R, R +1 als Ergebnis der Subtraktion 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn das Ergebnis zu einem negativen Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00111 und D00110 von D00101 und D00100 als 8-stellige BCD-Werte subtrahiert und das Ergebnis wird in D00121 und D00120 ausgegeben.



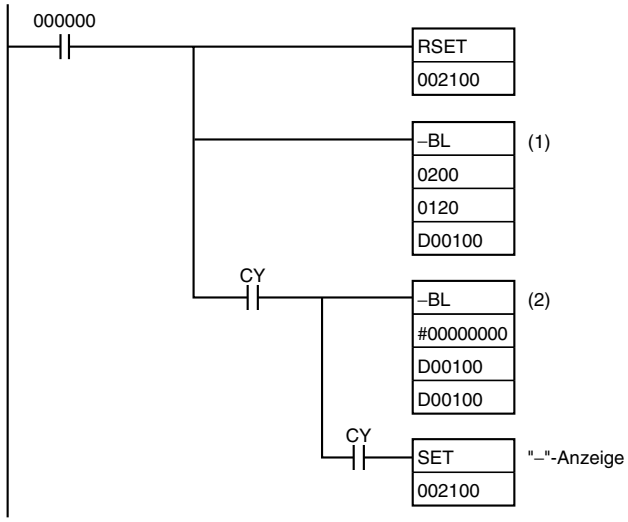
Wenn das Ergebnis der Subtraktion eine negative Zahl (Mi<Su oder Mi+1, Mi <Su+1, Su) ist, wird das Ergebnis als Zehnerkomplement ausgegeben. Das Übertrags-Flag (CY) wird auf EIN gesetzt. Um das Zehnerkomplement in die tatsächliche Zahl zu konvertieren, ist zum Subtrahieren des Ergebnisses von 0 ein Befehl erforderlich, der das Übertrags-Flag (CY) als Eingangsbedingung verwendet. Das auf EIN gesetzte Übertrags-Flag weist somit darauf hin, dass das Ergebnis der Subtraktion negativ ist.

Hinweis Zehnerkomplement

Ein Zehnerkomplement ist der Wert, der durch Subtrahieren der einzelnen Stellen von 9 und das Addieren von 1 zum Ergebnis ermittelt wird. Beispielsweise wird das Zehnerkomplement von 7556 folgendermaßen berechnet: 9999 – 7556 + 1 = 2444. Bei einer vierstelligen Zahl ist das Zehnerkomplement von A gleich 9999 – A + 1 = B. So erhält man die tatsächliche Zahl aus dem Zehnerkomplement B: A = 10000 – B. Beispielsweise erhält man die tatsächliche Zahl aus dem Zehnerkomplement 2444 wie folgt: 10000 – 2444 = 7556.

Programmbeispiel

9.583.960 – 17.072.641 = –7.488.681.
In diesem Beispiel wird der 8-stellige BCD-Inhalt von CIO 0121 und CIO 0120 von dem Inhalt von CIO 0201 und CIO 0200 subtrahiert und das Ergebnis wird als 8-stelliger BCD-Wert in D00101 und D00100 ausgegeben. Wenn das Ergebnis negativ ist, wird der Befehl bei (2) ausgeführt und der tatsächliche Wert anschließend in D00101 und D00100 ausgegeben.



Subtraktion bei 1

Mi+1: CIO 0201 Mi: CIO 0200

0	9	5	8		3	9	6	0

Su+1: CIO 0121 Su: CIO 0120

– | 1 | 7 | 0 | 7 | | 2 | 6 | 4 | 1 |

09583960 + (100000000 - 17072641)

CY R+1: D00101 R+1: D00100

1	9	2	5	1		1	3	1	9

Da das Übertrags-Flag (CY) auf EIN gesetzt ist, wird das Ergebnis von 0000 0000 subtrahiert.

Subtraktion bei 2

0	0	0	0		0	0	0	0

Su+1: D00101 Su: D00100

– | 9 | 2 | 5 | 1 | | 1 | 3 | 1 | 9 |

00000000 + (100000000 - 92511319)

CY R+1: D00101 R+1: D00100

1	0	7	4	8		8	6	8	1

Endergebnis der Subtraktion

Mi+1: CIO 0201 Mi: CIO 0200

2	0	F	5		5	A	1	0

Su+1: D00101 Su: D00100

– | 6 | 8 | 5 | 1 | | F | 9 | 2 | D |

CY R+1: D00101 R+1: D00100

1		0	7	4	8		8	6	8	1

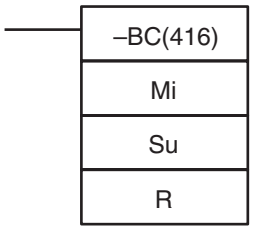
Wenn das Übertrags-Flag (CY) auf EIN gesetzt ist, lautet die tatsächliche Zahl –7.488.681. Da der Inhalt von D00101 und D00100 negativ ist, wird CIO 002100 durch CY auf EIN gesetzt, um dies anzugeben.

3-11-15 BCD SUBTRACT WITH CARRY: –BC(416)

Funktion

Subtrahiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten mit Übertrags-Flag (CY).

Kontaktplansymbol



Mi: Minuend-Wort

Su: Subtrahend-Wort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	–BC(416)
	Ausführung einmal bei steigender Flanke	@–BC(416)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

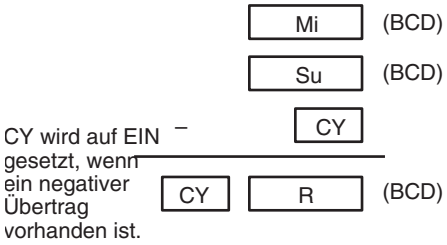
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Mi	Su	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis D32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #9999 (BCD)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl –BC(416) subtrahiert die BCD-Werte in Su und CY von dem in Mi und gibt das Ergebnis in R aus. Wenn das Ergebnis negativ ist, wird es als Zweierkomplement in R ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Mi kein BCD-Wert ist. EIN, wenn Su kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Subtraktion zu einem negativen Übertrag führt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn Mi und/oder Su keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

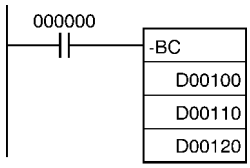
Wenn der Inhalt von R als Ergebnis der Subtraktion 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn das Ergebnis zu einem negativen Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Hinweis Zum Löschen des Übertrags-Flags (CY) führen Sie den Befehl CLEAR CARRY (CLC(041)) aus.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00110 und CY als 4-stellige BCD-Werte von D00100 subtrahiert und das Ergebnis wird in D00120 ausgegeben.

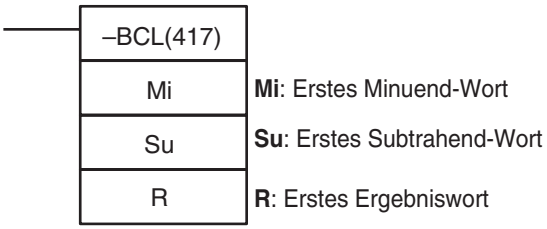


3-11-16 DOUBLE BCD SUBTRACT WITH CARRY: –BCL(417)

Funktion

Subtrahiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten mit Übertrags-Flag (CY).

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	–BCL(417)
	Ausführung einmal bei steigender Flanke	@–BCL(417)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

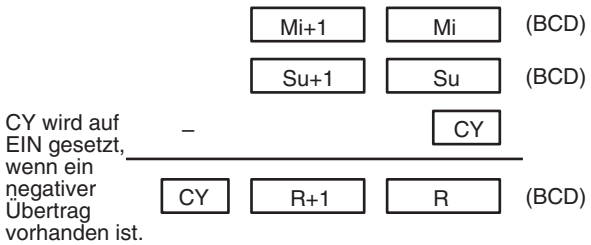
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Mi	Su	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich nicht remanent	W000 bis W510		
Arbeitsbereich remanent	H000 bis H510		
Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #99999999 (BCD)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl –BCL(417) subtrahiert die BCD-Werte in Su, Su+1 und CY von dem in Mi und Mi+1 und gibt das Ergebnis in R, R+1 aus. Wenn das Ergebnis negativ ist, wird es als Zehnerkomplement in R, R+1 ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Mi und/oder Mi +1 keine BCD-Werte sind. EIN, wenn Su und/oder Su +1 keine BCD-Werte sind. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn die Subtraktion zu einem negativen Übertrag führt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn Mi, Mi +1 und/oder Su, Su +1 keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

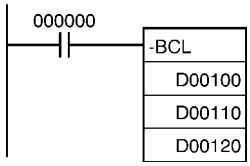
Wenn der Inhalt von R, R +1 als Ergebnis der Subtraktion 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn ein Subtraktionsergebnis zu einem negativen Übertrag führt, wird das Übertrags-Flag auf EIN gesetzt.

Hinweis Zum Löschen des Übertrags-Flags (CY) führen Sie den Befehl CLEAR CARRY (CLC(041)) aus.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00111, D00110 und CY von D00101 und D00100 als 8-stellige BCD-Werte subtrahiert und das Ergebnis wird in D00121 und D00120 ausgegeben.



Wenn das Ergebnis der Subtraktion eine negative Zahl (Mi<Su oder Mi+1, Mi <Su+1, Su) ist, wird das Ergebnis als Zehnerkomplement ausgegeben. Das Übertrags-Flag (CY) wird auf EIN gesetzt. Um das Zehnerkomplement in die tatsächliche Zahl zu konvertieren, ist zum Subtrahieren des Ergebnisses von 0 ein Befehl erforderlich, der das Übertrags-Flag (CY) als Eingangsbedingung verwendet. Das auf EIN gesetzte Übertrags-Flag weist somit darauf hin, dass das Ergebnis der Subtraktion negativ ist.

Hinweis **Zehnerkomplement**
Ein Zehnerkomplement ist der Wert, der durch Subtrahieren der einzelnen Stellen von 9 und das Addieren von 1 zum Ergebnis ermittelt wird. Beispielsweise wird das Zehnerkomplement von 7556 folgendermaßen berechnet: 9999 – 7556 + 1 = 2444. Bei einer vierstelligen Zahl ist das Zehnerkomplement von A gleich 9999 – A + 1 = B. So erhält man die tatsächliche Zahl aus dem Zehnerkomplement B: A = 10000 – B. Beispielsweise erhält man die tatsächliche Zahl aus dem Zehnerkomplement 2444 wie folgt: 10000 – 2444 = 7556.

3-11-17 SIGNED BINARY MULTIPLY: *(420)

Funktion

Multipliziert 4-stellige vorzeichenbehaftete Hexadezimaldaten und/oder -konstanten miteinander.

Kontaktplansymbol



Md: Multiplikand-Wort

Mr: Multiplikator-Wort

R: Ergebniswort

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	*(420)
	Ausführung einmal bei steigender Flanke	@*(420)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Md	Mr	R
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W511		W000 bis W510
Arbeitsbereich remanent	H000 bis H511		H000 bis H510
Systembereich	A000 bis A959		A448 bis A958
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4094
Zählerbereich	C0000 bis C4095		C0000 bis C4094
DM-Bereich	D00000 bis D32767		D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl *(420) multipliziert die vorzeichenbehafteten Binärwerte in Md and Mr miteinander und gibt das Ergebnis in R+1, R aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

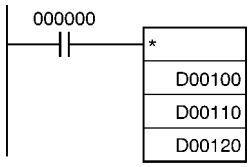
Nach der Ausführung von *(420) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R als Ergebnis der Multiplikation 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+1 und R als Ergebnis der Multiplikation 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00100 und D00110 als 4-stellige vorzeichenbehaftete Hexadezimalwerte miteinander multipliziert und das Ergebnis wird in D00120 ausgegeben.

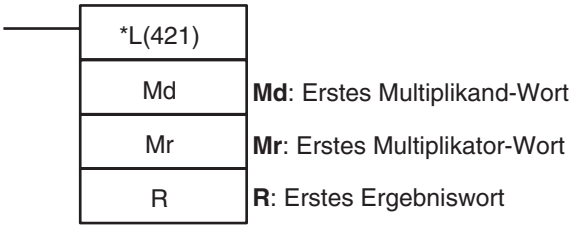


3-11-18 DOUBLE SIGNED BINARY MULTIPLY: *L(421)

Funktion

Multipliziert 8-stellige vorzeichenbehaftete Hexadezimaldaten und/oder -konstanten miteinander.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	*L(421)
	Ausführung einmal bei steigender Flanke	@*L(421)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

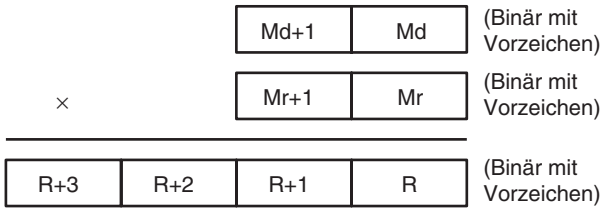
Operanden-Spezifikationen

Bereich	Md	Mr	R
CIO-Bereich	CIO 0000 bis CIO 6142		CIO 0000 bis CIO 6140
Arbeitsbereich nicht remanent	W000 bis W510		W000 bis W508
Arbeitsbereich remanent	H000 bis H510		H000 bis H508
Systembereich	A000 bis A958		A448 bis A956

Bereich	Md	Mr	R
Zeitgeberbereich	T0000 bis T4094		T0000 bis T4092
Zählerbereich	C0000 bis C4094		C0000 bis C4092
DM-Bereich	D00000 bis D32766		D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32766		E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl *L(421) multipliziert den vorzeichenbehafteten Binärwert in Md und Md+1 mit dem in Mr und Mr+1 und gibt das Ergebnis in R, R+1, R+2 und R+3 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

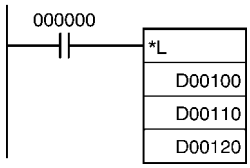
Nach der Ausführung von *L(421) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R, R+1, R+2, R+3 als Ergebnis der Multiplikation 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+1 als Ergebnis der Multiplikation 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00100, D00110, D00111 und D00110 als 8-stellige Hexadezimalwerte mit Vorzeichen miteinander multipliziert und das Ergebnis wird in D00121 und D00120 ausgegeben.

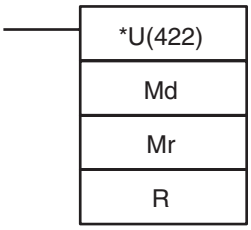


3-11-19 UNSIGNED BINARY MULTIPLY: *U(422)

Funktion

Multipliziert 4-stellige vorzeichenlose Hexadezimaldaten und/oder -konstanten miteinander.

Kontaktplansymbol



Md: Multiplikand-Wort

Mr: Multiplikator-Wort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	*U(422)
	Ausführung einmal bei steigender Flanke	@*U(422)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Md	Mr	R
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W511		W000 bis W510
Arbeitsbereich remanent	H000 bis H511		H000 bis H510
Systembereich	A000 bis A959		A448 bis A958
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4094
Zählerbereich	C0000 bis C4095		C0000 bis C4094
DM-Bereich	D00000 bis D32767		D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		

Bereich	Md	Mr	R
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_ 32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl *U(420) multipliziert die Binärwerte in Md and Mr miteinander und gibt das Ergebnis in R, R+1 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

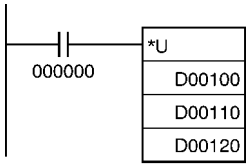
Nach der Ausführung von *U(422) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R, R+1 als Ergebnis der Multiplikation 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+1 als Ergebnis der Multiplikation 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00100 und D00110 als 4-stellige vorzeichenlose Binärwerte miteinander multipliziert und das Ergebnis wird in D00121 und D00120 ausgegeben.

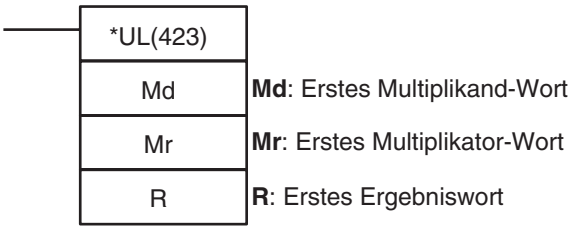


3-11-20 DOUBLE UNSIGNED BINARY MULTIPLY: *UL(423)

Funktion

Multipliziert 8-stellige vorzeichenlose Hexadezimaldaten und/oder -konstanten miteinander.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	*UL(423)
	Ausführung einmal bei steigender Flanke	@*UL(423)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

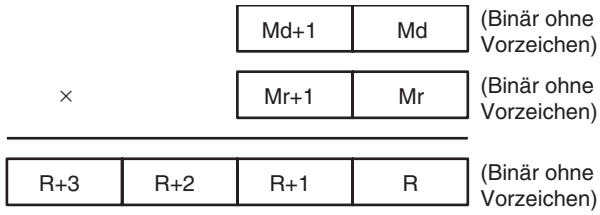
Operanden-Spezifikationen

Bereich	Md	Mr	R
CIO-Bereich	CIO 0000 bis CIO 6142		CIO 0000 bis CIO 6140
Arbeitsbereich nicht remanent	W000 bis W510		W000 bis W508
Arbeitsbereich remanent	H000 bis H510		H000 bis H508
Systembereich	A000 bis A958		A448 bis A956
Zeitgeberbereich	T0000 bis T4094		T0000 bis T4092
Zählerbereich	C0000 bis C4094		C0000 bis C4092
DM-Bereich	D00000 bis D32766		D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32766		E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		

Bereich	Md	Mr	R
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl *UL(423) multipliziert den vorzeichenlosen Binärwert in Md und Md+1 mit dem in Mr und Mr+1 und gibt das Ergebnis in R, R+1, R+2 und R+3 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

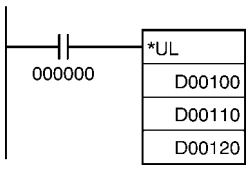
Nach der Ausführung von *UL(423) wird das Fehler-Flag auf AUS gesetzt.

Wenn der Inhalt von R, R+1, R+2, R+3 als Ergebnis der Multiplikation 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+3 als Ergebnis der Multiplikation 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00100, D00110, D00111 und D00110 als 8-stellige vorzeichenlose Binärwerte miteinander multipliziert und das Ergebnis wird in D00123, D00122, D00121 und D00120 ausgegeben.



3-11-21 BCD MULTIPLY: *B(424)

Funktion

Multipliziert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten miteinander.

Kontaktplansymbol



Md: Multiplikand-Wort

Mr: Multiplikator-Wort

R: Ergebniswort

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	*B(424)
	Ausführung einmal bei steigender Flanke	@*B(424)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Md	Mr	R
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W511		W000 bis W510
Arbeitsbereich remanent	H000 bis H511		H000 bis H510
Systembereich	A000 bis A959		A448 bis A958
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4094
Zählerbereich	C0000 bis C4095		C0000 bis C4094
DM-Bereich	D00000 bis D32767		D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #9999 (BCD)		---
Datenregister	DR0 bis DR15		---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl *B(424) multipliziert den BCD-Inhalt von Md mit dem von Mr und gibt das Ergebnis in R, R+1 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Md kein BCD-Wert ist. EIN, wenn Mr kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.

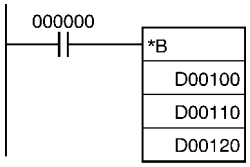
Vorsichtsmaßnahmen

Wenn Md und/oder Mr keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von R, R+1 als Ergebnis der Multiplikation 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00100 und D00110 als 4-stellige BCD-Werte miteinander multipliziert und das Ergebnis wird in D00121 und D00120 ausgegeben.



3-11-22 DOUBLE BCD MULTIPLY: *BL(425)

Funktion

Multipliziert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten miteinander.

Kontaktplansymbol



Md: Erstes Multiplikand-Wort

Mr: Erstes Multiplikator-Wort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	*BL(425)
	Ausführung einmal bei steigender Flanke	@*BL(425)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Md	Mr	R
CIO-Bereich	CIO 0000 bis CIO 6142		CIO 0000 bis CIO 6140
Arbeitsbereich nicht remanent	W000 bis W510		W000 bis W508
Arbeitsbereich remanent	H000 bis H510		H000 bis H508
Systembereich	A000 bis A958		A448 bis A956
Zeitgeberbereich	T0000 bis T4094		T0000 bis T4092

Bereich	Md	Mr	R
Zählerbereich	C0000 bis C4094		C0000 bis C4092
DM-Bereich	D00000 bis D32766		D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32766		E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #99999999 (BCD)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl *BL(425) multipliziert den BCD-Wert in Md und Md+1 mit dem in Mr und Mr+1 und gibt das Ergebnis in R, R+1, R+2 und R+3 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Md und/oder Md+1 keine BCD-Werte sind. EIN, wenn Mr und/oder Mr +1 keine BCD-Werte sind. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.

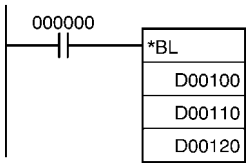
Vorsichtsmaßnahmen

Wenn Md, Md+1 und/oder Mr, Mr+1 keine BCD-Werte sind, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von R, R+1, R+2, R+3 als Ergebnis der Multiplikation 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden D00101, D00100 und D00111, D00110 als 8-stellige vorzeichenlose BCD-Werte miteinander multipliziert und das Ergebnis wird in D00123, D00122, D00121 und D00120 ausgegeben.

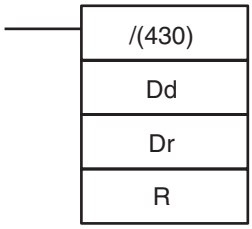


3-11-23 SIGNED BINARY DIVIDE: /(430)

Funktion

Dividiert vorzeichenbehaftete 4-stellige (Einzelwort-) Hexadezimaldaten und/oder -konstanten.

Kontaktplansymbol



Dd: Dividend-Wort

Dr: Divisor-Wort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	/(430)
	Ausführung einmal bei steigender Flanke	@/(430)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

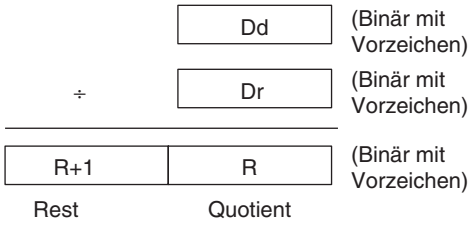
Operanden-Spezifikationen

Bereich	Dd	Dr	R
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W511		W000 bis W510
Arbeitsbereich remanent	H000 bis H511		H000 bis H510
Systembereich	A000 bis A959		A448 bis A958
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4094
Zählerbereich	C0000 bis C4095		C0000 bis C4094
DM-Bereich	D00000 bis D32767		D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		

Bereich	Dd	Dr	R
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	#0001 bis #FFFF (binär)	---
Datenregister	DR0 bis DR15		---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl /(430) dividiert den vorzeichenbehafteten Binärwert (16 Bit) in Dd durch den in Dr und gibt das Ergebnis in R, R+1 aus. Der Quotient wird in R und der Rest in R+1 geschrieben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn R als Ergebnis der Division 0 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

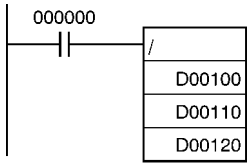
Wenn der Inhalt von Dr 0 ist, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von R als Ergebnis der Division 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R als Ergebnis der Division 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird D00100 als 4-stelliger Binärwert mit Vorzeichen durch D00110 dividiert, der Quotient in D00120 und der Rest in D00121 ausgegeben.

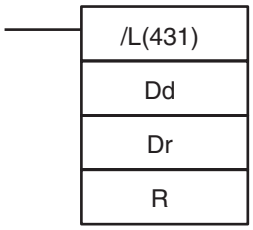


3-11-24 DOUBLE SIGNED BINARY DIVIDE: /L(431)

Funktion

Dividiert 8-stellige vorzeichenbehaftete (Doppelwort-) Hexadezimaldaten und/oder -konstanten.

Kontaktplansymbol



Dd: Erstes Dividend-Wort

Dr: Erstes Divisor-Wort

R: Erstes Ergebniswort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	/L(431)
	Ausführung einmal bei steigender Flanke	@/L(431)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

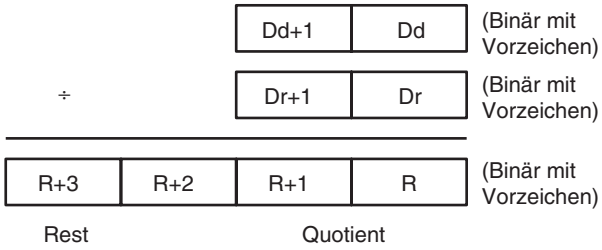
Operanden-Spezifikationen

Bereich	Dd	Dr	R
CIO-Bereich	CIO 0000 bis CIO 6142		CIO 0000 bis CIO 6140
Arbeitsbereich nicht remanent	W000 bis W510		W000 bis W508
Arbeitsbereich remanent	H000 bis H510		H000 bis H508
Systembereich	A000 bis A958		A448 bis A956
Zeitgeberbereich	T0000 bis T4094		T0000 bis T4092
Zählerbereich	C0000 bis C4094		C0000 bis C4092
DM-Bereich	D00000 bis D32766		D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32766		E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)	#00000001 to #FFFFFFFF (binär)	---
Datenregister	---		

Bereich	Dd	Dr	R
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl /L(431) dividiert den vorzeichenbehafteten Binärwert in Dd und Dd+1 durch den in Dr und Dr+1 und gibt das Ergebnis in R, R+1, R+2 und R+3 aus. Der Quotient wird in R und R+1 und der Rest in R+2 und R+3 ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn R+1, R als Ergebnis der Division 0 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit von R+1, R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

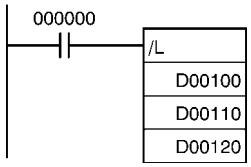
Wenn der Rest des Ergebnisses, R+3, R+2 gleich 0 ist, wird das Fehler-Flag auf EIN gesetzt.

Wenn der Inhalt von R+1, R als Ergebnis der Division 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+1 als Ergebnis der Division 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird D00101 und D00100 als 8-stelliger Hexadezimalwert mit Vorzeichen durch D00111 und D00110 dividiert und der Quotient wird in D00121 und D00120, der Rest in D00123 und D00122 ausgegeben.

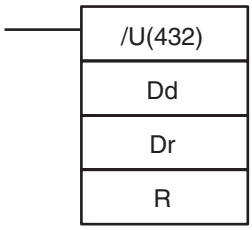


3-11-25 UNSIGNED BINARY DIVIDE: /U(432)

Funktion

Dividiert 4-stellige vorzeichenlose (Einzelwort-) Hexadezimaldaten und/oder -konstanten.

Kontaktplansymbol



Dd: Dividend-Wort

Dr: Divisor-Wort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	/U(432)
	Ausführung einmal bei steigender Flanke	@/U(432)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

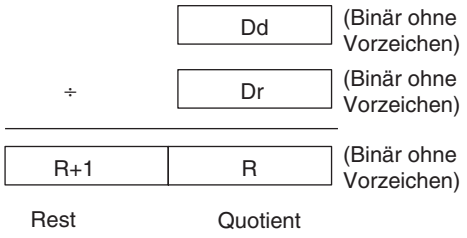
Operanden-Spezifikationen

Bereich	Dd	Dr	R
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W511		W000 bis W510
Arbeitsbereich remanent	H000 bis H511		H000 bis H510
Systembereich	A000 bis A959		A448 bis A958
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4094
Zählerbereich	C0000 bis C4095		C0000 bis C4094
DM-Bereich	D00000 bis D32767		D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	#0001 bis #FFFF (binär)	---
Datenregister	DR0 bis 15		---

Bereich	Dd	Dr	R
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl /U(432) dividiert den vorzeichenlosen Binärwert in Dd durch den in Dr und gibt den Quotienten in R und den Rest in R+1 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn R als Ergebnis der Division 0 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

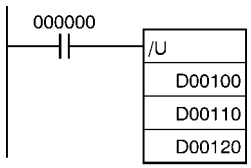
Wenn der Inhalt von R+1 als Ergebnis der Division 0 ist, wird das Fehler-Flag auf EIN gesetzt.

Wenn der Inhalt von R als Ergebnis der Division 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R als Ergebnis der Division 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird D00100 als 4-stelliger Binärwert ohne Vorzeichen durch D00110 dividiert, der Quotient in D00120 und der Rest in D00121 ausgegeben.

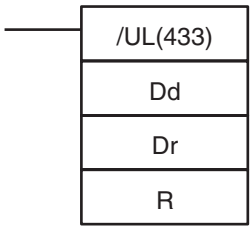


3-11-26 DOUBLE UNSIGNED BINARY DIVIDE: /UL(433)

Funktion

Dividiert 8-stellige vorzeichenlose (Doppelwort-) Hexadezimaldaten und/oder -konstanten.

Kontaktplansymbol



Dd: Erstes Dividend-Wort

Dr: Erstes Divisor-Wort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	/UL(433)
	Ausführung einmal bei steigender Flanke	@/UL(433)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

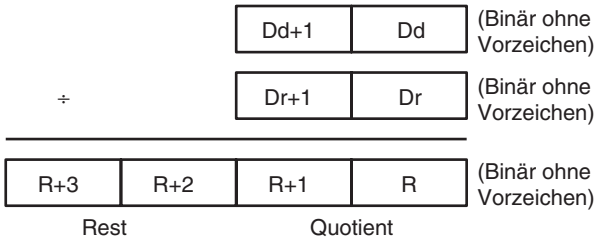
Operanden-Spezifikationen

Bereich	Dd	Dr	R
CIO-Bereich	CIO 0000 bis CIO 6142		CIO 0000 bis CIO 6140
Arbeitsbereich nicht remanent	W000 bis W510		W000 bis W508
Arbeitsbereich remanent	H000 bis H510		H000 bis H508
Systembereich	A000 bis A958		A448 bis A956
Zeitgeberbereich	T0000 bis T4094		T0000 bis T4092
Zählerbereich	C0000 bis C4094		C0000 bis C4092
DM-Bereich	D00000 bis D32766		D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32766		E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)	#00000001 to #FFFFFFFF (binär)	---
Datenregister	---		

Bereich	Dd	Dr	R
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl /UL(433) dividiert den vorzeichenlosen Binärwert in Dd und Dd+1 durch den in Dr und Dr+1 und gibt den Quotienten in R, R+1 und den Rest in R+2 und R+3 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn R+1, R als Ergebnis der Division 0 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn das höchstwertige Bit von R+1, R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

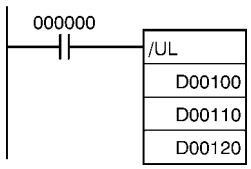
Wenn der Inhalt von Dr, Dr+1 gleich 0 ist, wird das Fehler-Flag auf EIN gesetzt.

Wenn der Inhalt von R, R+1 als Ergebnis der Division 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn der Inhalt des höchstwertigen Bits von R+1 als Ergebnis der Division 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird D00100 und D00101 als 8-stelliger Hexadezimalwert mit Vorzeichen durch D00111 und D00110 dividiert und der Quotient wird in D00121 und D00120, der Rest in D00123 und D00122 ausgegeben.

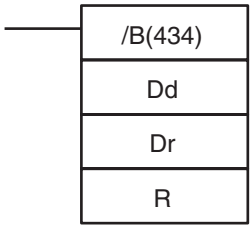


3-11-27 BCD DIVIDE: /B(434)

Funktion

Dividiert 4-stellige (Einzelwort-) BCD-Daten und/oder -Konstanten.

Kontaktplansymbol



Dd: Dividend-Wort

Dr: Divisor-Wort

R: Ergebniswort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	/B(434)
	Ausführung einmal bei steigender Flanke	@/B(434)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

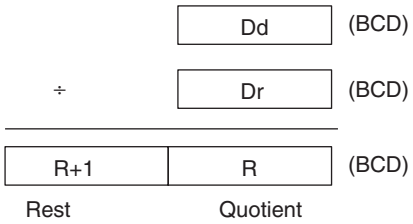
Operanden-Spezifikationen

Bereich	Dd	Dr	R
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W511		W000 bis W510
Arbeitsbereich remanent	H000 bis H511		H000 bis H510
Systembereich	A000 bis A959		A448 bis A958
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4094
Zählerbereich	C0000 bis C4095		C0000 bis C4094
DM-Bereich	D00000 bis D32767		D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #9999 (BCD)	#0001 bis #9999 (BCD)	---
Datenregister	DR0 bis DR15		---

Bereich	Dd	Dr	R
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl /B(434) dividiert den BCD-Inhalt von Dd durch den in Dr und gibt den Quotienten in R und den Rest in R+1 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Dd kein BCD-Wert ist. EIN, wenn Dr kein BCD-Wert ist. EIN, wenn der Rest 0 ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn R gleich 0 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

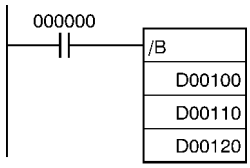
Wenn Dd oder Dr keine BCD-Werte sind oder der Rest (R+1) 0 ist, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von R als Ergebnis der Division 0000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Wenn das höchstwertige Bit von R als Ergebnis der Division 1 ist, wird das Negativ-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird D00100 als 4-stelliger BCD-Wert durch D00110 dividiert, der Quotient in D00120 und der Rest in D00121 ausgegeben.

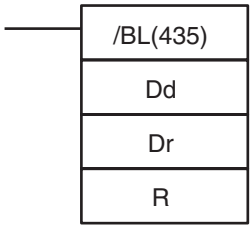


3-11-28 DOUBLE BCD DIVIDE: /BL(435)

Funktion

Dividiert 8-stellige (Doppelwort-) BCD-Daten und/oder -Konstanten.

Kontaktplansymbol



Dd: Erstes Dividend-Wort

Dr: Erstes Divisor-Wort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	/BL(435)
	Ausführung einmal bei steigender Flanke	@/BL(435)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

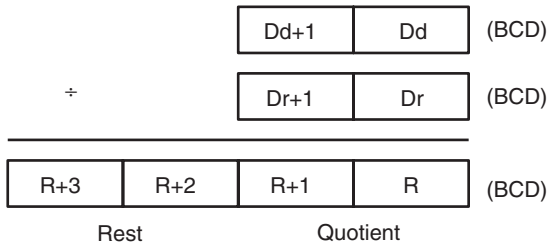
Operanden-Spezifikationen

Bereich	Dd	Dr	R
CIO-Bereich	CIO 0000 bis CIO 6142		CIO 0000 bis CIO 6140
Arbeitsbereich nicht remanent	W000 bis W510		W000 bis W508
Arbeitsbereich remanent	H000 bis H510		H000 bis H508
Systembereich	A000 bis A958		A448 bis A956
Zeitgeberbereich	T0000 bis T4094		T0000 bis T4092
Zählerbereich	C0000 bis C4094		C0000 bis C4092
DM-Bereich	D00000 bis D32766		D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32766		E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #99999999 (BCD)	#00000001 bis #99999999 (BCD)	---
Datenregister	---		

Bereich	Dd	Dr	R
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl /BL(435) dividiert den BCD-Wert in Dd und Dd+1 durch den in Dr und Dr+1 und gibt den Quotienten in R, R+1, den Rest in R+2, R+3 aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Dd, Dd+1 kein BCD-Wert ist. EIN, wenn Dr, Dr +1 kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.

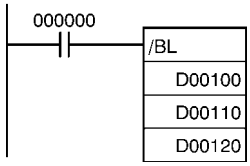
Vorsichtsmaßnahmen

Wenn Dd, Dd+1 und/oder Dr, Dr+1 keine BCD-Werte sind oder der Inhalt von Dr, Dr+1 gleich 0 ist, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn der Inhalt von R, R+1 als Ergebnis der Division 00000000 hex ist, wird das Gleich-Flag auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird D00101 und D00100 als 8-stelliger BCD-Wert durch D00111 und D00110 dividiert und der Quotient in D00121 und D00120, der Rest in D00123 und D00122 ausgegeben.



3-12 Konvertierungsbefehle

In diesem Abschnitt werden die Befehle für die Datenkonvertierung beschrieben.

Befehl	AWL	Funktionscode	Seite
BCD-TO-BINARY	BIN	023	466
DOUBLE BCD-TO-DOUBLE BINARY	BINL	058	467
BINARY-TO-BCD	BCD	024	469
DOUBLE BINARY-TO-DOUBLE BCD	BCDL	059	470
2'S COMPLEMENT	NEG	160	472
DOUBLE 2'S COMPLEMENT	NEGL	161	474

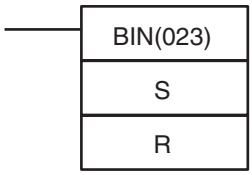
Befehl	AWL	Funktionscode	Seite
16-BIT TO 32-BIT SIGNED BINARY	SIGN	600	476
DATA DECODER	MLPX	076	477
DATA ENCODER	DMPX	077	482
ASCII CONVERT	ASC	086	486
ASCII TO HEX	HEX	162	490
COLUMN TO LINE	LINE	063	494
LINE TO COLUMN	COLM	064	496
SIGNED BCD-TO-BINARY	BINS	470	499
DOUBLE SIGNED BCD-TO-BINARY	BISL	472	502
SIGNED BINARY-TO-BCD	BCDS	471	505
DOUBLE SIGNED BINARY-TO-BCD	BDSL	473	507
GRAY CODE CONVERSION	GRY	474	511

3-12-1 BCD-TO-BINARY: BIN(023)

Funktion

Konvertiert BCD-Daten in Binärdaten.

Kontaktplansymbol



S: Quellwort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	BIN(023)
	Ausführung einmal bei steigender Flanke	@BIN(023)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	

Bereich	S	R
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

BIN(023) konvertiert BCD-Daten in S in Binärdaten und schreibt das Ergebnis in R.

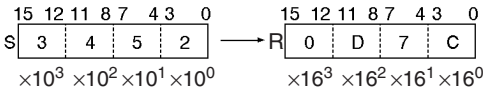


Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von S kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	AUS

Beispiel

Die folgende Abbildung zeigt ein Beispiel für eine Konvertierung von BCD-Daten in Binärdaten.

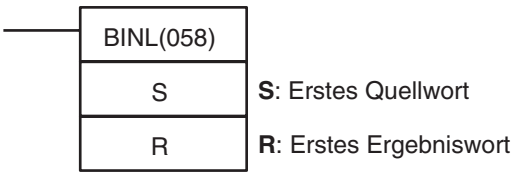


3-12-2 DOUBLE BCD-TO-DOUBLE BINARY: BINL(058)

Funktion

Konvertiert 8-stellige BCD-Daten in 8-stellige Hexadezimaldaten (32 Bits binär).

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	BINL(058)
	Ausführung einmal bei steigender Flanke	@BINL(058)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

BINL(058) konvertiert die 8-stelligen BCD-Daten in S und S+1 in 8-stellige Hexadezimaldaten (32 Bit binär) und schreibt das Ergebnis in R und R+1.

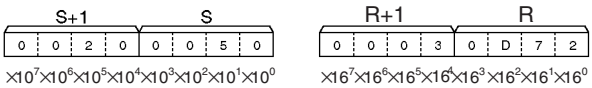


Flags

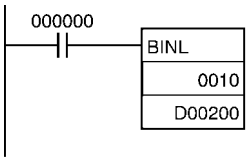
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von S+1, S kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	AUS

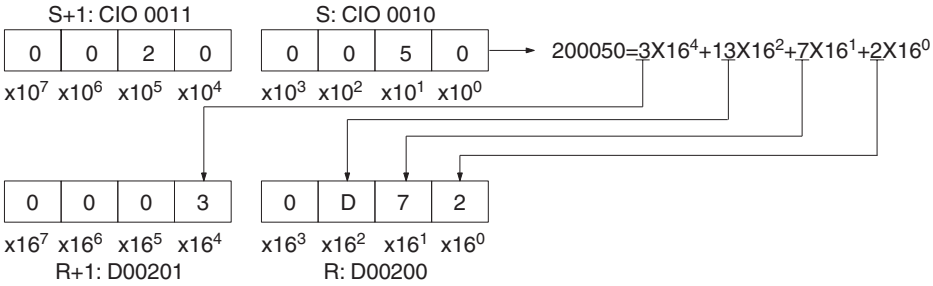
Beispiele

Die folgende Abbildung zeigt ein Beispiel für eine Konvertierung von 8-stelligen BCD-Daten in Binärdaten.



Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird der 8-stellige BCD-Wert in CIO 0010 und CIO 0011 in einen Hexadezimalwert konvertiert und in D00200 und D00201 gespeichert.



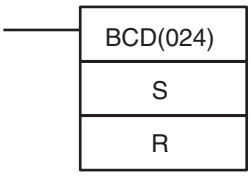


3-12-3 BINARY-TO-BCD: BCD(024)

Funktion

Konvertiert ein Wort mit Binärdaten in ein Wort BCD-Daten.

Kontaktplansymbol



S: Quellwort

R: Ergebniswort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	BCD(024)
	Ausführung einmal bei steigender Flanke	@BCD(024)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Quellwort

S muss zwischen 0000 und 270F hexadezimal (0000 und 9999 dezimal) liegen.

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	

Bereich	S	R
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl BCD(024) konvertiert die Binärdaten in S zu BCD-Daten und schreibt das Ergebnis in R.



Flags

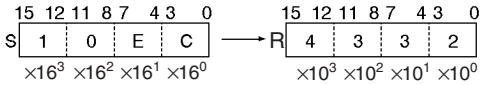
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von S größer als 270F (9999 dezimal) ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Inhalt von S darf 270F (9999 dezimal) nicht überschreiten.

Beispiel

Die folgende Abbildung zeigt ein Beispiel für eine Konvertierung von BCD-Daten in Binärdaten.

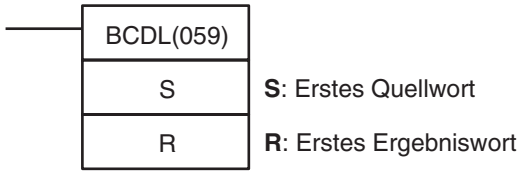


3-12-4 DOUBLE BINARY-TO-DOUBLE BCD: BCDL(059)

Funktion

Konvertiert 8-stellige Hexadezimaldaten (32 Bit binär) in 8-stellige BCD-Daten.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	BCDL(059)
	Ausführung einmal bei steigender Flanke	@BCDL(059)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Erstes Quellwort

Der Inhalt von S+1 und S muss zwischen 0000 0000 und 05F5 E0FF hexadezimal (0000 0000 und 9999 9999 dezimal) liegen.

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

BCDL(058) konvertiert die 8-stelligen Hexadezimaldaten (32 Bit binär) in S und S+1 in 8-stellige BCD-Daten und schreibt das Ergebnis in R und R+1.



Flags

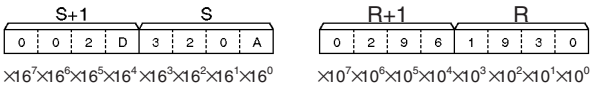
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von S und S+1 größer als 05F5 E0FF (9999 9999 dezimal) ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

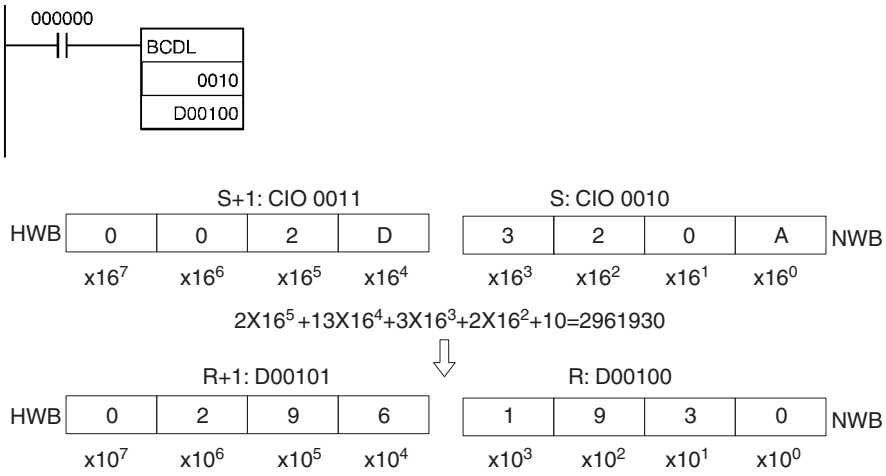
Der Inhalt von S+1 und S darf 05F5 E0FF (9999 9999 dezimal) nicht überschreiten.

Beispiele

Die folgende Abbildung zeigt ein Beispiel für eine 8-stellige BCD-BIN-Konvertierung.



Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird der Hexadezimalwert in CIO 0011 und CIO 0010 in einen BCD-Wert konvertiert und in D00200 und D00201 gespeichert.

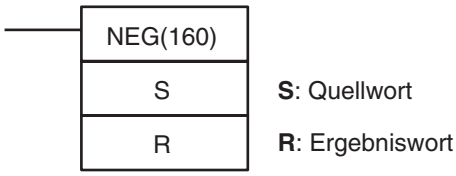


3-12-5 2'S COMPLEMENT: NEG(160)

Funktion

Berechnet das Zweierkomplement eines Worts von Hexadezimaldaten.

Kontaktplansymbol



S: Quellwort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	NEG(160)
	Ausführung einmal bei steigender Flanke	@NEG(160)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	

Bereich	S	R
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #FFFF (binär)	---
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

NEG(160) berechnet das Zweierkomplement von S und schreibt das Ergebnis in R. Bei der Zweierkomplementberechnung wird der Status der Bits in S invertiert und 1 dazuaddiert.



Hinweis Diese Funktion (Status der Bits umkehren und 1 addieren) entspricht der Subtraktion des Inhalts von S von 0000.

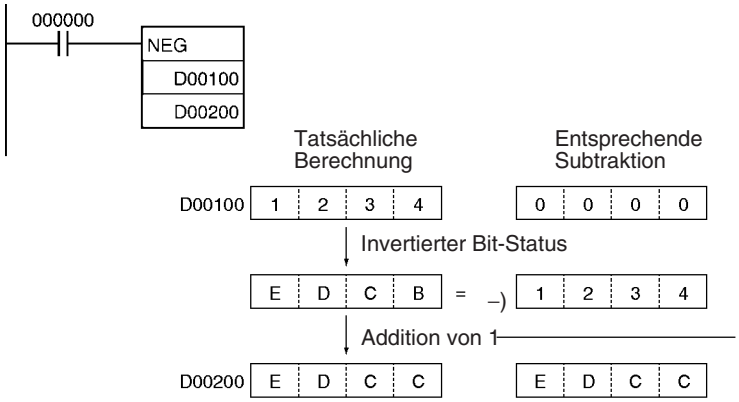
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn Bit 15 des Ergebnisses EIN ist. In allen anderen Fällen AUS.

Hinweis Das Ergebnis für 8000 hex ist 8000 hex.

Beispiel

Wenn CIO 000000 im folgenden Beispiel EIN ist, errechnet NEG(160) das Zweierkomplement des Inhalts von D00100 und schreibt das Ergebnis in D00200.

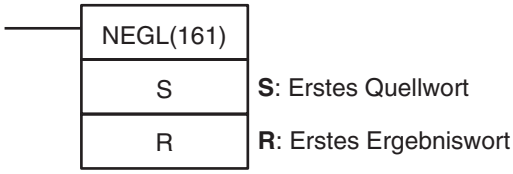


3-12-6 DOUBLE 2'S COMPLEMENT: NEGL(161)

Funktion

Errechnet das Zweierkomplement zweier Worte mit Hexadezimaldaten.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	NEGL(161)
	Ausführung einmal bei steigender Flanke	@NEGL(161)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	

Bereich	S	R
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Hinweis R und R+1 müssen sich im gleichen Datenbereich befinden.

Beschreibung

NEGL(161) errechnet das Zweierkomplement von S+1 und S und schreibt das Ergebnis in R+1. Bei der Zweierkomplementberechnung wird in der Regel der Status der Bits in S+1 und S invertiert und 1 addiert.

Zweierkomplement
(Komplement + 1)

$\overline{(S+1, S)} \longrightarrow (R+1, R)$

Hinweis Diese Funktion (Status der Bits umkehren und 1 addieren) entspricht der Subtraktion des Inhalts von S+1 und S von 0000 0000.

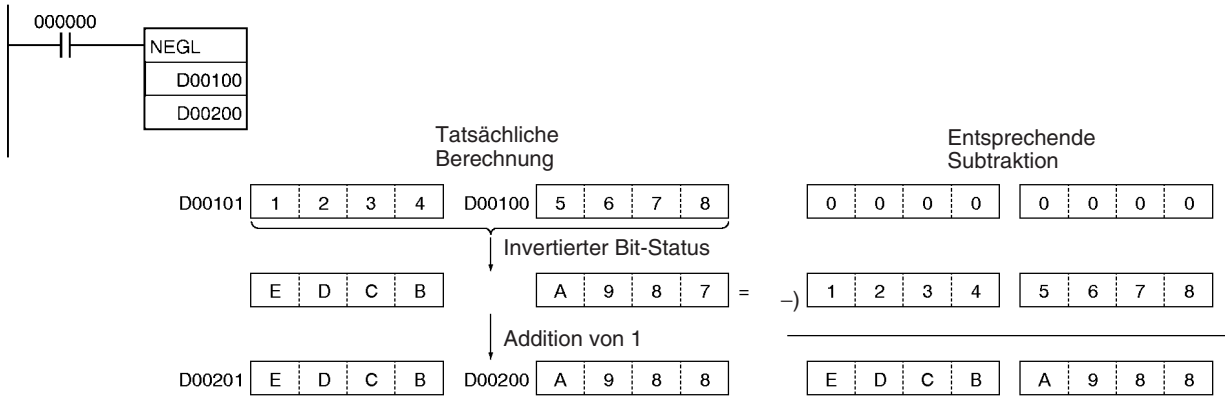
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0000 0000 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn Bit 15 von R+1 EIN ist. In allen anderen Fällen AUS.

Hinweis Das Ergebnis für 8000 hex ist 8000 hex.

Beispiel

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, berechnet NEGL(161) das Zweierkomplement des Inhalts von D00101 und D00100 und schreibt das Ergebnis in D00201 und D00200.

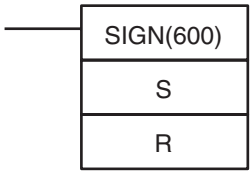


3-12-7 16-BIT TO 32-BIT SIGNED BINARY: SIGN(600)

Funktion

Erweitert einen vorzeichenbehafteten 16-Bit-Binärwert in dessen äquivalenten 32-Bit-Wert.

Kontaktplansymbol



S: Quellwort

R: Erstes Ergebniswort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SIGN(600)
	Ausführung einmal bei steigender Flanke	@SIGN(600)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

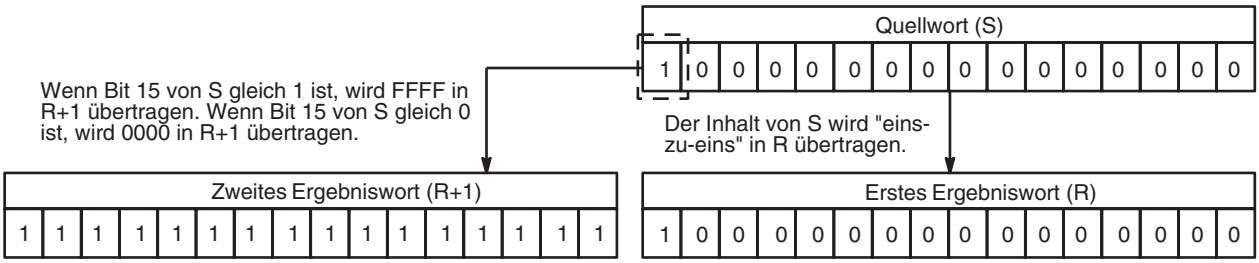
Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W510
Arbeitsbereich remanent	H000 bis H511	H000 bis H510
Systembereich	A000 bis A959	A448 bis A958
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094
Zählerbereich	C0000 bis C4095	C0000 bis C4094
DM-Bereich	D00000 bis D32767	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #FFFF (binär)	---
Datenregister	DR0 bis DR15	---
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Hinweis R und R+1 müssen im gleichen Datenbereich liegen.

Beschreibung

SIGN(600) konvertiert die vorzeichenbehaftete 16-Bit-Binärzahl in S in das vorzeichenbehaftete 32-Bit-Äquivalent und schreibt das Ergebnis in R+1 und R.

Die Konvertierung erfolgt durch Kopieren des Inhalts von S in R und durch Schreiben von FFFF in R+1, wenn Bit 15 von S gleich 1 ist, bzw. durch Schreiben von 0000 in R+1, wenn Bit 15 von S gleich 0 ist.

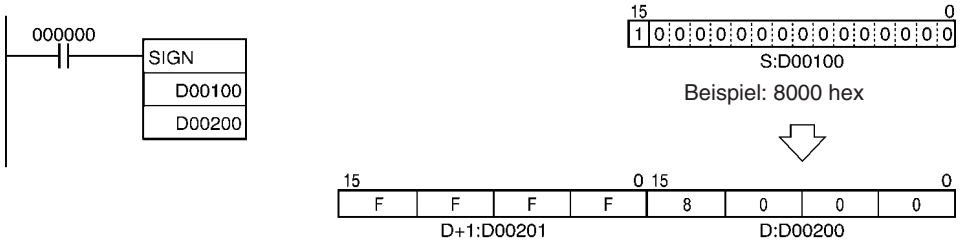


Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS
Gleich-Flag	=	EIN, wenn das Ergebnis 0000 0000 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn Bit 15 von R+1 EIN ist. In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, konvertiert SIGN(600) den vorzeichenbehafteten 16-Bit-Binärinhalt von D00100 (#8000 = –32.768 dezimal) in das 32-Bit-Äquivalent (#FFFF 8000 = –32.768 dezimal) und schreibt das Ergebnis in D00201 und D00200.

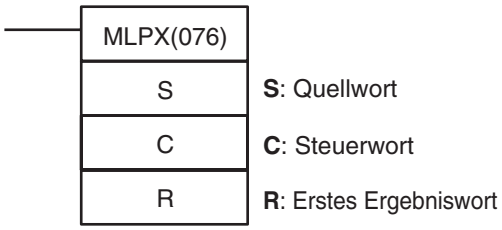


3-12-8 DATA DECODER: MLPX(076)

Funktion

Liest den numerischen Wert der spezifizierten Stelle (bzw. des Bytes) im Quellwort, setzt das entsprechende Bit im Ergebniswort (bzw. in dem 16-Wort-Bereich) auf EIN und setzt alle anderen Bits im Ergebniswort (bzw. in dem 16-Wort-Bereich) auf AUS.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MLPX(076)
	Ausführung einmal bei steigender Flanke	@MLPX(076)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

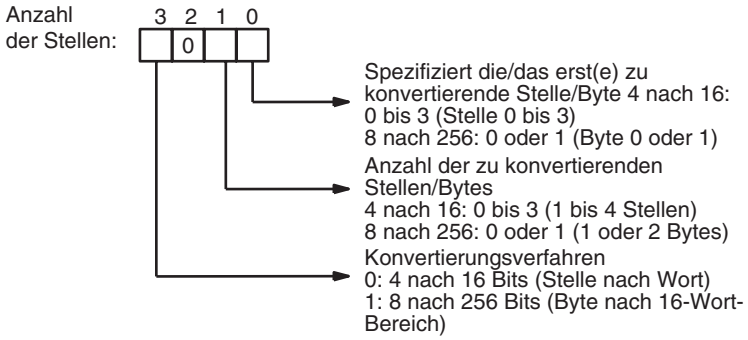
Operanden

S: Quellwort

Die Daten im Quellwort geben die Position des/der Bits an, die auf EIN gesetzt werden.

C: Steuerwort

Das Steuerwort spezifiziert, ob MLPX(076) eine 4-zu-16-Bit-Konvertierung oder eine 8-zu-256-Bit-Konvertierung durchführt, die Anzahl zu konvertierender Stellen oder Bytes und die Anfangsstelle bzw. das Anfangsbyte.



R: Erstes Ergebniswort

Je nach Art des Konvertierungsverfahrens und Anzahl der zu konvertierenden Stellen/Bytes liegt die Anzahl der Ergebnisworte zwischen 1 und 32. Die Ergebnisworte müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	S	C	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	Nur spezifizierte Werte	---
Datenregister	DR0 bis DR15		---

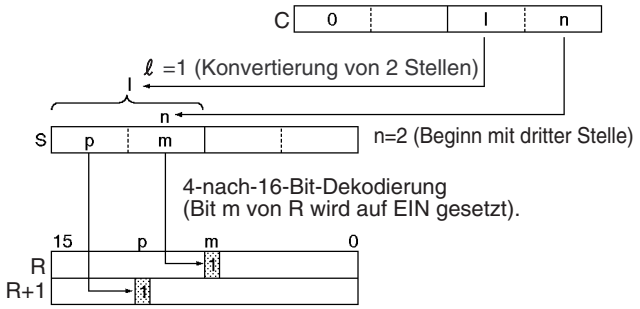
Bereich	S	C	R
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

MLPX(076) kann Konvertierungen von 4 nach 16 Bits oder von 8 nach 256 Bits durchführen. Setzen Sie die höchstwertige Stelle von C auf 0, um eine Konvertierung von 4 nach 16 Bits zu spezifizieren, und setzen Sie sie auf 1, um eine Konvertierung von 8 nach 256 Bits zu spezifizieren.

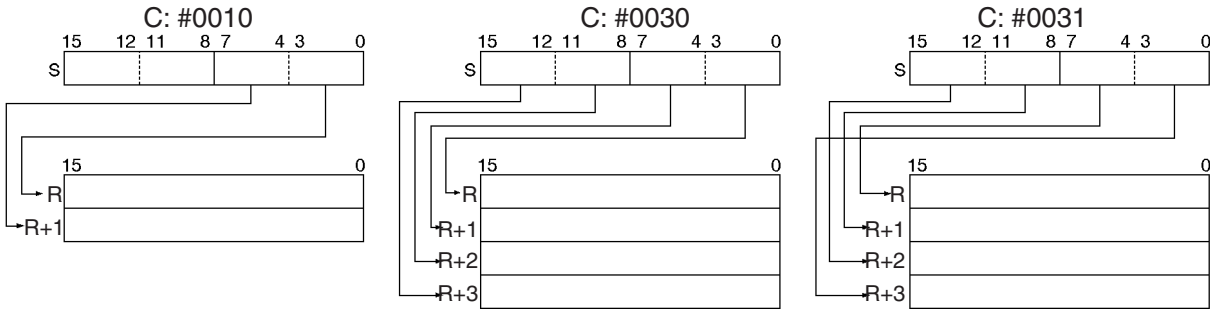
Konvertierung von 4 nach 16 Bits

Wenn die höchstwertige Stelle von C gleich 0 ist, verwendet MLPX(076) den Wert der spezifizierten Stelle in S (0 bis F) und setzt das entsprechende Bit im Ergebniswort auf EIN. Alle anderen Bits im Ergebniswort werden auf AUS gesetzt. Es können bis zu vier Stellen konvertiert werden.



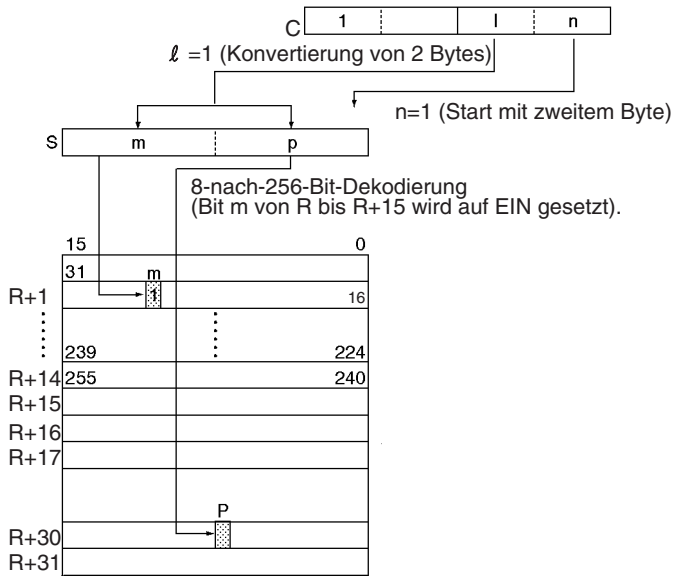
Wenn zwei oder mehr Stellen konvertiert werden, liest MLPX(076) die Stellen in S von rechts nach links und führt ggf. einen Übertrag in die nächste Zeile zur niederwertigsten (rechten) Stelle nach der höchstwertigen Stelle durch.

Die folgende Abbildung zeigt einige Beispielwerte für C und die Konvertierungen von 4 nach 16 Bits, die sie erzeugen.



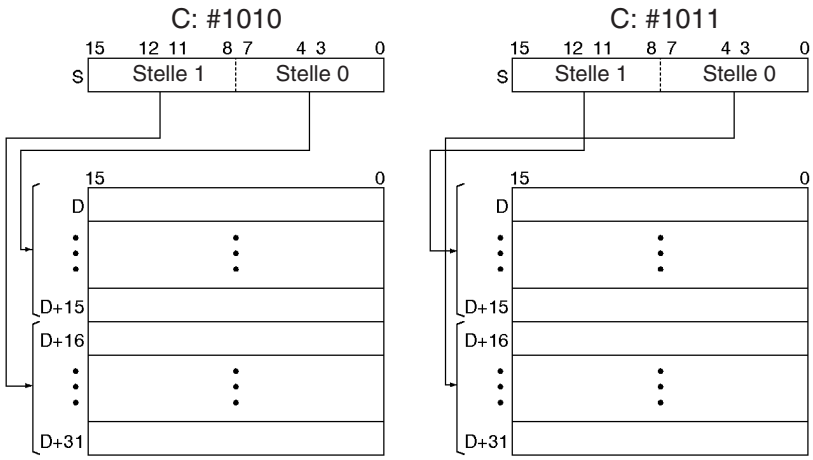
Konvertierung von 8 nach 256 Bits

Wenn die höchstwertige Stelle von C gleich 1 ist, verwendet MLPX(076) den Wert des spezifizierten Bytes in S (00 bis FF) und setzt das entsprechende Bit im Bereich von 16 Ergebnisworten auf EIN. Alle anderen Bits in den Ergebnisworten werden auf AUS gesetzt. Es können bis zu zwei Bytes konvertiert werden.



Wenn zwei Bytes konvertiert werden, liest MLPX(076) die Bytes in S von rechts nach links und führt einen Übertrag in die nächste Zeile zum niederwertigsten Byte durch, wenn das höchstwertige Byte (Byte 1) als Anfangsbyte spezifiziert wurde.

Die folgende Abbildung zeigt einige Beispielwerte für C und die Konvertierungen von 8 nach 256 Bits, die sie erzeugen.



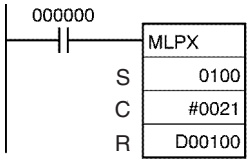
Flags

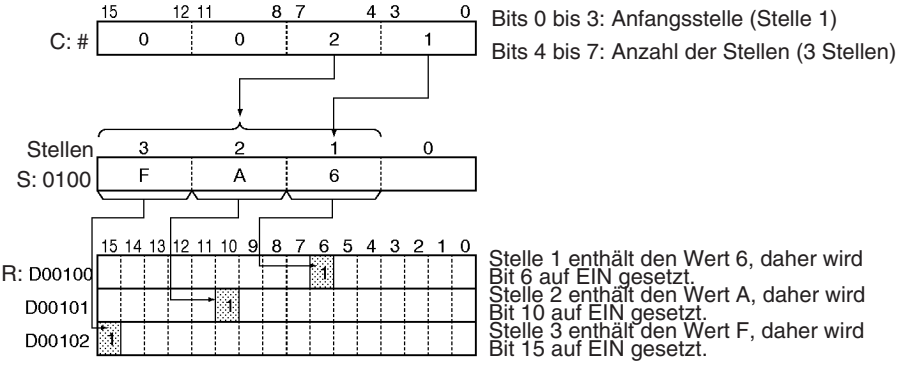
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn C nicht im spezifizierten Bereich liegt. In allen anderen Fällen AUS.

Beispiele

Konvertierung von 4 nach 16 Bits

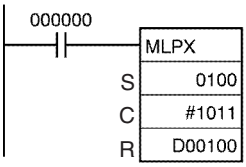
Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, konvertiert MLPX(076) 3 Stellen in S, mit Stelle 1 (die zweite Stelle) beginnend, wie durch C (#0021) spezifiziert. Die entsprechenden Bits in D00100, D00101 und D00102 werden auf EIN gesetzt.

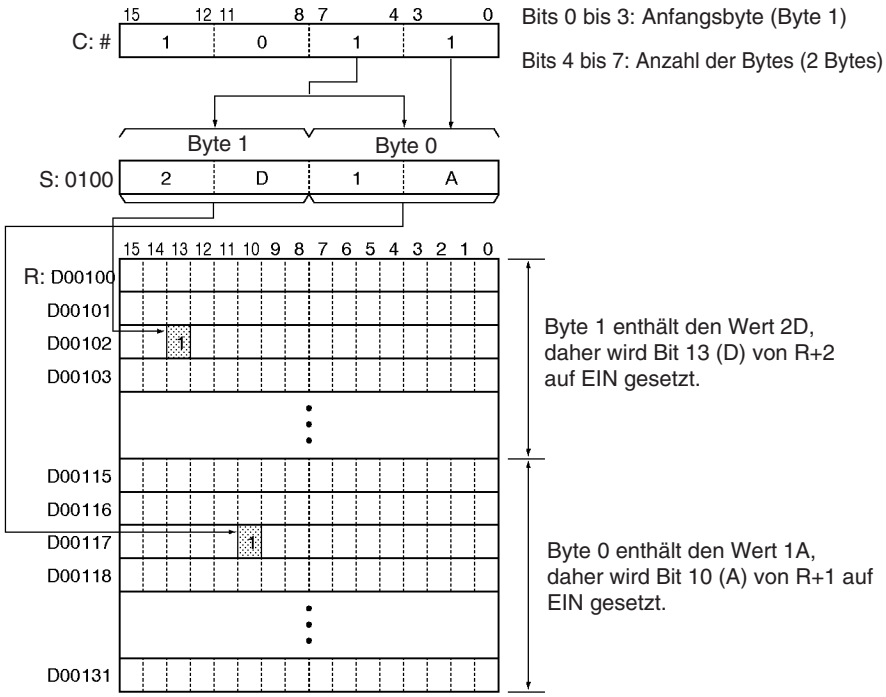




Konvertierung von 8 nach 256 Bits

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, konvertiert MLPX(076) die 2 Bytes in S, mit Byte 1 (die zweite Stelle) beginnend, wie durch C (#0021) spezifiziert. Die entsprechenden Bits in D00100 bis D00115 und D00116 bis D00131 werden auf EIN gesetzt.



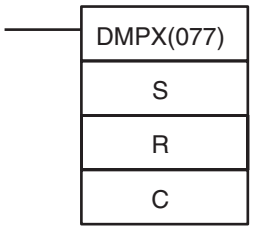


3-12-9 DATA ENCODER: DMPX(077)

Funktion

Sucht den Ort des ersten oder letzten auf EIN gesetzten Bits des Quellworts (bzw. des 16-Worte-Bereichs) und schreibt diesen Wert in die spezifizierte Stelle (bzw. das Byte) im Ergebniswort.

Kontaktplansymbol



S: Erstes Quellwort

R: Ergebniswort

C: Steuerwort

Variat5ionen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	DMPX(077)
	Ausführung einmal bei steigender Flanke	@DMPX(077)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Erstes Quellwort

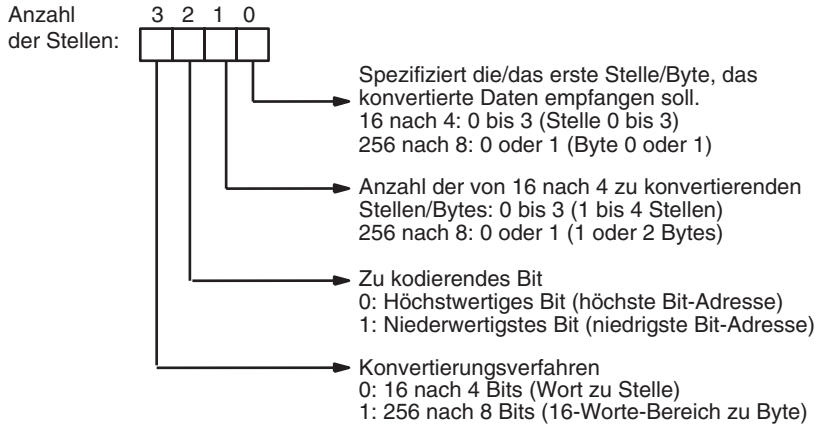
Je nach Art des Konvertierungsverfahrens und Anzahl der zu konvertierenden Stellen/Bytes liegt die Anzahl der Quellworte zwischen 1 und 32. Die Quellworte müssen im gleichen Datenbereich liegen.

R: Ergebniswort

Die Positionen der Bits, die in den Quellworten auf EIN gesetzt sind, werden mit der/dem spezifizierten ersten Stelle/Byte beginnend in die Stellen/Bytes in R geschrieben.

C: Steuerwort

Das Steuerwort spezifiziert, ob DMPX(077) eine Konvertierung von 16 nach 4 Bits oder eine Konvertierung von 256 nach 8 Bits durchführt und ob das höchstwertige oder niederwertigste auf EIN gesetzte Bit kodiert wird. Darüber hinaus spezifiziert es die Anzahl der zu konvertierenden Stellen bzw. Bytes sowie die Anfangsstelle bzw. das Anfangsbyte, ab der/dem das Ergebnis geschrieben wird.



Operanden-Spezifikationen

Bereich	S	R	C
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		

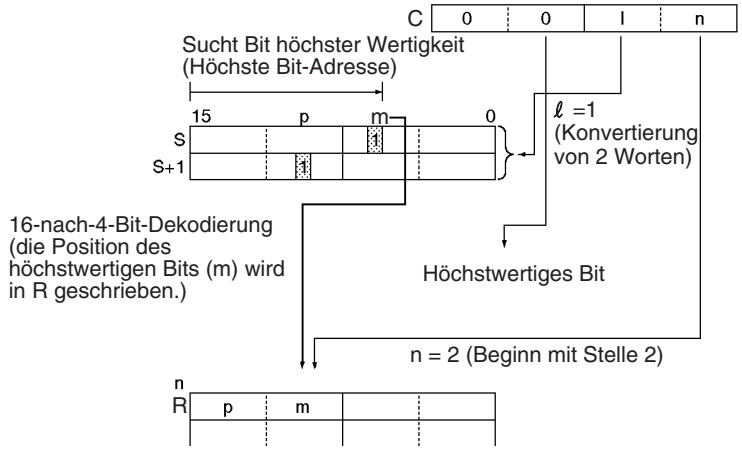
Bereich	S	R	C
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959	A448 bis A959	A000 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	---	Nur spezifizierte Werte
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

DMPX(077) kann Konvertierungen von 16 nach 4 Bits oder von 256 nach 8 Bits durchführen. Setzen Sie die höchstwertige Stelle von C auf 0, um eine Konvertierung von 16 nach 4 Bits zu spezifizieren, und setzen Sie sie auf 1, um eine Konvertierung von 256 nach 8 Bits zu spezifizieren.

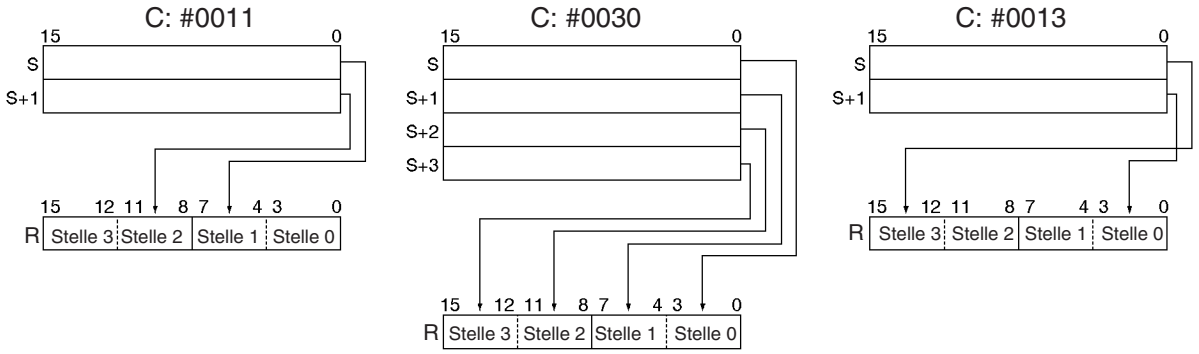
Konvertierung von 16 nach 4 Bits

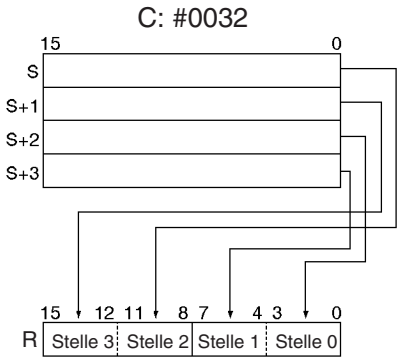
Wenn die vierte (höchstwertige) Stelle von C gleich 0 ist, sucht DMPX(077) die Positionen der höchstwertigen bzw. niederwertigsten auf EIN gesetzten Bits in bis zu 4 Quellworten und schreibt die Positionen, mit der spezifizierten Stelle beginnend, in R. (Setzen Sie die dritte Stelle von C auf 0, um die höchstwertigen auf EIN gesetzten Bits zu suchen, oder setzen Sie sie auf 1, um die niederwertigsten auf EIN gesetzten Bits zu suchen.)



Wenn zwei oder mehr Stellen konvertiert werden, schreibt DMPX(077) die Werte von rechts nach links in die Stellen von R und führt ggf. einen Übertrag in die nächste Zeile zur niederwertigsten (rechten) Stelle nach der höchstwertigen Stelle durch.

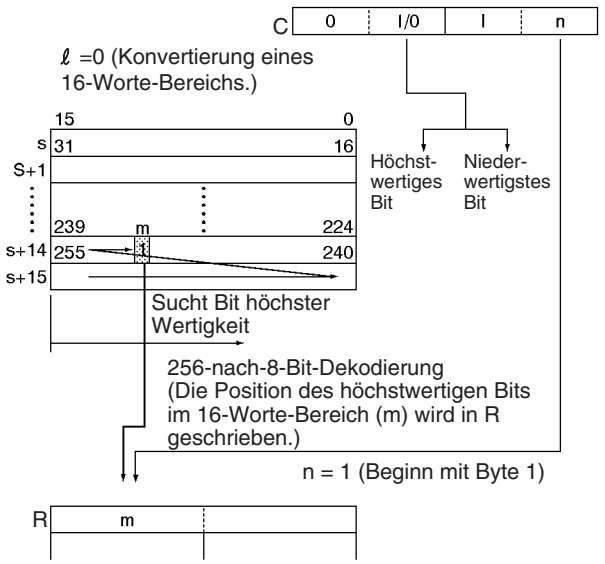
Die folgende Abbildung zeigt einige Beispielwerte für C und die Konvertierungen von 16 nach 4 Bits, die sie erzeugen.





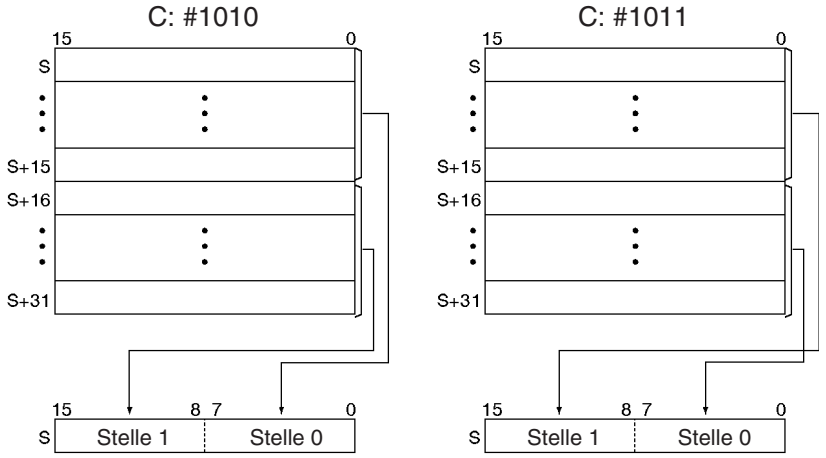
Konvertierung von 256 nach 8 Bits

Wenn die vierte (höchstwertige) Stelle von C gleich 1 ist, sucht DMPX(077) die Positionen der höchstwertigen (höchste Bit-Adresse) bzw. der niederwertigsten (niedrige Bit-Adresse) auf EIN gesetzten Bits in einem oder zwei 16-Worte-Bereichen von Quellworten. Die Positionen dieser Bits werden, mit dem spezifizierten Byte beginnend, in R geschrieben. (Setzen Sie die dritte Stelle von C auf 0, um die höchstwertigen auf EIN gesetzten Bits zu suchen, oder setzen Sie sie auf 1, um die niederwertigsten auf EIN gesetzten Bits zu suchen.)



Wenn zwei Bytes konvertiert werden, schreibt DMPX(077) die Werte von rechts nach links in die Bytes von R und führt einen Übertrag in die nächste Zeile zum niederwertigsten Byte durch, wenn das höchstwertige Byte (Byte 1) als Anfangsbyte spezifiziert wurde.

Die folgende Abbildung zeigt einige Beispielwerte für C und die resultierenden Konvertierungen von 256 nach 8 Bits.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn ein beliebiges Quellwort 0000 hex (d.h. kein Bit gesetzt) enthält. EIN, wenn C nicht im spezifizierten Bereich liegt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

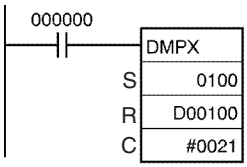
Wenn die Konvertierungsdaten 0000 hex enthalten, andere Daten jedoch zu kodieren sind, führen Sie separate Konvertierungen durch, indem Sie mehr als einen DMPX(077)-Befehl verwenden.

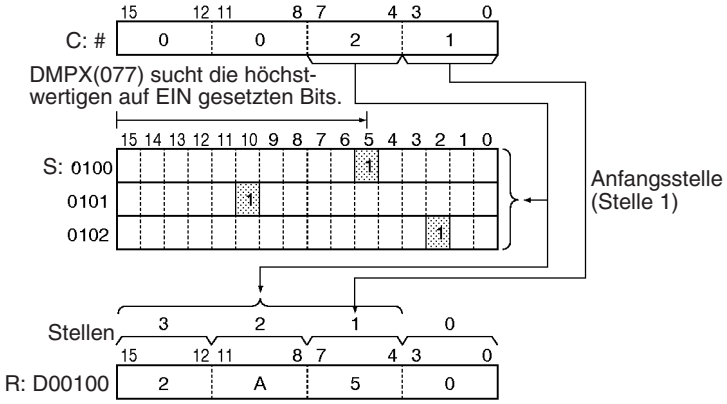
```
DMPX(077)  D0000  D0100  #0300
```

```
DMPX(077)  D0000  D0100  #0000
DMPX(077)  D0001  D0100  #0001
DMPX(077)  D0002  D0100  #0002
DMPX(077)  D0003  D0100  #0003
```

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, sucht DMPX(077) die höchstwertigen auf EIN gesetzten Bits in CIO 0100, CIO 0101 und CIO 0102 und schreibt die Positionen dieser Bits, mit Stelle 1 (der zweiten Stelle) beginnend, in 3 Stellen in R, wie durch C (#0021) spezifiziert.



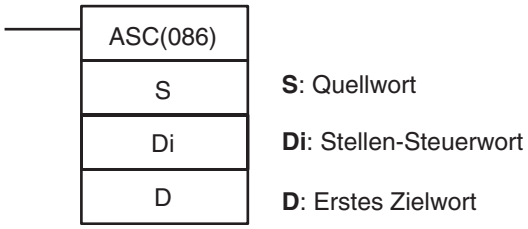


3-12-10 ASCII CONVERT: ASC(086)

Funktion

Konvertiert 4-Bit-Hexadezimalstellen im Quellwort in die entsprechenden 8-Bit-ASCII-Äquivalente.

Kontaktplansymbol



S: Quellwort

Di: Stellen-Steuerwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ASC(086)
	Ausführung einmal bei steigender Flanke	@ASC(086)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

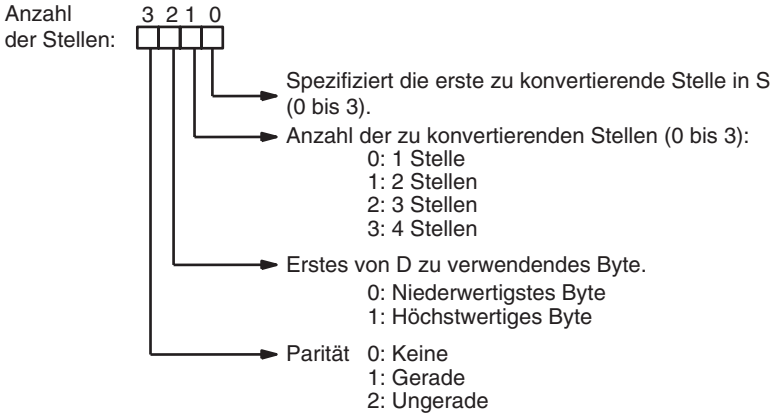
Operanden

S: Quellwort

Es können bis zu vier Stellen im Quellwort konvertiert werden. Die Stellen sind von 0 bis 3, von rechts nach links, nummeriert.

Di: Stellen-Steuerwort

Das Stellen-Steuerwort spezifiziert unterschiedliche Parameter für die Konvertierung, wie in der folgenden Abbildung gezeigt.



D: Erstes Zielwort

Die konvertierten ASCII-Daten werden, mit dem spezifizierten Byte in D beginnend, in das/die Zielwort(e) geschrieben. Wenn 4 Stellen konvertiert werden, sind drei Zielworte (D bis D+3) erforderlich und das höchstwertige Byte wird als erstes Byte in D ausgewählt. Die Zielworte müssen im gleichen Datenbereich liegen.

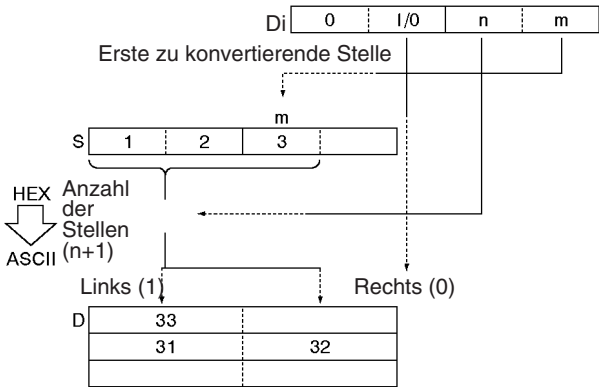
Alle Bytes in Zielworten, die nicht durch ASCII-Daten überschrieben werden, bleiben unverändert.

Operanden-Spezifikationen

Bereich	S	Di	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	Nur spezifizierte Werte	---
Datenregister	DR0 bis DR15		---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

ASC(086) behandelt den Inhalt von S als 4 Hexadezimalstellen, konvertiert die spezifizierten Stellen von S in die entsprechenden 8-Bit-Äquivalente und schreibt diese Daten, mit dem spezifizierten Byte in D beginnend, in das/die Zielworte.



Hinweis Eine Tabelle der erweiterten ASCII-Zeichen finden Sie in *Anhang A* im *CS/CJ-Serie Programmkonsolen Bedienerhandbuch (W341)*.

Parität

Die Parität der zur Fehlerkontrolle während der Datenübertragung verwendeten ASCII-Daten kann spezifiziert werden. Das höchstwertige Bit der jeweiligen ASCII-Zeichen wird automatisch auf gerade, ungerade oder keine Parität angepasst.

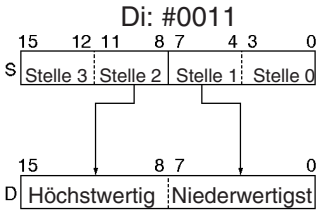
Wenn keine Parität (0) angegeben ist, ist das höchstwertige Bit immer Null. Wenn gerade Parität (1) angegeben ist, wird das höchstwertige Bit immer so angepasst, das die Gesamtzahl der auf EIN gesetzten Bits gerade ist. Wenn ungerade Parität (2) angegeben ist, wird das höchstwertige Bit der einzelnen ASCII-Zeichen so angepasst, dass eine ungerade Anzahl von auf EIN gesetzten Bits vorhanden ist. Der Status des Paritätsbits hat keine Auswirkungen auf die Bedeutung des ASCII-Codes.

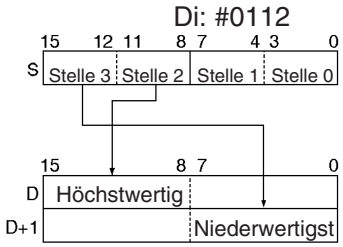
Beispiele für gerade Parität:
Bei Anpassung auf gerade Parität wird ASCII „31“ (00110001) zu „B1“ (10110001: das Paritätsbit ist auf EIN gesetzt, um eine gerade Anzahl auf EIN gesetzter Bits zu erhalten); ASCII „36“ (00110110) bleibt „36“ (00110110: das Paritätsbit bleibt auf AUS gesetzt, da die Anzahl der auf EIN gesetzten Bits bereits gerade ist).

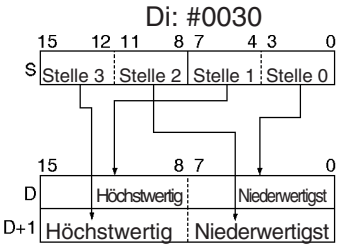
Beispiele für ungerade Parität:
Bei Anpassung auf ungerade Parität wird ASCII „36“ (00110110) zu „B6“ (10110110: das Paritätsbit ist auf EIN gesetzt, um eine ungerade Anzahl auf EIN gesetzter Bits zu erhalten); ASCII „46“ (01000110) bleibt „46“ (01000110: das Paritätsbit bleibt auf AUS gesetzt, da die Anzahl der auf EIN gesetzten Bits bereits ungerade ist).

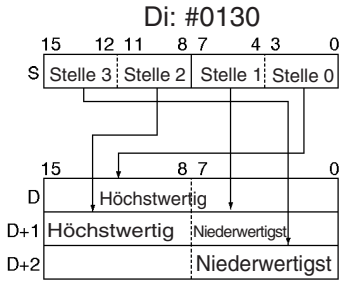
Beispiele für Di

Wenn zwei oder mehr Stellen konvertiert werden, liest ASC(086) die Bytes in S von rechts nach links und führt ggf. einen Übertrag in die nächste Zeile zum niederwertigsten Byte durch. Die folgende Abbildung zeigt einige Beispiele für Di und die Konvertierungen, die daraus resultieren.







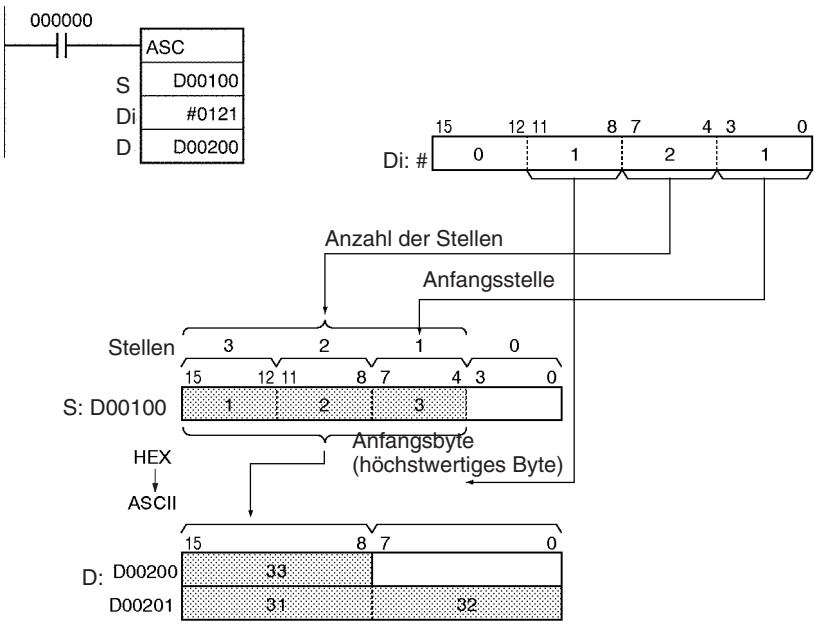


Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von Di nicht in den spezifizierten Bereichen liegt. In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 im folgenden Beispiel EIN ist, konvertiert ASC(086) drei Hexadezimalstellen in D00100 (mit Stelle 1 beginnend) in die entsprechenden ASCII-Äquivalente und schreibt diese Daten, mit dem höchstwertigen Byte in D00200 beginnend, in D00200 und D00201. In diesem Fall spezifiziert ein Stellen-Steuerwort von #0121 keine Parität, das Anfangsbyte (beim Schreiben) = höchstwertiges Byte, die Anzahl der zu lesenden Stellen = 3 und die Anfangsstelle (beim Lesen) = Stelle 1.

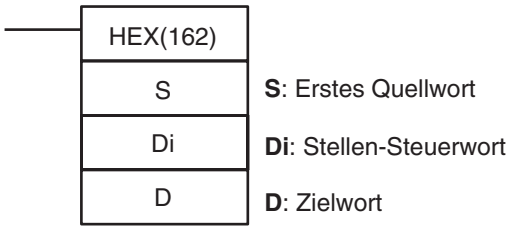


3-12-11 ASCII TO HEX: HEX(162)

Funktion

Konvertiert bis zu 4 Bytes ASCII-Daten im Quellwort in die Hexadezimal-Äquivalente und schreibt diese Stellen in das spezifizierte Zielwort.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	HEX(162)
	Ausführung einmal bei steigender Flanke	@HEX(162)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

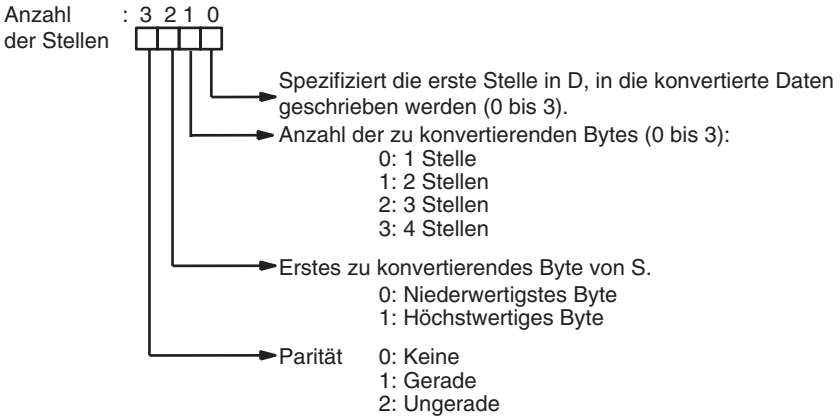
Operanden

S: Erstes Quellwort

Der Inhalt der Quellworte wird als ASCII-Daten behandelt. Es können bis zu drei Quellworte verwendet werden. (Drei Quellworte sind erforderlich, wenn 4 Bytes konvertiert werden und das höchstwertige Byte als erstes Byte in S ausgewählt wird.) Die Quellworte müssen im gleichen Datenbereich liegen.

Di: Stellen-Steuerwort

Das Stellen-Steuerwort spezifiziert unterschiedliche Parameter für die Konvertierung, wie in der folgenden Abbildung gezeigt.



D: Zielwort

Die konvertierten Hexadezimalstellen werden, mit der spezifizierten ersten Stelle beginnend, von rechts nach links in D geschrieben. Alle Stellen im Zielwort, die nicht durch die konvertierten Daten überschrieben werden, bleiben unverändert.

Operanden-Spezifikationen

Bereich	S	Di	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	Nur spezifizierte Werte	---
Datenregister	---	DR0 bis DR15	---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

HEX(162) behandelt den Inhalt des Quellwortes als ASCII-Daten, die Hexadezimalstellen (0 bis 9 und A bis F) darstellen, konvertiert die spezifizierte Anzahl von Bytes in das Hexadezimalformat und schreibt die Hexadezimaldaten, mit der spezifizierten Stelle beginnend, in das Zielwort.

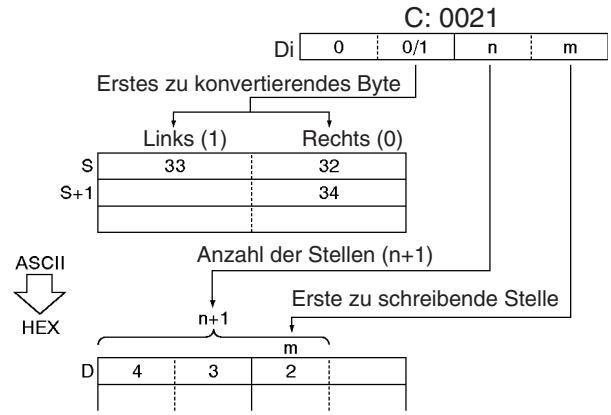
Es tritt ein Fehler auf, wenn die Quellworte Daten enthalten, die kein ASCII-Äquivalent von Hexadezimalstellen sind. Die folgende Tabelle zeigt Hexadezimalstellen und die entsprechenden ASCII-Äquivalente (ausgenommen Paritätsbits).

Flags

Hexadezimalstellen (4 Bits)	ASCII-Äquivalent (2 Hexadezimalstellen)
0 bis 9	30 bis 39
A bis F	41 bis 46

Hinweis Eine Tabelle der erweiterten ASCII-Zeichen entnehmen Sie bitte *Anhang A* des *CS/CJ-Serie Programmkonsolen-Bedienerhandbuchs (W341)*.

In der folgenden Abbildung ist die Grundfunktion von HEX(162) mit Di=0021 dargestellt.



Parität

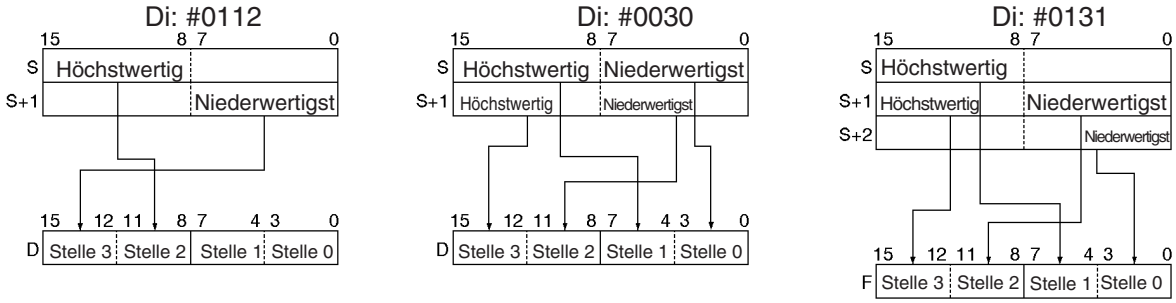
Die Parität der ASCII-Daten kann für die Fehlerüberwachung während der Datenübertragung spezifiziert werden. Das höchstwertige Bit in jedem Byte ist das Paritätsbit. Wenn keine Parität eingestellt ist, muss das Paritätsbit immer Null sein, bei gerader Parität muss der Status des Paritätsbits zu einer geraden Anzahl von auf EIN gesetzten Bits führen und bei ungerader Parität muss der Status des Paritätsbits zu einer ungeraden Anzahl von auf EIN gesetzten Bits führen.

Die folgende Tabelle zeigt die Funktion von HEX(162) für die einzelnen Paritätseinstellungen.

Paritätseinstellung (höchstwertige Stelle von Di)	Funktion von HEX(162)
Keine Parität (0)	HEX(162) wird nur dann ausgeführt, wenn das Paritätsbit in jedem Byte 0 ist. Wenn ein Paritätsbit nicht Null ist, tritt ein Fehler auf.
Gerade Parität (1)	HEX(162) wird nur dann ausgeführt, wenn in jedem Byte eine gerade Anzahl von auf EIN gesetzten Bits vorhanden ist. Wenn ein Byte über eine ungerade Anzahl von auf EIN gesetzten Bits verfügt, tritt ein Fehler auf.
Ungerade Parität (2)	HEX(162) wird nur dann ausgeführt, wenn in jedem Byte eine ungerade Anzahl von auf EIN gesetzten Bits vorhanden ist. Wenn ein Byte über eine gerade Anzahl von auf EIN gesetzten Bits verfügt, tritt ein Fehler auf.

Beispiele für Di

Wenn zwei oder mehr Byte konvertiert werden, schreibt HEX(162) die konvertierten Stellen von rechts nach links in das Zielwort und führt ggf. einen Übertrag in die nächste Zeile zur niederwertigsten Stelle durch. Die folgende Abbildung zeigt einige Beispiele für Di und die Konvertierungen, die daraus resultieren.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn ein Paritätsfehler in den ASCII-Daten vorliegt. EIN, wenn die ASCII-Daten in den Quellworten keine Äquivalente als Hexadezimalstellen haben. EIN, wenn der Inhalt von Di nicht in den spezifizierten Bereichen liegt. In allen anderen Fällen AUS.

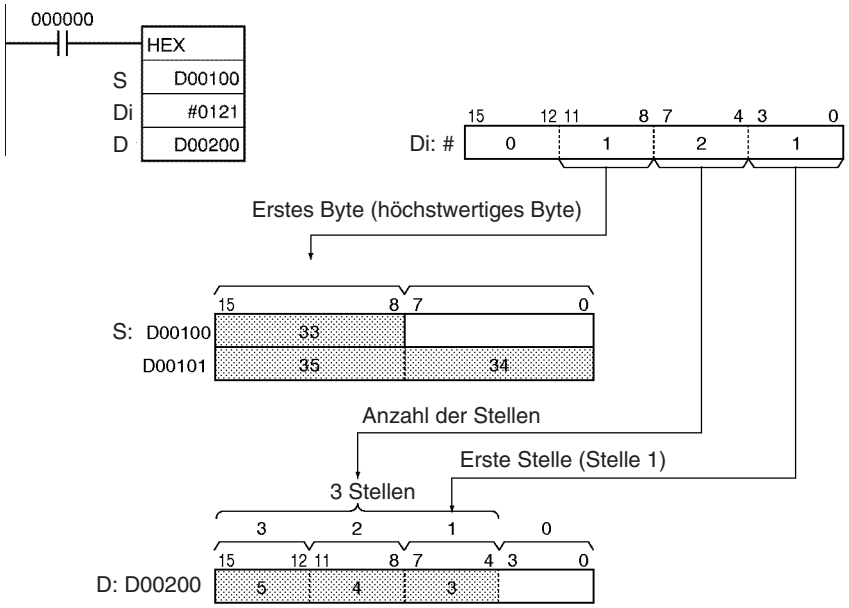
Vorsichtsmaßnahmen

Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn ein Paritätsfehler in den ASCII-Daten vorliegt, die ASCII-Daten in den Quellworten nicht den hexadezimalen Stellen entsprechen oder der Inhalt von Di nicht in den spezifizierten Bereichen liegt.

Beispiele

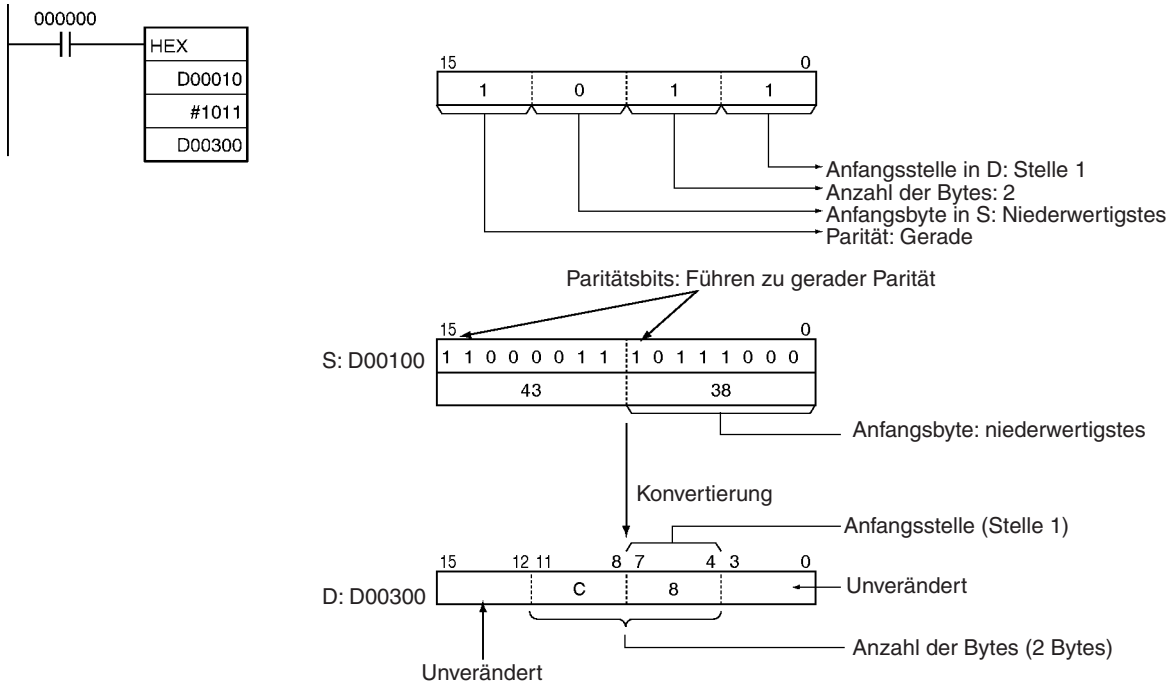
Wenn CIO 000000 im folgenden Beispiel EIN ist, konvertiert HEX(162) die ASCII-Daten in D00100 und D00101 entsprechend den Einstellungen des Steuerwortes. (Di = #0121 spezifiziert keine Parität, das Anfangsbyte (beim Lesen) = höchstwertiges (linkes) Byte, die Anzahl der zu lesenden Bytes = 3 und die Anfangsstelle (beim Schreiben) = Stelle 1.)

HEX(162) konvertiert drei Bytes von ASCII-Daten (3 Zeichen), mit dem höchstwertigen (linken) Byte von D00100 beginnend, in die entsprechenden Hexadezimal-Äquivalente und schreibt diese Daten, mit Stelle 1 beginnend, in D00200.



Wenn CIO 000000 im folgenden Beispiel EIN ist, konvertiert HEX(162) die ASCII-Daten in D00010 mit dem niederwertigsten (rechten) Byte beginnend und schreibt die Hexadezimal-Äquivalente mit Stelle 1 beginnend in D00300.

Das Steuerworts von #1011 spezifiziert die gerade Parität, das Anfangsbyte (beim Lesen) = höchstwertiges (linkes) Byte, die Anzahl der zu lesenden Bytes = 2 und die Anfangsstelle (beim Schreiben) = Stelle 1.

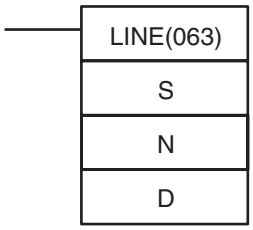


3-12-12 COLUMN TO LINE: LINE(063)

Funktion

Konvertiert eine Spalte von Bits eines 16-Worte-Bereichs (die gleiche Bit-Nummer in 16 aufeinander folgenden Worten) in die 16 Bits des Zielworts.

Kontaktplansymbol



S: Erstes Quellwort

N: Bitnummer

D: Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	LINE(063)
	Ausführung einmal bei steigender Flanke	@LINE(063)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Erstes Quellwort

Spezifiziert das erste Quellwort. S und S+15 müssen im gleichen Datenbereich liegen.

N: Bitnummer

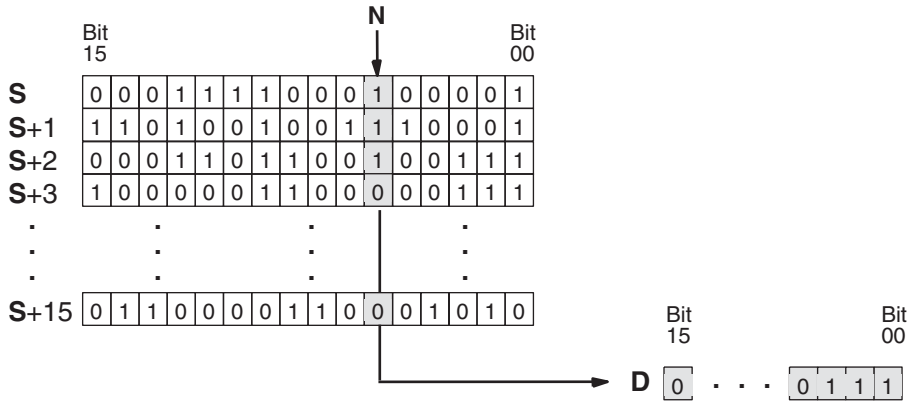
Spezifiziert die Bitnummer (0000 bis 000F oder &0 bis &15), aus den Quellworten kopiert werden soll.

Operanden-Spezifikationen

Bereich	S	N	D
CIO-Bereich	CIO 0000 bis CIO 6128	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W496	W000 bis W511	
Arbeitsbereich remanent	H000 bis H496	H000 bis H511	
Systembereich	A000 bis A944	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4080	T0000 bis T4095	
Zählerbereich	C0000 bis C4080	C0000 bis C4095	
DM-Bereich	D00000 bis D32752	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32752	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32752 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0000 bis 000F (binär) oder &0 bis &15	---
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl LINE(063) kopiert die 16 Bits mit Bitnummer N aus dem 16-Worte-Bereich S bis S+15 in das Zielwort D. Bit N von S+m wird in Bit m von D kopiert, d.h. Bit N von S wird in Bit 00 von D und Bit N von S+15 in Bit 15 von D kopiert.

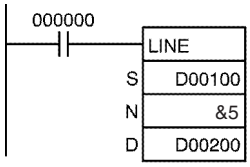


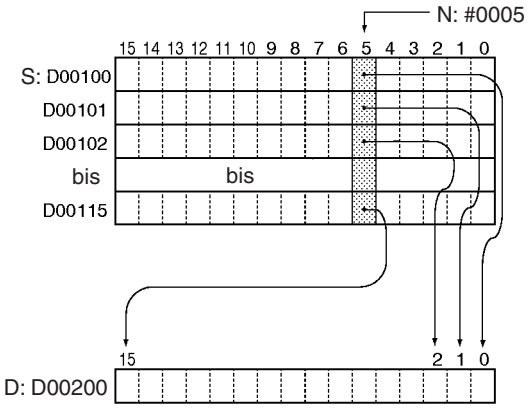
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N nicht im spezifizierten Bereich von 0000 bis 000F liegt. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn D nach der Ausführung 0000 ist. In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 im folgenden Beispiel EIN ist, kopiert LINE(063) das Bit 5 von D00100 bis D00115 in die 16 Bits von D00200.



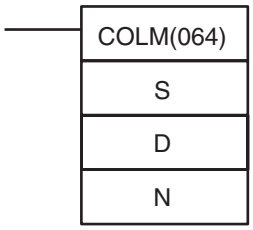


3-12-13 LINE TO COLUMN: COLM(064)

Funktion

Konvertiert die 16 Bits eines Quellwortes in eine Bitspalte in einen 16-Worte-Bereich der Zielworte (gleiche Bitnummer in 16 aufeinander folgenden Worten).

Kontaktplansymbol



S: Quellwort

D: Erstes Zielwort

N: Bitnummer

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	COLM(064)
	Ausführung einmal bei steigender Flanke	@COLM(064)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

D: Erstes Zielwort

Spezifiziert das erste Zielwort. D und D+15 müssen im gleichen Datenbereich liegen.

N: Bitnummer

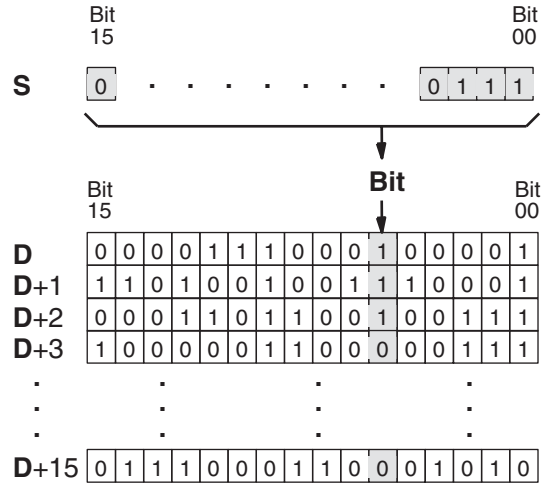
Spezifiziert die Bitnummern (0000 bis 000F oder &0 bis &15), die mit den Quellwort-Bits überschrieben werden sollen.

Operanden-Spezifikationen

Bereich	S	D	N
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6128	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W496	W000 bis W511
Arbeitsbereich remanent	H000 bis H511	H000 bis H496	H000 bis H511
Systembereich	A000 bis A959	A448 bis A944	A000 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4080	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4080	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32752	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32752	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32752 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	#0000 bis #000F (binär) oder &0 bis &15
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

COLM(064) kopiert die 16 Bits aus S in die 16 Bits mit der Bitnummer N im 16-Worte-Bereich D bis D+15. Bit m von S wird in Bit N von D+m kopiert, d.h. Bit 00 von S wird in Bit N von D kopiert und Bit 15 von S wird in Bit N von D+15 kopiert.

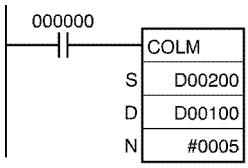


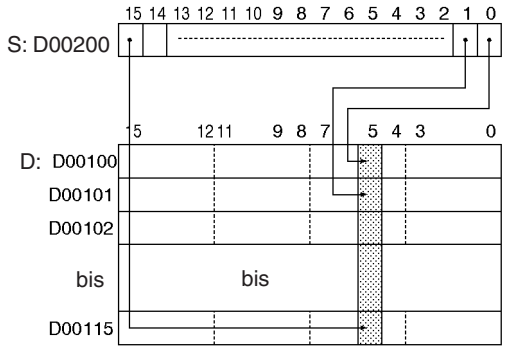
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N nicht im spezifizierten Bereich von 0000 bis 000F liegt. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn Bit N nach der Ausführung in allen 16 Worten von D bis D+15 gleich 0 ist. In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 im folgenden Beispiel EIN ist, kopiert COLM(064) die 16 Bits in D00200 (Bits 00 bis 15) in Bit 5 von D00100 bis D00115.



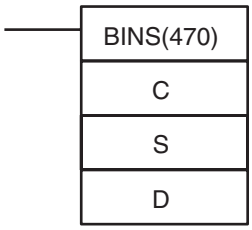


3-12-14 SIGNED BCD-TO-BINARY: BINS(470)

Funktion

Konvertiert ein Wort vorzeichenbehafteter BCD-Daten in ein Wort vorzeichenbehafteter Binärdaten.

Kontaktplansymbol



C: Steuerwort

S: Quellwort

D: Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	BINS(470)
	Ausführung einmal bei steigender Flanke	@BINS(470)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

Spezifiziert das vorzeichenbehaftete BCD-Format. C muss 0000 bis 0003 sein.

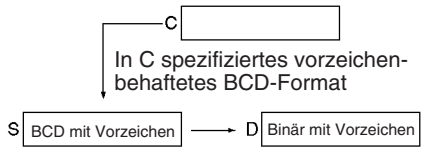
Operanden-Spezifikationen

Bereich	C	S	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #0003 (binär)	---	
Datenregister	DR0 bis DR15		

Bereich	C	S	D
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

BINS(470) konvertiert vorzeichenbehaftete BCD-Daten in vorzeichenbehaftete Binärdaten. Zuerst werden das Format und der Bereich der vorzeichenbehafteten BCD-Daten im Wort S mit der Einstellung im Steuerwort (C) überprüft. Wenn die Quelldaten korrekt sind, werden die vorzeichenbehafteten BCD-Daten in S in vorzeichenbehaftete Binärdaten konvertiert und in D ausgegeben. Wenn die Quelldaten nicht korrekt sind, wird das Fehler-Flag auf EIN gesetzt und der Befehl nicht ausgeführt.



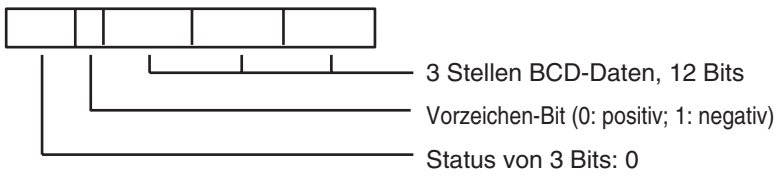
Wenn die konvertierten Daten negativ sind, werden sie als Zweierkomplement ausgegeben und das Negativ-Flag wird auf EIN gesetzt. NEG(160) kann verwendet werden, um den Absolutwert einer mit einem negativen Vorzeichen behafteten Binärzahl zu bestimmen. Detaillierte Informationen hierzu finden Sie unter *3-12-52'S COMPLEMENT: NEG(160)*.

Der Wert –0 in den Quelldaten wird als 0 behandelt und führt nicht zu einem Fehler. Der Status der Bits 13 bis 15 von S wird nicht überprüft, wenn C = 0000 ist.

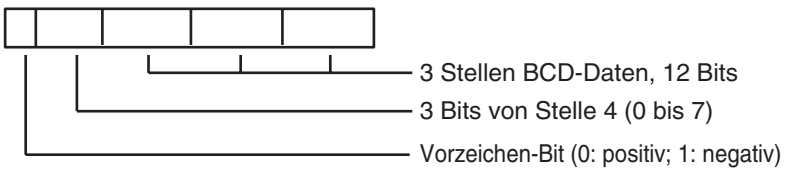
Hinweis Einige Spezial-E/A-Baugruppen geben vorzeichenbehaftete BCD-Daten aus. Berechnungen, mit diesen Daten sind in der Regel einfacher, wenn sie zuerst mit BINS(470) in vorzeichenbehaftete Binärdaten konvertiert werden.

Das Steuerwort spezifiziert das vorzeichenbehaftete BCD-Format (siehe unten).

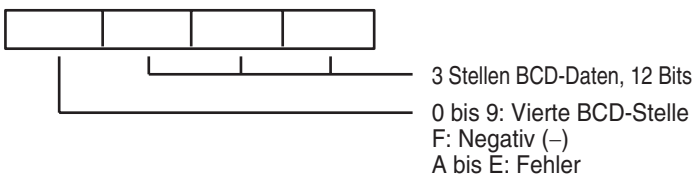
C = 0000 (Eingangsdatenbereich: -999 bis 999 BCD)



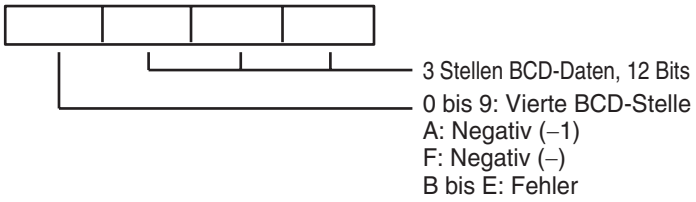
C = 0001 (Eingangsdatenbereich: -7999 bis 7999 BCD)



C = 0002 (Eingangsdatenbereich: -999 bis 9999 BCD)



C = 0003 (Eingangsdatenbereich: -1999 bis 9999 BCD)



Die folgende Tabelle zeigt die möglichen BCD-Werte für jedes vorzeichenbehaftete BCD-Format und die entsprechenden vorzeichenbehafteten Binärwerte.

Einstellung	Vorzeichenbehaftete BCD-Werte	Vorzeichenbehaftete Binärwerte
C = 0000	-999 bis –1 und 0 bis 999	FC19 bis FFFF und 0000 bis 03E7
C = 0001	-7999 bis –1 und 0 bis 7999	E0C1 bis FFFF und 0000 bis 1F3F
C = 0002	-999 bis –1 und 0 bis 9999	FC19 bis FFFF und 0000 bis 270F
C = 0003	-1999 bis –1 und 0 bis 9999	F831 bis FFFF und 0000 bis 270F

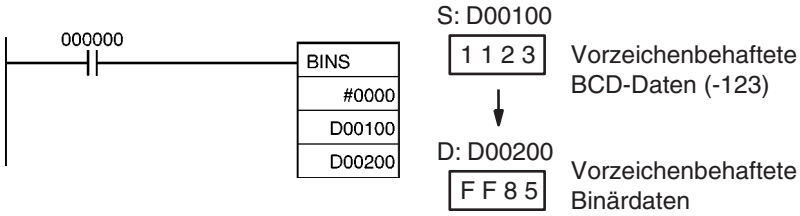
Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn C nicht im spezifizierten Bereich von 0000 bis 0003 liegt. EIN, wenn C = 0002 und die höchstwertige (linke) Stelle von S gleich A bis E ist. EIN, wenn C = 0003 und die höchstwertige (linke) Stelle von S gleich B bis E ist. EIN, wenn der Inhalt von S kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn D nach der Ausführung 0000 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn Bit 15 von D nach der Ausführung auf EIN gesetzt ist. In allen anderen Fällen AUS.

Beispiele

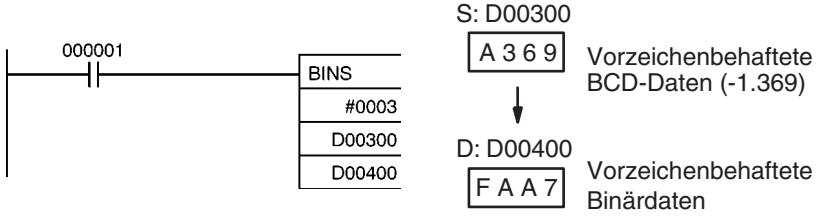
BCD-Format 0 (C = #0000)

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden das Format und der Bereich der vorzeichenbehafteten BCD-Daten in D00100 mit dem im Steuerwort (0000) spezifizierten Format abgeglichen. Die Quelldaten sind korrekt, daher werden die vorzeichenbehafteten BCD-Daten in D00100 in vorzeichenbehaftete Binärwerte konvertiert und in D00200 ausgegeben.



BCD-Format 0 (C = #0003)

Wenn CIO 00001 im folgenden Beispiel auf EIN gesetzt ist, werden das Format und der Bereich der vorzeichenbehafteten BCD-Daten in D00100 mit dem im Steuerwort (0003) spezifizierten Format abgeglichen. Die Quelldaten sind korrekt, daher werden die vorzeichenbehafteten BCD-Daten in D00300 in vorzeichenbehaftete Binärwerte konvertiert und in D00400 ausgegeben.

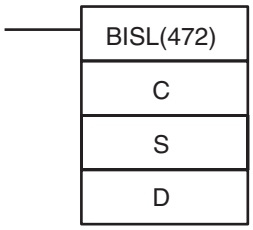


3-12-15 DOUBLE SIGNED BCD-TO-BINARY: BISL(472)

Funktion

Konvertiert vorzeichenbehaftete BCD-Doppelwort-Daten zu vorzeichenbehafteten Binär-Doppelwort-Daten.

Kontaktplansymbol



C: Steuerwort

S: Erstes Quellwort

D: Erstes Zielwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	BISL(472)
	Ausführung einmal bei steigender Flanke	@BISL(472)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

Spezifiziert das Format der vorzeichenbehafteten BCD-Daten. C muss 0000 bis 0003 sein.

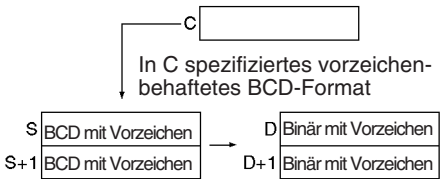
Operanden-Spezifikationen

Bereich	C	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W510	
Arbeitsbereich remanent	H000 bis H511	H000 bis H510	
Systembereich	A000 bis A959	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094	
Zählerbereich	C0000 bis C4095	C0000 bis C4094	
DM-Bereich	D00000 bis D32767	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #0003 (binär)	---	
Datenregister	DR0 bis DR15	---	

Bereich	C	S	D
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

BISL(472) konvertiert die vorzeichenbehafteten BCD-Doppelwort-Daten in S+1 und S in vorzeichenbehaftete Binär-Doppelwort-Daten und schreibt das Ergebnis in D+1 und D. Zuerst werden das Format und der Bereich der vorzeichenbehafteten BCD-Daten in den Worten S+1 und S mit der Einstellung im Steuerwort (C) verglichen. Wenn die Quelldaten korrekt sind, werden die vorzeichenbehafteten BCD-Daten in S+1 und S in vorzeichenbehaftete Binärdaten konvertiert und in D+1 und D ausgegeben. Wenn die Quelldaten nicht korrekt sind, wird das Fehler-Flag auf EIN gesetzt und der Befehl nicht ausgeführt.



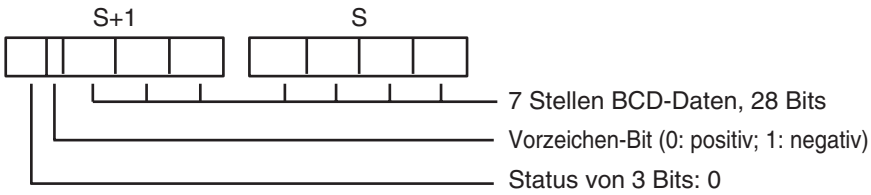
Wenn die konvertierten Daten negativ sind, werden sie als Zweierkomplement ausgegeben und das Negativ-Flag wird auf EIN gesetzt. NEGL(161) kann verwendet werden, um den Absolutwert einer negativen vorzeichenbehafteten Binär-Doppelwort-Zahl zu bestimmen. Detaillierte Informationen hierzu finden Sie unter *3-12-6 DOUBLE 2'S COMPLEMENT: NEGL(161)*.

Der Wert –0 in den Quelldaten wird als 0 behandelt und führt nicht zu einem Fehler. Der Status der Bits 13 bis 15 von S+1 wird nicht überprüft, wenn C = 0000.

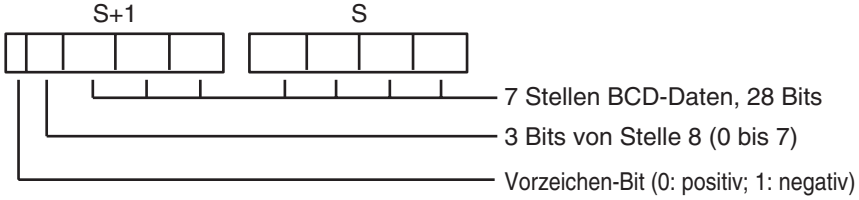
Hinweis Einige Spezial-E/A-Baugruppen geben vorzeichenbehaftete BCD-Daten aus. Berechnungen, bei denen diese Daten verwendet werden, können normalerweise leichter durchgeführt werden, wenn sie zuerst mit BISL(472) in vorzeichenbehaftete Binärdaten konvertiert werden.

Das Steuerwort spezifiziert das vorzeichenbehaftete BCD-Format (siehe unten).

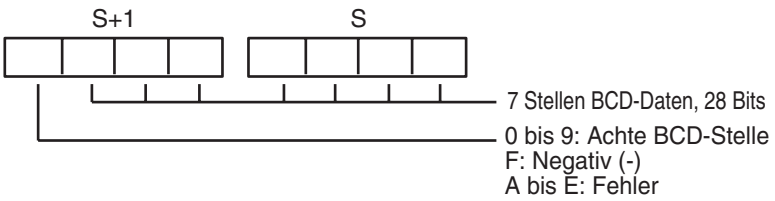
C = 0000 (Eingangsdatenbereich: -999 9999 bis 999 9999 BCD)



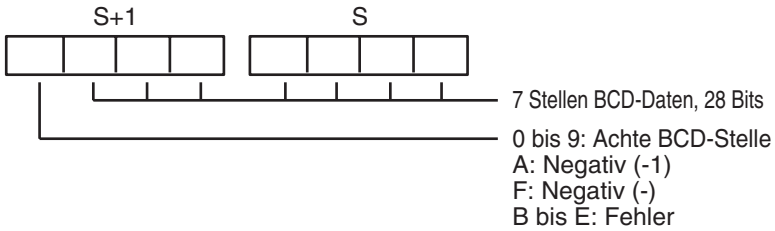
C = 0001 (Eingangsdatenbereich: -7999 9999 bis 7999 9999 BCD)



C = 0002 (Eingangsdatenbereich: -999 9999 bis 7999 9999 BCD)



C = 0003 (Eingangsdatenbereich: -1999 9999 bis 7999 9999 BCD)



Die folgende Tabelle zeigt die möglichen BCD-Werte für jedes vorzeichenbehaftete BCD-Format und die entsprechenden vorzeichenbehafteten Binärwerte.

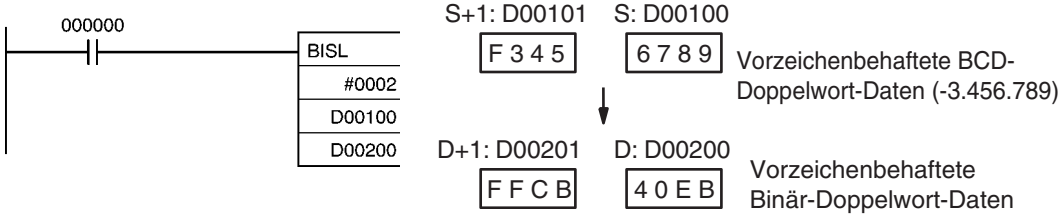
Einstellung	Vorzeichenbehaftete BCD-Werte	Vorzeichenbehaftete Binärwerte
C = 0000	-999 9999 bis –1	FF67 6981 bis FFFF FFFF
	0 bis 999 9999	0000 0000 bis 0098 967F
C = 0001	-7999 9999 bis –1	FB3B 4C01 bis FFFF FFFF
	0 bis 7999 9999	0000 0000 bis 04C4 B3FF
C = 0002	-999 9999 bis –1	FF67 6981 bis FFFF FFFF
	0 bis 999 9999	0000 0000 bis 05F5 E0FF
C = 0003	-1999 9999 bis –1	FECE D301 bis FFFF FFFF
	0 bis 999 9999	0000 0000 bis 05F5 E0FF

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn C nicht im spezifizierten Bereich von 0000 bis 0003 liegt. EIN, wenn C = 0002 und die höchstwertige (linke) Stelle von S+1 gleich A bis E ist. EIN, wenn C = 0003 und die höchstwertige (linke) Stelle von S+1 gleich B bis E ist. EIN, wenn der Inhalt von S+1 kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn D+1 nach der Ausführung 0000 0000 enthält. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn Bit 15 von D+1 nach der Ausführung auf EIN gesetzt ist. In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 im folgenden Beispiel EIN ist, werden das vorzeichenbehaftete BCD-Doppelwort-Format und der Bereich D00101 und D00100 mit dem im Steuerwort (0002) spezifizierten Format verglichen. Die Quelldaten sind korrekt, daher werden die vorzeichenbehafteten BCD-Doppelwort-Daten in D00101 und D00100 in vorzeichenbehaftete Binär-Doppelwort-Werte konvertiert und in D00201 und D00200 ausgegeben.

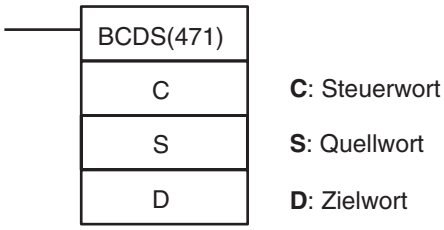


3-12-16 SIGNED BINARY-TO-BCD: BCDS(471)

Funktion

Konvertiert ein Wort vorzeichenbehaftete Binärdaten in ein Wort vorzeichenbehaftete BCD-Daten.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	BCDS(471)
	Ausführung einmal bei steigender Flanke	@BCDS(471)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operand

C: Steuerwort

Spezifiziert das Format der vorzeichenbehafteten BCD-Daten. C muss 0000 bis 0003 sein.

S: Quellwort

Enthält die zu konvertierenden vorzeichenbehafteten Binärdaten. Der Inhalt von S muss im gültigen Bereich des in C spezifizierten BCD-Formats liegen.

Einstellung	Gültige Werte für S
C = 0000	FC19 bis FFFF oder 0000 bis 03E7
C = 0001	E0C1 bis FFFF oder 0000 bis 1F3F
C = 0002	FC19 bis FFFF oder 0000 bis 270F
C = 0003	F831 bis FFFF oder 0000 bis 270F

D: Zielwort

Enthält die konvertierten vorzeichenbehafteten BCD-Daten. Eine Erläuterung der BCD-Formate finden Sie im Abschnitt unten.

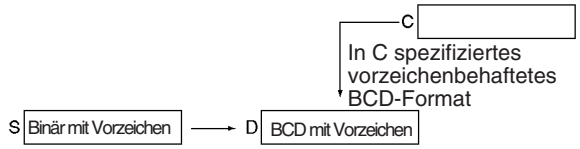
Operanden-Spezifikationen

Bereich	C	S	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		

Bereich	C	S	D
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #0003 (binär)	---	
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis 1–2048 bis +2047 ,IR5 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

BCDS(471) konvertiert vorzeichenbehaftete Binärdaten in vorzeichenbehaftete BCD-Daten. Zuerst wird überprüft, ob die vorzeichenbehafteten Binärdaten in Wort S im gültigen Bereich für das im Steuerwort (C) spezifizierte vorzeichenbehaftete BCD-Format liegen. Wenn die Quelldaten korrekt sind, werden die vorzeichenbehafteten Binärdaten in S in vorzeichenbehaftete BCD-Daten konvertiert und in D ausgegeben. Wenn die Quelldaten nicht korrekt sind, wird das Fehler-Flag auf EIN gesetzt und der Befehl nicht ausgeführt.

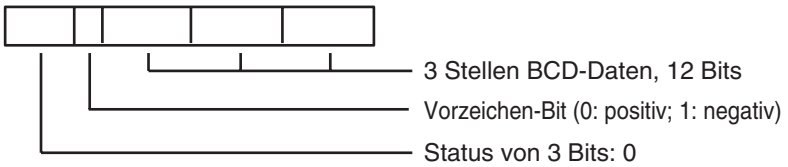


Hinweis

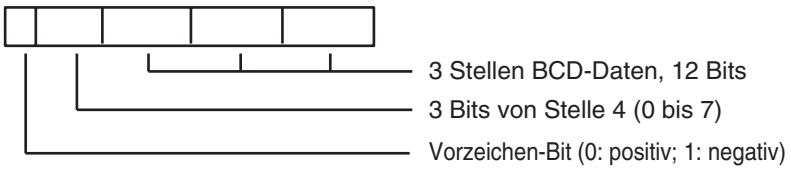
1. Der Wert –0 in den Quelldaten wird als 0 behandelt und führt nicht zu einem Fehler.
2. Einige Spezial-E/A-Baugruppen erfordern vorzeichenbehaftete BCD-Daten als Eingabe. BCDS(471) kann verwendet werden, um vorzeichenbehaftete Binärdaten für die Ausgabe an diese Baugruppen zu konvertieren.

Das Steuerwort spezifiziert das vorzeichenbehaftete BCD-Format, das für das Ergebnis verwendet wird (siehe unten).

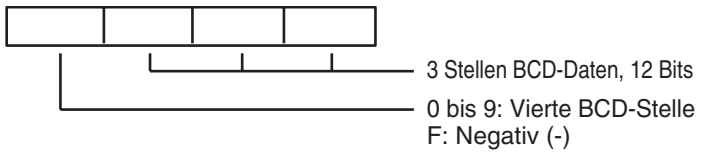
C = 0000 (Ausgangsdatenbereich: -999 bis 999 BCD)



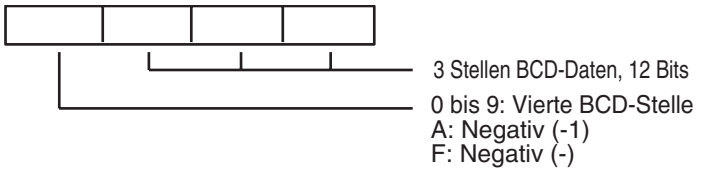
C = 0001 (Ausgangsdatenbereich: -7999 bis 7999 BCD)



C = 0002 (Ausgangsdatenbereich: -999 bis 9999 BCD)



C = 0003 (Ausgangsdatenbereich: -1999 bis 9999 BCD)



Die folgende Tabelle zeigt die möglichen vorzeichenbehafteten Binärwerte für jedes vorzeichenbehaftete BCD-Format. Es tritt ein Fehler auf, wenn die Quelldaten nicht im gültigen Bereich für das spezifizierte vorzeichenbehaftete BCD-Format liegen.

Einstellung	Vorzeichenbehaftete Binärwerte	Vorzeichenbehaftete BCD-Werte
C = 0000	FC19 bis FFFF und 0000 bis 03E7	-999 bis –1 und 0 bis 999
C = 0001	E0C1 bis FFFF und 0000 bis 1F3F	-7999 bis –1 und 0 bis 7999
C = 0002	FC19 bis FFFF und 0000 bis 270F	-999 bis –1 und 0 bis 9999
C = 0003	F831 bis FFFF und 0000 bis 270F	-1999 bis –1 und 0 bis 9999

Flags

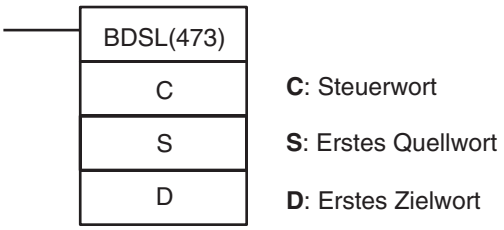
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn C nicht im spezifizierten Bereich von 0000 bis 0003 liegt. EIN, wenn C = 0000 ist und die Quelldaten nicht in den gültigen Bereichen (FC19 bis FFFF oder 0000 bis 03E7) liegen. EIN, wenn C = 0001 ist und die Quelldaten nicht in den gültigen Bereichen (E0C1 bis FFFF oder 0000 bis 1F3F) liegen. EIN, wenn C = 0002 ist und die Quelldaten nicht in den gültigen Bereichen (FC19 bis FFFF oder 0000 bis 270F) liegen. EIN, wenn C = 0003 ist und die Quelldaten nicht in den gültigen Bereichen (F831 bis FFFF oder 0000 bis 270F) liegen. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn D nach der Ausführung 0000 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn C = 0000 oder 0001 ist und das Vorzeichen-Bit des Ergebnisses nach der Ausführung auf EIN gesetzt ist. EIN, wenn C = 0002 und die höchstwertige Stelle des Ergebnisses F ist. EIN, wenn C = 0003 und die höchstwertige Stelle des Ergebnisses A oder F ist. In allen anderen Fällen AUS.

3-12-17 DOUBLE SIGNED BINARY-TO-BCD: BDSL(473)

Funktion

Konvertiert vorzeichenbehaftete Binär-Doppelwort-Daten zu vorzeichenbehafteten BCD-Doppelwort-Daten.

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	BDSL(473)
	Ausführung einmal bei steigender Flanke	@BDSL(473)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Operanden

C: Steuerwort

Spezifiziert das Format der vorzeichenbehafteten BCD-Daten. C muss 0000 bis 0003 sein.

S: Erstes Quellwort

Die Quellworte S+1 und S enthalten zu konvertierende vorzeichenbehaftete Binär-Doppelwort-Daten. Deren Inhalt muss im gültigen Bereich für das in C spezifizierte BCD-Format liegen.

Einstellung	Gültige Werte für S+1 und S
C = 0000	FF67 6981 bis FFFF FFFF oder 0000 0000 bis 0098 967F
C = 0001	FB3B 4C01 bis FFFF FFFF oder 0000 0000 bis 04C4 B3FF
C = 0002	FF67 6981 bis FFFF FFFF oder 0000 0000 bis 05F5 E0FF
C = 0003	FECE D301 bis FFFF FFFF oder 0000 0000 bis 05F5 E0FF

D: Erstes Zielwort

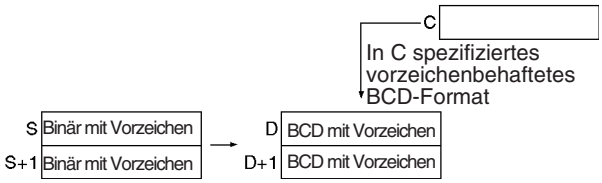
Die Zielworte D+1 und D enthalten die konvertierten vorzeichenbehafteten BCD-Doppelwort-Daten. Eine Erläuterung der BCD-Formate finden Sie im Abschnitt unten.

Operanden-Spezifikationen

Bereich	C	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W510	
Arbeitsbereich remanent	H000 bis H511	H000 bis H510	
Systembereich	A000 bis A959	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094	
Zählerbereich	C0000 bis C4095	C0000 bis C4094	
DM-Bereich	D00000 bis D32767	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767(n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767(n = 0 bis C)		
Konstanten	#0000 bis #0003 (binär)	---	
Datenregister	DR0 bis DR15	---	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

BDSL(473) konvertiert vorzeichenbehaftete Binär-Doppelwort-Daten zu vorzeichenbehafteten BCD-Doppelwort-Daten. Zuerst wird überprüft, ob die vorzeichenbehafteten Binär-Doppelwort-Daten in den Worten S+1 und S im gültigen Bereich für das im Steuerwort (C) spezifizierte vorzeichenbehaftete BCD-Format liegen. Wenn die Quelldaten korrekt sind, werden die vorzeichenbehafteten Binär-Doppelwort-Daten in S+1 und S in vorzeichenbehaftete Binärdaten konvertiert und in D+1 und D ausgegeben. Wenn die Quelldaten nicht korrekt sind, wird das Fehler-Flag auf EIN gesetzt und der Befehl nicht ausgeführt.

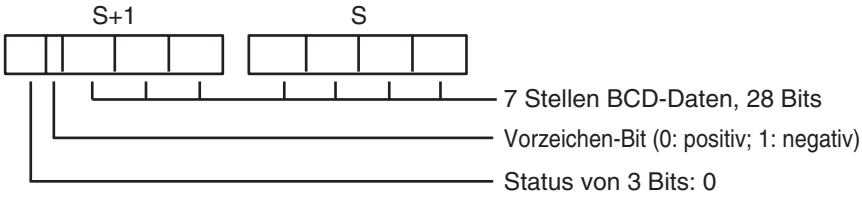


Hinweis

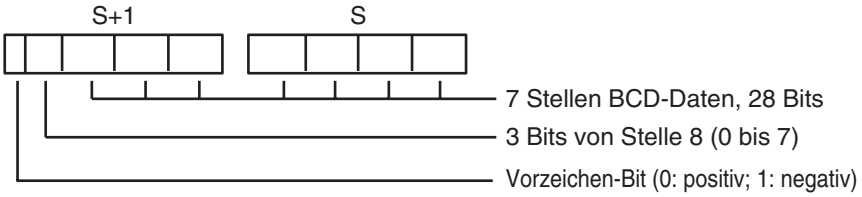
1. Der Wert –0 in den Quelldaten wird als 0 behandelt und führt nicht zu einem Fehler.
2. Einige Spezial-E/A-Baugruppen erfordern vorzeichenbehaftete BCD-Daten als Eingabe. BCDL(473) kann verwendet werden, um vorzeichenbehaftete Binärdaten für die Ausgabe an diese Baugruppen zu konvertieren.

Das Steuerwort spezifiziert das vorzeichenbehaftete BCD-Format, das für das Ergebnis verwendet wird (siehe unten).

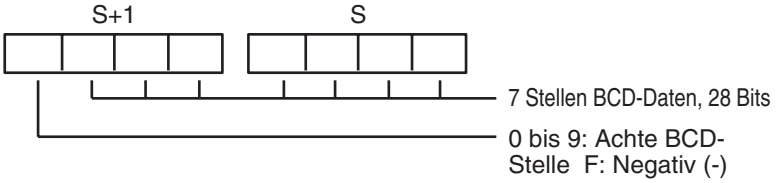
C = 0000 (Ausgangsdatenbereich: -999 9999 bis 999 9999 BCD)



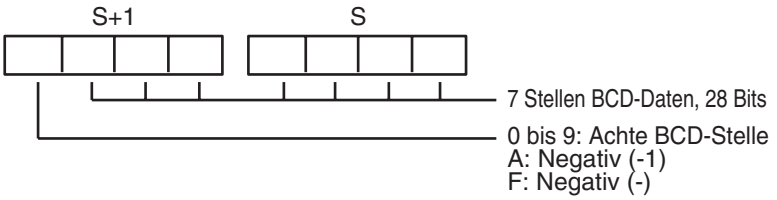
C = 0001 (Ausgangsdatenbereich: -7999 9999 bis 7999 9999 BCD)



C = 0002 (Ausgangsdatenbereich: -999 9999 bis 7999 9999 BCD)



C = 0003 (Ausgangsdatenbereich: -1999 9999 bis 7999 9999 BCD)



Die folgende Tabelle zeigt die möglichen vorzeichenbehafteten Binär-Doppelwort-Werte für jedes vorzeichenbehaftete BCD-Format. Es tritt ein Fehler auf, wenn die Quelldaten nicht im gültigen Bereich für das spezifizierte vorzeichenbehaftete BCD-Format liegen.

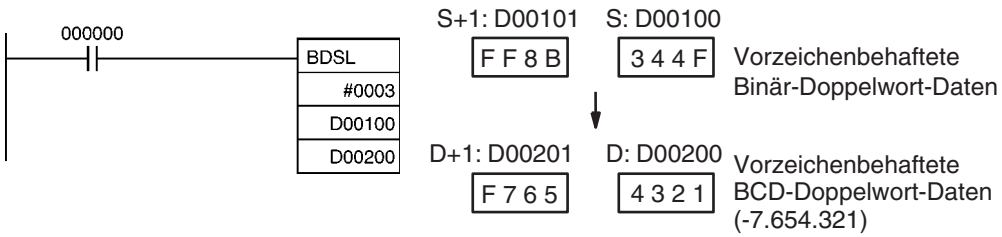
Einstellung	Vorzeichenbehaftete Binärwerte	Vorzeichenbehaftete BCD-Werte
C = 0000	FF67 6981 bis FFFF FFFF	-999 9999 bis –1
	0000 0000 bis 0098 967F	0 bis 999 9999
C = 0001	FB3B 4C01 bis FFFF FFFF	-7999 9999 bis –1
	0000 0000 bis 04C4 B3FF	0 bis 7999 9999
C = 0002	FF67 6981 bis FFFF FFFF	-999 9999 bis –1
	0000 0000 bis 05F5 E0FF	0 bis 999 9999
C = 0003	FECE D301 bis FFFF FFFF	-1999 9999 bis –1
	0000 0000 bis 05F5 E0FF	0 bis 999 9999

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn C nicht im spezifizierten Bereich von 0000 bis 0003 liegt. EIN, wenn C = 0000 ist und die Quelldaten nicht innerhalb des folgenden Bereichs liegen: FF67 6981 bis FFFF FFFF oder 0000 0000 bis 0098 967F. EIN, wenn C = 0001 ist und die Quelldaten nicht innerhalb des folgenden Bereichs liegen: FB3B 4C01 bis FFFF FFFF oder 0000 0000 bis 04C4 B3FF. EIN, wenn C = 0002 ist und die Quelldaten nicht innerhalb des folgenden Bereichs liegen: FF67 6981 bis FFFF FFFF oder 0000 0000 bis 05F5 E0FF. EIN, wenn C = 0003 ist und die Quelldaten nicht innerhalb des folgenden Bereichs liegen: FECE D301 bis FFFF FFFF oder 0000 0000 bis 05F5 E0FF. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn D nach der Ausführung 0000 ist. In allen anderen Fällen AUS.
Negativ-Flag	N	EIN, wenn C = 0000 oder 0001 ist und das Vorzeichen-Bit des Ergebnisses nach der Ausführung auf EIN gesetzt ist. EIN, wenn C = 0002 und die höchstwertige Stelle des Ergebnisses F ist. EIN, wenn C = 0003 und die höchstwertige Stelle des Ergebnisses A oder F ist. In allen anderen Fällen AUS.

Beispiel

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, werden die vorzeichenbehafteten Binär-Doppelwort-Daten in D00101 und D00100 mit dem im Steuerwort (0003) spezifizierten Format abgeglichen. Die Quelldaten sind korrekt, daher werden die vorzeichenbehafteten Binär-Doppelwort-Daten in D00101 und D00100 in vorzeichenbehaftete BCD-Doppelwort-Werte konvertiert und in D00201 und D00200 ausgegeben.



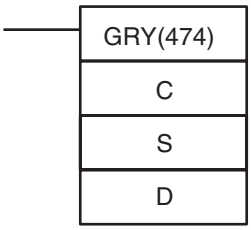
3-12-18 GRAY CODE CONVERT: GRY(474)

Funktion

Konvertiert Gray-Code-Daten im spezifizierten Wort in Binär-, BCD- oder Winkeldaten mit der spezifizierten Auflösung.

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Baugruppenversion 2.0 (einschließlich CS1-H, CJ1-H und CJ1M CPU-Baugruppen ab Lot-Nummer 030201) unterstützt.

Kontaktplansymbol



C: Erstes Steuerwort

S: Quellwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	GRY(474)
	Ausführung einmal bei steigender Flanke	@GRY(474)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

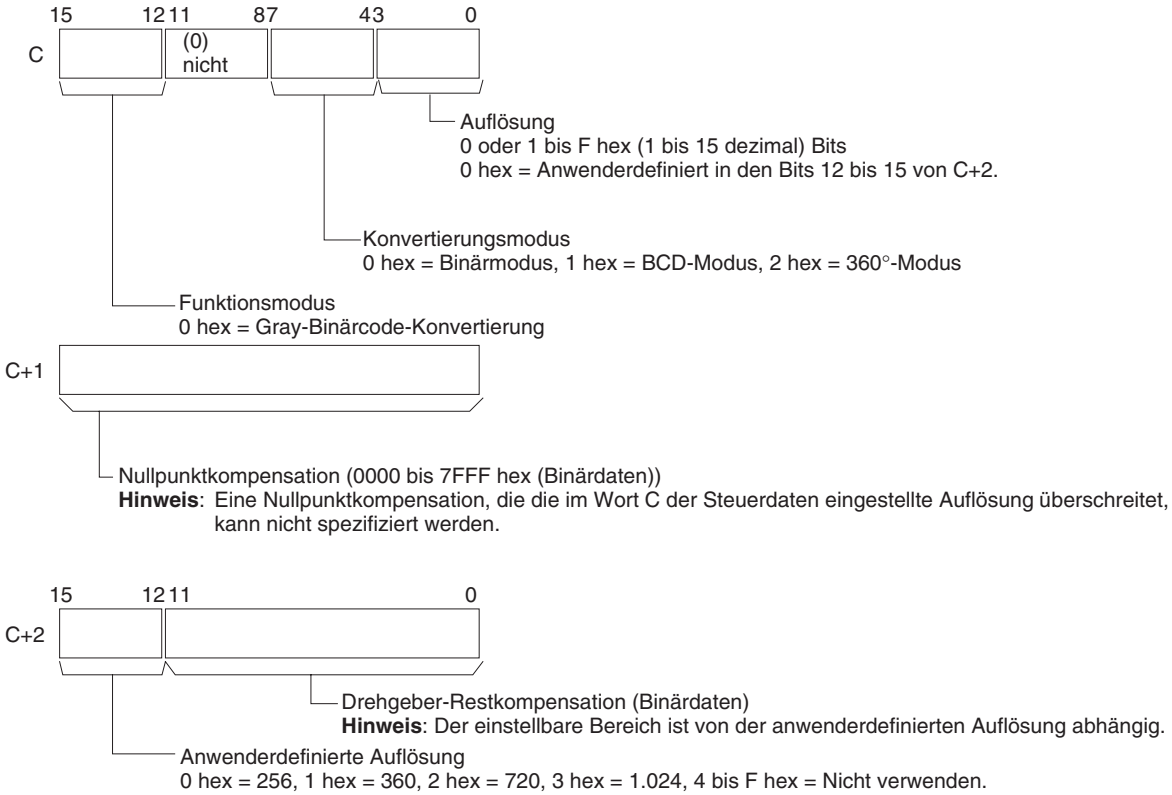
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

Spezifiziert die Parameter für die Konvertierung wie unten gezeigt.



Hinweis: Die Einstellung oben ist gültig, wenn die Auflösung in den Bits 00 bis 03 von C auf 0 hex gesetzt ist.

S: Quellwort

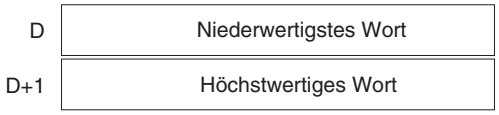
Enthält die zu konvertierenden vorzeichenbehafteten Gray-Binärcode-Daten. Der Bereich muss innerhalb der Anzahl von Bits liegen, die durch die in den Bits 00 bis 03 von C spezifizierte Auflösung bestimmt wird. Alle Bits außerhalb der der spezifizierten Auflösung entsprechenden Anzahl von Bits werden ignoriert. Wenn die spezifizierte Auflösung beispielsweise 08 hex ist und S den Wert FFFF hex enthält, wird 00FF hex als binärer Gray-Code verwendet.



D: Erstes Zielwort

Die Zielworte D+1 und D enthalten die Ergebnisse der Konvertierung von binären Gray-Code-Daten mit der in den Bits 00 bis 03 des Steuerdatenworts C spezifizierten Auflösung sowie den in den Bits 04 bis 07 des Steuerdatenworts C spezifizierten Konvertierungsmodus. Die ausgegebenen Daten haben folgende Bereiche:

Binärmodus: 0000 0000 bis 0000 7FFF hex
BCD-Modus: 0000 0000 bis 0003 2767
360°-Modus: 0000 0000 bis 0000 3599
(0,0° bis 359,9° in Schritten von 0,1°, BCD)



Operanden-Spezifikationen

Bereich	C	S	D
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142
Arbeitsbereich nicht remanent	W000 bis W510	W000 bis W511	W000 bis W510
Arbeitsbereich remanent	H000 bis H510	H000 bis H511	H000 bis H510
Systembereich	A000 bis A958	A000 bis A959	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095	T0000 bis T4094
Zählerbereich	C0000 bis C4094	C0000 bis C4095	C0000 bis C4094
DM-Bereich	D00000 bis D32766	D00000 bis D32767	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0000 bis #FFFF (binär)	---
Datenregister	---	DR0 bis DR15	---

Bereich	C	S	D
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

GRY(474) konvertiert die binären Gray-Code-Daten in dem in S spezifizierten Wort mit der in C spezifizierten Auflösung unter Verwendung eines der ebenfalls in C spezifizierten Konvertierungsmodi (binär, BCD oder 360°) und schreibt das Ergebnis in D und D+1..

Konvertierungs-modus	Funktion
Binärmodus	Binäre Gray-Code-Daten werden in Binärdaten im Bereich zwischen 0000 0000 und 0000 7FFF hex konvertiert. Nullpunkt-Offset und Restkompensation werden angewendet und das Ergebnis wird in D und D+1 ausgegeben.
BCD-Modus	Binäre Gray-Code-Daten werden in BCD-Daten konvertiert. Nullpunkt-Offset und Restkompensation werden angewendet, die Daten werden in BCD-Daten im Bereich zwischen 0000 0000 und 0003 2767 konvertiert und das Ergebnis wird anschließend in D und D+1 ausgegeben.
360° Modus	Binäre Gray-Code-Daten werden in BCD-Daten konvertiert. Nullpunkt-Offset und Restkompensation werden angewendet, die Daten werden in einen Winkelwert zwischen 0000 0000 und 0000 3599 (0,0° bis 359,9° in Schritten von 0,1°) konvertiert und das Ergebnis wird anschließend in D und D+1 ausgegeben.

Hinweis

1. Der Befehl GRY(474) wird normalerweise verwendet, wenn ein Parallelsignal (2^n) von einem Absolutwert-Drehgeber, der binären Gray-Code ausgibt, über eine DC-Eingangsbaugruppe eingegeben wird.
2. Wenn das in S spezifizierte Wort einer Eingangsbaugruppe zugewiesen ist, handelt es sich bei den durch GRY(474) konvertierten Eingangsdaten und die binären Gray-Code-Daten aus dem vorhergehenden CPU-Zyklus, die Daten sind also eine Zykluszeit alt.

Einschränkungen

Für den Befehl GRY(474) gelten die folgenden Einschränkungen.

■ **Einschränkungen bei der CPU-Baugruppe**

GRY(474) kann nur bei den folgenden Modellen von CPU-Baugruppen und nur bei CPU-Baugruppen verwendet werden, die nach dem 1. Februar 2003 hergestellt wurden (Lot-Nummer ab 030201, einschließlich CPU-Baugruppenversion ab 2.0).

- CJ1M-CPU□□
- CJ1G-CPU□□H
- CJ1H-CPU□□H
- CS1G-CPU□□H
- CS1H-CPU□□H
- CS1D-CPU□□S

Das Herstellungsdatum kann anhand der Lot-Nummer geprüft werden, die an der Seite oder Unterseite der CPU-Baugruppe angegeben ist. Lot-Nummern geben das Herstellungsdatum im folgenden Format an:

JJMMTT nnnn

JJ = beide rechten Stellen der Jahreszahl, MM = Monat als Zahl, TT = Tag des Monats, nnnn = Seriennummer

Hinweis Wenn der Befehl GRY(474) an eine CPU-Baugruppe übertragen wird, die diesen Befehl nicht unterstützt, und das Programm von einer Programmierkonsole gelesen wird, erfolgt die Anzeige von "?" für den Befehl GRY(474) als Kennzeichnung für einen unzulässigen Befehl. Wenn GRY(474) mit einer EIN-Ausführungsbedingung in einer CPU-Baugruppe ausgeführt wird, die den Befehl nicht unterstützt, tritt ein Fehler aus und die Programmausführung wird angehalten.

■ **Einschränkungen bei CX-Programmer**

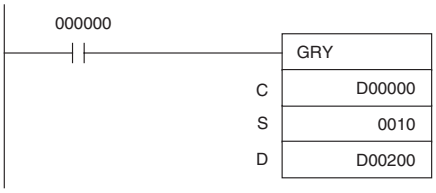
GRY(474) kann nur mit CX-Programmer ab Version 3.2 verwendet werden.

Flags

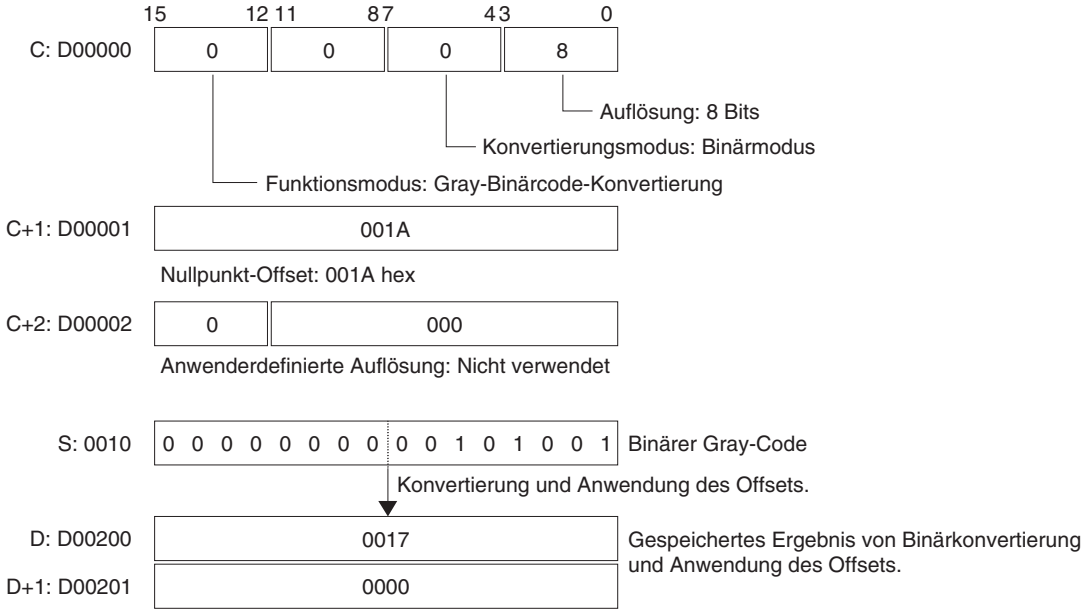
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn Bits 12 bis 15 von C nicht den Wert 0 hex haben (Funktionsmodus = Gray-Binärcode-Konvertierung). EIN, wenn der in C+1 spezifizierte Nullpunkt-Offset nicht innerhalb der spezifizierten Auflösung liegt (einschließlich anwenderdefinierte Auflösungen). EIN, wenn Bits 04 bis 07 von C nicht den Wert 0 hex (= Binärmodus), 1 hex (= BCD-Modus) oder 2 hex (= 360°-Modus) enthalten. EIN, wenn die spezifizierte Drehgeber-Restkompensation größer als die anwenderdefinierte Auflösung ist (falls Bits 00 bis 03 von C auf 0 hex (= anwenderdefinierte Auflösung) gesetzt sind). EIN, wenn der konvertierte Binärwert kleiner ist als die Drehgeber-Restkompensation (falls Bits 00 bis 03 von C auf 0 hex (= anwenderdefinierte Auflösung) gesetzt sind). EIN, wenn der konvertierte Binärwert kleiner ist als die Auflösung (falls Bits 00 bis 03 von C auf 0 hex (= anwenderdefinierte Auflösung) gesetzt sind). In allen anderen Fällen AUS.
Gleich-Flag	=	In allen Fällen AUS.
Negativ-Flag	N	In allen Fällen AUS.

Beispiele

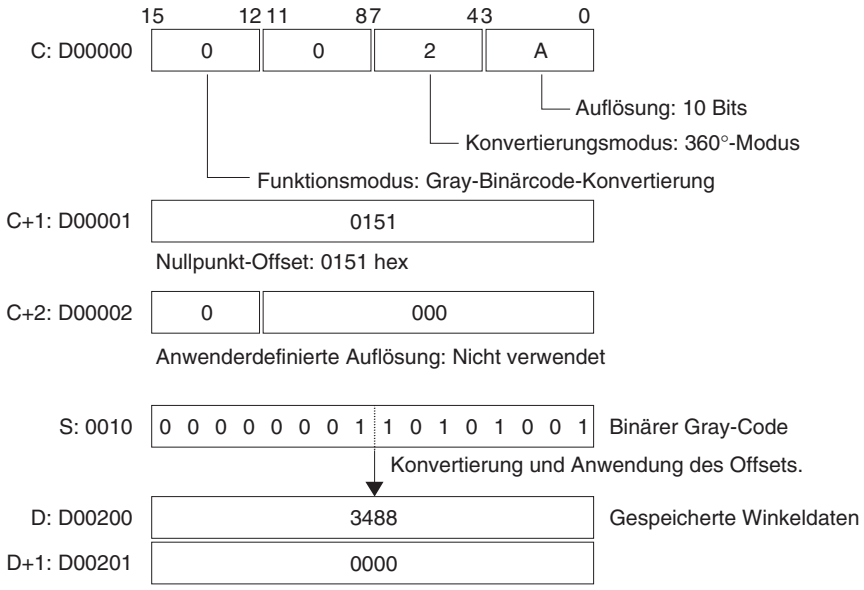
Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, wird der binäre Gray-Code in CIO 0010 entsprechend der Einstellungen für die Steuerdaten in D00000 bis D00002 konvertiert und das Ergebnis wird in D00200 und D00201 ausgegeben.



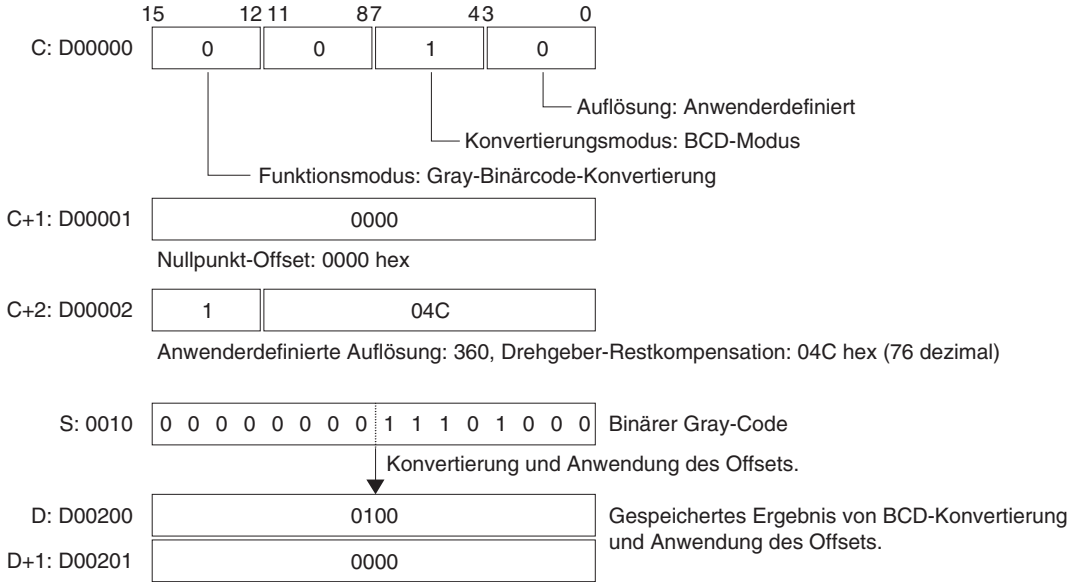
■ **Beispiel 1: Konvertierung in Binärdaten mit 8-Bit-Auflösung und einem Nullpunkt-Offset von 001A hex.**



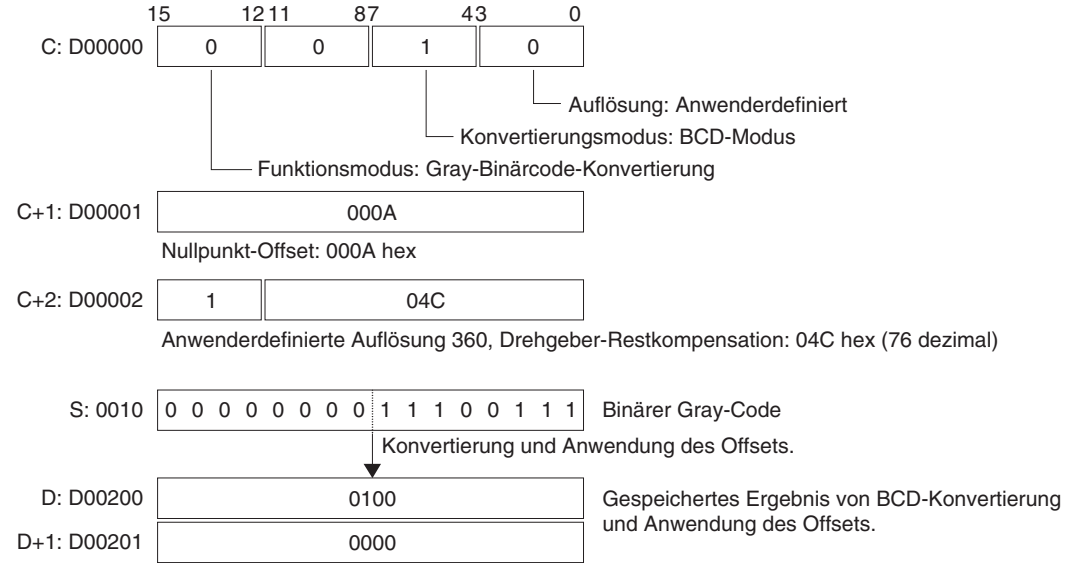
■ **Beispiel 2: Konvertierung in Winkeldaten mit 10-Bit-Auflösung und einem Nullpunkt-Offset von 0151 hex.**



■ Beispiel 3: Konvertierung der Daten eines OMRON E6C2-AG5C Absolutwert-Drehgebers (Auflösung: 360/Umdrehung, Drehgeber-Restkompensation: 76) in BCD-Daten mit einem Nullpunkt-Offset von 0000 hex



■ Beispiel 4: Konvertierung der Daten eines OMRON E6C2-AG5C Absolutwert-Drehgebers (Auflösung: 360/Umdrehung, Drehgeber-Restkompensation: 76) in BCD-Daten mit einem Nullpunkt-Offset von 000A hex



3-13 Logikbefehle

In diesem Abschnitt werden Befehle für Logikoperationen an Wortdaten beschrieben.

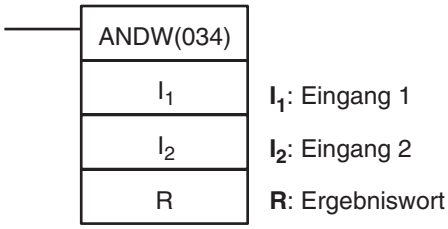
Befehl	AWL	Funktionscode	Seite
LOGICAL AND	ANDW	034	517
DOUBLE LOGICAL AND	ANDL	610	519
LOGICAL OR	ORW	035	520
DOUBLE LOGICAL OR	ORWL	611	522
EXCLUSIVE OR	XORW	036	524
DOUBLE EXCLUSIVE OR	XORL	612	526
EXCLUSIVE NOR	XNRW	037	528
DOUBLE EXCLUSIVE NOR	XNRL	613	529
COMPLEMENT	COM	029	531
DOUBLE COMPLEMENT	COML	614	533

3-13-1 LOGICAL AND: ANDW(034)

Zweck

Bildet eine logische AND-Verknüpfung zwischen entsprechenden Bits in Einzelworten von Wortdaten und/oder Konstanten.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	ANDW(034)
	Ausführung einmal bei steigender Flanke	@ANDW(034)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	I_1	I_2	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerker	H000 bis H511		
Systemmerker	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		

Bereich	I_1	I_2	R
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

ANDW(034) bildet eine logische AND-Verknüpfung zwischen den in I_1 und I_2 spezifizierten Daten und gibt das Ergebnis nach R aus.

- Die logische AND-Verknüpfung wird zwischen den entsprechenden Bits in I_1 und I_2 der Reihe nach gebildet.
- Wenn der Inhalt beider entsprechenden Bits in I_1 und I_2 1 ist, wird eine 1, wenn mindestens eines der Bits 0 ist, wird eine 0 an das entsprechende Bit in R ausgegeben.

I_1, I_2 → R

I_1	I_2	R
1	1	1
1	0	0
0	1	0
0	0	0

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von ANDW(034) wird der Fehlermerker auf AUS gesetzt.

Wenn der Inhalt von R als Ergebnis der AND-Verknüpfung 0000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

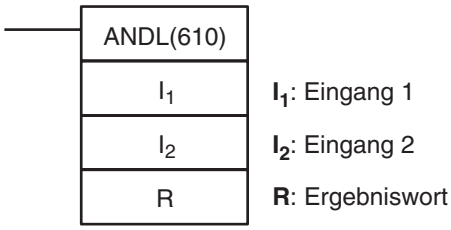
Wenn das höchstwertige Bit von R als Ergebnis der AND-Verknüpfung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

3-13-2 DOUBLE LOGICAL AND: ANDL(610)

Zweck

Bildet eine logische AND-Verknüpfung zwischen entsprechenden Bits in Doppelworten von Wortdaten und/oder Konstanten.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	ANDL(610)
	Ausführung einmal bei steigender Flanke	@ANDL(610)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	I_1	I_2	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerker	H000 bis H510		
Systemmerker	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

ANDL(610) bildet eine logische AND-Verknüpfung der in I_1, I_1+1 und I_2, I_2+1 spezifizierten Daten und gibt das Ergebnis in R, R+1 aus.

(I_1, I_1+1), (I_2, I_2+1) → (R, R+1)

I_1, I_1+1	I_2, I_2+1	R, R+1
1	1	1
1	0	0
0	1	0
0	0	0

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

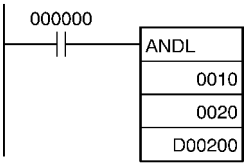
Nach der Ausführung von ANDL(610) wird der Fehlermerker auf AUS gesetzt.

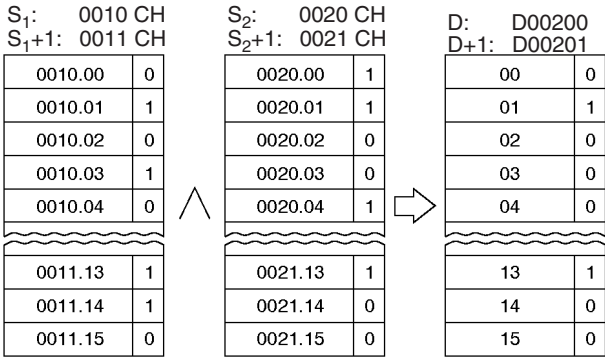
Wenn der Inhalt von R, R+1 als Ergebnis der AND-Verknüpfung 00000000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

Wenn das höchstwertige Bit von R+1 als Ergebnis der AND-Verknüpfung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

Beispiele

Wenn die Ausführungsbedingung CIO 00000000 auf EIN gesetzt ist, wird das logische AND der entsprechenden Bits in CIO 0011, CIO 0010 und CIO 0021, CIO 0020 gebildet und das Ergebnis wird in die entsprechenden Bits in D00201 und D00200 ausgegeben.





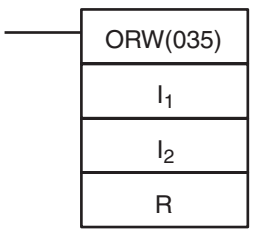
Hinweis: Der vertikal Pfeil steht für ein logisches AND.

3-13-3 LOGICAL OR: ORW(035)

Zweck

Bildet eine logische OR-Verknüpfung zwischen entsprechenden Bits in Einzelworten von Wortdaten und/oder Konstanten.

Kontaktplansymbol



I_1: Eingang 1

I_2: Eingang 2

R: Ergebniswort

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	ORW(035)
	Ausführung einmal bei steigender Flanke	@ORW(035)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	I_1	I_2	R
CIO Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerker	H000 bis H511		
Systemmerker	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

ORW(034) bildet eine logische OR-Verknüpfung zwischen den in I_1 und I_2 spezifizierten Daten und gibt das Ergebnis nach R aus.

- Die logische OR-Verknüpfung wird zwischen den entsprechenden Bits in I_1 und I_2 der Reihe nach gebildet.
- Wenn eines der entsprechenden Bits in I_1 und I_2 gleich 1 ist oder wenn beide Bits 1 sind, wird eine 1 an das entsprechende Bit in R ausgegeben.

$I_1 + I_2 \rightarrow R$

I_1	I_2	R
1	1	1
1	0	1

I_1	I_2	R
0	1	1
0	0	0

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von ORW(035) wird der Fehlermerker auf AUS gesetzt.

Wenn der Inhalt von R als Ergebnis der OR-Verknüpfung 0000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

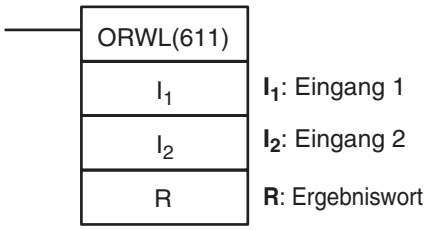
Wenn das höchstwertige Bit von R als Ergebnis der OR-Verknüpfung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

3-13-4 DOUBLE LOGICAL OR: ORWL(611)

Zweck

Bildet eine logische OR-Verknüpfung zwischen entsprechenden Bits in Doppelworten von Wortdaten und/oder Konstanten.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	ORWL(611)
	Ausführung einmal bei steigender Flanke	@ORWL(611)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	I_1	I_2	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerker	H000 bis H510		
Systemmerker	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		

Bereich	I_1	I_2	R
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

ORWL(611) bildet eine logische OR-Verknüpfung zwischen den als Doppelwortdaten in I_1, I_1+1 und I_2, I_2+1 spezifizierten Daten und gibt das Ergebnis nach R, R+1 aus.

- Wenn beliebige der entsprechenden Bits in I_1, I_1+1 und I_2, I_2+1 gleich 1 sind, wird eine 1 in das entsprechende Bit in R, R+1 ausgegeben. Wenn beide entsprechenden Bits den Wert 0 haben, wird eine 0 in das entsprechende Bit in R, R+1 ausgegeben.

(I_1, I_1+1) + (I_2, I_2+1) → (R, R+1)

I_1, I_1+1	I_2, I_2+1	R, R+1
1	1	1
1	0	1
0	1	1
0	0	0

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

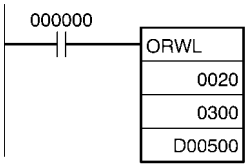
Nach der Ausführung von ORWL(611) wird der Fehlermerker auf AUS gesetzt.

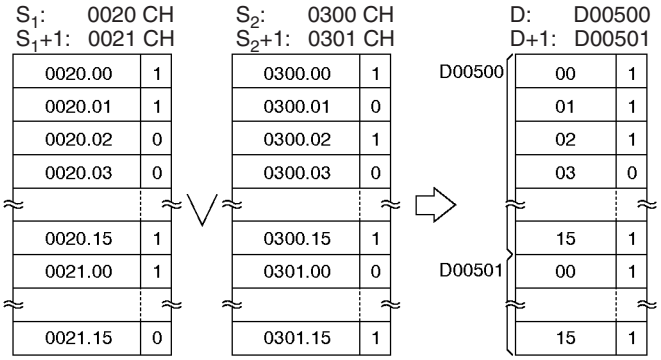
Wenn der Inhalt von R, R+1 als Ergebnis der OR-Verknüpfung 00000000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

Wenn das höchstwertige Bit von R+1 als Ergebnis der OR-Verknüpfung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

Beispiele

Wenn die Ausführungsbedingung CIO 00000000 auf EIN gesetzt ist, wird das logische OR der entsprechenden Bits in CIO 0021, CIO 0020 und CIO 0301, CIO 0300 gebildet und das Ergebnis wird in die entsprechenden Bits in D00501 und D00500 ausgegeben.





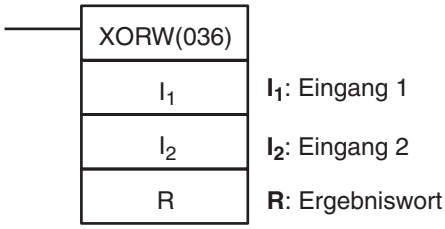
Hinweis: Der vertikale Pfeil steht für ein logisches OR.

3-13-5 EXCLUSIVE OR: XORW(036)

Zweck

Bildet eine logische Exklusiv-OR-Verknüpfung zwischen entsprechenden Bits in Einzelworten von Wortdaten und/oder Konstanten.

Kontaktplansymbol



I1: Eingang 1

I2: Eingang 2

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	XORW(036)
	Ausführung einmal bei steigender Flanke	@XORW(036)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	I_1	I_2	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerker	H000 bis H511		
Systemmerker	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		

Bereich	I_1	I_2	R
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

XORW(036) bildet eine logische Exklusiv-OR-Verknüpfung zwischen den in I_1 und I_2 spezifizierten Daten und gibt das Ergebnis nach R aus.

- Die logische Exklusiv-OR-Verknüpfung wird zwischen den entsprechenden Bits in I_1 und I_2 der Reihe nach gebildet.
- Wenn der Inhalt der entsprechenden Bits von I_1 und I_2 voneinander verschieden ist, wird in das entsprechende Bit von R eine 1 ausgegeben. Wenn die entsprechenden Bits gleich sind, wird eine 0 in das entsprechende Bit von R ausgegeben.

$$I_1, \overline{I_2} + \overline{I_1}, I_2 \rightarrow R$$

I_1	I_2	R
1	1	0
1	0	1
0	1	1
0	0	0

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von XORW(036) wird der Fehlermerker auf AUS gesetzt.

Wenn der Inhalt von R als Ergebnis der OR-Verknüpfung 0000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

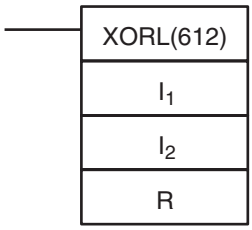
Wenn das höchstwertige Bit von R als Ergebnis der OR-Verknüpfung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

3-13-6 DOUBLE EXCLUSIVE OR: XORL(612)

Zweck

Bildet eine logische exklusive OR-Verknüpfung zwischen entsprechenden Bits in Doppelworten von Wortdaten und/oder Konstanten.

Kontaktplansymbol



I_1: Eingang 1

I_2: Eingang 2

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	XORL(612)
	Ausführung einmal bei steigender Flanke	@XORL(612)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	I_1	I_2	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerker	H000 bis H510		
Systemmerker	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

XORL(612) bildet eine logische exklusive OR-Verknüpfung zwischen den als Doppelwortdaten in I_1, I_1+1 und I_2, I_2+1 spezifizierten Daten und gibt das Ergebnis nach R, R+1 aus.

- Wenn der Inhalt beliebiger der entsprechenden Bits in I_1, I_1+1 und I_2, I_2+1 voneinander verschieden ist, wird eine 1 in das entsprechende Bit in R, R+1 ausgegeben. Wenn beliebige entsprechende Bits gleich sind, wird eine 0 in das entsprechende Bit in R, R+1 ausgegeben.

$(I_1, I_1+1), \overline{(I_2, I_2+1)} + \overline{(I_1, I_1+1)}, (I_2, I_2+1) \rightarrow (R, R+1)$

I_1, I_1+1	I_2, I_2+1	R, R+1
1	1	0
1	0	1
0	1	1
0	0	0

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

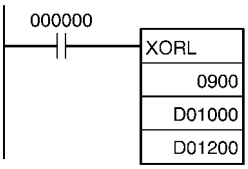
Nach der Ausführung von XORL(612) wird der Fehlermerker auf AUS gesetzt.

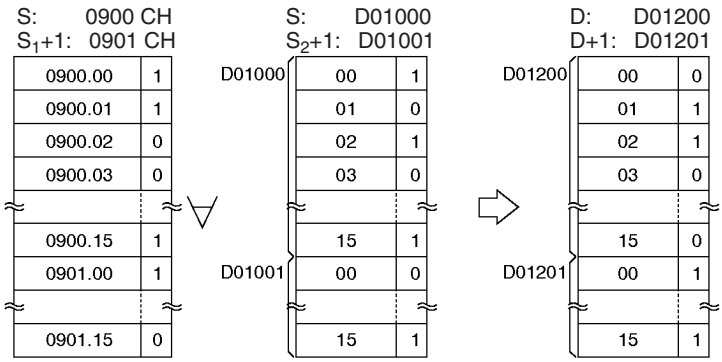
Wenn der Inhalt von R, R+1 als Ergebnis der exklusiven OR-Verknüpfung 00000000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

Wenn das höchstwertige Bit von R+1 als Ergebnis der exklusiven OR-Verknüpfung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

Beispiele

Wenn die Ausführungsbedingung CIO 00000000 auf EIN gesetzt ist, wird das logische exklusive OR der entsprechenden Bits in CIO 0901, CIO 0900 und D01001, D01000 gebildet und das Ergebnis wird in die entsprechenden Bits in D01201 und D01200 ausgegeben.





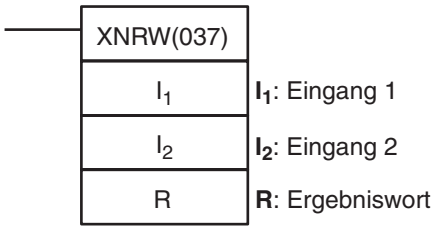
Hinweis: Das Symbol steht für ein exklusives logisches OR.

3-13-7 EXCLUSIVE NOR: XNRW(037)

Zweck

Bildet eine logische exklusive NOR-Verknüpfung zwischen entsprechenden Einzelworten von Wortdaten und/oder Konstanten.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	XNRW(037)
	Ausführung einmal bei steigender Flanke	@XNRW(037)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	I_1	I_2	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerker	H000 bis H511		
Systemmerker	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)		---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

XNRW(037) bildet eine logische exklusive NOR-Verknüpfung zwischen den in I_1 und I_2 spezifizierten Daten und gibt das Ergebnis nach R aus.

- Die logische exklusive NOR-Verknüpfung wird zwischen den entsprechenden Bits in I_1 und I_2 der Reihe nach gebildet.
- Wenn der Inhalt der entsprechenden Bits von I_1 und I_2 voneinander verschieden ist, wird in das entsprechende Bit von R eine 0 ausgegeben. Wenn die entsprechenden Bits gleich sind, wird eine 1 in das entsprechende Bit von R ausgegeben.

$\mathbf{I_1, I_2 + \overline{T_1}, \overline{T_2} \rightarrow R}$

I_1	I_2	R
1	1	1
1	0	0
0	1	0
0	0	1

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Nach der Ausführung von XNRW(037) wird der Fehlermerker auf AUS gesetzt.

Wenn der Inhalt von R als Ergebnis der NOR-Verknüpfung 0000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

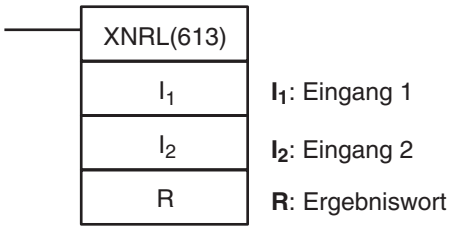
Wenn das höchstwertige Bit von R als Ergebnis der NOR-Verknüpfung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

3-13-8 DOUBLE EXCLUSIVE NOR: XNRL(613)

Zweck

Bildet eine logische exklusive NOR-Verknüpfung zwischen entsprechenden Bits in Doppelworten von Wortdaten und/oder Konstanten.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	XNRL(613)
	Ausführung einmal bei steigender Flanke	@XNRL(613)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	I_1	I_2	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerker	H000 bis H510		
Systemmerker	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

XNRL(613) bildet eine logische exklusive NOR-Verknüpfung zwischen den in I_1, I_1+1 und I_2, I_2+1 spezifizierten Daten und gibt das Ergebnis nach R, R+1 aus.

- Wenn der Inhalt beliebiger der entsprechenden Bits in I_1, I_1+1 und I_2, I_2+1 voneinander verschieden ist, wird eine 0 in das entsprechende Bit in R, R+1 ausgegeben. Wenn beliebige entsprechende Bits gleich sind, wird eine 1 in das entsprechende Bit in R, R+1 ausgegeben.

$(I_1, I_1+1), (I_2, I_2+1) + \overline{(I_1, I_1+1)}, \overline{(I_2, I_2+1)} \rightarrow (R, R+1)$

I_1, I_1+1	I_2, I_2+1	R, R+1
1	1	1
1	0	0
0	1	0
0	0	1

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

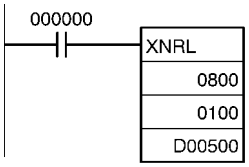
Nach der Ausführung von XNRL(613) wird der Fehlermerker auf AUS gesetzt.

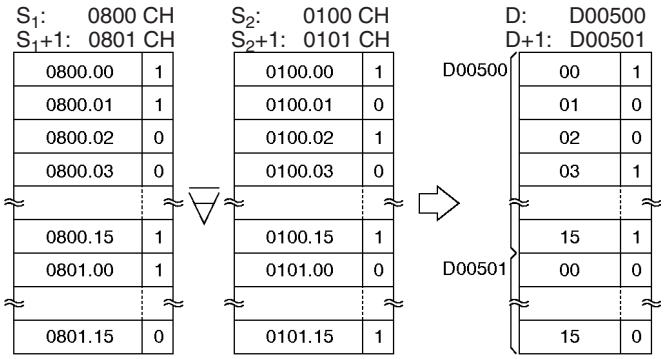
Wenn der Inhalt von R, R+1 als Ergebnis der exklusiven NOR-Verknüpfung 00000000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

Wenn das höchstwertige Bit von R+1 als Ergebnis der exklusiven NOR-Verknüpfung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

Beispiele

Wenn die Ausführungsbedingung CIO 00000000 auf EIN gesetzt ist, wird das logische exklusive NOR der entsprechenden Bits in CIO 0801, CIO 0800 und CIO 0101, CIO 0100 gebildet und das Ergebnis wird in die entsprechenden Bits in D00501 und D00500 ausgegeben.





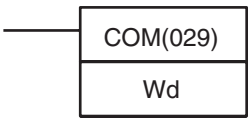
Hinweis: Das Symbol steht für ein exklusives logisches NOR.

3-13-9 COMPLEMENT: COM(029)

Zweck

Setzt alle auf EIN gesetzten Bits in Wd auf AUS und alle auf AUS gesetzten Bits auf EIN.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	COM(029)
	Ausführung einmal bei steigender Flanke	@COM(029)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511
Haftmerker	H000 bis H511
Systemmerker	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767

Bereich	Wd
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

COM(029) invertiert den Status aller spezifizierten Bits in Wd.
$\overline{Wd} \rightarrow Wd$: 1 → 0 und 0 → 1

Hinweis Achten Sie bei Verwendung des COM-Befehls darauf, dass der Status aller Bits in jedem Zyklus geändert wird, in dem die Ausführungsbedingung EIN ist.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

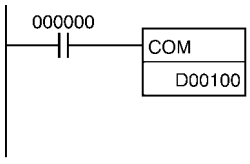
Nach der Ausführung von COM(029) wird der Fehlermerker auf AUS gesetzt.

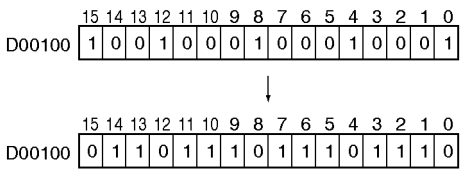
Wenn der Inhalt von R als Ergebnis der Komplementbildung 0000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

Wenn das höchstwertige Bit von R als Ergebnis der Komplementbildung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird der Status aller Bits in D00100 invertiert.



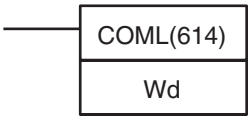


3-13-10 DOUBLE COMPLEMENT: COML(614)

Zweck

Setzt in Wd und Wd+1 alle auf EIN gesetzten Bits auf AUS und alle auf AUS gesetzten Bits auf EIN.

Kontaktplansymbol



Wd: Wort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	COML(614)
	Ausführung einmal bei steigender Flanke	@COML(614)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Wd
CIO-Bereich	CIO 0000 bis CIO 6142
Arbeitsbereich	W000 bis W510
Haftmerker	H000 bis H510
Systemmerker	A448 bis A958
Zeitgeberbereich	T0000 bis T4094
Zählerbereich	C0000 bis C4094
DM-Bereich	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

COML(614) invertiert den Status aller spezifizierten Bits in Wd und Wd+1.
$\overline{(Wd+1, Wd)} \rightarrow (Wd+1, Wd)$

Hinweis Achten Sie bei Verwendung des Befehls zur Komplementbildung darauf, dass der Status aller Bits in jedem Zyklus geändert wird, in dem die Ausführungsbedingung EIN ist.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R gleich 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

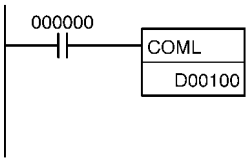
Nach der Ausführung von COML(614) wird der Fehlermerker auf AUS gesetzt.

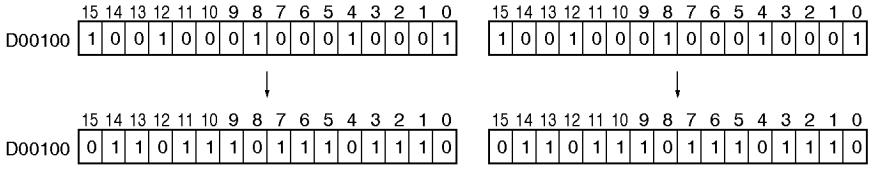
Wenn der Inhalt von R, R+1 als Ergebnis der Doppelwort-Komplementbildung 00000000 hex ist, wird der Gleich-Merker auf EIN gesetzt.

Wenn das höchstwertige Bit von R+1 als Ergebnis der Doppelwort-Komplementbildung 1 ist, wird der Negativ-Merker auf EIN gesetzt.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, wird der Status aller Bits in D00100 und D00101 invertiert.





3-14 Spezielle arithmetische Befehle

In diesem Abschnitt werden Befehle für spezielle arithmetische Berechnungen beschrieben.

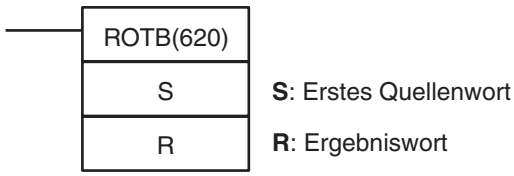
Befehl	AWL	Funktionscode	Seite
BINARY ROOT	ROTB	620	534
BCD SQUARE ROOT	ROOT	072	536
ARITHMETIC PROCESS	APR	069	540
FLOATING POINT DIVIDE	FDIV	079	552
BIT COUNTER	BCNT	067	556

3-14-1 BINARY ROOT: ROTB(620)

Zweck

Berechnet die Quadratwurzel aus dem vorzeichenbehafteten 32-Bit-Binärinhalt (positiver Wert) der spezifizierten Worte und gibt den ganzzahligen Teil des Ergebnisses in das spezifizierte Ergebniswort aus.

Kontaktplansymbol



S: Erstes Quellenwort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ROTB(620)
	Ausführung einmal bei steigender Flanke	@ROTB(620)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

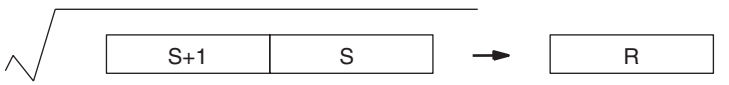
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W510	W000 bis W511
Haftmerker	H000 bis H510	H000 bis H511
Systemmerker	A000 bis A958	A448 bis A959
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	DR0 bis DR15	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

ROTB(620) berechnet die Quadratwurzel der 32-Bit-Binärzahl in S+1 und S und gibt den ganzzahligen Teil des Ergebnisses in R aus. Der nicht ganzzahlige Rest wird verworfen.



Binärdaten (32 Bits) Binärdaten (16 Bits)

Der Datenbereich, der für die Worte S+1 und S spezifiziert werden kann, beträgt 0000 0000 bis 3FFF FFFF. Wenn eine Zahl zwischen 4000 0000 und 7FFF FFFF spezifiziert wird, so wird sie für die Berechnung der Quadratwurzel als 3FFF FFFF behandelt. Ein Fehler tritt auf, wenn der Inhalt der Quellenworte größer als 7FFF FFFF ist, d. h., wenn Bit 15 von S+1 eine 1 ist.

Merker

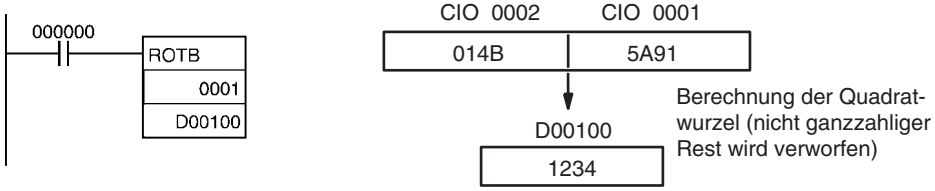
Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn Bit 15 von S+1 gleich 1 (EIN) ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Inhalt von S+1 und S zwischen 4000 0000 und 7FFF FFFF liegt. In allen anderen Fällen AUS.
Unterlauf-Merker	UF	AUS
Negativ-Merker	N	AUS

Vorsichtsmaßnahmen

Der Inhalt von S+1 und S muss kleiner sein als 8000 0000.
Die Operanden dieses Befehls (S+1, S und R) werden alle als Binärwerte behandelt. Verwenden Sie bei BCD-Daten als Eingangsdaten den Befehl ROOT(072).

Beispiel

Wenn CIO 000000 im folgenden Beispiel EIN ist, berechnet ROTB(620) die Quadratwurzel der Daten in CIO 0002 und CIO 0001 und schreibt den ganzzahligen Teil des Ergebnisses in D00100.

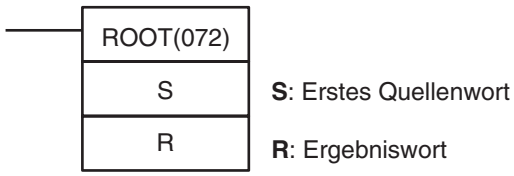


3-14-2 BCD SQUARE ROOT: ROOT(072)

Zweck

Berechnet die Quadratwurzel aus einer 8-stelligen BCD-Zahl und gibt den ganzzahligen Teil des Ergebnisses in das spezifizierte Ergebniswort aus.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ROOT(072)
	Ausführung einmal bei steigender Flanke	@ROOT(072)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

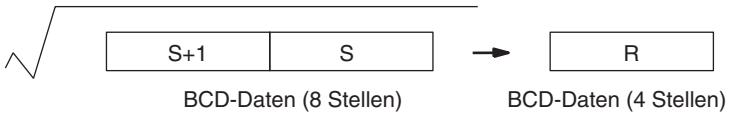
Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W510	W000 bis W511
Haftmerker	H000 bis H510	H000 bis H511
Systemmerker	A000 bis A958	A448 bis A959

Bereich	S	R
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #99999999 (BCD)	---
Datenregister	---	DR0 bis DR15
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

ROOT(072) berechnet die Quadratwurzel der 8-Bit-BCD-Zahl in S+1 und S und gibt den ganzzahligen Teil des Ergebnisses in R aus. Der nicht ganzzahlige Rest wird verworfen.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Daten in S+1 und S keine BCD-Daten sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

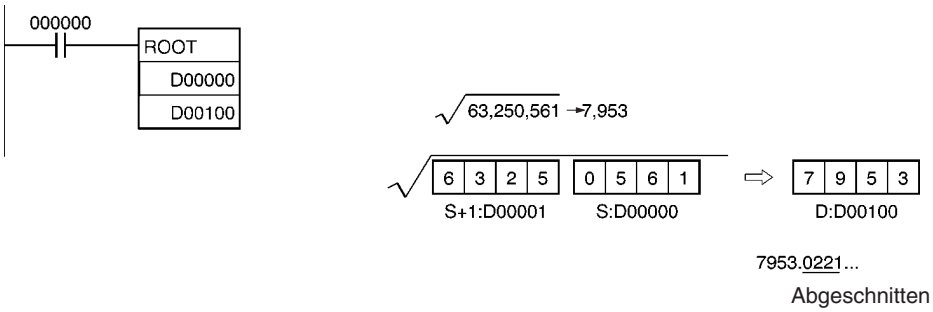
Die Operanden dieses Befehls (S+1, S und R) werden alle als BCD-Werte behandelt. Verwenden Sie bei Binärdaten als Eingangsdaten den Befehl ROTB(620).

Beispiele

Quadratwurzel einer 8-stelligen Zahl

Wenn CIO 000000 im folgenden Beispiel EIN ist, berechnet ROOT(072) die Quadratwurzel der Daten in D00001 und D00000 und schreibt den ganzzahligen Teil des Ergebnisses in D00100.

Hinweis Nachkommawerte von 8-stelligen Zahlen werden abgeschnitten.



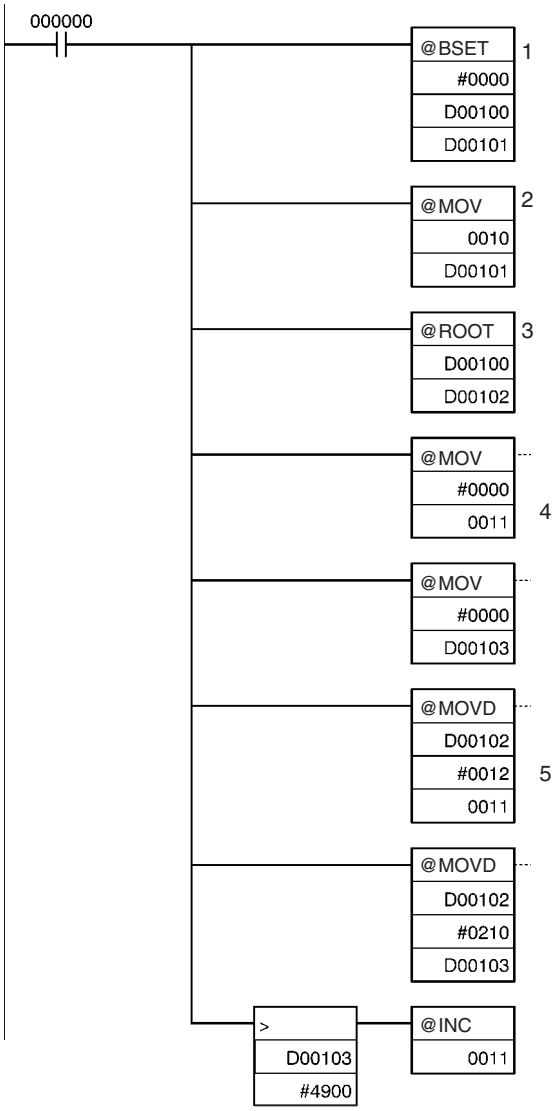
Quadratwurzel einer 4-stelligen Zahl

Das folgende Beispiel zeigt, wie die Quadratwurzel einer 4-stelligen Zahl berechnet wird und das Ergebnis gerundet wird. Dieses Programm berechnet die Quadratwurzel der 4-stelligen Zahl in CIO 0010, rundet das Ergebnis und schreibt es in CIO 0011. (Grundsätzlich wird die 4-stellige Zahl mit 10.000 (100^2) multipliziert und das Ergebnis wird durch 100 geteilt, wodurch die Genauigkeit der Berechnung um den Faktor 100 erhöht wird.)

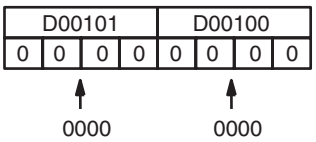
Hinweis Nachkommawerte von 4-stelligen Zahlen werden gerundet.

$\sqrt{6017}$=77.56....→78

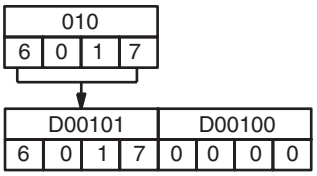
Die Nachkommawerte werden gerundet.



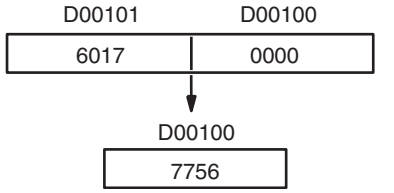
1,2,3... 1. Die Quellenworte (D00101 und D00100) werden auf 0000 0000 gelöscht.



2. Die 4-stellige Zahl wird in D00101 übertragen.



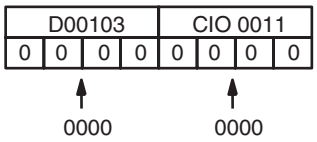
3. ROOT(072) berechnet die Quadratwurzel aus D00101 und D00100 und schreibt das Ergebnis in D00102.



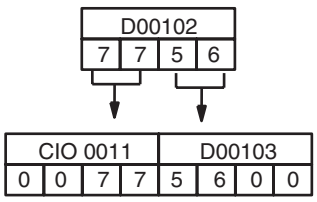
$\sqrt{60,170,000} = 7,756,932\ldots$

Berechnung der Quadratwurzel
(Rest wird verworfen)

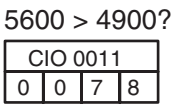
4. D00103 und das Ergebniswort, CIO 0011, werden auf 0000 0000 gelöscht.



5. Das Ergebnis der Berechnung der Quadratwurzel wird durch 100 dividiert. Der ganzzahlige Teil des Ergebnisses wird in CIO 0011 geschrieben, der Rest in D00103.



6. Wenn der Inhalt von D00103 größer als 4900 ist, wird CIO 0011 um 1 inkrementiert. In diesem Fall ist das Ergebnis 78.

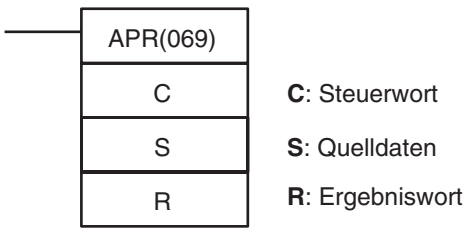


3-14-3 ARITHMETIC PROCESS: APR(069)

Zweck

Berechnet den Sinus, Kosinus oder eine lineare Extrapolation der Quelldaten. Mit der linearen Extrapolation können alle Beziehungen zwischen X und Y durch Geradensegmente angenähert werden.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	APR(069)
	Ausführung einmal bei steigender Flanke	@APR(069)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Sinus-Funktion (C = 0000 hex)

Operand	Wert	Datenbereich
C	0000 hex	---
S	0000 bis 0900 (BCD)	0° bis 90°
D	0000 bis 9999 (BCD)	0,0000 bis 0,9999
	9999 (BCD)	1,0000

Kosinus-Funktion (C = 0001 hex)

Operand	Wert	Datenbereich
C	0001 hex	---
S	0000 bis 0900 (BCD)	0° bis 90°
D	0000 bis 9999 (BCD)	0,0000 bis 0,9999
	9999 (BCD)	1,0000

Lineare Extrapolationsfunktion (C = Datenbereichsadresse)

Operand	Wert	Datenbereich
C	Datenbereichsadresse	---
S	vorzeichenlose 16-Bit-BCD-Daten	0000 bis 9999
	vorzeichenlose 16-Bit-Binärdaten	0 bis 65.535
	vorzeichenbehaftete 16-Bit-Binärdaten[1]	– 32.768 bis 32.767
	vorzeichenbehaftete 32-Bit-Binärdaten[1]	– 2.147.483.648 bis 2.147.483.647
	Gleitkommadaten[1]	$-\infty$, $-3{,}402823 \times 10^{38}$ bis $-1{,}175494 \times 10^{-38}$, $1{,}175494 \times 10^{-38}$ bis $3{,}402823 \times 10^{38}$, $+\infty$
D	vorzeichenlose 16-Bit-BCD-Daten	0000 bis 9999
	vorzeichenlose 16-Bit-Binärdaten	0 bis 65.535
	vorzeichenbehaftete 16-Bit-Binärdaten[1]	– 32.768 bis 32.767
	vorzeichenbehaftete 32-Bit-Binärdaten[1]	– 2.147.483.648 bis 2.147.483.647
	Gleitkommadaten[1]	$-\infty$, $-3{,}402823 \times 10^{38}$ bis $-1{,}175494 \times 10^{-38}$, $1{,}175494 \times 10^{-38}$ bis $3{,}402823 \times 10^{38}$, $+\infty$

Hinweis

1. Vorzeichenbehaftete Binärdaten und Gleitkommadaten werden nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.
2. Falls C eine Wort-Adresse ist, extrapoliert APR(069) den Y-Wert für den X-Wert in S anhand der Koordinaten (Bildung von Geradensegmenten), die im voraus in eine bei C beginnende Tabelle eingegeben wurden. Eine detaillierte Beschreibung finden Sie unter *Beschreibung*.

Operanden-Spezifikationen

Bereich	C	S	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerker	H000 bis H511		
Systemmerker	A000 bis A959		A448 bis A959

Bereich	C	S	R
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	Nur spezifizierte Werte		---
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Die Funktion von APR(069) hängt vom Steuerwort C ab. Ist C gleich 0000 oder 0001, berechnet APR(069) den Sinus bzw. den Kosinus von S (Einheit von S: zehntel Grad).

Falls C eine Wort-Adresse ist, extrapoliert APR(069) den Y-Wert für den X-Wert in S anhand der Koordinaten (Bildung von Geradensegmenten), die im voraus in eine bei C beginnende Tabelle eingegeben wurden.

Sinus-Funktion (C = 0000)

Wenn C gleich 0000 ist, berechnet APR(069) den SIN(S) und schreibt das Ergebnis in R. Der Wertebereich für S beträgt 0000 bis 0900 BCD (0,0° bis 90,0°) und der Wertebereich für R beträgt 0000 bis 9999 BCD (0,0000 bis 0,9999). Der Rest des Ergebnisses, der über die vierte Dezimalstelle hinausgeht, wird verworfen.

Kosinus-Funktion (C = 0001)

Wenn C gleich 0001 ist, berechnet APR(069) den COS(S) und schreibt das Ergebnis in R. Der Wertebereich für S beträgt 0000 bis 0900 BCD (0,0° bis 90,0°) und der Wertebereich für R beträgt 0000 bis 9999 BCD (0,0000 bis 0,9999). Der Rest des Ergebnisses, der über die vierte Dezimalstelle hinausgeht, wird verworfen.

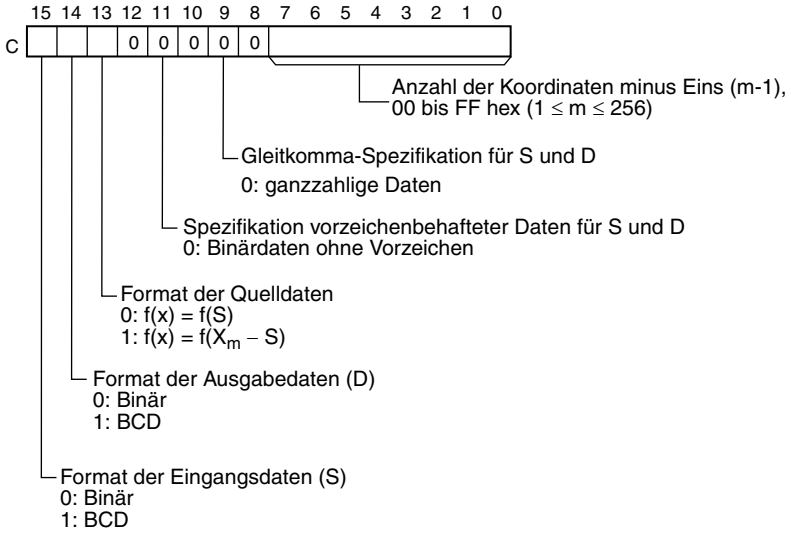
Lineare Extrapolation

Die lineare Extrapolation bei APR(069) wird durch Angabe einer Wort-Adresse für C spezifiziert.

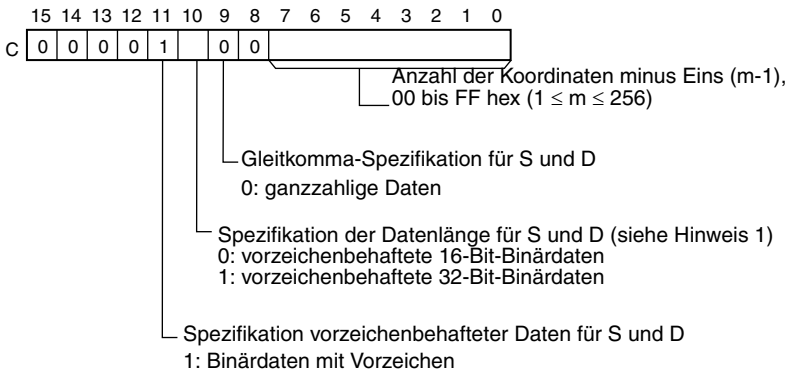
Der Inhalt des Worts in C spezifiziert die Anzahl der Koordinaten in einer Datentabelle, die bei C+2 beginnt, das Format der Quelldaten und ob es sich dabei um BCD- oder Binärdaten handelt. Bei CPU-Baugruppen der SPS-

Systeme CS1-H, CJ1-H, CJ1M and CS1D darf es sich bei den Quelldaten auch um vorzeichenbehaftete Binärdaten oder Gleitkommadaten handeln.

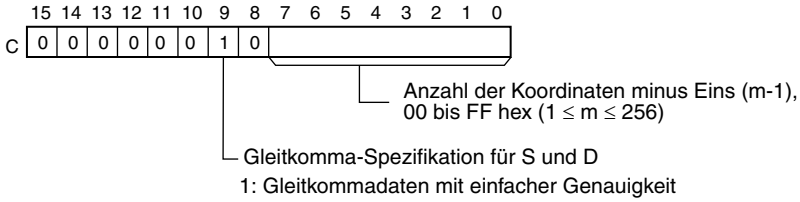
Vorzeichenlose ganzzahlige Daten (binär oder BCD)



Vorzeichenbehaftete ganzzahlige Daten (binär)



Gleitkommadaten mit einfacher Genauigkeit



Wenn 16-Bit-Bnär- oder BCD-Daten verwendet werden, sind die Geradensegmentdaten in den Worten C+ 1 bis C+2m+2 enthalten. Wenn 32-Bit-Binärder Gleitkommadaten verwendet werden (nur CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M), sind die Geradensegmentdaten in den Worten C+ 1 bis C+4m+4 enthalten.

Die Bits 00 bis 07 enthalten die Anzahl (binär) der Geradenkoordinaten minus 1, m–1. Die Bits 08 bis 12 werden nicht verwendet. Bit 13 spezifiziert entweder f(x)=f(S) oder f(x)=f(X_m–S): AUS spezifiziert f(x)=f(S) und EIN spezifiziert f(x)=f(X_m–S). Bit 14 legt fest, ob die Ausgabe im BCD- oder Binärformat erfolgt: AUS spezifiziert binär, EIN spezifiziert BCD. Bit 15 legt

fest, ob es sich beim Eingang um BCD- oder Binärdaten handelt: AUS spezifiziert binär, EIN spezifiziert BCD.

16-Bit-Binär- (mit oder ohne Vorzeichen) oder 16-Bit-BCD-Daten

C+1	X0 (*1)
C+2	Y0
C+3	X1
C+4	Y1
C+5	X2
C+6	Y2
	Xn
	Yn
C+ (2m+1)	Xm
C+ (2m+2)	Ym

Hinweis: Schreiben Sie X_m (max. X-Wert in der Tabelle) in das Wort C+1, wenn die E/A-Daten in S und D vorzeichenbehaftete Daten enthalten (Bit 11 von C = 0).

Vorzeichenbehaftete 32-Bit-Binärdaten

C+1	X0 (niederwertigste (rechte) 16 Bits)
C+2	X0 (höchstwertige (linke) 16 Bits)
C+3	Y0 (niederwertigste 16 Bits)
C+4	Y0 (höchstwertige 16 Bits)
C+5	X1 (niederwertigste 16 Bits)
C+6	X1 (höchstwertige 16 Bits)
C+7	Y1 (niederwertigste 16 Bits)
C+8	Y1 (höchstwertige 16 Bits)
bis	bis
C+ (4n+1)	Xn (niederwertigste 16 Bits)
C+ (4n+2)	Xn (höchstwertige 16 Bits)
C+ (4n+3)	Yn (niederwertigste 16 Bits)
C+ (4n+4)	Yn (höchstwertige 16 Bits)
bis	bis
C+ (4m+1)	Xm (niederwertigste 16 Bits)
C+ (4m+2)	Xm (höchstwertige 16 Bits)
C+ (4m+3)	Ym (niederwertigste 16 Bits)
C+ (4m+4)	Ym (höchstwertige 16 Bits)

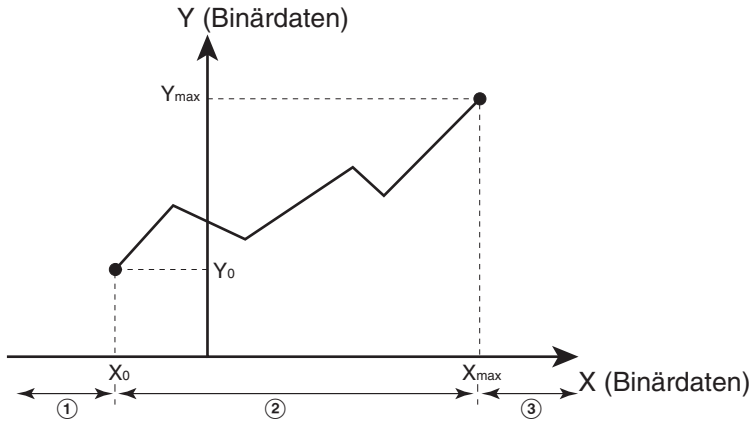
Gleitkommadaten

C+1	X0 (niederwertigste (rechte) 16 Bits)
C+2	X0 (höchstwertige (linke) 16 Bits)
C+3	Y0 (niederwertigste 16 Bits)
C+4	Y0 (höchstwertige 16 Bits)
C+5	X1 (niederwertigste 16 Bits)
C+6	X1 (höchstwertige 16 Bits)
C+7	Y1 (niederwertigste 16 Bits)
C+8	Y1 (höchstwertige 16 Bits)
bis	bis
C+ (4n+1)	Xn (niederwertigste 16 Bits)
C+ (4n+2)	Xn (höchstwertige 16 Bits)
C+ (4n+3)	Yn (niederwertigste 16 Bits)
C+ (4n+4)	Yn (höchstwertige 16 Bits)
bis	bis
C+ (4m+1)	Xm (niederwertigste 16 Bits)
C+ (4m+2)	Xm (höchstwertige 16 Bits)
C+ (4m+3)	Ym (niederwertigste 16 Bits)
C+ (4m+4)	Ym (höchstwertige 16 Bits)

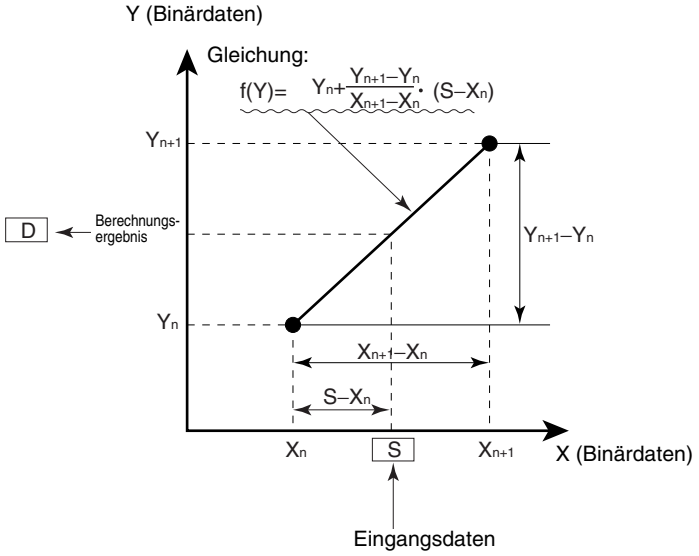
Hinweis Die X-Koordinaten müssen in aufsteigender Reihenfolge vorliegen: $X_1 < X_2 < ... < X_m$. Geben Sie unabhängig vom im Steuerwort C spezifizierten Datenformat alle Werte von (X_n, Y_n) als Binärdaten ein.

Funktionsweise der linearen Extrapolationsfunktion

APR(069) verarbeitet die in S spezifizierten Eingangsdaten gemäß der folgenden Gleichung und mit den in der mit C+1 beginnenden Tabelle spezifizierten Geradensegmentdaten (X_n, Y_n). Das Ergebnis wird in das/die für D spezifizierte(n) Zielwort(e) ausgegeben.



1. Bei $S < X_0$
   Konvertierter Wert = Y_0
2. Bei $X_0 \leq S \leq X_{max}$, wenn $X_n < S < X_{n+1}$
   $\text{Konvertierter Wert} = Y_n + [\{Y_n + 1 - Y_n\}/\{X_n + 1 - X_n\}] \times [\textit{Eingangsdaten}\, S - X_n\}$



3. $X_{max} < S$
   Konvertierter Wert = Y_{max}

In der bei C+1 beginnenden Tabelle der Geradensegmentdaten können bis zu 256 Endpunkte gespeichert werden. Die folgenden 5 Arten von E/A-Daten können verwendet werden:

- vorzeichenlose 16-Bit-BCD-Daten
- vorzeichenlose 16-Bit-Binärdaten
- vorzeichenbehaftete 16-Bit-Binärdaten (nur CS1-H/CJ1-H/CJ1M)
- vorzeichenbehaftete 32-Bit-Binärdaten (nur CS1-H/CJ1-H/CJ1M)
- Gleitkommadaten einfacher Präzision (nur CS1-H/CJ1-H/CJ1M)

Einstellen den Datenformats in Steuerwort C

- Vorzeichenlose 16-Bit-BCD-Daten
  Die Eingangs- und/oder Ausgangsdaten können vorzeichenlose 16-Bit-BCD-Daten sein. Zudem kann die Funktion zur linearen Extrapolation so eingerichtet werden, dass sie direkt den in S spezifizierten Wert verwendet oder den Wert von X_m–S. (X_m ist der größte X-Wert in den Geradensegmentdaten.)

Bezeichnung der Einstellung	Bit in C	Einstellung
Format der Eingangsdaten (S)	15	0: Binär 1: BCD
Format der Ausgabedaten (D)	14	0: Binär 1: BCD
Format der Quelldaten	13	0: Verwendung von S 1: Verwendung von X_m–S
Spezifikation vorzeichenbehafteter Daten für S und D	11	0: vorzeichenlose Daten
Spezifikation der Datenlänge für S und D	10	Ungültig (fest auf 16 Bits)
Spezifikation von Gleitkommadaten	09	0: ganzzahlige Daten

• Vorzeichenlose 16-Bit-Binärdaten
Die Eingangs- und/oder Ausgangsdaten können vorzeichenlose 16-Bit-Binärdaten sein. Zudem kann die Funktion zur linearen Extrapolation so eingerichtet werden, dass sie direkt den in S spezifizierten Wert verwendet oder den Wert von X_m–S. (X_m ist der größte X-Wert in den Geradensegmentdaten.)

Bezeichnung der Einstellung	Bit in C	Einstellung
Format der Eingangsdaten (S)	15	0: Binär 1: BCD
Format der Ausgabedaten (D)	14	0: Binär 1: BCD
Format der Quelldaten	13	0: Verwendung von S 1: Verwendung von X_m–S
Spezifikation vorzeichenbehafteter Daten für S und D	11	0: vorzeichenlose Daten
Spezifikation der Datenlänge für S und D	10	Ungültig (fest auf 16 Bits)
Spezifikation von Gleitkommadaten	09	0: ganzzahlige Daten

• Vorzeichenbehaftete 16-Bit-Binärdaten (nur CS1-H, CJ1-H, CJ1M und CS1D)

Bezeichnung der Einstellung	Bit in C	Einstellung
Format der Eingangsdaten (S)	15	0: Binär
Format der Ausgabedaten (D)	14	0: Binär
Format der Quelldaten	13	0
Spezifikation vorzeichenbehafteter Daten für S und D	11	1: vorzeichenbehaftete Daten
Spezifikation der Datenlänge für S und D	10	0: vorzeichenbehaftete 16-Bit-Binärdaten
Spezifikation von Gleitkommadaten	09	0: ganzzahlige Daten

• Vorzeichenbehaftete 32-Bit-Binärdaten (nur CS1-H, CJ1-H, CJ1M und CS1D)

Bezeichnung der Einstellung	Bit in C	Einstellung
Format der Eingangsdaten (S)	15	0: Binär
Format der Ausgabedaten (D)	14	0: Binär
Format der Quelldaten	13	0
Spezifikation vorzeichenbehafteter Daten für S und D	11	1: vorzeichenbehaftete Daten
Spezifikation der Datenlänge für S und D	10	1: vorzeichenbehaftete 32-Bit-Binärdaten
Spezifikation von Gleitkommadaten	09	0: ganzzahlige Daten

Hinweis Wenn die Einstellung „Spezifikation der Datenlänge für S und D“ in Bit 10 von C auf 1 gesetzt ist und eine 16-Bit-Konstante für S eingegeben wird, werden die Eingangsdaten vor der Berechnung zur linearen Extrapolation in vorzeichenbehaftete 32-Bit-Binärdaten konvertiert.

• Gleitkommadaten (nur CS1-H, CJ1-H, CJ1M und CS1D)

Bezeichnung der Einstellung	Bit in C	Einstellung
Format der Eingangsdaten (S)	15	0: Binär
Format der Ausgabedaten (D)	14	0: Binär
Format der Quelldaten	13	0
Spezifikation vorzeichenbehafteter Daten für S und D	11	0
Spezifikation der Datenlänge für S und D	10	0
Spezifikation von Gleitkommadaten	09	1: Gleitkommadaten

Hinweis Wenn die Einstellung „Spezifikation von Gleitkommadaten“ in Bit 09 von C auf 1 gesetzt ist, kann für S keine Konstante eingegeben werden.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn C eine Konstante größer als 0001 ist. EIN, wenn C eine Wort-Adresse ist, die X-Koordinaten jedoch nicht in aufsteigender Reihenfolge vorliegen ($X_1 \leq X_2 \leq ... \leq X_m$). EIN, wenn C eine Wortadresse ist und die Bits 9, 11 und 15 von C BCD-Daten als Eingang spezifizieren, die Daten in S jedoch keine BCD-Daten sind. EIN, wenn C eine Wortadresse ist und Bit 9 von C Gleitkommadaten spezifiziert, S jedoch eine Ein-Wort-Konstante ist. EIN, wenn C gleich 0000 oder 0001 ist, S jedoch kein BCD-Wert zwischen 0000 und 0900. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn Bit 15 von R EIN ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Das tatsächliche Ergebnis von SIN(90°) und COS(0°) ist 1, jedoch wird 9999 (0,9999) in R ausgegeben.

Ein Fehler tritt auf, wenn C eine Konstante größer als 0001 ist.

Es tritt ein Fehler auf, wenn die lineare Extrapolation spezifiziert ist (C ist eine Wort-Adresse), die X-Koordinaten jedoch nicht in aufsteigender Reihenfolge vorliegen ($X_1 < X_2 < ... < X_m$).

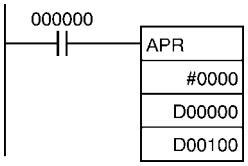
Es tritt ein Fehler auf, wenn die lineare Extrapolation und ein BCD-Wert als Eingang spezifiziert ist (Bit 15 von C ist EIN), S jedoch kein BCD-Wert ist.

Ein Fehler tritt auf, wenn eine trigonometrische Funktion spezifiziert ist (C = 0000 oder 0001), S jedoch keinen BCD-Wert zwischen 0000 und 0900 enthält.

Beispiele

Sinus-Funktion (C: #0000)

Das folgende Beispiel zeigt, wie der Befehl APR(069) zur Berechnung des Sinus von 30° verwendet wird.



Quelldaten

S: D00000			
0	10^1	10^0	10^{-1}
0	3	0	0

Format der Quelldaten: 10^{-1} Grad. (0000 bis 0900, BCD)

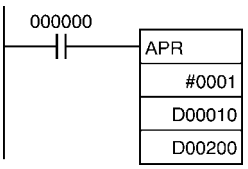
Ergebnis

R: D00100			
10^{-1}	10^{-2}	10^{-3}	10^{-4}
5	0	0	0

Die Ergebnisdaten bestehen aus vier gültige Stellen, die fünfte und weitere Stellen werden ignoriert. (0000 bis 9999, BCD)

Kosinus-Funktion (C: #0001)

Das folgende Beispiel zeigt, wie der Befehl APR(069) zur Berechnung des Kosinus von 30° verwendet wird.
(SIN(30) = 0,8660)



Quelldaten

S: D00010			
0	10^1	10^0	10^{-1}
0	3	0	0

Format der Quelldaten: 10^{-1} Grad. (0000 bis 0900, BCD)

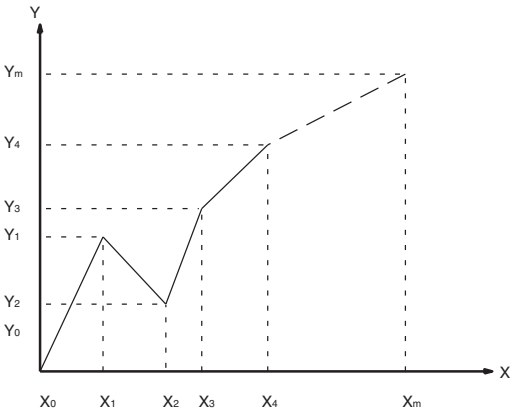
Ergebnis

R: D00200			
10^{-1}	10^{-2}	10^{-3}	10^{-4}
8	6	6	0

Die Ergebnisdaten bestehen aus vier gültige Stellen, die fünfte und weitere Stellen werden ignoriert. (0000 bis 9999, BCD)

Lineare Extrapolation (C: Wort-Adresse)
Verwendung von 16-Bit-BCD- oder Binärdaten ohne Vorzeichen

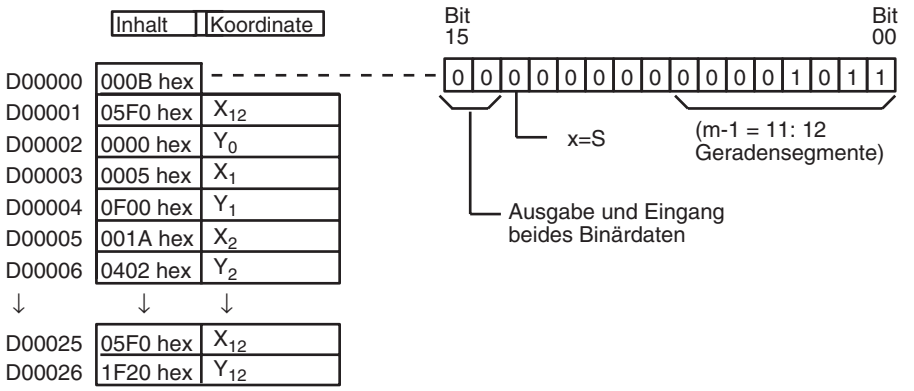
APR(069) verarbeitet die in S spezifizierten Eingangsdaten gemäß der Steuerdaten in C und der Geradensegmentdaten in der bei C+1 beginnenden Tabelle. Das Ergebnis wird nach D ausgegeben.



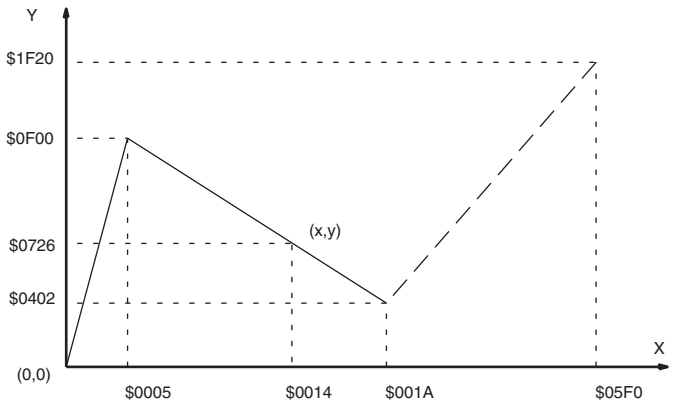
Wort	Koordinate
C+1	Xm (max. X-Wert)
C+2	Y_0
C+3	X_1
C+4	Y_1
C+5	X_2
C+6	Y_2
↓	↓
C+(2m+1)	X_m (max. X-Wert)
C+(2m+2)	Y_m

- $Y_n = f(X_n)$, $Y_0 = f(X_0)$
- In jedem Fall muss die Bedingung $X_{n-1} < X_n$ erfüllt sein.
- Geben Sie alle Werte für (X_n, Y_n) als Binärdaten ein.

Dieses Beispiel zeigt die Erstellung einer linearen Extrapolation mit 12 Koordinaten. Der Datenblock ist, wie erforderlich, kontinuierlich von D00000 bis D00026 (C bis C + (2 × 12 + 2)). Die Eingangsdaten werden aus CIO 0010 gelesen und das Ergebnis wird in CIO 0011 ausgegeben.



In diesem Fall enthält das Quellenwort, CIO 0010, den Wert 0014, und f(0014) = 0726 wird nach R, CIO 0011, ausgegeben.



Nachfolgend ist die Berechnung der linearen Extrapolation gezeigt.

$$Y = 0F00 + \frac{0402 - 0F00}{001A - 0005} \times (0014 - 0015)$$

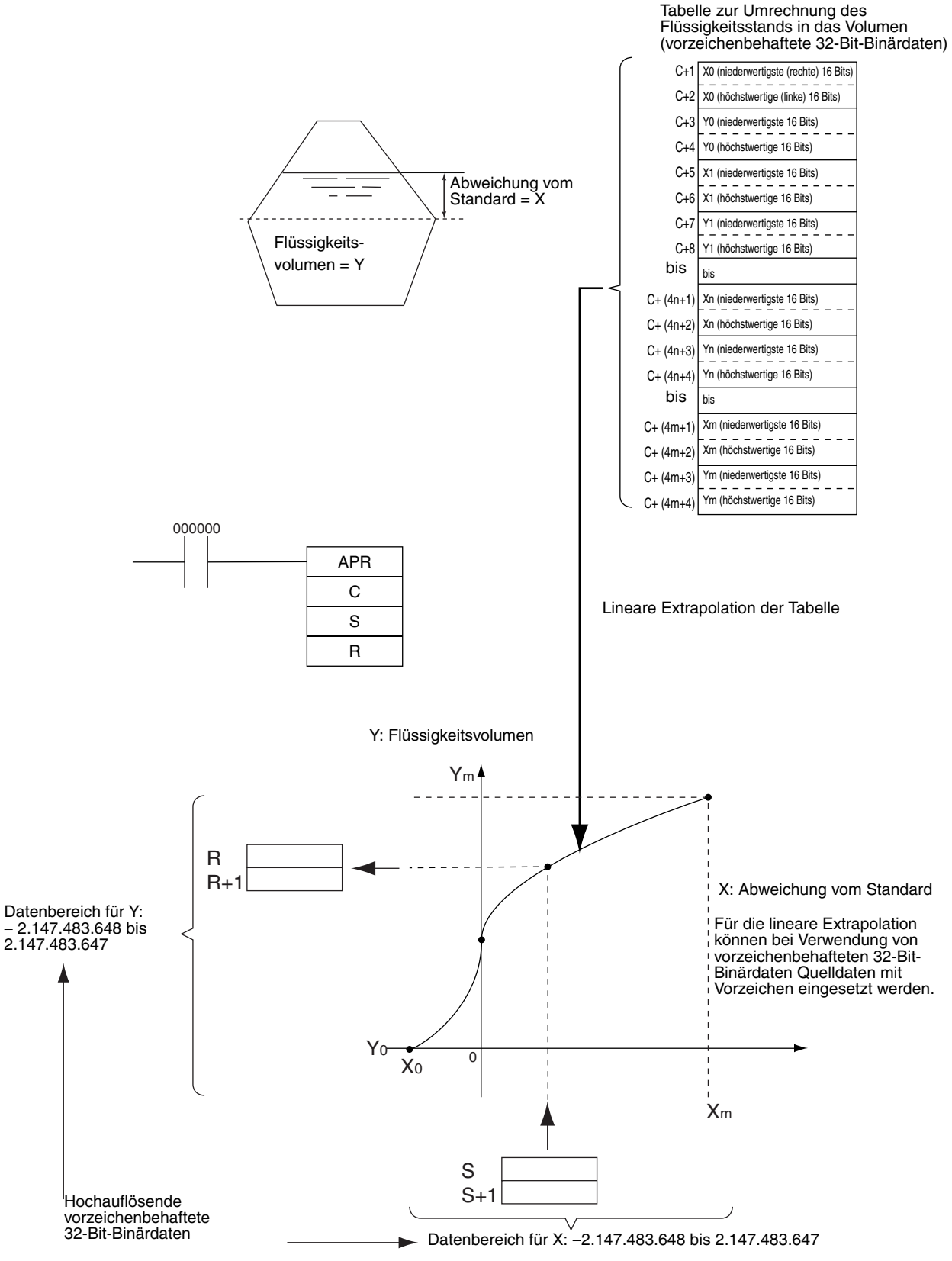
$$= 0F00 - (0086 \times 000F)$$

$$= 0726$$

Alle Werte sind Hexadezimalwerte (hex).

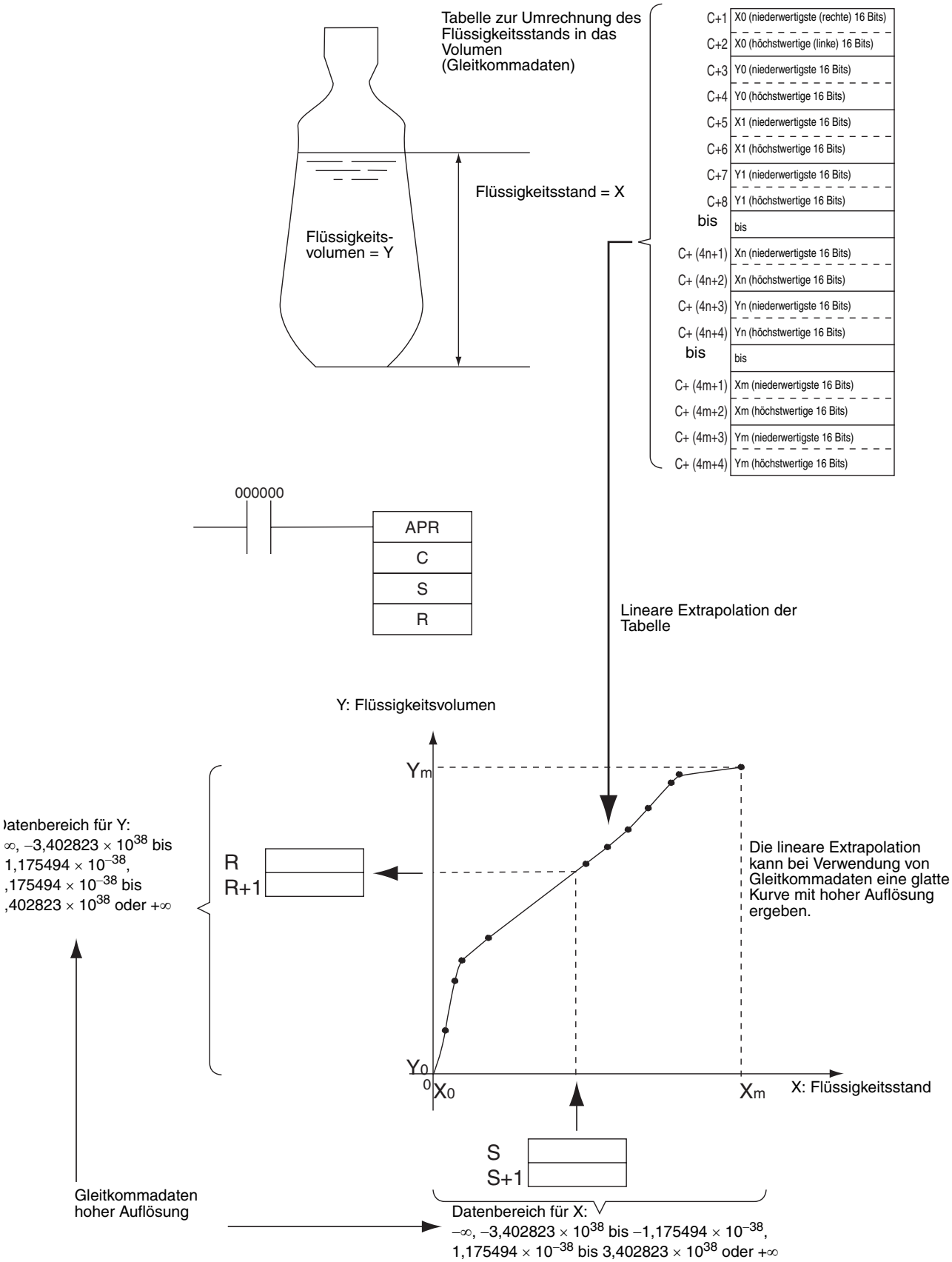
Lineare Extrapolation (C: Wort-Adresse)
Verwendung vorzeichenbehafteter 32-Bit-Binärdaten (nur CS1-H, CJ1-H, CJ1M und CS1D)

In diesem Beispiel wird der Befehl APR(069) verwendet, um den Flüssigkeitsstand in einem Behälter anhand der Form des Behälters in das Flüssigkeitsvolumen umzurechnen.



Lineare Extrapolation (C: Wort-Adresse)
Verwendung von Gleitkommadaten (nur CS1-H, CJ1-H, CJ1M und CS1D)

In diesem Beispiel wird der Befehl APR(069) verwendet, um den Flüssigkeitsstand in einem Behälter anhand der Form des Behälters in das Flüssigkeitsvolumen umzurechnen.

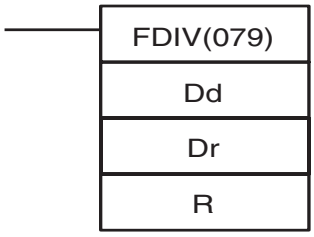


3-14-4 FLOATING POINT DIVIDE: FDIV(079)

Zweck

Dividiert eine 7-stellige Gleitkommazahl durch eine andere. Die Gleitkommazahlen werden in wissenschaftlicher Schreibweise ausgedrückt (7-stellige Mantisse und 1-stelliger Exponent).

Kontaktplansymbol



Dd: Erstes Dividendwort

Dr: Erstes Divisorwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FDIV(079)
	Ausführung einmal bei steigender Flanke	@FDIV(079)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

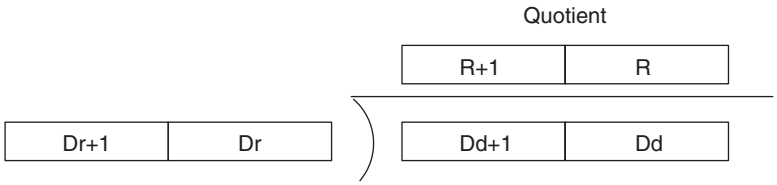
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

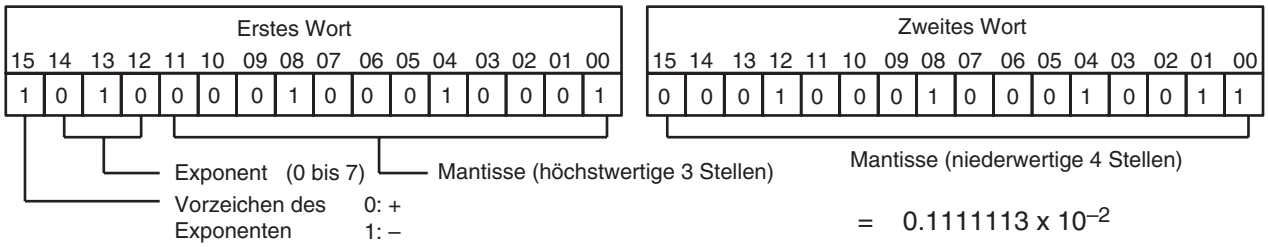
Bereich	Dd	Dr	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerker	H000 bis H510		
Systemmerker	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

FDIV(079) dividiert den Gleitkommawert in Dd und Dd+1 durch den in Dr und Dr+1 und schreibt das Ergebnis in R und R+1.



Als Schreibweise für Gleitkommawerte werden die sieben niederwertigen (rechten) Stellen für die Mantisse und die höchstwertige (linke) Stelle für den Exponenten verwendet, wie in der folgenden Abbildung zu ersehen. Die höchstwertige Stelle kann Werte zwischen 0 und F annehmen; positive Exponenten im Bereich von 0 bis 7 und negative Exponenten im Bereich von 8 bis F (0 bis –7). Die niederwertigen (rechten) Stellen müssen BCD-Werte enthalten.



Zwei weitere Beispiele für Gleitkommawerte:

6123 4567: 0,1234567 × 10^{6} (6 = 0110 binär)

B123 4567: 0,1234567 × 10^{-3} (B = 1011 binär)

In der folgenden Tabelle sind die zulässigen Maximal- und Minimalwerte aufgeführt.

Limit	8-stellig hexadezimal	Gleitkomma
Maximalwert	7999 9999	0,9999999 × 10^{7}
Minimalwert (Divisor und Dividend)	F000 0001	0,0000001 × 10^{-7}
Minimalwert (Ergebnis)	F100 0000	0,1000000 × 10^{-7}

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Mantisse (niederwertige 7 Stellen) in Dd+1 und Dd nicht in BCD vorliegt. EIN, wenn die Mantisse (niederwertige 7 Stellen) in Dr+1 und Dr nicht in BCD vorliegt. EIN, wenn der Divisor (Dr+1 und Dr) 0 ist. EIN, wenn das Ergebnis nicht im Bereich zwischen 0,1000000 × 10^{-7} und 0,9999999 × 10^{7} liegt. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

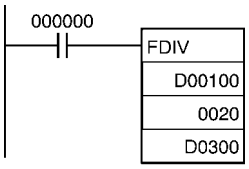
Das Ergebnis wird als Gleitkommawert ausgegeben, besitzt also 7 gültige Stellen. Die achte und weitere Stellen werden verworfen.

Das Ergebnis muss im Bereich zwischen 0,1000000 × 10^{-7} und 0,9999999 × 10^{7} liegen.

Beispiele

Grundlegende Gleitkommadivision

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, führt der Befehl FDIV(079) eine Division der Gleitkommazahl in D00101 und D00100 durch die Gleitkommazahl in CIO 0021 und CIO 0020 aus und schreibt das Ergebnis in D00301 und D00300.

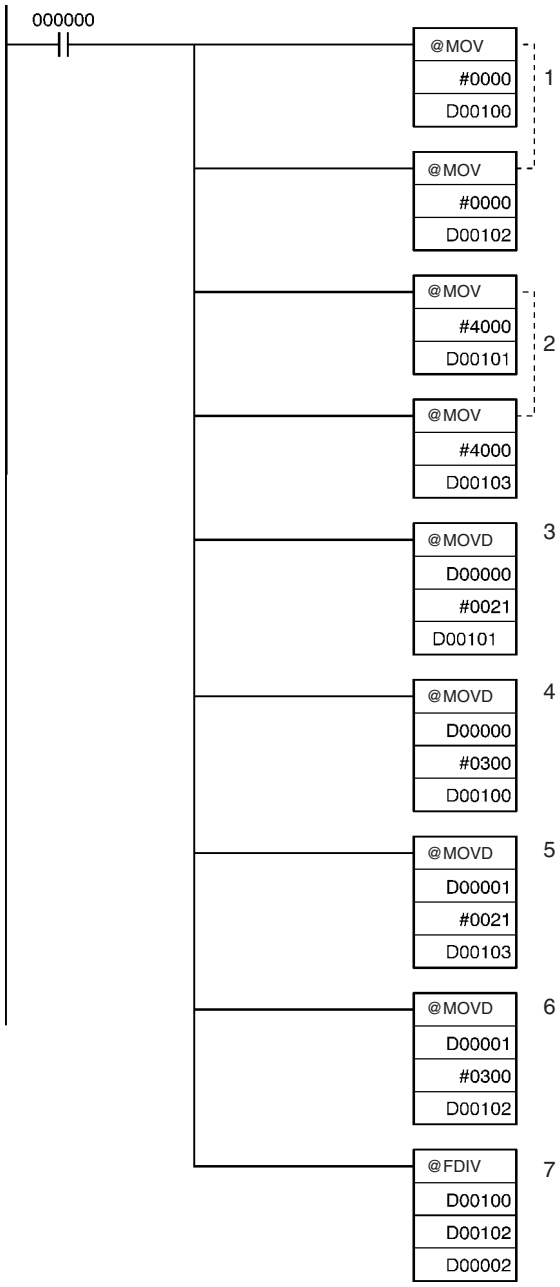


	D00101				D00100				
	A	5	6	7	0	0	0	0	0.5670000×10^{-2}
÷	CIO 0021				CIO 0020				
	B	1	2	3	4	5	6	7	0.1234567×10^{-3}
	D00301				D00300				
	2	4	5	9	2	7	0	3	0.4592703×10^{2}

Gleitkommawert-Division von zwei BCD-Zahlen

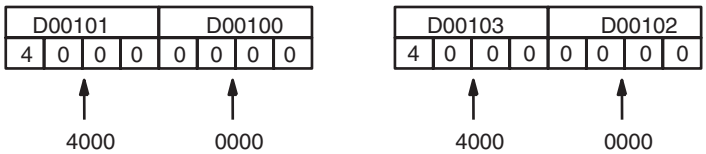
In diesem Beispiel wird die 4-stellige BCD-Zahl in D00000 durch die 4-stellige BCD-Zahl in D00001 dividiert und der Ergebnis-Gleitkommawert wird in D00003 und D00002 geschrieben.

Zur Durchführung der Gleitkommawert-Division wird der BCD-Wert in D00000 in das Gleitkommaformat konvertiert und in D00101 und D00100 geschrieben und der BCD-Wert in D00001 in das Gleitkommaformat konvertiert und in D00103 und D00102 geschrieben.

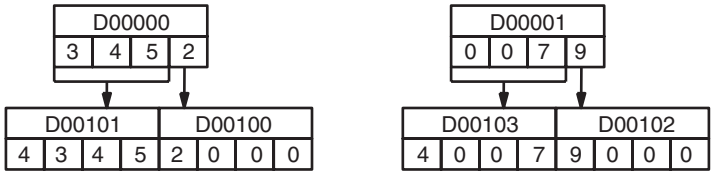


1,2,3... 1. D00100 und D00102 werden auf 0000 gesetzt.

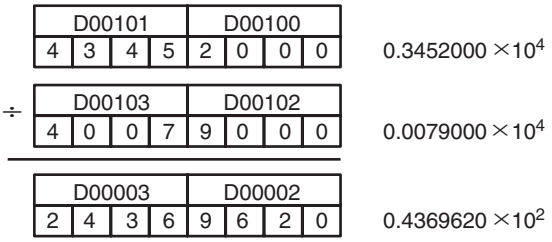
2. D00101 und D00103 werden auf 4000 gesetzt.



3. Der Befehl MOVD(083) wird verwendet, um die Stellen der ursprünglichen Quellenworte in die korrekten Stellen des 2-Wort-Gleitkommaformats zu übertragen.



4. Der Befehl FDIV(079) dividiert die Gleitkommazahl in D00101 und D00100 durch die Gleitkommazahl in D00103 und D00102.

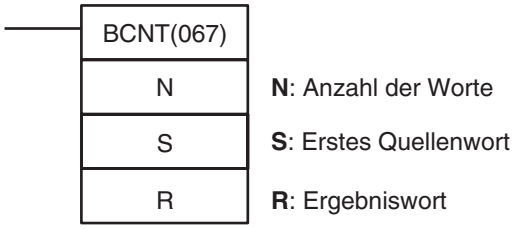


3-14-5 BIT COUNTER: BCNT(067)

Zweck

Zählt die Gesamtanzahl der auf EIN gesetzten Bits in den spezifizierten Worten.

Kontaktplansymbol



Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	BCNT(067)
	Ausführung einmal bei steigender Flanke	@BCNT(067)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Anzahl der Worte

Die Anzahl der Worte muss im Bereich zwischen 0001 und FFFF liegen (1 bis 65.535 Worte).

S: Erstes Quellenwort

S und S+(N–1) müssen im gleichen Datenbereich liegen.

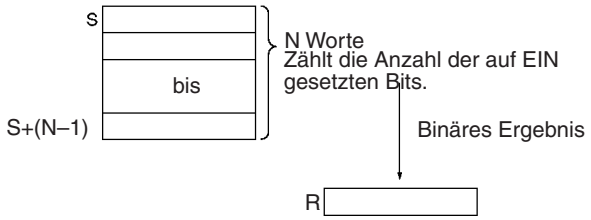
Operanden-Spezifikationen

Bereich	N	S	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerker	H000 bis H511		
Systemmerker	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		

Bereich	N	S	R
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0001 bis #FFFF (binär) oder &1 bis &65.535	---	
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl BCNT(067) zählt die Gesamtzahl der Bits im EIN-Zustand in allen Worten zwischen S und S+(N–1) und gibt das Ergebnis nach R aus.



Merker

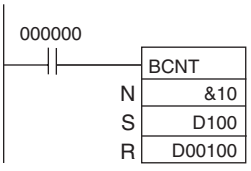
Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn N gleich 0000 ist. EIN, wenn das Ergebnis FFFF übersteigt. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.

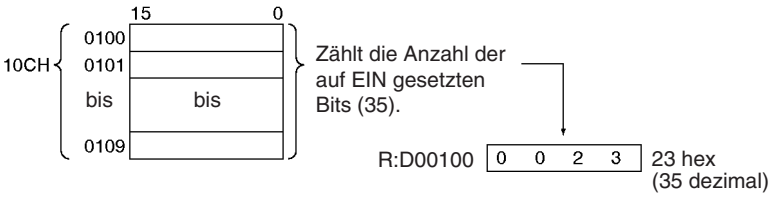
Vorsichtsmaßnahmen

Ein Fehler tritt auf, wenn N = 0000 oder wenn das Ergebnis FFFF übersteigt.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, zählt der Befehl BCNT(067) die Gesamtzahl der Bits im EIN-Zustand in den 10 Worten von CIO 0100 bis CIO 0109 und schreibt das Ergebnis in D00100.





3-15 Gleitkomma-Arithmetikbefehle

Die Gleitkomma-Arithmetikbefehle konvertieren Daten und führen Gleitkomma-Arithmetikoperationen aus. Die CPU-Baugruppen der CS/CJ-Serie unterstützen die folgenden Befehle.

Befehl	AWL	Funktionscode	Seite
FLOATING TO 16-BIT	FIX	450	563
FLOATING TO 32-BIT	FIXL	451	565
16-BIT TO FLOATING	FLT	452	566
32-BIT TO FLOATING	FLTL	453	568
FLOATING-POINT ADD	+F	454	570
FLOATING-POINT SUBTRACT	–F	455	572
FLOATING-POINT MULTIPLY	*F	456	574
FLOATING-POINT DIVIDE	/F	457	576
DEGREES TO RADIANS	RAD	458	578
RADIANS-TO-DEGREES	DEG	459	579
SINE	SIN	460	581
COSINE	COS	461	583
TANGENT	TAN	462	585
ARC SINE	ASIN	463	587
ARC COSINE	ACOS	464	589
ARC TANGENT	ATAN	465	591
SQUARE ROOT	SQRT	466	593
EXPONENT	EXP	467	595
LOGARITHM	LOG	468	597
EXPONENTIAL POWER	PWR	840	599

Zusätzlich zu den oben aufgeführten Befehlen unterstützen die CPU-Baugruppen der CS1-H/CJ1-H-Serie die folgenden Gleitkomma-Vergleichs- und Konvertierungsbefehle. Weitere Informationen zu Gleitkommabefehlen doppelter Präzision finden Sie unter *3-16-21 Symbolvergleichsbefehle für Gleitkommadaten doppelter Genauigkeit*.

Befehl	AWL	Funktionscode	Seite
Symbolvergleichsbefehle für Gleitkommadaten einfacher Genauigkeit (*nur CS1-H/CJ1-H/CJ1M)	LD, AND, OR + =F, <>F, <F, <=F, >F oder >=F	329 bis 334	600
FLOATING-POINT TO ASCII (*nur CS1-H/CJ1-H/CJ1M)	FSTR	448	604
ASCII TO FLOATING-POINT (*nur CS1-H/CJ1-H/CJ1M)	FVAL	449	609

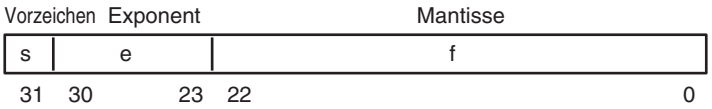
Datenformat

Gleitkommadaten drücken reelle Zahlen mittels Vorzeichen, Exponent und Mantisse aus. Wenn Daten im Gleitkommaformat ausgedrückt werden, gilt die folgende Formel.

Reelle Zahl = $(-1)^s\ 2^{e-127}\ (1.f)$

s: Vorzeichen
e: Exponent
f: Mantisse

Das Gleitkommadatenformat entspricht der Norm IEEE754. Daten werden wie folgt in 32 Bits ausgedrückt:



Daten	Anzahl Bits	Inhalt
s: Vorzeichen	1	0: positiv; 1: negativ
e: Exponent	8	Die Werte für den Exponenten (e) haben einen Bereich von 0 bis 255. Der tatsächliche Exponent ist der Wert, der erhalten wird, indem 127 von e subtrahiert wird, wodurch sich ein Bereich von -127 bis 128 ergibt. „e = 0“ und „e = 255“ drücken spezielle Zahlen aus.
f: Mantisse	23	Der Mantissenteil der binären Gleitkommadaten entspricht dem Format 2,0 > 1,f ≥1,0.

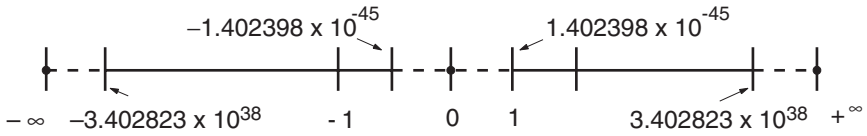
Anzahl der Stellen

Die Anzahl der effektiven Stellen von Gleitkommadaten beträgt 24 Bits im Binärformat (ca. sieben Stellen in dezimal).

Gleitkommadaten

Die folgenden Daten können durch Gleitkommadaten ausgedrückt werden:

- $-\infty$
- $-3{,}402823 \times 10^{38} \leq \text{Wert} \leq -1{,}402398 \times 10^{-45}$
- 0
- $1{,}402398 \times 10^{-45} \leq \text{Wert} \leq 3{,}402823 \times 10^{38}$
- $+\infty$
- Keine Zahl (Not a Number, NaN)



Spezielle Zahlen

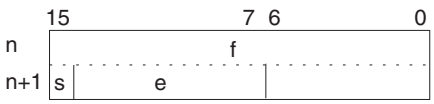
Die Formate für NaN, $\pm\infty$ und 0 sind wie folgt:

NaN*: e = 255, f ≠ 0
+∞: e = 255, f = 0, s = 0
–∞: e = 255, f = 0, s = 1
0: e = 0

*NaN (keine Zahl) ist keine gültige Gleitkommazahl. Die Ausführung von Gleitkomma-Rechenbefehlen kann nicht zu dem Ergebnis NaN führen.

Schreiben von Gleitkommadaten

Wenn in der E/A-Speicher-Bearbeitungsansicht von CX-Programmer Gleitkommadaten als Datenformat spezifiziert sind, werden die standardmäßig in diesem Fenster eingegebenen Dezimalzahlen automatisch in das zuvor beschriebene Gleitkommaformat (IEEE754-Format) konvertiert und in den E/A-Speicher geschrieben. Im IEEE754-Format geschriebene Daten werden zur Überwachung in der Anzeige automatisch in das Standard-Dezimalformat konvertiert.



Es ist für den Benutzer beim Lesen und Schreiben von Gleitkommadaten nicht erforderlich, das IEEE754-Datenformat zu beachten. Es ist nur erforderlich daran zu denken, dass Gleitkommadaten jeweils 2 Worte belegen.

Als Gleitkommawerte ausgedrückte Zahlen

Die folgenden Arten von Gleitkommazahlen können verwendet werden.

Mantisse (f)	Exponent (e)		
	0	Nicht 0 und nicht alle Bits 1	Alle Bits 1 (255)
0	0	Normalisierte Zahlen	Unendlich
Nicht 0	Nicht normalisierte Zahlen		NaN

Hinweis Eine nicht normalisierte Zahl ist eine Zahl, deren Absolutwert zu klein ist, um als normalisierte Zahl ausgedrückt zu werden. Nicht normalisierte Zahlen besitzen weniger gültige Stellen. Wenn das Ergebnis von Berechnungen (einschließlich Zwischenergebnissen) eine nicht normalisierte Zahl ergibt, wird die Anzahl der gültigen Stellen reduziert.

Normalisierte Zahlen

Normalisierte Zahlen stellen reelle Zahlen dar. Das Vorzeichenbit ist bei einer positiven Zahl auf 1, bei einer negativen Zahl auf 0 gesetzt.

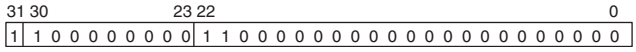
Der Exponent (e) wird durch eine Zahl zwischen 1 und 254 ausgedrückt. Der tatsächliche Exponent ist diese Zahl minus 127, also –126 bis 127.

Die Mantisse (f) wird durch eine Zahl zwischen 0 und $2^{33} – 1$ ausgedrückt, und es wird davon ausgegangen, dass Bit 2^{33} der tatsächlichen Mantisse gleich 1 ist und das Dezimalkomma unmittelbar dahinter steht.

Normalisierte Zahlen werden wie folgt ausgedrückt:

$(–1)^{(\text{Vorzeichen s})} \times 2^{(\text{Exponent e})–127} \times (1 + \text{Mantisse} \times 2^{–23})$

Beispiel



Vorzeichen: –
Exponent: 128 – 127 = 1
Mantisse: $1 + (2^{22} + 2^{21}) \times 2^{–23} = 1 + (2^{–1} + 2^{–2}) = 1 + 0{,}75 = 1{,}75$
Wert: $–1{,}75 \times 2^{1} = –3{,}5$

Nicht normalisierte Zahlen

Nicht normalisierte Zahlen stellen reelle Zahlen mit sehr kleinem Absolutwert dar. Das Vorzeichenbit ist bei einer positiven Zahl auf 1, bei einer negativen Zahl auf 0 gesetzt.

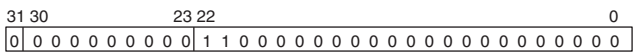
Der Exponent (e) ist 0 und der tatsächliche Exponent ist –126.

Die Mantisse (f) wird durch eine Zahl zwischen 1 und $2^{33} – 1$ ausgedrückt, und es wird davon ausgegangen, dass Bit 2^{33} der tatsächlichen Mantisse gleich 0 ist und das Dezimalkomma unmittelbar dahinter steht.

Nicht normalisierte Zahlen werden wie folgt ausgedrückt:

$(–1)^{(\text{Vorzeichen s})} \times 2^{–126} \times (\text{Mantisse} \times 2^{–23})$

Beispiel



Vorzeichen: –
Exponent: –126
Mantisse: $0 + (2^{22} + 2^{21}) \times 2^{–23} = 0 + (2^{–1} + 2^{–2}) = 0 + 0{,}75 = 0{,}75$
Wert: $–0{,}75 \times 2^{–126}$

Null

Werte von +0,0 und –0,0 können ausgedrückt werden, indem das Vorzeichen für positiv auf 1 und für negativ auf 0 gesetzt wird. Der Exponent und die Mantisse haben in diesem Fall beide den Wert 0. Sowohl +0,0 als auch –0,0 entsprechen 0,0. Beschreibungen der Unterschiede, die sich durch ein Vorzeichen beim Wert 0,0 ergeben, finden Sie nachfolgend unter *Ergebnisse der Gleitkommaarithmetik*.

Unendlich

Werte von $+\infty$ und $–\infty$ können ausgedrückt werden, indem das Vorzeichen für positiv auf 1 und für negativ auf 0 gesetzt wird. Der Exponent hat in diesem Fall den Wert 255 ($2^{8} – 1$) und die Mantisse den Wert 0.

NaN

NaN (keine Zahl) ergibt sich, wenn das Ergebnis einer Berechnung, wie z. B. 0,0/0,0, ∞/∞ oder ∞–∞, keiner Zahl und nicht unendlich entspricht. Der Exponent hat in diesem Fall den Wert 255 ($2^8 - 1$) und die Mantisse ist nicht 0.

Hinweis Es gibt keine Spezifikationen hinsichtlich des Vorzeichens von NaN oder des Werts im Mantissenfeld (außer, dass sie nicht 0 sein darf).

Ergebnisse der Gleitkommaarithmetik

Rundung von Ergebnissen

Die folgenden Methoden werden zur Rundung von Ergebnissen angewandt, wenn die Anzahl der Stellen im genauen Ergebnis der Gleitkommaarithmetik die Anzahl der gültigen Stellen von internen Verarbeitungsausdrücken überschreitet.

Wenn das Ergebnis nahe bei einem von zwei internen Gleitkommaausdrükken liegt, wird der näher liegende Ausdruck verwendet. Wenn das Ergebnis genau in der Mitte zwischen zwei Gleitkommaausdrücken liegt, wird das Ergebnis so gerundet, dass die letzte Stelle der Mantisse eine 0 ist.

Überläufe, Unterläufe und ungültige Berechnungen

Ein Überlauf wird, je nach Vorzeichen des Ergebnisses, entweder als positiv oder negativ Unendlich ausgegeben. Ein Unterlauf wird, je nach Vorzeichen des Ergebnisses, entweder als positive oder negative Null ausgegeben.

Ungültige (nicht definierte) Berechnungen führen zu dem Ergebnis NaN. Zu den ungültigen Berechnungen gehört die Addition von Unendlich zu einer Zahl mit gegengesetztem Vorzeichen, die Subtraktion von Unendlich von einer Zahl mit gegengesetztem Vorzeichen, die Multiplikation von Null und Unendlich, die Division von Null durch Null oder die Division von Unendlich durch Unendlich.

Beim Auftreten eines Überlaufs ist das Ergebnis nach der Konvertierung einer Gleitkommazahl in eine Ganzzahl möglicherweise nicht korrekt.

Vorsichtsmaßnahmen beim Umgang mit speziellen Werten

Die folgenden Vorsichtsmaßnahmen gelten für den Umgang mit Null, Unendlich und NaN.

- Die Summe aus positiver Null und negativer Null ist eine positive Null.
- Die Differenz von zwei Nullen mit gleichem Vorzeichen ist eine positive Null.
- Wenn eine beliebiger Operand NaN ist, so ist auch das Ergebnis NaN.
- Die positive Null und die negative Null werden bei Vergleichen als äquivalent behandelt.
- Vergleiche oder Äquivalenzprüfungen mit einer oder mehreren NaN haben beim Befehl != immer wahr als Ergebnis und bei allen anderen Befehlen stets falsch.

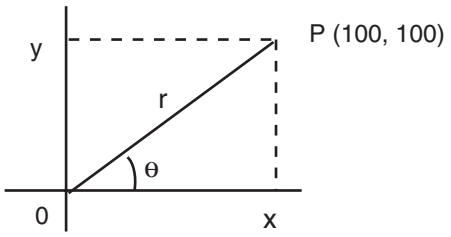
Ergebnisse von Gleitkommaberechnungen

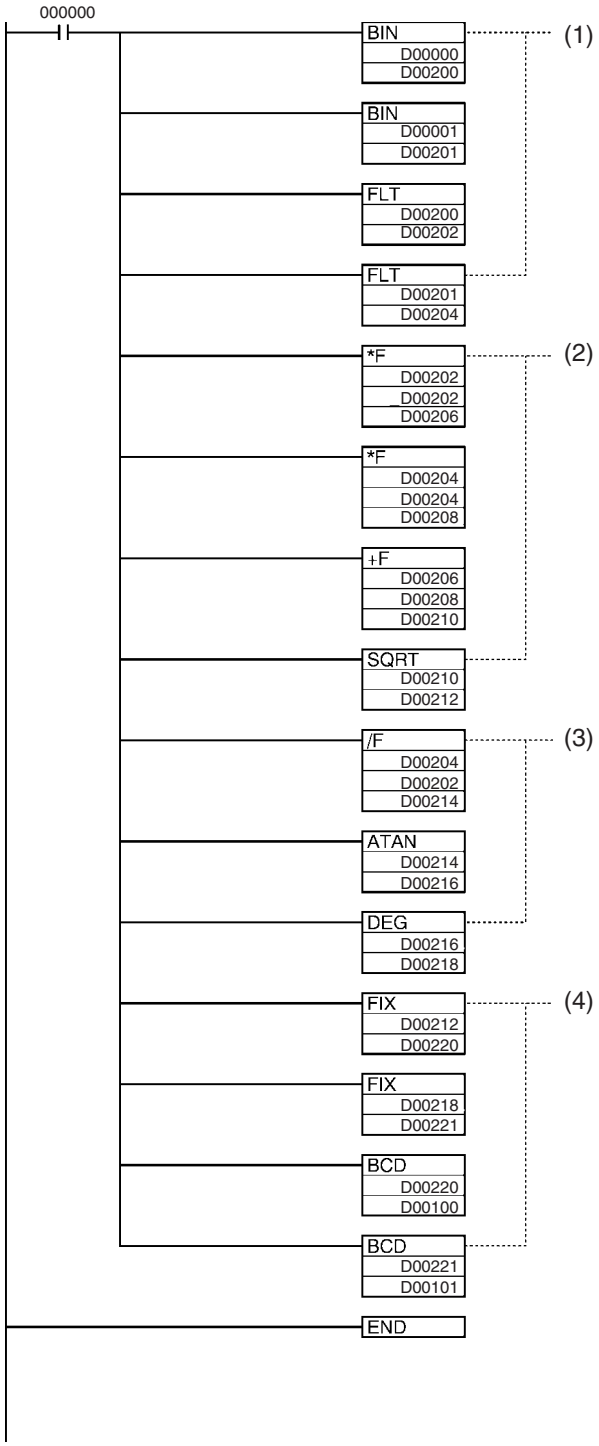
Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben. Ist das Ergebnis positiv, wird es als +∞ ausgegeben, ist es negativ, als –∞.

Der Gleich-Merker wird nur auf EIN gesetzt, wenn sowohl der Exponent (e) als auch die Mantisse (f) nach der Berechnung gleich Null sind. Ein Berechnungsergebnis wird auch als Null ausgegeben, wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich mit Gleitkommadaten ausdrücken lässt. In diesem Fall wird der Unterlaufmerker auf EIN gesetzt.

Beispiel

In diesem Programmbeispiel werden die Koordinaten von X-Achse und Y-Achse (x, y) als 4-stelliger BCD-Inhalt von D00000 und D00001 zur Verfügung gestellt. Der Abstand (r) vom Ursprung und der Winkel (θ, in Grad) werden ermittelt und nach D00100 und D00101 ausgegeben. Im Ergebnis werden alle Nachkommastellen abgeschnitten.


y
P (100, 100)
r
θ
0
x


000000
BIN
D00000
D00200
(1)
BIN
D00001
D00201
FLT
D00200
D00202
FLT
D00201
D00204
*F
D00202
D00202
D00206
(2)
*F
D00204
D00204
D00208
+F
D00206
D00208
D00210
SQRT
D00210
D00212
/F
D00204
D00202
D00214
(3)
ATAN
D00214
D00216
DEG
D00216
D00218
FIX
D00212
D00220
(4)
FIX
D00218
D00221
BCD
D00220
D00100
BCD
D00221
D00101
END

Berechnung

Abstand r = $\sqrt{x^2 + y^2}$

Winkel = $\tan^{-1}\left(\frac{y}{x}\right)$

Beispiel

Abstand r = $\sqrt{100^2 + 100^2} = 141{,}4214$

Winkel = $\tan^{-1}\left(\frac{100}{100}\right) = 45{,}0$

Inhalt von DM

D00000	#0100 (BCD)	x	→	D00100	0 1 4 1 (BCD) r
D00001	#0100 (BCD)	y		D00101	0 0 4 5 (BCD)

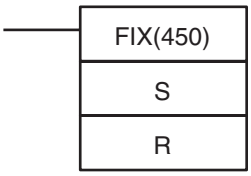
1. Dieser Abschnitt des Programms konvertiert die Daten aus dem BCD-Format in Gleitkommadaten.
   a) Der Datenbereich ab D00200 aufwärts wird als Arbeitsbereich verwendet.
   b) Zunächst wird der Befehl BIN(023) verwendet, um die BCD-Daten temporär in Binärdaten zu konvertieren, anschließend der Befehl FLT(452), um die Binärdaten in Gleitkommadaten zu konvertieren.
   c) Der in Gleitkommadaten konvertierte Wert von x wird nach D00203 und D00202 ausgegeben.
   d) Der in Gleitkommadaten konvertierte Wert von y wird nach D00205 und D00204 ausgegeben.
2. Zur Ermittlung des Abstands r werden Gleitkomma-Arithmetikbefehle verwendet, um die Quadratwurzel von x^2+y^2 zu berechnen. Das Ergebnis wird als Gleitkommadaten nach D00213 und D00212 ausgegeben.
3. Zur Ermittlung des Winkels θ werden Gleitkomma-Arithmetikbefehle verwendet, um $\tan^{-1}$ (y/x) zu berechnen. Der Befehl ATAN(465) gibt ein Ergebnis in Radianten aus. Daher wird der Befehl DEG(459) zur Konvertierung des Ergebnisses in Grad verwendet. Das Ergebnis wird als Gleitkommadaten nach D00219 und D00218 ausgegeben.
4. Das Ergebnis wird wieder aus Gleitkommadaten in BCD rückkonvertiert.
   a) Zunächst wird der Befehl FIX(450) verwendet, um die Gleitkommadaten temporär in Binärdaten zu konvertieren, anschließend der Befehl BCD(024), um die Binärdaten in BCD-Daten zu konvertieren.
   b) Der Abstand r wird nach D00100 ausgegeben.
   c) Der Winkel θ wird nach D00101 ausgegeben.

3-15-1 FLOATING TO 16-BIT: FIX(450)

Zweck

Konvertiert einen 32-Bit-Gleitkommawert in 16-Bit-Binärdaten mit Vorzeichen, und gibt das Ergebnis in das spezifizierte Ergebniswort aus.

Kontaktplansymbol



S: Erstes Quellenwort

R: Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FIX(450)
	Ausführung einmal bei steigender Flanke	@FIX(450)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

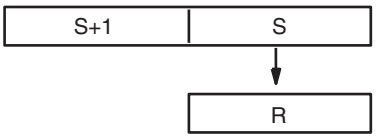
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W510	W000 bis W511
Haftmerkerbereich	H000 bis H510	H000 bis H511
Zusatz-Systembereich	A000 bis A958	A448 bis A959
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	DR0 bis DR15
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl FIX(450) konvertiert den ganzzahligen Teil der 32-Bit-Gleitkommazahl in S+1 und S (IEEE754-Format) in 16-Bit-Binärdaten mit Vorzeichen und schreibt das Ergebnis in R.



Gleitkommadaten (32 Bits)

Binärdaten mit Vorzeichen (16 Bits)

Nur der ganzzahlige Teil der Gleitkommazahl wird konvertiert, der Dezimalbruchteil wird abgeschnitten. Der ganzzahlige Teil der Gleitkommazahl muss im Bereich zwischen –32.768 und 32.767 liegen.

Beispiele für Konvertierungen:
Ein Gleitkommawert von 3,5 wird zu 3 konvertiert.
Ein Gleitkommawert von –3,5 wird zu –3 konvertiert.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Daten in S+1 und S keine Zahl (NaN) sind. EIN, wenn der ganzzahlige Teil von S+1 und S nicht innerhalb des Bereichs von –32.768 bis 32.767 liegt. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn Bit 15 des Ergebnisses EIN ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

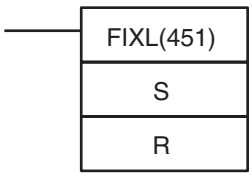
Beim Inhalt von S+1 und S muss es sich um Gleitkommadaten handeln, deren ganzzahliger Teil im Bereich zwischen –32.768 und 32.767 liegt.

3-15-2 FLOATING TO 32-BIT: FIXL(451)

Zweck

Konvertiert einen 32-Bit-Gleitkommawert in 32-Bit-Binärdaten mit Vorzeichen, und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



S: Erstes Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FIXL(451)
	Ausführung einmal bei steigender Flanke	@FIXL(451)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

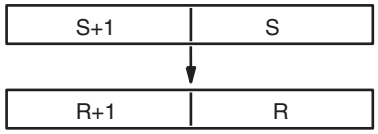
Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	

Bereich	S	R
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl FIXL(451) konvertiert den ganzzahligen Teil der 32-Bit-Gleitkommazahl in S+1 und S (IEEE754-Format) in 32-Bit-Binärdaten mit Vorzeichen und schreibt das Ergebnis in R+1 und R.



Gleitkommadaten (32 Bits)

Binärdaten mit Vorzeichen (32 Bits)

Nur der ganzzahlige Teil der Gleitkommazahl wird konvertiert, der Dezimalbruchteil wird abgeschnitten. (Der ganzzahlige Teil der Gleitkommazahl muss im Bereich zwischen -2.147.483.648 und 2.147.483.647 liegen.)

Beispiele für Konvertierungen:
Ein Gleitkommawert von 2.147.483.640,5 wird zu 2.147.483.640 konvertiert.
Ein Gleitkommawert von -214.748.340,5 wird zu -214.748.340 konvertiert.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Daten in S+1 und S keine Zahl (NaN) sind. EIN, wenn der ganzzahlige Teil von S+1 und S nicht innerhalb des Bereichs von -2.147.483.648 bis 2.147.483.647 liegt. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0000 0000 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn Bit 15 von R+1 nach der Ausführung auf EIN gesetzt ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

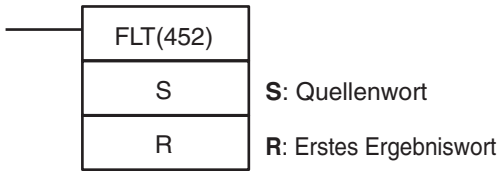
Beim Inhalt von S+1 und S muss es sich um Gleitkommadaten handeln, deren ganzzahliger Teil im Bereich zwischen -2.147.483.648 und 2.147.483.647 liegt.

3-15-3 16-BIT TO FLOATING: FLT(452)

Zweck

Konvertiert einen 16-Bit-Binärwert mit Vorzeichen in 32-Bit-Gleitkommadaten und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



S: Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FLT(452)
	Ausführung einmal bei steigender Flanke	@FLT(452)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

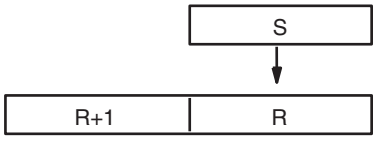
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142
Arbeitsbereich	W000 bis W511	W000 bis W510
Haftmerkerbereich	H000 bis H511	H000 bis H510
Zusatz-Systembereich	A000 bis A959	A448 bis A958
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094
Zählerbereich	C0000 bis C4095	C0000 bis C4094
DM-Bereich	D00000 bis D32767	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #FFFF (binär)	---
Datenregister	DR0 bis DR15	---
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

FLT(452) konvertiert die 16-Bit-Binärdaten mit Vorzeichen in S zu 32-Bit-Gleitkommadaten (IEEE754-Format) und schreibt das Ergebnis in R+1 und R. Bei der resultierenden Gleitkommazahl wird eine einzelne 0 als Nachkommastelle hinzugefügt.



Binärdaten mit Vorzeichen (16 Bits)

Gleitkommadaten (32 Bits)

Für S können nur Werte innerhalb des Bereichs von –32.768 bis 32.767 spezifiziert werden. Zur Konvertierung von Binärdaten außerhalb dieses Bereich kann der Befehl FLTL(453) verwendet werden.

Beispiele für Konvertierungen:
Ein vorzeichenbehafteter Binärwert von 3 wird zu 3,0 konvertiert.
Ein vorzeichenbehafteter Binärwert von –3 wird zu –3,0 konvertiert.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

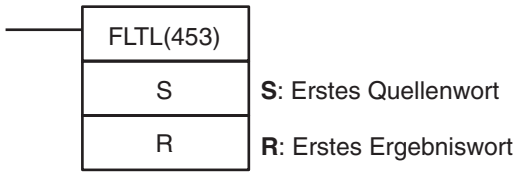
Beim Inhalt von S muss sich um vorzeichenbehaftete Binärdaten mit einem Wert (dezimal) im Bereich von –32.768 bis 32.767 handeln.

3-15-4 32-BIT TO FLOATING: FLTL(453)

Zweck

Konvertiert einen 32-Bit-Binärwert mit Vorzeichen in 32-Bit-Gleitkommadaten und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	FLTL(453)
	Ausführung einmal bei steigender Flanke	@FLTL(453)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

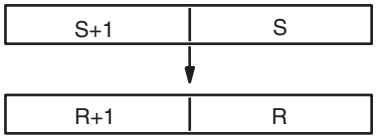
Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	

Bereich	S	R
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

FLTL(453) konvertiert die 32-Bit-Binärdaten mit Vorzeichen in S+1 und S zu 32-Bit-Gleitkommadaten (IEEE754-Format) und schreibt das Ergebnis in R+1 und R. Bei der resultierenden Gleitkommazahl wird eine einzelne 0 als Nachkommastelle hinzugefügt.



Binärdaten mit Vorzeichen (32 Bits)

Gleitkommadaten (32 Bits)

Für S+1 und S können vorzeichenbehaftete Binärdaten im Bereich von –2.147.483.648 bis 2.147.483.647 spezifiziert werden. Der Gleitkommawert hat 24 wertige Binärstellen (Bits). Das Ergebnis ist nicht genau, wenn durch den Befehl FLTL(453) eine Zahl konvertiert wird, die größer als 16.777.215 (der Maximalwert, der sich mit 24 Bits ausdrücken lässt) ist.

Beispiele für Konvertierungen:

Ein vorzeichenbehafteter Binärwert von 16.777.215 wird zu 16.777.215,0 konvertiert.

Ein vorzeichenbehafteter Binärwert von –16.777.215 wird zu –15.777.215,0 konvertiert.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

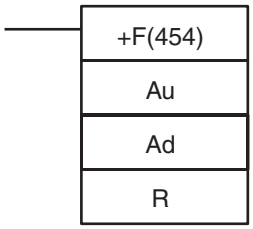
Das Ergebnis ist nicht genau, wenn eine Zahl konvertiert wird, deren Absolutwert größer als 16.777.215 (der Maximalwert, der sich mit 24 Bits ausdrücken lässt) ist.

3-15-5 FLOATING-POINT ADD: +F(454)

Zweck

Addiert zwei 32-Bit-Gleitkommazahlen und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



Au: 1. Summand, erstes Wort

AD: 2. Summand, erstes Wort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	+F(454)
	Ausführung einmal bei steigender Flanke	@+F(454)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

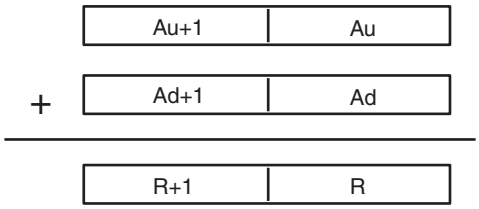
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Au	Ad	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerkerbereich	H000 bis H510		
Zusatz-Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

+F(454) addiert die 32-Bit-Gleitkommazahl in Ad+1 und Ad zu der 32-Bit-Gleitkommazahl in Au+1 und Au und schreibt das Ergebnis in R+1 und R. (Die Gleitkommadaten müssen im IEEE754-Format vorliegen.)



Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Die verschiedenen Kombinationen von Daten für den 1. und 2. Summanden führen zu den in der folgenden Tabelle aufgeführten Ergebnissen.

	1. Summand				
2. Summand	**0**	**Zahl**	**+∞**	**–∞**	**NaN**
0	0	Zahl	+∞	–∞	
Zahl	Zahl	Siehe Hinweis 1	+∞	–∞	
+∞	+∞	+∞	+∞	Siehe Hinweis 2	
–∞	–∞	–∞	Siehe Hinweis 2	–∞	
NaN					Siehe Hinweis 2

Hinweis

1. Das Ergebnis kann Null (einschließlich Unterlauf), eine Zahl, +∞ oder –∞ sein.
2. Der Fehlermerker wird auf EIN gesetzt und der Befehl wird nicht ausgeführt.

Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn die Daten für den 1. oder 2. Summanden nicht als Gleitkommadaten erkannt werden. EIN, wenn die Daten für den 1. oder 2. Summanden keine Zahl (NaN) sind. EIN, wenn +∞ und –∞ addiert werden. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

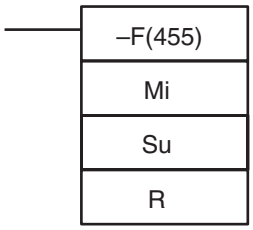
Die Daten des 1. Summanden (Au+1 und Au) und des 2. Summanden (Ad+1 und Ad) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-6 FLOATING-POINT SUBTRACT: –F(455)

Zweck

Subtrahiert eine 32-Bit-Gleitkommazahl von einer anderen und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	–F(455)
	Ausführung einmal bei steigender Flanke	@–F(455)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

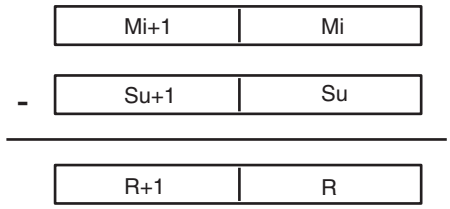
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Mi	Su	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerkerbereich	H000 bis H510		
Zusatz-Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

–F(455) subtrahiert die 32-Bit-Gleitkommazahl in Su+1 und Su von der 32-Bit-Gleitkommazahl in Mi+1 und Mi und schreibt das Ergebnis in R+1 und R. (Die Gleitkommadaten müssen im IEEE754-Format vorliegen.)



Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Die verschiedenen Kombinationen von Daten für Minuend und Subtrahend führen zu den in der folgenden Tabelle aufgeführten Ergebnissen.

	Minuend				
Subtrahend	**0**	**Zahl**	**+∞**	**–∞**	**NaN**
0	0	Zahl	+∞	–∞	
Zahl	Zahl	Siehe Hinweis 1	+∞	–∞	
+∞	–∞	–∞	Siehe Hinweis 2	–∞	
–∞	+∞	+∞	+∞	Siehe Hinweis 2	
NaN					Siehe Hinweis 2

Hinweis

1. Das Ergebnis kann Null (einschließlich Unterlauf), eine Zahl, +∞ oder –∞ sein.
2. Der Fehlermerker wird auf EIN gesetzt und der Befehl wird nicht ausgeführt.

Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn die Minuenden- und Subtrahendendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn der Minuend oder Subtrahend keine Zahl (NaN) ist. EIN, wenn +∞ von +∞ subtrahiert wird. EIN, wenn –∞ von –∞ subtrahiert wird. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

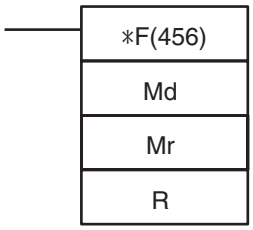
Die Minuendendaten (Mi+1 und Mi) und die Subtrahendendaten (Su+1 und Su) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-7 FLOATING-POINT MULTIPLY: ∗F(456)

Zweck

Multipliziert zwei 32-Bit-Gleitkommazahlen miteinander und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



Md: Erstes Multiplikandenwort

Mr: Erstes Multiplikatorwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	*F(456)
	Ausführung einmal bei steigender Flanke	@*F(456)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

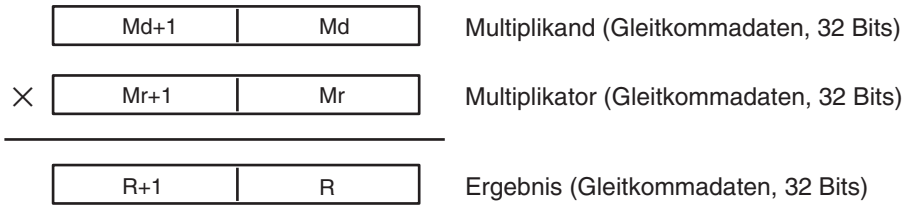
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Md	Mr	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerkerbereich	H000 bis H510		
Zusatz-Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

∗F(456) multipliziert die 32-Bit-Gleitkommazahl in Md+1 und Md mit der 32-Bit-Gleitkommazahl in Mr+1 und Mr und schreibt das Ergebnis in R+1 und R. (Die Gleitkommadaten müssen im IEEE754-Format vorliegen.)



Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Die verschiedenen Kombinationen von Multiplikanden- und Multiplikatordaten führen zu den in der folgenden Tabelle aufgeführten Ergebnissen.

	Multiplikand				
Multiplikator	**0**	**Zahl**	**+∞**	**–∞**	**NaN**
0	0	0	Siehe Hinweis 2	Siehe Hinweis 2	
Zahl	0	Siehe Hinweis 1	+/–∞	+/–∞	
+∞	Siehe Hinweis 2	+/–∞	+∞	–∞	
–∞	Siehe Hinweis 2	+/–∞	–∞	+∞	
NaN					Siehe Hinweis 2

Hinweis

1. Das Ergebnis kann Null (einschließlich Unterlauf), eine Zahl, +∞ oder –∞ sein.
2. Der Fehlermerker wird auf EIN gesetzt und der Befehl wird nicht ausgeführt.

Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn die Multiplikanden- oder Multiplikatordaten nicht als Gleitkommadaten erkannt werden. EIN, wenn Multiplikand oder Multiplikator keine Zahl (NaN) ist. EIN, wenn +∞ und 0 miteinander multipliziert werden. EIN, wenn –∞ und 0 miteinander multipliziert werden. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

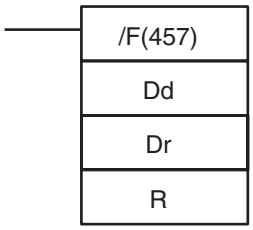
Die Daten von Multiplikand (Md+1 und Md) und Multiplikator (Mr+1 und Mr) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-8 FLOATING-POINT DIVIDE: /F(457)

Zweck

Dividiert eine 32-Bit-Gleitkommazahl durch eine andere und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



Dd: Erstes Dividendwort

Dr: Erstes Divisorwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	/F(457)
	Ausführung einmal bei steigender Flanke	@/F(457)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

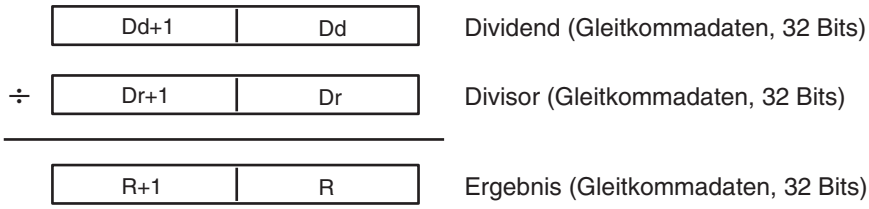
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	Dd	Dr	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerkerbereich	H000 bis H510		
Zusatz-Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

/F(457) dividiert die 32-Bit-Gleitkommazahl in Dd+1 und Dd durch die 32-Bit-Gleitkommazahl in Dr+1 und Dr und schreibt das Ergebnis in R+1 und R. (Die Gleitkommadaten müssen im IEEE754-Format vorliegen.)



Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Die verschiedenen Kombinationen von Daten für Dividend und Divisor führen zu den in der folgenden Tabelle aufgeführten Ergebnissen.

	Dividend				
Divisor	**0**	**Zahl**	**+∞**	**–∞**	**NaN**
0	Siehe Hinweis 3	+/–∞	+∞	–∞	
Zahl	0	Siehe Hinweis 1	+/–∞	+/–∞	
+∞	0	Siehe Hinweis 2	Siehe Hinweis 3	Siehe Hinweis 3	
–∞	0	Siehe Hinweis 2	Siehe Hinweis 3	Siehe Hinweis 3	
NaN					Siehe Hinweis 3

Hinweis

1. Das Ergebnis kann Null (einschließlich Unterlauf), eine Zahl, +∞ oder –∞ sein.
2. Bei einem Unterlauf ist das Ergebnis Null.
3. Der Fehlermerker wird auf EIN gesetzt und der Befehl wird nicht ausgeführt.

Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn die Dividend- und Divisordaten nicht als Gleitkommadaten erkannt werden. EIN, wenn der Dividend oder der Divisor keine Zahl (NaN) ist. EIN, wenn Dividend und Divisor beide 0 sind. EIN, wenn Dividend und Divisor beide +∞ oder –∞ sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

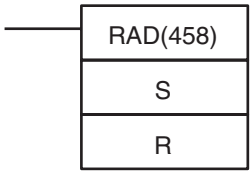
Die Daten von Dividend (Dd+1 und Dd) und Divisor (Dr+1 und Dr) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-9 DEGREES TO RADIANS: RAD(458)

Zweck

Konvertiert eine 32-Bit-Gleitkommazahl von Grad zu Radianten und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



S: Erstes Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	RAD(458)
	Ausführung einmal bei steigender Flanke	@RAD(458)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

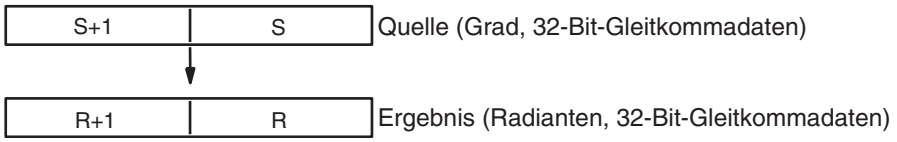
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

RAD(458) konvertiert die 32-Bit-Gleitkommazahl in S+1 und S von Grad zu Radianten und schreibt das Ergebnis in R und R+1. (Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Die Konvertierung von Grad in Radianten erfolgt anhand der folgenden Formel:

Grad $\times$ π/180 = Radianten

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

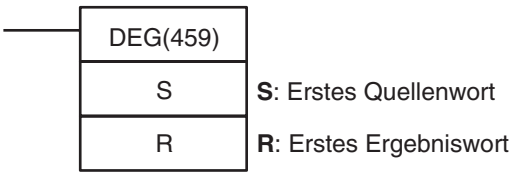
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-10 RADIANS TO DEGREES: DEG(459)

Zweck

Konvertiert eine 32-Bit-Gleitkommazahl von Radianten zu Grad und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	DEG(459)
	Ausführung einmal bei steigender Flanke	@DEG(459)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

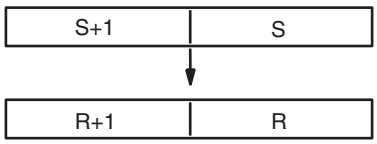
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

DEG(459) konvertiert die 32-Bit-Gleitkommazahl in S+1 und S von Radianten zu Grad und schreibt das Ergebnis in R+1 und R. (Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Quelle (Radianten, 32-Bit-Gleitkommadaten)

Ergebnis (Grad, 32-Bit-Gleitkommadaten)

Die Konvertierung von Radianten zu Grad erfolgt anhand der folgenden Formel:

Radianten $\times$ 180/π = Grad

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

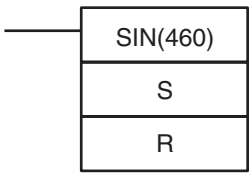
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-11 SINE: SIN(460)

Zweck

Berechnet den Sinus einer 32-Bit-Gleitkommazahl (in Radianten) und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



S: Erstes Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SIN(460)
	Ausführung einmal bei steigender Flanke	@SIN(460)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

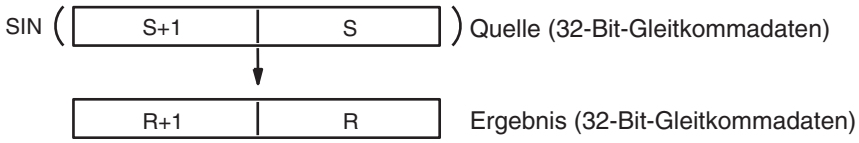
Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	

Bereich	S	R
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

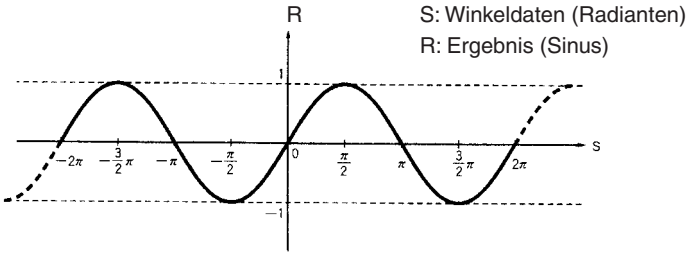
Beschreibung

SIN(460) berechnet den Sinus des Winkels (in Radianten), der als 32-Bit-Gleitkommawert ausgedrückt in S+1 und S steht, und schreibt das Ergebnis in R+1 und R.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Spezifizieren Sie den gewünschten Winkel (–65.535 bis 65.535) in Radianten in den Worten S+1 und S. Wenn der Winkel außerhalb des Bereichs zwischen –65.535 und 65.535 liegt, tritt ein Fehler auf und der Befehl wird nicht ausgeführt. Informationen zum Konvertieren von Grad in Radianten finden Sie unter *3-15-19 LOGARITHM: LOG(468) DEGREES-TO-RADIANS: RAD(458)*.

Die folgende Abbildung zeigt die Beziehung zwischen Winkel und Ergebnis.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 65.535 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	AUS

Name	Bezeich-nung	Funktion
Unterlauf-Merker	UF	AUS
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

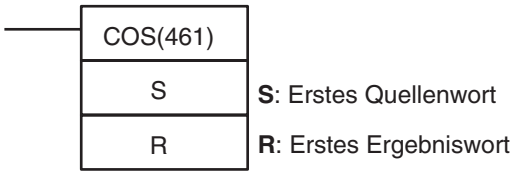
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-12 COSINE: COS(461)

Zweck

Berechnet den Kosinus einer 32-Bit-Gleitkommazahl (in Radianten) und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	COS(461)
	Ausführung einmal bei steigender Flanke	@COS(461)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

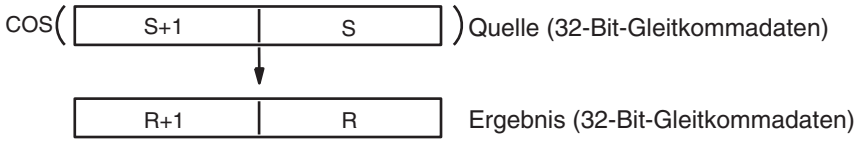
Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	

Bereich	S	R
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

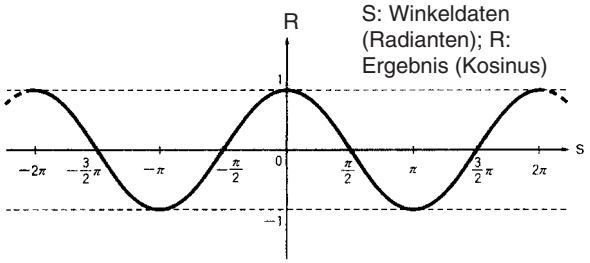
Beschreibung

COS(461) berechnet den Kosinus des Winkels (in Radianten), der als 32-Bit-Gleitkommawert ausgedrückt in S+1 und S steht, und schreibt das Ergebnis in R+1 und R.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Spezifizieren Sie den gewünschten Winkel (–65.535 bis 65.535) in Radianten in den Worten S+1 und S. Wenn der Winkel außerhalb des Bereichs zwischen –65.535 und 65.535 liegt, tritt ein Fehler auf und der Befehl wird nicht ausgeführt. Informationen zum Konvertieren von Grad in Radianten finden Sie unter *3-15-9 DEGREES TO RADIANS: RAD(458)*.

Die folgende Abbildung zeigt die Beziehung zwischen Winkel und Ergebnis.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 65.535 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	AUS
Unterlauf-Merker	UF	AUS
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

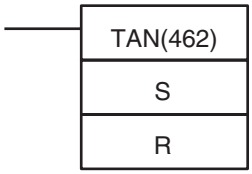
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-13 TANGENT: TAN(462)

Zweck

Berechnet den Tangens einer 32-Bit-Gleitkommazahl (in Radianten) und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



S: Erstes Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TAN(462)
	Ausführung einmal bei steigender Flanke	@TAN(462)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

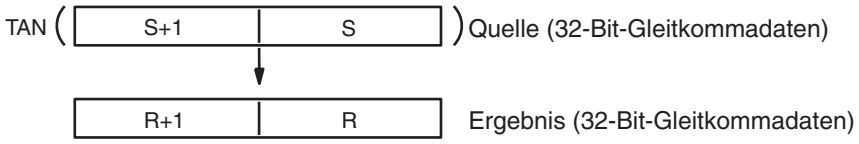
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

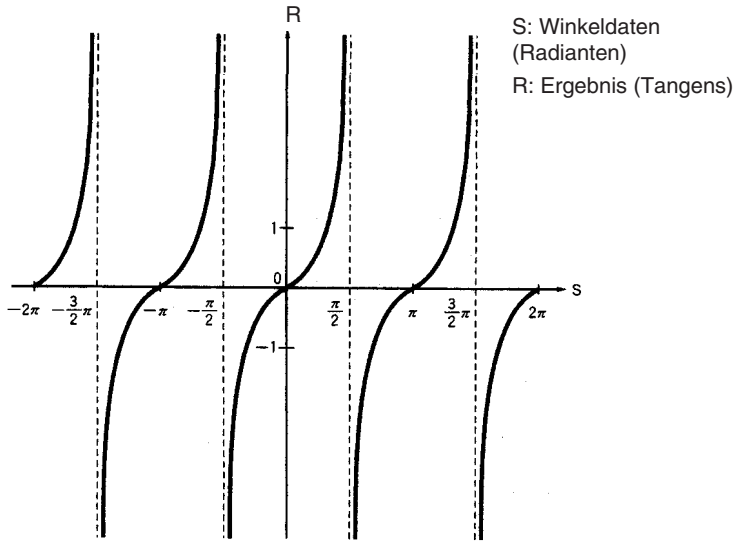
TAN(462) berechnet den Tangens des Winkels (in Radianten), der als 32-Bit-Gleitkommawert ausgedrückt in S+1 und S steht, und schreibt das Ergebnis in R+1 und R.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Spezifizieren Sie den gewünschten Winkel (–65.535 bis 65.535) in Radianten in den Worten S+1 und S. Wenn der Winkel außerhalb des Bereichs zwischen –65.535 und 65.535 liegt, tritt ein Fehler auf und der Befehl wird nicht ausgeführt. Informationen zum Konvertieren von Grad in Radianten finden Sie unter *3-15-9 DEGREES TO RADIANS: RAD(458)*.

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Winkel und Ergebnis.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 65.535 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	AUS
Unterlauf-Merker	UF	AUS
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

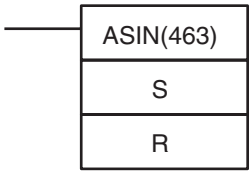
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-14 ARC SINE: ASIN(463)

Zweck

Berechnet den Arkussinus einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. (Die Arkussinusfunktion ist die Umkehrung der Sinusfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Sinuswert zwischen –1 und 1 entspricht.)

Kontaktplansymbol



S: Erstes Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ASIN(463)
	Ausführung einmal bei steigender Flanke	@ASIN(463)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

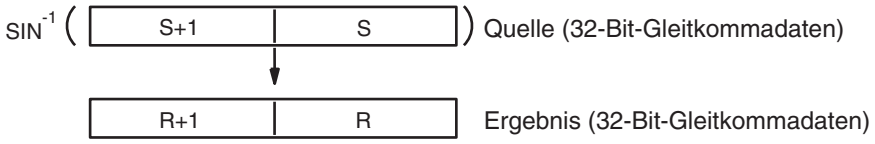
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

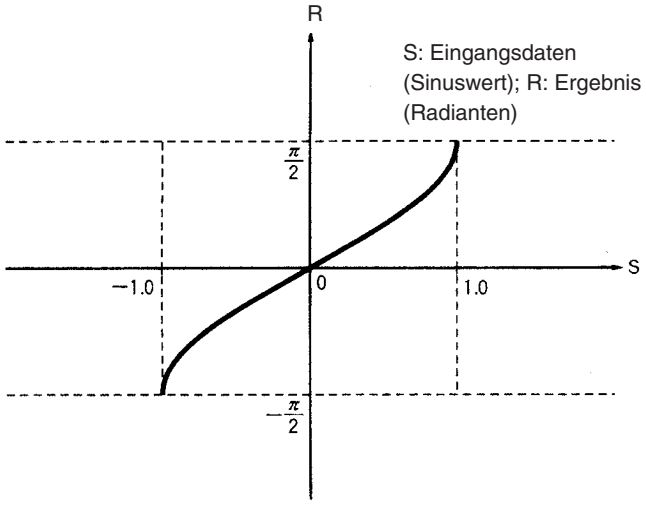
ASIN(463) berechnet den Winkel (in Radianten) für einen Sinuswert, der als 32-Bit-Gleitkommawert ausgedrückt in S+1 und S steht, und schreibt das Ergebnis in R+1 und R.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Die Quellendaten müssen im Bereich zwischen –1,0 und 1,0 liegen. Wenn der Absolutwert der Quellendaten größer als 1,0 ist, tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

Das Ergebnis wird als Winkel (in Radianten) im Bereich zwischen $-\pi/2$ und $\pi/2$ in die Worte R+1 und R ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 1,0 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	AUS
Unterlauf-Merker	UF	AUS
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

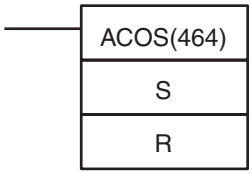
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-15 ARC COSINE: ACOS(464)

Zweck

Berechnet den Arkuskosinus einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. (Die Arkuskosinusfunktion ist die Umkehrung der Kosinusfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Kosinuswert zwischen -1 und 1 entspricht.)

Kontaktplansymbol



S: Erstes Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ACOS(464)
	Ausführung einmal bei steigender Flanke	@ACOS(464)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

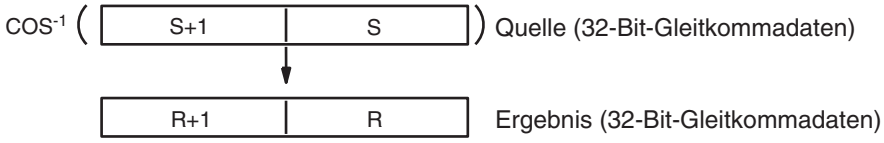
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

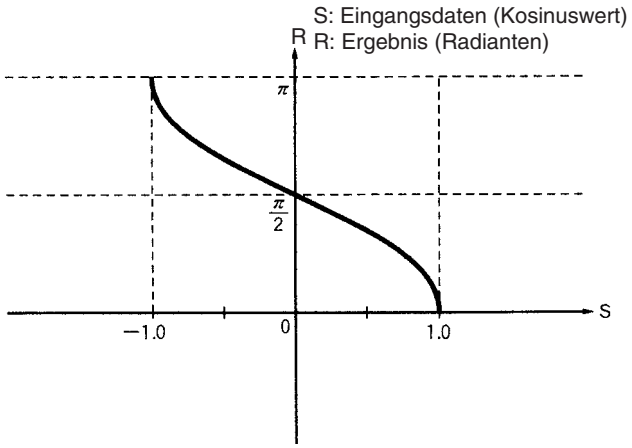
ACOS(464) berechnet den Winkel (in Radianten) für einen Kosinuswert, der als 32-Bit-Gleitkommawert ausgedrückt in S+1 und S steht, und schreibt das Ergebnis in R+1 und R.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Die Quellendaten müssen im Bereich zwischen –1,0 und 1,0 liegen. Wenn der Absolutwert der Quellendaten größer als 1,0 ist, tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

Das Ergebnis wird als Winkel (in Radianten) im Bereich zwischen 0 und π in die Worte R+1 und R ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 1,0 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	AUS
Unterlauf-Merker	UF	AUS
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

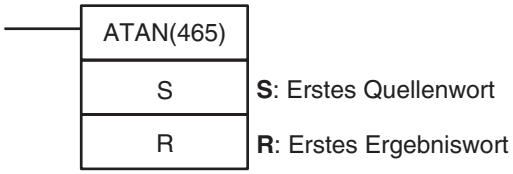
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-16 ARC TANGENT: ATAN(465)

Zweck

Berechnet den Arkustangens einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. (Die Arkustangensfunktion ist die Umkehrung der Tangensfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Tangenswert entspricht.)

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ATAN(465)
	Ausführung einmal bei steigender Flanke	@ATAN(465)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

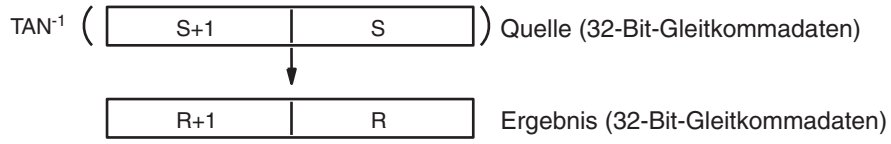
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

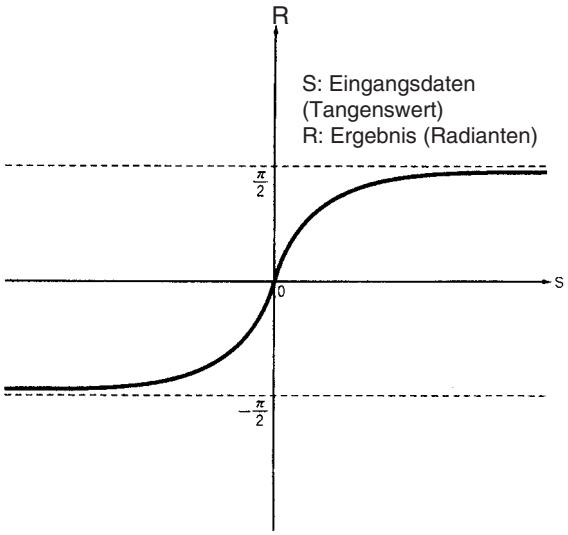
Beschreibung

ATAN(465) berechnet den Winkel (in Radianten) für einen Tangenswert, der als 32-Bit-Gleitkommawert ausgedrückt in S+1 und S steht, und schreibt das Ergebnis in R+1 und R.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Das Ergebnis wird als Winkel (in Radianten) im Bereich zwischen $-\pi/2$ und $\pi/2$ in die Worte R+1 und R ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	AUS
Unterlauf-Merker	UF	AUS
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

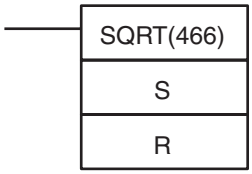
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-17 SQUARE ROOT: SQRT(466)

Zweck

Berechnet die Quadratwurzel einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



S: Erstes Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SQRT(466)
	Ausführung einmal bei steigender Flanke	@SQRT(466)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

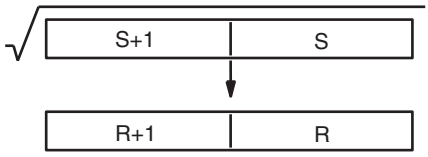
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

SQRT(466) berechnet die Quadratwurzel der 32-Bit-Gleitkommazahl in S+1 and S und schreibt das Ergebnis in R+1 und R. (Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



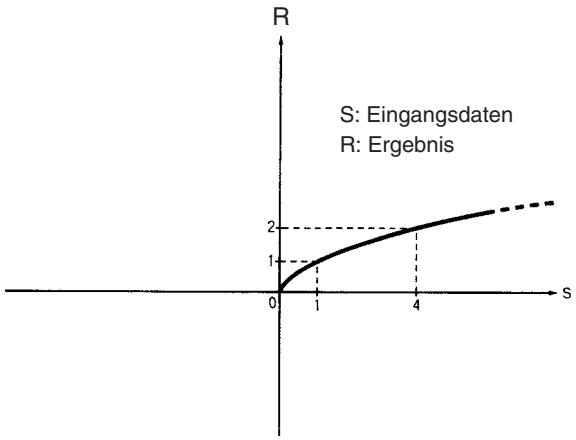
Quelle (32-Bit-Gleitkommadaten)

Ergebnis (32-Bit-Gleitkommadaten)

Die Quellendaten müssen positiv sein. Bei negativen Quellendaten tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten negativ sind. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	AUS
Negativ-Merker	N	AUS

Vorsichtsmaßnahmen

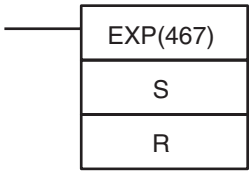
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-18 EXPONENT: EXP(467)

Zweck

Berechnet die natürliche Exponentialgröße (Basis e) einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



S: Erstes Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	EXP(467)
	Ausführung einmal bei steigender Flanke	@EXP(467)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

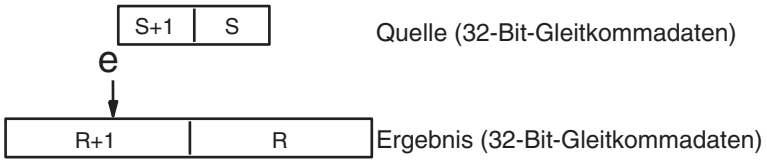
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

EXP(467) berechnet die natürliche Exponentialgröße (Basis e) der 32-Bit-Gleitkommazahl in S+1 and S und schreibt das Ergebnis in R+1 und R. Mit anderen Worten: EXP(467) berechnet e^x (x = Quelle) und schreibt das Ergebnis in R+1 und R.

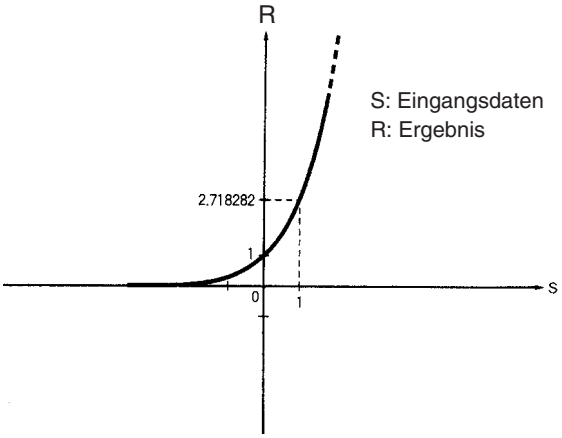


Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Hinweis Die Konstante e ist 2,718282.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Negativ-Merker	N	AUS

Vorsichtsmaßnahmen

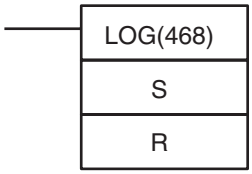
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-19 LOGARITHM: LOG(468)

Zweck

Berechnet den natürlichen Logarithmus (Basis e) einer 32-Bit-Gleitkommazahl und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Kontaktplansymbol



S: Erstes Quellenwort

R: Erstes Ergebniswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	LOG(468)
	Ausführung einmal bei steigender Flanke	@LOG(468)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

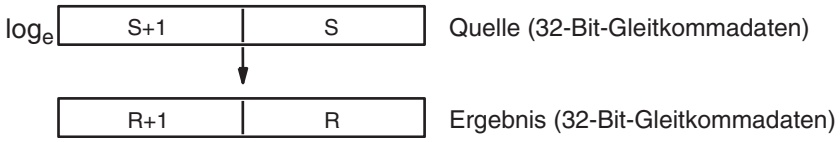
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	R
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

LOG(468) berechnet den natürlichen Logarithmus (Basis e) der 32-Bit-Gleitkommazahl in S+1 and S und schreibt das Ergebnis in R+1 und R.

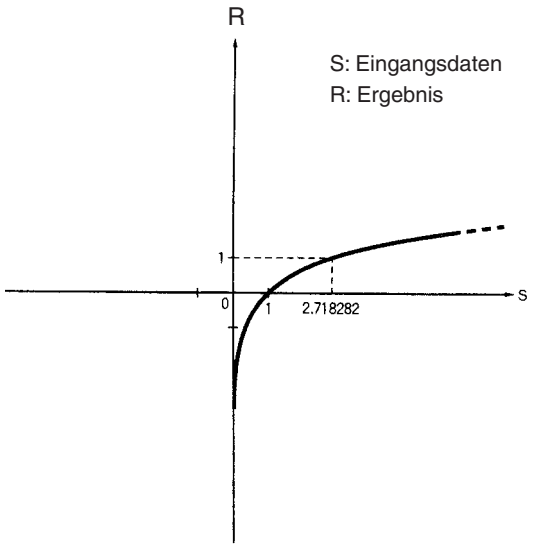


Die Quellendaten müssen positiv sein. Bei negativen Quellendaten tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Hinweis Die Konstante e ist 2,718282.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten negativ sind. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	AUS
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

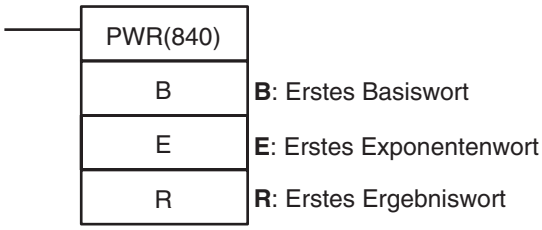
Die Quellendaten in S+1 und S müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-20 EXPONENTIAL POWER: PWR(840)

Zweck

Erhebt eine 32-Bit-Gleitkommazahl in die Potenz einer anderen 32-Bit-Gleitkommazahl.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	PWR(840)
	Ausführung einmal bei steigender Flanke	@PWR(840)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

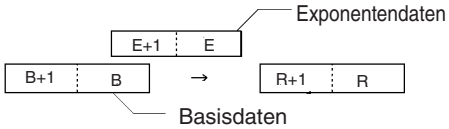
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	B	E	R
CIO-Bereich	CIO 0000 bis CIO 6142		
Arbeitsbereich	W000 bis W510		
Haftmerkerbereich	H000 bis H510		
Zusatz-Systembereich	A000 bis A958		A448 bis A958
Zeitgeberbereich	T0000 bis T4094		
Zählerbereich	C0000 bis C4094		
DM-Bereich	D00000 bis D32766		
EM-Bereich ohne Bank	E00000 bis E32766		
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#00000000 bis #FFFFFFFF (binär)		---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

PWR(840) erhebt die 32-Bit-Gleikommazahl in B+1 und B in die Potenz der 32-Bit-Gleikommazahl in E+1 und E. Mit anderen Worten: PWR(840) berechnet X^Y (X = B+1 und B; Y = E+1 und E).



Wenn beispielsweise die Basisworte (B+1 und B) den Wert 3,1 enthalten und die Exponentenworte (E+1 und E) den Wert 3 enthalten, ist das Ergebnis $3{,}1^3$ bzw. 29,791.

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Basis (B+1 und B) oder der Exponent (E+1 und E) nicht als Gleitkommadaten erkannt werden. EIN, wenn die Basis (B+1 und B) oder der Exponent (E+1 und E) keine Zahl (NaN) ist. EIN, wenn die Basis (B+1 und B) oder der Exponent (E+1 und E) kleiner als 0 ist. (Division durch 0) EIN, wenn die Basis (B+1 und B) negativ ist und der Exponent (E+1 und E) keine Ganzzahl ist. (Wurzel einer negativen Zahl) In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als 32-Bit-Gleitkommawert ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Basis (B+1 und B) und der Exponent (E+1 und E) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-15-21 Vergleichsbefehle für Gleitkommadaten einfacher Genauigkeit

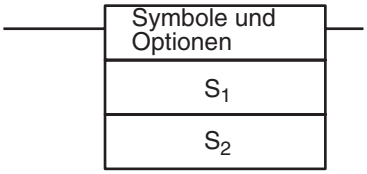
Zweck

Diese Symbolvergleichsbefehle vergleichen zwei Gleitkommawerte einfacher Genauigkeit (32-Bit-Gleitkommadaten im IEEE754-Format, Konstanten und/oder Inhalt der spezifizierten Worte) und erzeugen eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist.

Diese Befehle werden nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Hinweis Weitere Informationen zu Binär-Symbolvergleichsbefehlen (mit/ohne Vorzeichen) finden Sie unter *3-7-1 Symbolvergleichsbefehle (300 bis 328)*. Weitere Informationen zu Gleitkomma-Symbolvergleichsbefehlen doppelter Genauigkeit finden Sie unter *3-16-21 Symbolvergleichsbefehle für Gleitkommadaten doppelter Genauigkeit*.

Kontaktplansymbol



S1: Vergleichsdaten 1

S2: Vergleichsdaten 2

Variationen

Variationen	Erzeugt in jedem Zyklus ein EIN, in dem die Vergleichsbedingung wahr ist.	Symbol-vergleichsbefehl
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S_1	S_2
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich	W000 bis W510	
Haftmerkerbereich	H000 bis H510	
Zusatz-Systembereich	A000 bis A958	
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	
Datenregister	---	
Indexregister	IR0 bis IR15 (nur bei Daten ohne Vorzeichen)	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

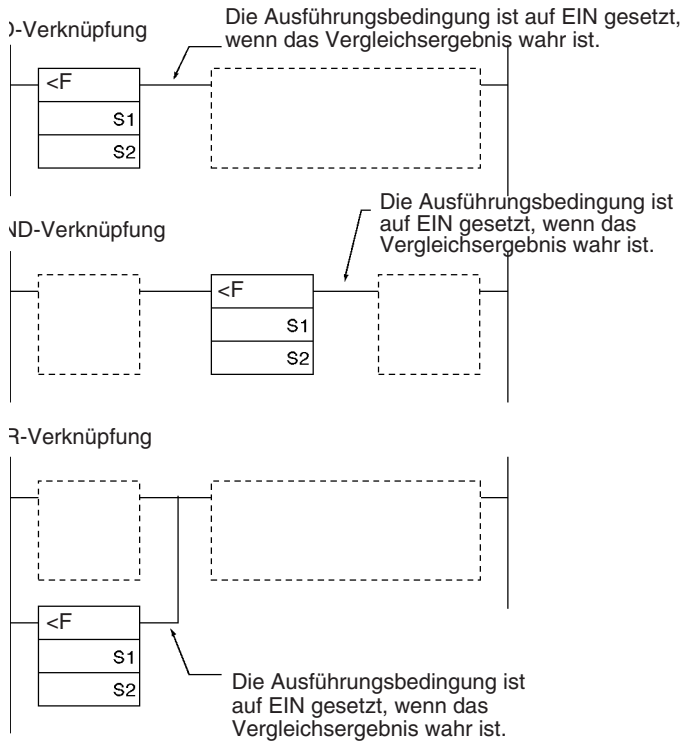
Beschreibung

Der Symbolvergleichsbefehl vergleicht die in S_1 und S_2 spezifizierten Gleitkommawerte einfacher Präzision (32-Bit-Daten im IEEE754-Format) und erzeugt eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Wenn die Daten in Worten gespeichert sind, spezifizieren S_1 und S_2 jeweils die ersten der zwei Worte, die die 32-Bit-Daten enthalten. Eine Eingabe der Gleitkommadaten als 8-stellige Hexadezimalkonstante ist ebenfalls möglich.

Verwendung der Symbolvergleichsbefehle

Die Symbolvergleichsbefehle werden genau wie die Befehl LD, AND und OR behandelt, um die Ausführung nachfolgender Befehle zu steuern.

Eingangs-typ	Funktion
LD	Der Befehl kann direkt mit der linken Stromschiene verbunden werden.
AND	Der Befehl kann nicht direkt mit der linken Stromschiene verbunden werden.
OR	Der Befehl kann direkt mit der linken Stromschiene verbunden werden.



Optionen

Es gibt 18 verschiedene Kombinationsmöglichkeiten aus diesen drei Eingangstypen und sechs Symbolen.

Symbol		Option (Datenformat)
=	(Gleich)	F: Gleitkommadaten mit einfacher Präzision
< >	(Ungleich)	
<	(Kleiner als)	
<=	(Kleiner als oder gleich)	
>	(Größer als)	
>=	(Größer als oder gleich)	

Übersicht über Symbolvergleichsbefehle

In der folgenden Tabelle sind Funktionscodes, AWL-Kürzel. Namen und Funktionen der Symbolvergleichsbefehle für Gleitkommadaten einfacher Präzision aufgeführt. (C1 = S_1+1, S_1 und C2 = S_2+1, S_2.)

Code	AWL	Name	Funktion
329	LD=F	LOAD FLOATING EQUAL	Wahr, wenn C1 = C2
	AND=F	AND FLOATING EQUAL	
	OR=F	OR FLOATING EQUAL	

Code	AWL	Name	Funktion
330	LD <>F	LOAD FLOATING NOT EQUAL	Wahr, wenn C1 ≠ C2
	AND <>F	AND FLOATING NOT EQUAL	
	OR <>F	OR FLOATING NOT EQUAL	
331	LD <F	LOAD FLOATING LESS THAN	Wahr, wenn C1 < C2
	AND <F	AND FLOATING LESS THAN	
	OR <F	OR FLOATING LESS THAN	
332	LD <=F	LOAD FLOATING LESS THAN OR EQUAL	Wahr, wenn C1 ≤ C2
	AND <=F	AND FLOATING LESS THAN OR EQUAL	
	OR <=F	OR FLOATING LESS THAN OR EQUAL	
333	LD >F	LOAD FLOATING GREATER THAN	Wahr, wenn C1 > C2
	AND >F	AND FLOATING GREATER THAN	
	OR >F	OR FLOATING GREATER THAN	
325	LD >=F	LOAD FLOATING GREATER THAN OR EQUAL	Wahr, wenn C1 ≥ C2
	AND >=F	AND FLOATING GREATER THAN OR EQUAL	
	OR >=F	OR FLOATING GREATER THAN OR EQUAL	

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn S_1+1, S_1 oder S_2+1, S_2 keine gültige Gleitkommazahl (NaN) ist. EIN, wenn S_1+1, S_1 oder S_2+1, S_2 den Wert $+\infty$ hat. EIN, wenn S_1+1, S_1 oder S_2+1, S_2 den Wert $-\infty$ hat. In allen anderen Fällen AUS.
Größer-als-Merker	>	EIN, wenn S_1+1, $S_1 > S_2+1$, S_2. In allen anderen Fällen AUS.
Größer-oder-gleich-Merker	> =	EIN, wenn S_1+1, $S_1 \geq S_2+1$, S_2. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn S_1+1, $S_1 = S_2+1$, S_2. In allen anderen Fällen AUS.
Ungleich-Merker	=	EIN, wenn S_1+1, $S_1 \neq S_2+1$, S_2. In allen anderen Fällen AUS.
Kleiner-als-Merker	<	EIN, wenn S_1+1, $S_1 < S_2+1$, S_2. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Merker	< =	EIN, wenn S_1+1, $S_1 \leq S_2+1$, S_2. In allen anderen Fällen AUS.
Negativ-Merker	N	Unverändert

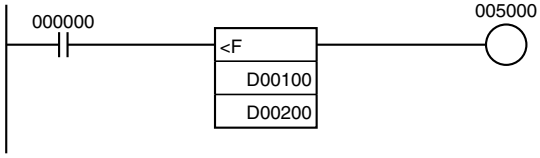
Vorsichtsmaßnahmen

Symbolvergleichsbefehle können nicht mit Verbindung zur rechten Stromschiene verwendet werden, d. h., zwischen ihnen und der rechten Stromschiene muss ein weiterer Befehl eingesetzt werden.

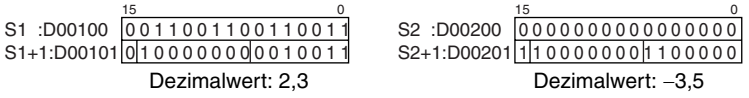
Beispiel

AND FLOATING LESS THAN: AND<F(331)

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, werden die Gleitkommadaten in D00101, D00100 mit den Gleitkommadaten in D00201, D00200 verglichen. Wenn der Inhalt von D00101, D00100 kleiner ist als der von D00201, D00200, wird die Programmausführung mit der nächsten Zeile fortgesetzt und CIO 005000 wird auf EIN gesetzt. Wenn der Inhalt von D00101, D00100 nicht kleiner ist als der von D00201, D00200, wird CIO 005000 auf AUS gesetzt.

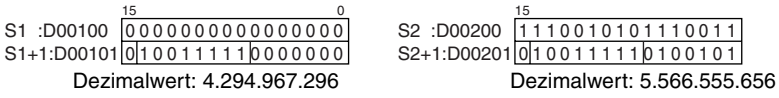


FLOATING LESS THAN-Vergleich (<F)



2.3>-3.5

Erzeugt keine EIN-Bedingung.



4294967296<5566555656

Erzeugt eine EIN-Bedingung.

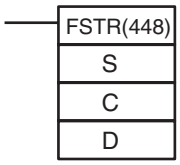
3-15-22 FLOATING-POINT TO ASCII: FSTR(448)

Zweck

Drückt einen 32-Bit-Gleitkommawert (IEEE754-Format) in dezimaler Standardschreibweise oder wissenschaftlicher Schreibweise aus und konvertiert diesen Wert in ASCII-Text.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Erstes Quellenwort
C: Erstes Steuerwort
D: Erstes Zielwort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	FSTR(448)
	Ausführung einmal bei steigender Flanke	@FSTR(448)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

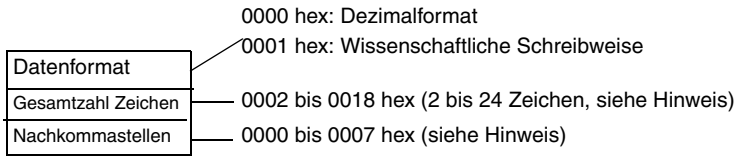
Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6141	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W510	W000 bis W509	W000 bis W511
Haftmerkerbereich	H000 bis H510	H000 bis H509	H000 bis H511
Zusatz-Systembereich	A000 bis A958	A000 bis A957	A448 bis A959
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4093	T0000 bis T4095

Bereich	S	C	D
Zählerbereich	C0000 bis C4094	C0000 bis C4093	C0000 bis C4095
DM-Bereich	D00000 bis D32766	D00000 bis D32765	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32765	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32765 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	#00000000 bis #FFFFFFFF (binär)	---	
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15 ,IR0 bis ,IR15		

Beschreibung

FSTR(448) drückt die 32-Bit-Gleitkommazahl in S+1 und S (IEEE754-Format) entsprechend der Steuerdaten in den Worten C bis C+2 in dezimaler oder wissenschaftlicher Schreibweise aus, konvertiert die Zahl in ASCII-Text und gibt das Ergebnis in die mit D beginnenden Zielworte aus.

In der folgenden Abbildung ist der Inhalt der 3 Steuerworte dargestellt.

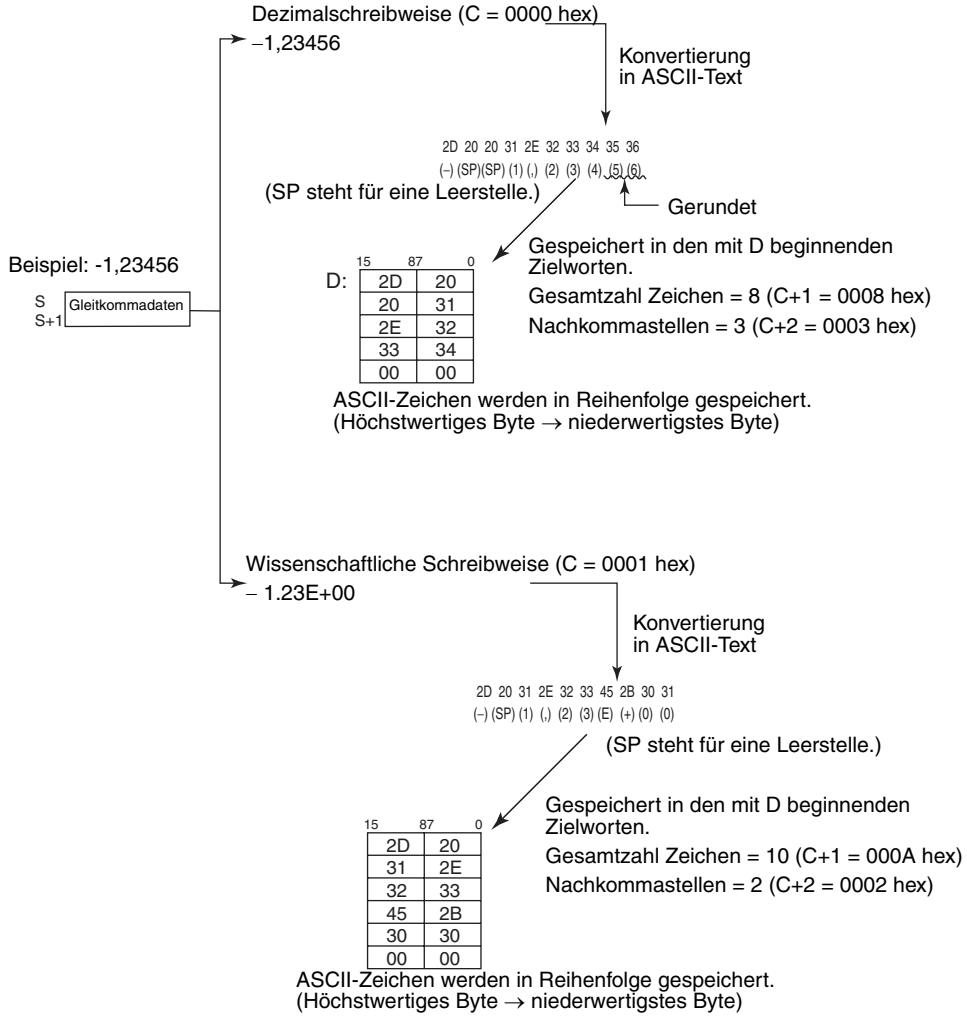


Hinweis: Es gibt Begrenzungen für die Gesamtzahl der Zeichen und die Anzahl der Nachkommastellen. Detaillierte Informationen hierzu finden Sie unter *Begrenzungen der Anzahl von ASCII-Zeichen* auf Seite 607.

- Der Inhalt von C (Datenformat) spezifiziert, ob die Zahl in S+1 und S in Dezimalschreibweise oder wissenschaftlicher Schreibweise ausgedrückt werden soll.
  - Dezimalschreibweise
    Drückt eine reelle Zahl als Ganzzahl mit Dezimalbruchteil aus.
    Beispiel: 124,56
  - Wissenschaftliche Schreibweise
    Drückt eine reelle Zahl als ganzzahligen Teil, Dezimalbruchteil und Exponent aus.
    Beispiel: 1,2456E-2 ($1{,}2456 \times 10^{-2}$)

- Der Inhalt von C+1 (Gesamtzahl Zeichen) spezifiziert die Anzahl der ASCII-Zeichen nach Konvertierung einschließlich Vorzeichen, Zahlen, Dezimalkomma und Leerstellen.
- Der Inhalt von C+2 (Nachkommastellen) spezifiziert die Anzahl von Stellen (Zeichen) hinter dem Dezimalkomma.

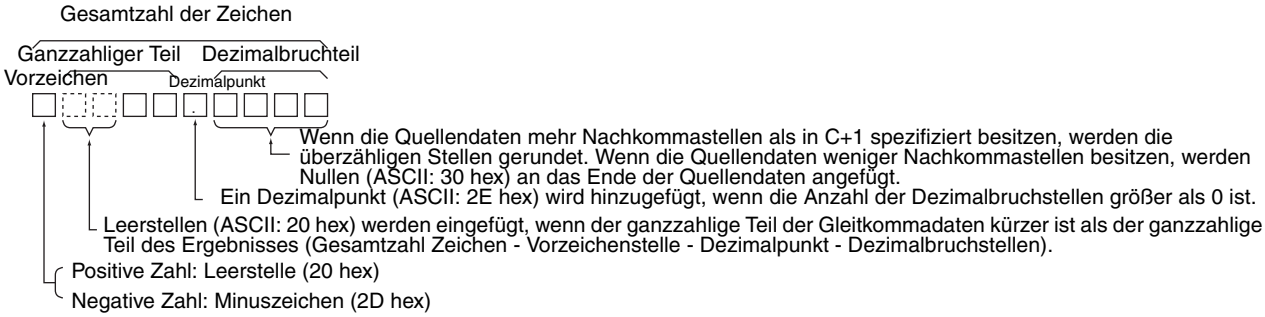
Der ASCII-Text wird in D und den nachfolgenden Worten in der folgenden Reihenfolge gespeichert: höchstwertiges Byte von D, niederwertigstes Byte von D, höchstwertiges Byte von D+1, niederwertigstes Byte von D+1 usw.



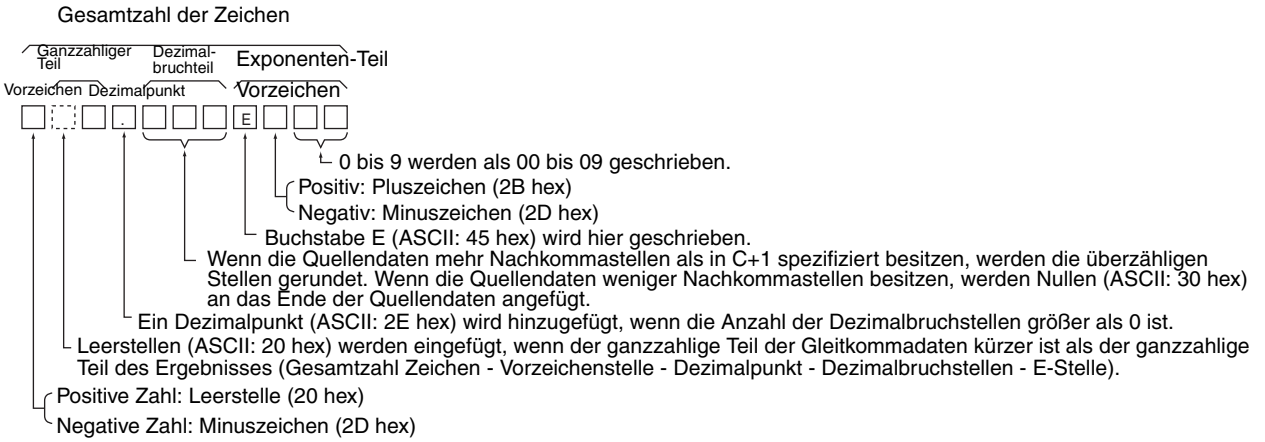
Speicherung von ASCII-Text

Nach der Konvertierung der Gleitkommazahl in ASCII-Text werden die ASCII-Zeichen in den mit D beginnenden Zielworten gespeichert (siehe folgende Abbildungen). Für die Dezimalschreibweise und die wissenschaftliche Schreibweise werden verschiedene Speichermethoden genutzt.

Dezimalschreibweise (C = 0000 hex)



Wissenschaftliche Schreibweise (C = 0000 hex)



Hinweis Als Endcode werden ein oder zwei Bytes mit Nullen an das Ende des ASCII-Textes angefügt.
Gesamtzahl der Zeichen ungerade: hinter dem ASCII-Text wird 00 hex gespeichert.
Gesamtzahl der Zeichen gerade: hinter dem ASCII-Text wird 0000 hex gespeichert.

Begrenzungen der Anzahl von ASCII-Zeichen

In der konvertierten Zahl gibt es Begrenzungen für die Anzahl der ASCII-Zeichen. Der Fehlermerker wird auf EIN gesetzt, wenn die Anzahl der Zeichen die zulässige Höchstzahl übersteigt.

1. Begrenzungen der Gesamtzahl von ASCII-Zeichen
   a) Dezimalschreibweise (C = 0000 hex)
   - Wenn kein Dezimalbruchteil vorhanden ist (C+2 = 0000 hex):
     $2 \leq$ Gesamtzahl Zeichen ≤ 24
   - Wenn ein Dezimalbruchteil vorhanden ist (C+2 = 0001 bis 0007 hex):
     (Dezimalbruchstellen + 3) $\leq$ Gesamtzahl Zeichen ≤ 24

   b) Wissenschaftliche Schreibweise (C = 0001 hex)
   - Wenn kein Dezimalbruchteil vorhanden ist (C+2 = 0000 hex):
     $6 \leq$ Gesamtzahl Zeichen ≤ 24
   - Wenn ein Dezimalbruchteil vorhanden ist (C+2 = 0001 bis 0007 hex):
     (Dezimalbruchstellen +7) $\leq$ Gesamtzahl Zeichen ≤ 24
2. Begrenzungen der Stellenzahl im ganzzahligen Teil
   a) Dezimalschreibweise (C = 0000 hex)

- Wenn kein Dezimalbruchteil vorhanden ist (C+2 = 0000 hex):
  1 ≤ Anzahl ganzzahlige Stellen ≤ 24
- Wenn ein Dezimalbruchteil vorhanden ist (C+2 = 0001 bis 0007 hex):
  1 ≤ Anzahl ganzzahlige Stellen ≤ (24 − Dezimalbruchstellen − 2)

b) Wissenschaftliche Schreibweise (C = 0001 hex)
1 Stelle (fest)

3. Begrenzungen der Stellenzahl im Dezimalbruchteil
   a) Dezimalschreibweise (C = 0000 hex)
   - Dezimalbruchstellen ≤ 7
   - Sowie: Dezimalbruchstellen ≤ (Gesamtzahl der ASCII-Zeichen − 3)

   b) Wissenschaftliche Schreibweise (C = 0001 hex)
   - Dezimalbruchstellen ≤ 7
   - Sowie: Dezimalbruchstellen ≤ (Gesamtzahl der ASCII-Zeichen − 3)

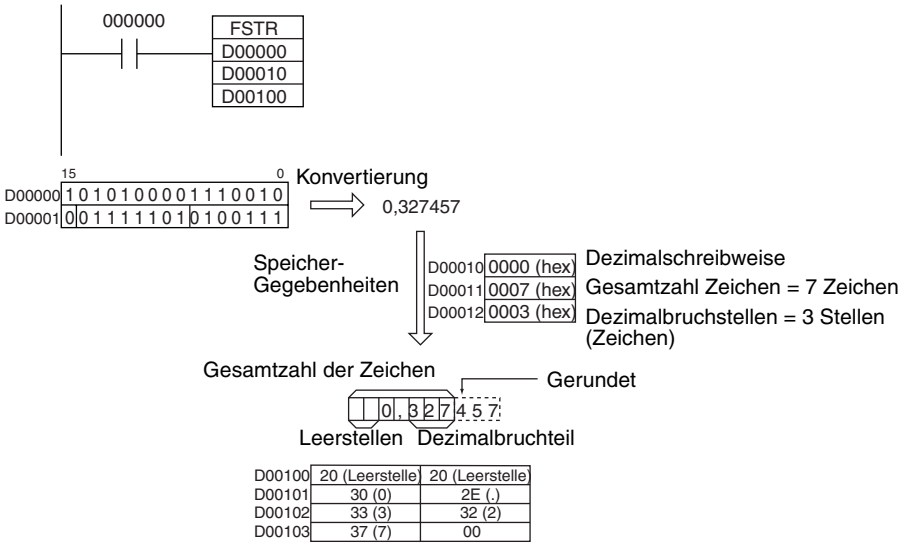
Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Daten in S+1 und S keine gültige Gleitkommazahl (NaN) sind. EIN, wenn die Daten in S+1 und S den Wert +∞ oder −∞ besitzen. EIN, wenn die Einstellung für das Datenformat in C nicht 0000 oder 0001 ist. EIN, wenn die Einstellung für die Gesamtzahl der Zeichen in C+1 nicht innerhalb des zulässigen Bereichs liegt. (Detaillierte Informationen hierzu finden Sie unter *1. Begrenzungen der Gesamtzahl von ASCII-Zeichen* weiter oben.) EIN, wenn die Einstellung für die Anzahl der Dezimalbruchstellen in C+2 nicht innerhalb des zulässigen Bereichs liegt. (Detaillierte Informationen hierzu finden Sie unter *3. Begrenzungen der Stellenzahl im Dezimalbruchteil* weiter oben.) In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Konvertierungsergebnis 0 ist. In allen anderen Fällen AUS.

Beispiele

Konvertierung in ASCII-Text mit Dezimalschreibweise

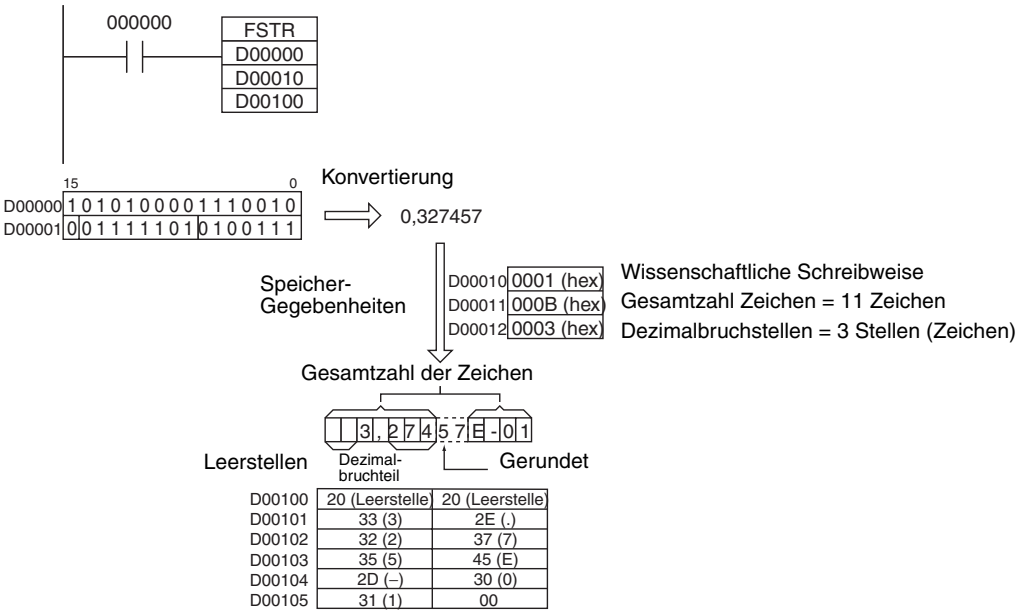
Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, konvertiert der Befehl FSTR(448) die Gleitkommadaten in D00001 und D00000 in ASCII-Text mit Dezimalschreibweise und schreibt dem ASCII-Text in die mit D00100 beginnenden Zielworte. Der Inhalt der Steuerworte (D00010 bis D00012) spezifiziert die Details des Datenformats (Dezimalschreibweise, 7 Zeichen gesamt, 3 Dezimalbruchstellen).



D00100	20 (Leerstelle)	20 (Leerstelle)
D00101	30 (0)	2E (.)
D00102	33 (3)	32 (2)
D00103	37 (7)	00

Konvertierung in ASCII-Text mit wissenschaftlicher Schreibweise

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, konvertiert der Befehl FSTR(448) die Gleitkommadaten in D00001 und D00000 in ASCII-Text mit wissenschaftlicher Schreibweise und schreibt den ASCII-Text in die mit D00100 beginnenden Zielworte. Der Inhalt der Steuerworte (D00010 bis D00012) spezifiziert die Details des Datenformats (wissenschaftliche Schreibweise, 11 Zeichen gesamt, 3 Dezimalbruchstellen).



D00100	20 (Leerstelle)	20 (Leerstelle)
D00101	33 (3)	2E (.)
D00102	32 (2)	37 (7)
D00103	35 (5)	45 (E)
D00104	2D (–)	30 (0)
D00105	31 (1)	00

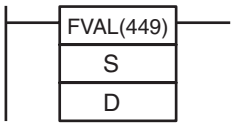
3-15-23 ASCII TO FLOATING-POINT: FVAL(449)

Zweck

Konvertiert eine als ASCII-Text ausgedrückte Zahl (in Dezimal- oder wissenschaftlicher Schreibweise) in einen 32-Bit-Gleitkommawert (IEEE754-Format) und gibt den Gleitkommawert in die spezifizierten Worte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Erstes Quellenwort

D: Erstes Zielwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	FVAL(449)
	Ausführung einmal bei steigender Flanke	@FVAL(449)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142
Arbeitsbereich	W000 bis W511	W000 bis W510
Haftmerkerbereich	H000 bis H511	H000 bis H510
Zusatz-Systembereich	A000 bis A959	A448 bis A958
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094
Zählerbereich	C0000 bis C4095	C0000 bis C4094
DM-Bereich	D00000 bis D32767	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15 ,IR0 bis ,IR15	

Beschreibung

FVAL(449) konvertiert die spezifizierte als ASCII-Text vorhandene Zahl (beginnend mit Wort S) in eine 32-Bit-Gleikommazahl (IEEE754-Format) und gibt das Ergebnis in die mit D beginnenden Worte aus.

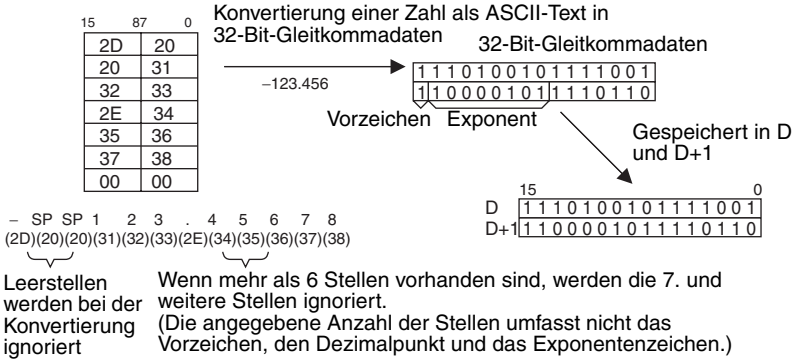
FVAL(449) kann ASCII-Text mit Zahlen in Dezimal- oder wissenschaftlicher Schreibweise konvertieren, wenn sie die folgenden Bedingungen erfüllen:

- Dezimalschreibweise
  Reelle Zahlen, die durch einen ganzzahligen Teil und einen Dezimalbruchteil ausgedrückt sind.
  Beispiel: 124,56
- Wissenschaftliche Schreibweise
  Reelle Zahlen, die durch einen ganzzahligen Teil, einen Dezimalbruchteil und einen Exponenten-Teil ausgedrückt sind.
  Beispiel: 1.2456E-2 ($1{,}2456 \times 10^{-2}$)

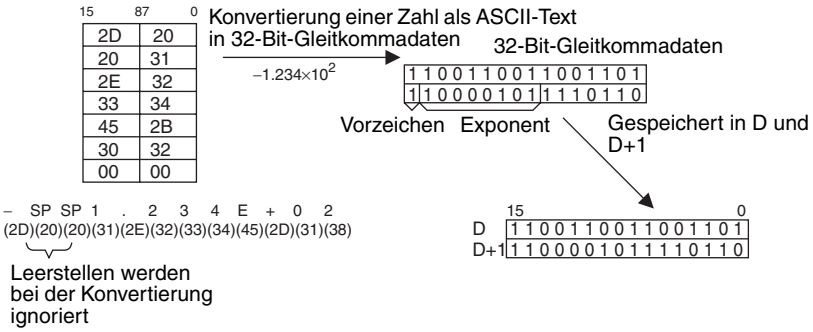
Das Datenformat (Dezimal- oder wissenschaftliche Schreibweise) wird automatisch erkannt.

Der ASCII-Text muss in D und den nachfolgenden Worten in der folgenden Reihenfolge gespeichert sein: höchstwertiges Byte von S, niederwertigstes Byte von S, höchstwertiges Byte von S+1, niederwertigstes Byte von S+1 usw.

Dezimalschreibweise



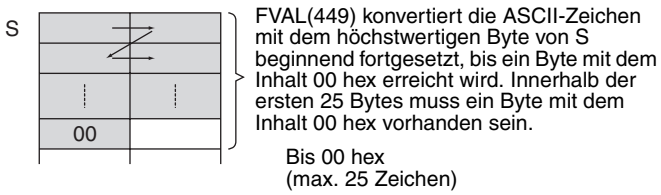
Wissenschaftliche Schreibweise



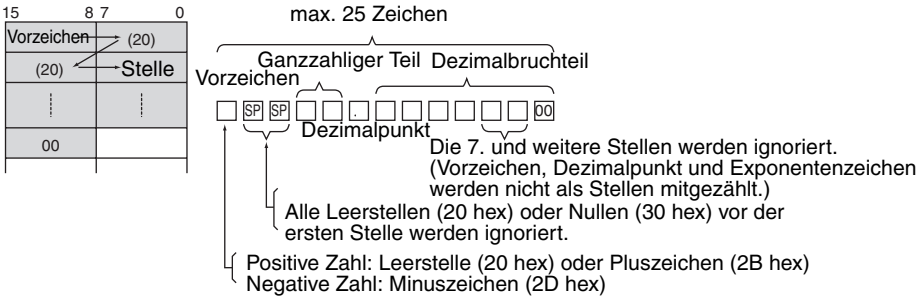
Speicherung von ASCII-Text

In den folgenden Abbildungen wird gezeigt, wie eine Zahl als ASCII-Text in Gleitkommadaten konvertiert wird. Für die in Dezimalschreibweise und die wissenschaftlicher Schreibweise gespeicherten Zahlen werden verschiedene Konvertierungsmethoden genutzt.

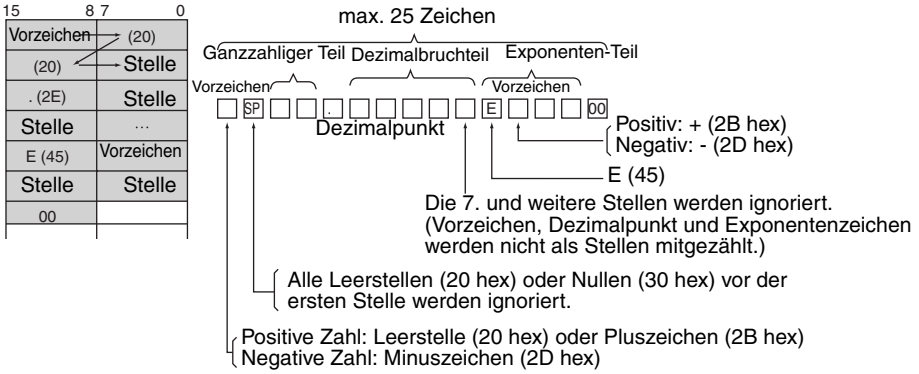
ASCII-Zeichen-Speicher



Dezimalschreibweise



Wissenschaftliche



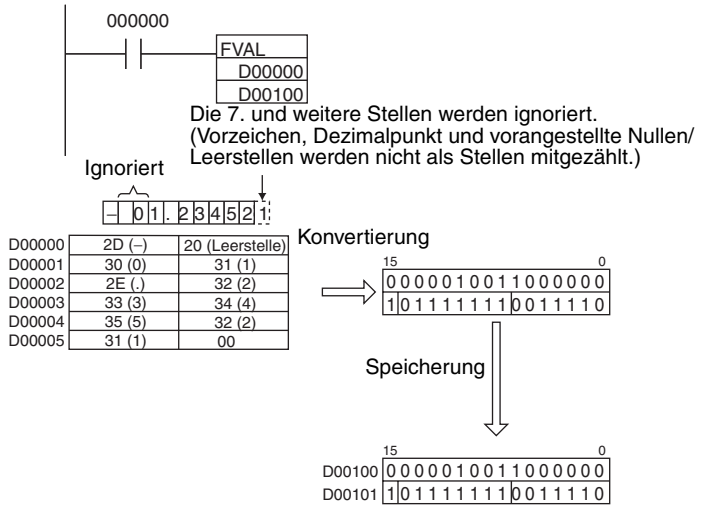
Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Stellen (ganzzahliger und Dezimalbruchteil) in den mit S beginnenden Quellendaten nicht 30 bis 39 hex (0 bis 9) sind. EIN, wenn die ersten beiden Stellen des Exponententeils nicht 45 und 2B hex (E+) oder 45 und 2D hex (E-) enthalten. EIN, wenn zwei oder mehr Exponententeile in den Quellendaten vorhanden sind. Ein, wenn die Daten nach Konvertierung den Wert +∞ oder −∞ haben. EIN, wenn 0 Zeichen in den Textdaten vorhanden sind. EIN, wenn innerhalb der ersten 25 Zeichen kein Byte mit dem Inhalt 00 hex vorkommt. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Konvertierungsergebnis 0 ist. In allen anderen Fällen AUS.

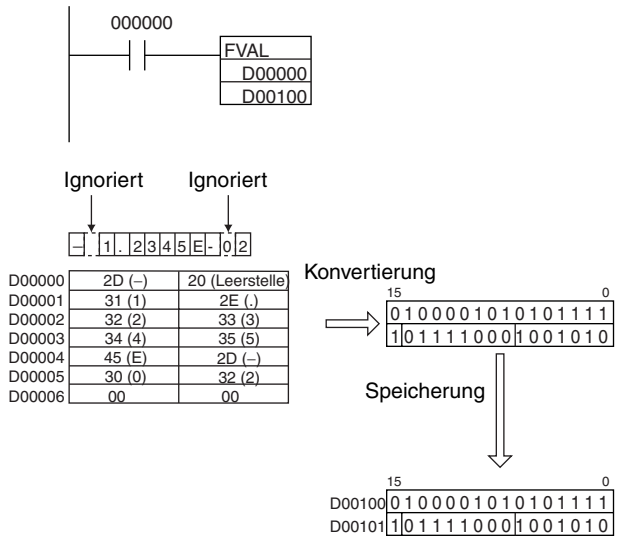
Beispiele

Konvertierung in ASCII-Text in Dezimalschreibweise zu Gleitkommadaten

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, konvertiert der Befehl FVAL(449) die spezifizierte ASCII-Text-Zahl (in Dezimalschreibweise) in den bei D00000 beginnenden Quellenworten zu Gleitkommadaten und schreibt das Ergebnis in die Zielworte D00100 und D00101.



Konvertierung von ASCII-Text in wissenschaftlicher Schreibweise
Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, konvertiert der Befehl FVAL(449) die spezifizierte ASCII-Text-Zahl (in wissenschaftlicher Schreibweise) in den bei D00000 beginnenden Quellenworten zu Gleitkommadaten und schreibt das Ergebnis in die Zielworte D00100 und D00101.



3-16 Befehle für Gleitkommadaten doppelter Genauigkeit (nur CS1-H, CJ1-H, CJ1M und CS1D)

Die Befehle für Gleitkommadaten doppelter Genauigkeit konvertieren Daten und führen Gleitkommaarithmetik-Operationen mit Gleitkommadaten doppelter Genauigkeit durch. Die CPU-Baugruppen der SPS-Systeme CS1-H/CJ1-H unterstützen die folgenden 20 Befehle.

Befehl	AWL	Funktionscode	Seite
DOUBLE FLOATING TO 16-BIT	FIXD	841	620
DOUBLE FLOATING TO 32-BIT	FIXLD	842	621
16-BIT TO DOUBLE FLOATING	DBL	843	623
32-BIT TO DOUBLE FLOATING	DBLL	844	624
DOUBLE FLOATING-POINT ADD	+D	845	626
DOUBLE FLOATING-POINT SUBTRACT	–D	846	628
DOUBLE FLOATING-POINT MULTIPLY	∗D	847	630

Befehl	AWL	Funktionscode	Seite
DOUBLE FLOATING-POINT DIVIDE	/D	848	632
DOUBLE DEGREES TO RADIANS	RADD	849	634
DOUBLE RADIANS TO DEGREES	DEGD	850	636
DOUBLE SINE	SIND	851	637
DOUBLE COSINE	COSD	852	639
DOUBLE TANGENT	TAND	853	641
DOUBLE ARC SINE	ASIND	854	643
DOUBLE ARC COSINE	ACOSD	855	645
DOUBLE ARC TANGENT	ATAND	856	647
DOUBLE SQUARE ROOT	SQRTD	857	649
DOUBLE EXPONENT	EXPD	858	651
DOUBLE LOGARITHM	LOGD	859	653
DOUBLE EXPONENTIAL POWER	PWRD	860	655
Symbolvergleichsbefehle für Gleitkommadaten doppelter Genauigkeit	LD, AND, OR + =D, <>D, <D, <=D, >D oder >=D	335 bis 340	657

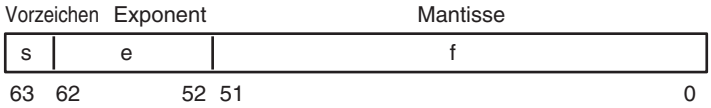
Datenformat

Gleitkommadaten drücken reelle Zahlen mittels Vorzeichen, Exponent und Mantisse aus. Wenn Daten im Gleitkommaformat ausgedrückt werden, gilt die folgende Formel.

Reelle Zahl = $(-1)^s \, 2^{e-1.023} \, (1.f)$

s: Vorzeichen
e: Exponent
f: Mantisse

Das Gleitkommadatenformat entspricht der Norm IEEE754. Daten werden wie folgt in 32 Bits ausgedrückt:



Daten	Anzahl Bits	Inhalt
s: Vorzeichen	1	0: positiv; 1: negativ
e: Exponent	11	Die Werte für den Exponenten (e) haben einen Bereich von 0 bis 2.047. Der tatsächliche Exponent ist der Wert, der erhalten wird, indem 1.023 von e subtrahiert wird, wodurch sich ein Bereich von –1.023 bis 1.024 ergibt. „e = 0“ und „e = 2.047“ drücken spezielle Zahlen aus.
f: Mantisse	52	Der Mantissenteil der binären Gleitkommadaten entspricht dem Format 2,0 > 1,f ≥1,0.

Anzahl der Stellen

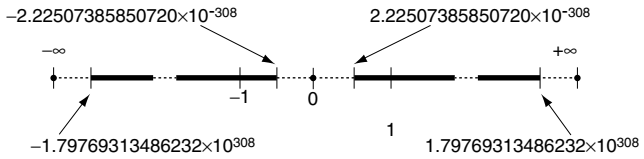
Die Anzahl der effektiven Stellen von Gleitkommadaten beträgt 53 Bits im Binärformat (ca. 15 Stellen in dezimal).

Gleitkommadaten

Die folgenden Daten können durch Gleitkommadaten ausgedrückt werden:

- $-\infty$
- $-1{,}79769313486232 \times 10^{308} \leq$ Wert $\leq -2{,}22507385850720 \times 10^{-308}$
- 0
- $2{,}22507385850720 \times 10^{-308} \leq$ Wert $\leq 1{,}79769313486232 \times 10^{30}$

- +∞
- Keine Zahl (Not a Number, NaN)



Spezielle Zahlen

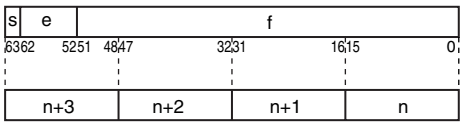
Die Formate für NaN, ±∞ und 0 sind wie folgt:

NaN*: e = 1.024, f ≠ 0
+∞: e = 1.024, f = 0 und s = 0
−∞: e = 1.024, f = 0 und s = 1
0: e = 0 und f = 0

*NaN (keine Zahl) ist keine gültige Gleitkommazahl. Die Ausführung von Befehlen für Gleitkommadaten doppelter Genauigkeit kann nicht zu dem Ergebnis NaN führen.

Schreiben von Gleitkommadaten

Wenn in der E/A-Speicher-Bearbeitungsansicht von CX-Programmer Gleitkommadaten doppelter Genauigkeit als Datenformat spezifiziert sind, werden die in diesem Fenster eingegebenen Standard-Dezimalzahlen automatisch in das zuvor beschriebene Gleitkommaformat doppelter Genauigkeit (IEEE754-Format) konvertiert und in den E/A-Speicher geschrieben. Im IEEE754-Format geschriebene Daten werden zur Überwachung in der Anzeige automatisch in das Standard-Dezimalformat konvertiert.



Es ist für den Benutzer beim Lesen und Schreiben von Gleitkommadaten doppelter Genauigkeit nicht erforderlich, das IEEE754-Datenformat zu beachten. Es ist nur erforderlich daran zu denken, dass Gleitkommadaten doppelter Genauigkeit jeweils 4 Worte belegen.

Als Gleitkommawerte ausgedrückte Zahlen

Die folgenden Arten von Gleitkommazahlen können verwendet werden.

Mantisse (f)	Exponent (e)		
	0	Nicht 0 und nicht alle Bits 1 (1.024)	Alle Bits 1 (1.024)
0	0	Normalisierte Zahlen	Unendlich
Nicht 0	Nicht normalisierte Zahlen		NaN

Hinweis Eine nicht normalisierte Zahl ist eine Zahl, deren Absolutwert zu klein ist, um als normalisierte Zahl ausgedrückt zu werden. Nicht normalisierte Zahlen besitzen weniger wertige Stellen. Wenn das Ergebnis von Berechnungen (einschließlich Zwischenergebnissen) eine nicht normalisierte Zahl ergibt, wird die Anzahl der wertigen Stellen reduziert.

Normalisierte Zahlen

Normalisierte Zahlen stellen reelle Zahlen dar. Das Vorzeichenbit ist bei einer positiven Zahl auf 1, bei einer negativen Zahl auf 0 gesetzt.

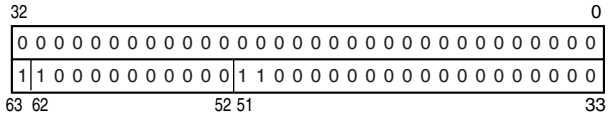
Der Exponent (e) wird durch eine Zahl zwischen 1 und 2.046 ausgedrückt. Der tatsächliche Exponent ist diese Zahl minus 1.023, also –1.022 bis 1.023.

Die Mantisse (f) wird durch eine Zahl zwischen 0 und $(2^{52} – 1)$ ausgedrückt, und es wird davon ausgegangen, dass Bit 2^{52} der tatsächlichen Mantisse gleich 1 ist und das Dezimalkomma unmittelbar dahinter steht.

Normalisierte Zahlen werden wie folgt ausgedrückt:

$(–1)^{(\text{Vorzeichen s})} \times 2^{(\text{Exponent e}) - 1.023} \times (1 + \text{Mantisse} \times 2^{-52})$

Beispiel



Vorzeichen: –
Exponent: 1.024 -1.023 = 1
Mantisse: $1 + (2^{51} + 2^{50}) \times 2^{-52} = 1 + (2^{-1} + 2^{-2}) = 1 + (0{,}75) = 1{,}75$
Wert: $-1{,}75 \times 2^{1} = -3{,}5$

Nicht normalisierte Zahlen

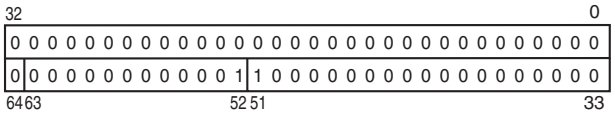
Nicht normalisierte Zahlen stellen reelle Zahlen mit sehr kleinem Absolutwert dar. Das Vorzeichenbit ist bei einer positiven Zahl auf 1, bei einer negativen Zahl auf 0 gesetzt.

Der Exponent (e) ist 0 und der tatsächliche Exponent ist –1.022.

Die Mantisse (f) wird durch eine Zahl zwischen 1 und $(2^{52} - 1)$ ausgedrückt, und es wird davon ausgegangen, dass Bit 2^{52} der tatsächlichen Mantisse gleich 0 ist und das Dezimalkomma unmittelbar dahinter steht.

Nicht normalisierte Zahlen werden wie folgt ausgedrückt:
$(-1)^{(\text{Vorzeichen s})} \times 2^{-1.022} \times (\text{Mantisse} \times 2^{-52})$

Beispiel



Vorzeichen: –
Exponent: –1.022
Mantisse: $0 + (2^{51} + 2^{50}) \times 2^{-52} = 0 + (2^{-1} + 2^{-2}) = 0 + (0{,}75) = 0{,}75$
Wert: $-0{,}75 \times 2^{-1.022} = 1{,}668805 \times 10^{-308}$

Null

Werte von +0,0 und –0,0 können ausgedrückt werden, indem das Vorzeichen für positiv auf 1 und für negativ auf 0 gesetzt wird. Der Exponent und die Mantisse haben in diesem Fall beide den Wert 0. Sowohl +0,0 als auch –0,0 entsprechen 0,0. Beschreibungen der Unterschiede, die sich durch ein Vorzeichen beim Wert 0,0 ergeben, finden Sie nachfolgend unter *Ergebnisse der Gleitkommaarithmetik*.

Unendlich

Werte von $+\infty$ und $-\infty$ können ausgedrückt werden, indem das Vorzeichen für positiv auf 1 und für negativ auf 0 gesetzt wird. der Exponent muss den Wert 2.047 $(2^{11} - 1)$ und die Mantisse den Wert 0 haben.

NaN

NaN (keine Zahl) ergibt sich, wenn das Ergebnis einer Berechnung, wie z. B. 0,0/0,0, ∞/∞ oder $\infty-\infty$, keiner Zahl und nicht unendlich entspricht. der Exponent hat in diesem Fall den Wert 255 $(2^8 - 1)$ und die Mantisse ist nicht 0.

Hinweis Es gibt keine Spezifikationen hinsichtlich den Vorzeichens von NaN oder des Werts im Mantissenfeld (außer, dass sie nicht 0 sein darf).

Ergebnisse der Gleitkommaarithmetik

Rundung von Ergebnissen

Die folgenden Methoden werden zur Rundung von Ergebnissen angewandt, wenn die Anzahl der Stellen im genauen Ergebnis der Gleitkommaarithmetik die Anzahl der wertigen Stellen von internen Verarbeitungsausdrücken überschreitet.

Wenn das Ergebnis nahe bei einem von zwei internen Gleitkommaausdrükken liegt, wird der näher liegende Ausdruck verwendet. Wenn das Ergebnis genau in der Mitte zwischen zwei Gleitkommaausdrücken liegt, wird das Ergebnis so gerundet, dass die letzte Stelle der Mantisse eine 0 ist.

Überläufe, Unterläufe und ungültige Berechnungen

Ein Überlauf wird, je nach Vorzeichen des Ergebnisses, entweder als positiv oder negativ Unendlich ausgegeben. Ein Unterlauf wird, je nach Vorzeichen des Ergebnisses, entweder als positive oder negative Null ausgegeben.

Ungültige (nicht definierte) Berechnungen führen zu dem Ergebnis NaN. Zu den ungültigen Berechnungen gehört die Addition von Unendlich zu einer Zahl mit umgekehrtem Vorzeichen, die Subtraktion von Unendlich von einer Zahl mit umgekehrtem Vorzeichen, die Multiplikation von Null und Unendlich, die Division von Null durch Null oder die Division von Unendlich durch Unendlich.

Beim Auftreten eines Überlaufs ist das Ergebnis nach der Konvertierung einer Gleitkommazahl in eine Ganzzahl möglicherweise nicht korrekt.

Vorsichtsmaßnahmen beim Umgang mit speziellen Werten

Die folgenden Vorsichtsmaßnahmen gelten für den Umgang mit Null, Unendlich und NaN.

- Die Summe aus positiver Null und negativer Null ist eine positive Null.
- Die Differenz von zwei Nullen mit gleichem Vorzeichen ist eine positive Null.
- Wenn eine beliebiger Operand NaN ist, so ist auch das Ergebnis NaN.
- Die positive Null und die negative Null werden bei Vergleichen als äquivalent behandelt.
- Vergleiche oder Äquivalenzprüfungen mit einer oder mehreren NaN haben beim Befehl != immer wahr als Ergebnis und bei allen anderen Befehlen stets falsch.

Ergebnisse von Gleitkomma-Berechnungen doppelter Genauigkeit

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben. Ist das Ergebnis positiv, wird es als $+\infty$ ausgegeben, ist es negativ, als $-\infty$.

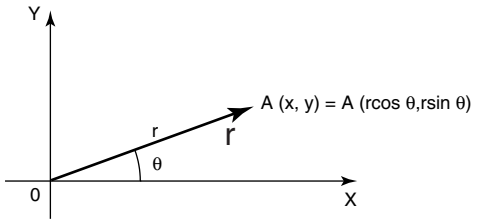
Der Gleich-Merker wird nur auf EIN gesetzt, wenn sowohl der Exponent (e) als auch die Mantisse (f) nach der Berechnung gleich Null sind. Ein Berechnungsergebnis wird auch als Null ausgegeben, wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich mit Gleitkommadaten ausdrücken lässt. In diesem Fall wird der Unterlaufmerker auf EIN gesetzt.

Vergleich zwischen Berechnungen mit einfacher und doppelter Genauigkeit

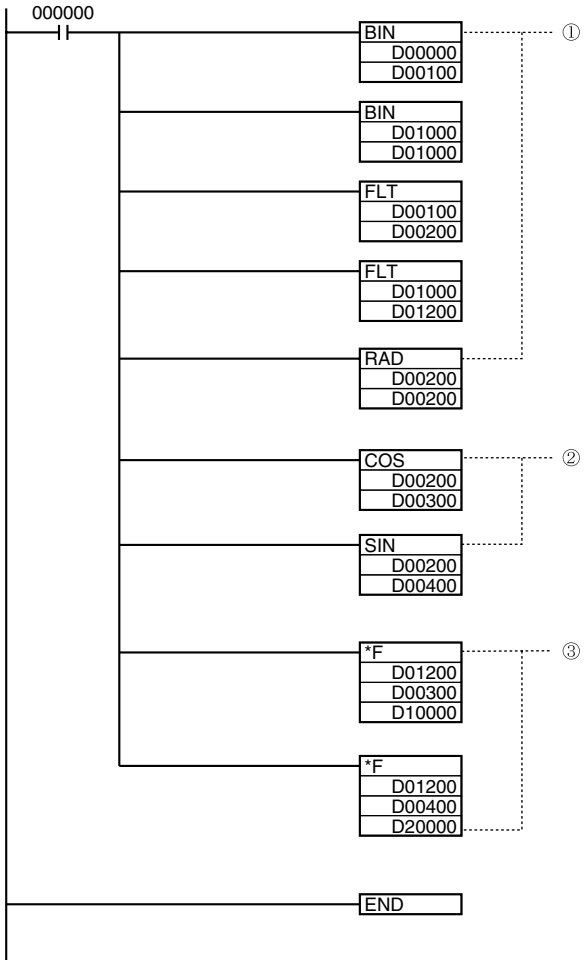
Dieses Beispiel zeigt die Unterschiede zwischen Berechnungen einfacher und doppelter Genauigkeit, wenn der folgende, in Polarkoordinaten ausgedrückte Vektor in Rechteckkoordinaten A (x,y) konvertiert wird.

$$r = re^{j\left(\frac{\pi}{360}\right)\theta}$$

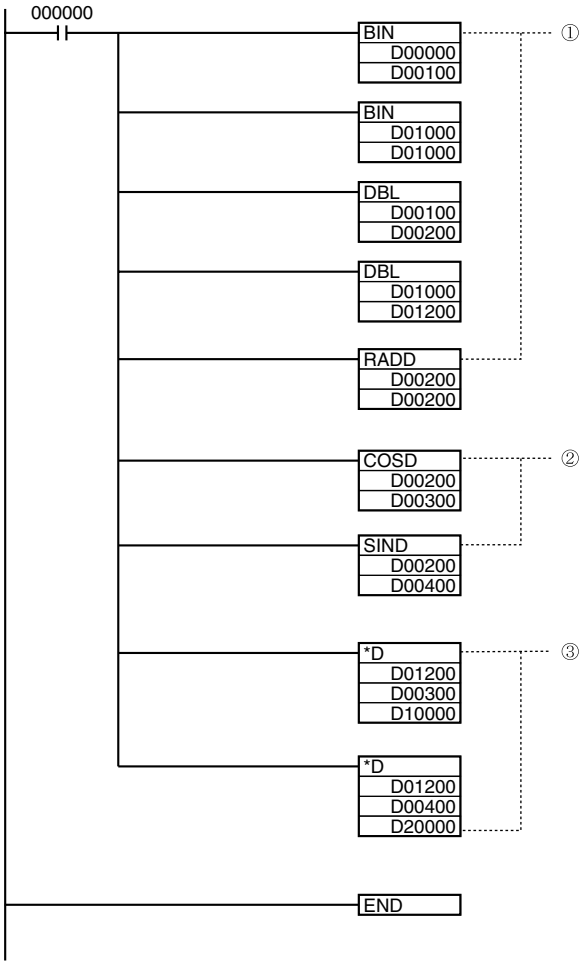
In diesem Beispiel wird der 4-stellige BCD-Winkel (θ, in Grad) aus D00000 und der 4-stellige BCD-Abstand (r) aus D01000 gelesen.



- Kontaktplanprogramm für die Berechnung einfacher Genauigkeit



- Kontaktplanprogramm für die Berechnung doppelter Genauigkeit



1. Dieser Programmabschnitt konvertiert die BCD-Daten in Gleitkommadaten einfacher Genauigkeit (32 Bits, IEEE754-Format).
   a) Die BIN(023)-Befehle konvertieren die BCD-Daten in Binärdaten, und die FLT(452)-Befehle konvertieren die Binärdaten in Gleitkommadaten einfacher Genauigkeit.
   b) Die Gleitkommadaten für den Winkel θ werden in D00200 und D00201 ausgegeben.
   c) RAD(458) konvertiert die Winkeldaten in D00200 und D00201 zu Radianten.
   d) Die Gleitkommadaten für den Radius r werden in D01200 und D01201 ausgegeben.
2. Dieser Programmabschnitt berechnet den sin θ und den cos θ als Gleitkommawerte einfacher Genauigkeit.
   a) Der Wert für cos θ wird nach D00300 und D00301 ausgegeben.
   b) Der Wert für sin θ wird nach D00400 und D00401 ausgegeben.
3. Dieser Programmabschnitt berechnet x ($r \times \cos \theta$) und y ($r \times \sin \theta$).
   a) Der Wert für x ($r \times \cos \theta$) wird nach D10000 und D10001 ausgegeben.
   b) Der Wert für y ($r \times \sin \theta$) wird nach D20000 und D20001 ausgegeben.

Koordinate	Gleitkommazahl	Reelle Zahl
x	4116 59CF	3,4202015399933
y	405A E495	9,3969259262085

1. Dieser Programmabschnitt konvertiert die BCD-Daten in Gleitkommadaten doppelter Genauigkeit (64 Bits, IEEE754-Format).
   a) Die BIN(023)-Befehle konvertieren die BCD-Daten in Binärdaten, und die DBL(843)-Befehle konvertieren die Binärdaten in Gleitkommadaten doppelter Genauigkeit.
   b) Die Gleitkommadaten für den Winkel θ werden in die Worte D00200 bis D00203 ausgegeben.
   c) RADD(849) konvertiert die Winkeldaten in den Worten D00200 bis D00203 zu Radianten.
   d) Die Gleitkommadaten für den Radius r werden in die Worte D01200 bis D01203 ausgegeben.
2. Dieser Programmabschnitt berechnet den sin θ und den cos θ als Gleitkommawerte doppelter Genauigkeit.
   a) Der Wert für cos θ wird in die Worte D00300 bis D00303 ausgegeben.
   b) Der Wert für sin θ wird in die Worte D00400 bis D00403 ausgegeben.
3. Dieser Programmabschnitt berechnet x ($r \times \cos \theta$) und y ($r \times \sin \theta$).
   a) Der Wert für x ($r \times \cos \theta$) wird in die Worte D10000 bis D10003 ausgegeben.
   b) Der Wert für y ($r \times \sin \theta$) wird in die Worte D20000 bis D20003 ausgegeben.

Koordinate	Gleitkommazahl	Reelle Zahl
x	4022 CB39 E973 5C32	3,4202014332567
y	400B 5C92 91AC 8EEB	9,3969262078591

Vergleich der Berechnungsergebnisse

Bei Vergleich der reellen Ergebniszahlen wird deutlich, dass die Berechnung doppelter Genauigkeit zu einen genaueren Ergebnis führt.

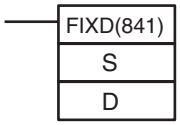
3-16-1 DOUBLE FLOATING TO 16-BIT: FIXD(841)

Zweck

Konvertiert einen (64-Bit-) Gleitkommawert doppelter Genauigkeit in 16-Bit-Binärdaten mit Vorzeichen, und gibt das Ergebnis in das spezifizierte Ergebniswort aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Erstes Quellenwort

D: Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FIXD(841)
	Ausführung einmal bei steigender Flanke	@FIXD(841)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

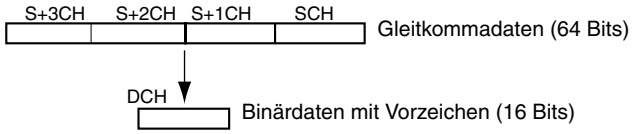
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W508	W000 bis W511
Haftmerkerbereich	H000 bis H508	H000 bis H511
Zusatz-Systembereich	A000 bis A956	A448 bis A959
Zeitgeberbereich	T0000 bis T4092	T0000 bis T4095
Zählerbereich	C0000 bis C4092	C0000 bis C4095
DM-Bereich	D00000 bis D32764	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32764	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	DR0 bis DR15
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl FIXD(841) konvertiert den ganzzahligen Teil der (64-Bit-) Gleitkommazahl doppelter Genauigkeit in S bis S+3 (IEEE754-Format) in 16-Bit-Binärdaten mit Vorzeichen und schreibt das Ergebnis in D.



Nur der ganzzahlige Teil der Gleitkommazahl wird konvertiert, der Dezimalbruchteil wird abgeschnitten. Der ganzzahlige Teil der Gleitkommazahl muss im Bereich zwischen –32.768 und 32.767 liegen.

Beispiele für Konvertierungen:
Ein Gleitkommawert von 3,5 wird zu 3 konvertiert.
Ein Gleitkommawert von –3,5 wird zu –3 konvertiert.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten (S bis S+3) keine Zahl (NaN) sind. EIN, wenn der ganzzahlige Teil der Quelldaten (S bis S+3) nicht innerhalb des Bereichs von –32.768 bis 32.767 liegt. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn Bit 15 des Ergebnisses EIN ist. In allen anderen Fällen AUS.

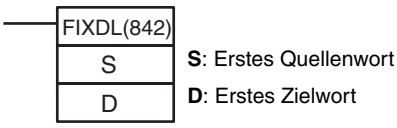
3-16-2 DOUBLE FLOATING TO 32-BIT: FIXLD(842)

Zweck

Konvertiert einen (64-Bit-) Gleitkommawert doppelter Genauigkeit in 32-Bit-Binärdaten mit Vorzeichen, und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FIXLD(842)
	Ausführung einmal bei steigender Flanke	@FIXLD(842)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

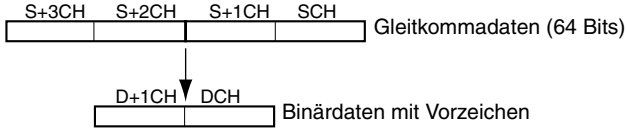
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	CIO 0000 bis CIO 6142
Arbeitsbereich	W000 bis W508	W000 bis W510

Bereich	S	D
Haftmerkerbereich	H000 bis H508	H000 bis H510
Zusatz-Systembereich	A000 bis A956	A448 bis A958
Zeitgeberbereich	T0000 bis T4092	T0000 bis T4094
Zählerbereich	C0000 bis C4092	C0000 bis C4094
DM-Bereich	D00000 bis D32764	D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32764	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl FIXLD(842) konvertiert den ganzzahligen Teil der (64-Bit-) Gleitkommazahl doppelter Genauigkeit in S bis S+3 (IEEE754-Format) in 32-Bit-Binärdaten mit Vorzeichen und schreibt das Ergebnis in D+1 und D.



Nur der ganzzahlige Teil der Gleitkommazahl wird konvertiert, der Dezimalbruchteil wird abgeschnitten. (Der ganzzahlige Teil der Gleitkommazahl muss im Bereich zwischen –2.147.483.648 und 2.147.483.647 liegen.)

Beispiele für Konvertierungen:
Ein Gleitkommawert von 2.147.483.640,5 wird zu 2.147.483.640 konvertiert.
Ein Gleitkommawert von –2.147.483.640,5 wird zu -2.147.483.640 konvertiert.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Daten in den Worten S bis S+3 keine Zahl (NaN) sind. EIN, wenn der ganzzahlige Teil der Worte S bis S+3 nicht innerhalb des Bereichs von -2.147.483.648 bis 2.147.483.647 liegt. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0000 0000 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn Bit 15 von D+1 nach der Ausführung auf EIN gesetzt ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Beim Inhalt der Worte S bis S+3 muss es sich um Gleitkommadaten handeln, deren ganzzahliger Teil im Bereich zwischen –2.147.483.648 und 2.147.483.647 liegt.

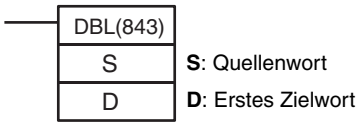
3-16-3 16-BIT TO DOUBLE FLOATING: DBL(843)

Zweck

Konvertiert einen 16-Bit-Binärwert mit Vorzeichen in (64-Bit-) Gleitkommadaten doppelter Genauigkeit und gibt das Ergebnis in die spezifizierten Zielworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	DBL(843)
	Ausführung einmal bei steigender Flanke	@DBL(843)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

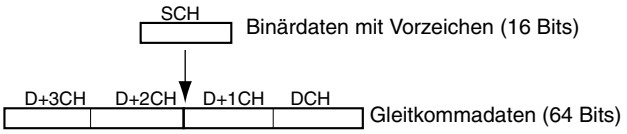
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6140
Arbeitsbereich	W000 bis W511	W000 bis W508
Haftmerkerbereich	H000 bis H511	H000 bis H508
Zusatz-Systembereich	A000 bis A959	A448 bis A956
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4092
Zählerbereich	C0000 bis C4095	C0000 bis C4092
DM-Bereich	D00000 bis D32767	D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #FFFF (binär)	---
Datenregister	DR0 bis DR15	---

Bereich	S	D
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl DBL(843) konvertiert die 16-Bit-Binärdaten mit Vorzeichen in S zu (64-Bit-) Gleitkommadaten doppelter Genauigkeit (IEEE754-Format) und schreibt das Ergebnis in die Worte D bis D+3. Bei der resultierenden Gleitkommazahl wird eine einzelne 0 als Nachkommastelle hinzugefügt.



Für S können nur Werte innerhalb des Bereichs von –32.768 bis 32.767 spezifiziert werden. Zur Konvertierung von Binärdaten außerhalb dieses Bereich kann der Befehl DBLL(844) verwendet werden.

Beispiele für Konvertierungen:
Ein vorzeichenbehafteter Binärwert von 3 wird zu 3,0 konvertiert.
Ein vorzeichenbehafteter Binärwert von -3 wird zu -3,0 konvertiert.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Beim Inhalt von S muss es sich um vorzeichenbehaftete Binärdaten mit einem Wert (dezimal) im Bereich von –32.768 bis 32.767 handeln.

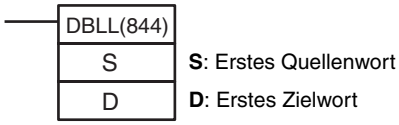
3-16-4 32-BIT TO DOUBLE FLOATING: DBLL(844)

Zweck

Konvertiert einen 32-Bit-Binärwert mit Vorzeichen in (64-Bit-) Gleitkommadaten doppelter Genauigkeit und gibt das Ergebnis in die spezifizierten Zielworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	DBLL(844)
	Ausführung einmal bei steigender Flanke	@DBLL(844)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

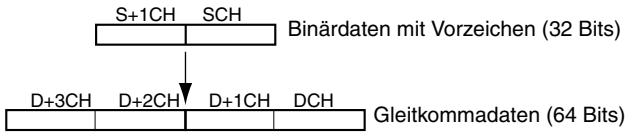
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6140
Arbeitsbereich	W000 bis W510	W000 bis W508
Haftmerkerbereich	H000 bis H510	H000 bis H508
Zusatz-Systembereich	A000 bis A958	A448 bis A956
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4092
Zählerbereich	C0000 bis C4094	C0000 bis C4092
DM-Bereich	D00000 bis D32766	D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#00000000 bis #FFFFFFFF (binär)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl DBLL(844) konvertiert die 32-Bit-Binärdaten mit Vorzeichen in S+1 und S zu (64-Bit-) Gleitkommadaten doppelter Genauigkeit (IEEE754-Format) und schreibt das Ergebnis in die Worte D bis D+3. Bei der resultierenden Gleitkommazahl wird eine einzelne 0 als Nachkommastelle hinzugefügt.



Für S+1 und S können vorzeichenbehaftete Binärdaten im Bereich von –2.147.483.648 bis 2.147.483.647 spezifiziert werden. Der Gleitkommawert hat 24 wertige Binärstellen (Bits). Das Ergebnis ist nicht genau, wenn durch den Befehl DBLL(844) eine Zahl konvertiert wird, die größer als 16.777.215 (der Maximalwert, der sich mit 24 Bits ausdrücken lässt) ist.

Beispiele für Konvertierungen:

Ein vorzeichenbehafteter Binärwert von 16.777.215 wird zu 16.777.215,0 konvertiert.

Ein vorzeichenbehafteter Binärwert von –16.777.215 wird zu –15.777.215,0 konvertiert.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	AUS
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Das Ergebnis ist nicht genau, wenn eine Zahl konvertiert wird, deren Absolutwert größer als 16.777.215 (der Maximalwert, der sich mit 24 Bits ausdrücken lässt) ist.

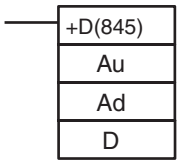
3-16-5 DOUBLE FLOATING-POINT ADD: +D(845)

Zweck

Addiert zwei (64-Bit-) Gleitkommazahlen doppelter Genauigkeit und gibt das Ergebnis in die spezifizierten Zielworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Au: Erstes Wort des 1. Summanden
Ad: Erstes Wort des 2. Summanden
D: Erstes Zielwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	+D(845)
	Ausführung einmal bei steigender Flanke	@+D(845)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

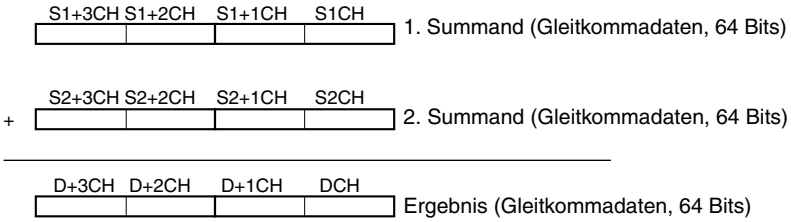
Operanden-Spezifikationen

Bereich	Au	Ad	D
CIO-Bereich	CIO 0000 bis CIO 6140		
Arbeitsbereich	W000 bis W508		
Haftmerkerbereich	H000 bis H508		
Zusatz-Systembereich	A000 bis A956		A448 bis A956
Zeitgeberbereich	T0000 bis T4092		
Zählerbereich	C0000 bis C4092		
DM-Bereich	D00000 bis D32764		
EM-Bereich ohne Bank	E00000 bis E32764		
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)		

Bereich	Au	Ad	D
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl +D(845) addiert die (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten Ad bis Ad+3 zu der (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten Au bis Au+3 und schreibt das Ergebnis in die Worte D bis D+3. (Die Gleitkommadaten müssen im IEEE754-Format vorliegen.)



Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Die verschiedenen Kombinationen von Daten für den 1. und 2. Summanden führen zu den in der folgenden Tabelle aufgeführten Ergebnissen.

	1. Summand				
2. Summand	**0**	**Zahl**	**+∞**	**–∞**	**NaN**
0	0	Zahl	+∞	–∞	
Zahl	Zahl	Siehe Hinweis 1	+∞	–∞	
+∞	+∞	+∞	+∞	Siehe Hinweis 2	
–∞	–∞	–∞	Siehe Hinweis 2	–∞	
NaN					Siehe Hinweis 2

Hinweis

1. Das Ergebnis kann Null (einschließlich Unterlauf), eine Zahl, +∞ oder –∞ sein.
2. Der Fehlermerker wird auf EIN gesetzt und der Befehl wird nicht ausgeführt.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Daten für den 1. oder 2. Summanden nicht als Gleitkommadaten erkannt werden. EIN, wenn die Daten für den 1. oder 2. Summanden keine Zahl (NaN) sind. EIN, wenn +∞ zu –∞ addiert wird. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Daten des 1. Summanden (Au bis Au+3) und des 2. Summanden (Ad bis Ad+3) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

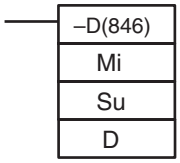
3-16-6 DOUBLE FLOATING-POINT SUBTRACT: –D(846)

Zweck

Subtrahiert eine (64-Bit-) Gleitkommazahl doppelter Genauigkeit von einer anderen und gibt das Ergebnis in die spezifizierten Zielworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Mi: Erstes Minuendenwort

Su: Erstes Subtrahendenwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	–D(846)
	Ausführung einmal bei steigender Flanke	@–D(846)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

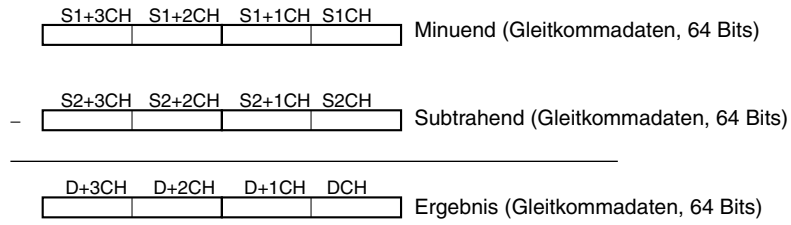
Operanden-Spezifikationen

Bereich	Mi	Su	D
CIO-Bereich	CIO 0000 bis CIO 6140		
Arbeitsbereich	W000 bis W508		
Haftmerkerbereich	H000 bis H508		
Zusatz-Systembereich	A000 bis A956		A448 bis A956
Zeitgeberbereich	T0000 bis T4092		
Zählerbereich	C0000 bis C4092		
DM-Bereich	D00000 bis D32764		

Bereich	Mi	Su	D
EM-Bereich ohne Bank	E00000 bis E32764		
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl –D(846) subtrahiert die (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten Su bis Su+3 von der (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten Mi bis Mi+3 und schreibt das Ergebnis in die Worte D bis D+3. (Die Gleitkommadaten müssen im IEEE754-Format vorliegen.)



Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Die verschiedenen Kombinationen von Daten für Minuend und Subtrahend führen zu den in der folgenden Tabelle aufgeführten Ergebnissen.

	Minuend				
Subtrahend	**0**	**Zahl**	**+∞**	**–∞**	**NaN**
0	0	Zahl	+∞	–∞	
Zahl	Zahl	Siehe Hinweis 1	+∞	–∞	
+∞	–∞	–∞	Siehe Hinweis 2	–∞	
–∞	+∞	+∞	+∞	Siehe Hinweis 2	
NaN					Siehe Hinweis 2

Hinweis

1. Das Ergebnis kann Null (einschließlich Unterlauf), eine Zahl, +∞ oder –∞ sein.
2. Der Fehlermerker wird auf EIN gesetzt und der Befehl wird nicht ausgeführt.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Minuenden- und Subtrahendendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Minuenden- oder Subtrahendendaten keine Zahl (NaN) sind. EIN, wenn +∞ von +∞ subtrahiert wird. EIN, wenn –∞ von –∞ subtrahiert wird. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Minuendendaten (Mi bis Mi+3) und die Subtrahendendaten (Su bis Su+3) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

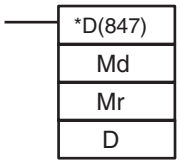
3-16-7 DOUBLE FLOATING-POINT MULTIPLY: ∗D(847)

Zweck

Multipliziert zwei (64-Bit-) Gleitkommazahlen doppelter Genauigkeit miteinander und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Md: Erstes Multiplikandenwort

Mr: Erstes Multiplikatorwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	*D(847)
	Ausführung einmal bei steigender Flanke	@*D(847)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

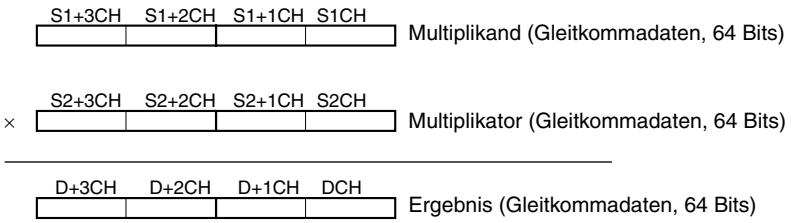
Operanden-Spezifikationen

Bereich	Md	Mr	D
CIO-Bereich	CIO 0000 bis CIO 6140		
Arbeitsbereich	W000 bis W508		
Haftmerkerbereich	H000 bis H508		
Zusatz-Systembereich	A000 bis A956		A448 bis A956
Zeitgeberbereich	T0000 bis T4092		
Zählerbereich	C0000 bis C4092		
DM-Bereich	D00000 bis D32764		

Bereich	Md	Mr	D
EM-Bereich ohne Bank	E00000 bis E32764		
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

∗D(847) multipliziert die (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten Md bis Md+3 mit der (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten Mr bis Mr+3 und schreibt das Ergebnis in die Worte D bis D+3. (Die Gleitkommadaten müssen im IEEE754-Format vorliegen.)



Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Die verschiedenen Kombinationen von Multiplikanden- und Multiplikatordaten führen zu den in der folgenden Tabelle aufgeführten Ergebnissen.

	Multiplikand				
Multiplikator	**0**	**Zahl**	**+∞**	**–∞**	**NaN**
0	0	0	Siehe Hinweis 2	Siehe Hinweis 2	
Zahl	0	Siehe Hinweis 1	+/–∞	+/–∞	
+∞	Siehe Hinweis 2	+/–∞	+∞	–∞	
–∞	Siehe Hinweis 2	+/–∞	–∞	+∞	
NaN					Siehe Hinweis 2

Hinweis

1. Das Ergebnis kann Null (einschließlich Unterlauf), eine Zahl, +∞ oder –∞ sein.
2. Der Fehlermerker wird auf EIN gesetzt und der Befehl wird nicht ausgeführt.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Multiplikanden- oder Multiplikatordaten nicht als Gleitkommadaten erkannt werden. EIN, wenn Multiplikand oder Multiplikator keine Zahl (NaN) ist. EIN, wenn +∞ und 0 miteinander multipliziert werden. EIN, wenn –∞ und 0 miteinander multipliziert werden. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Daten von Multiplikand (Md bis Md+3) und Multiplikator (Mr bis Mr+3) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

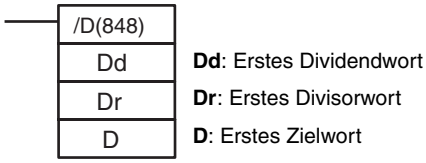
3-16-8 DOUBLE FLOATING-POINT DIVIDE: /D(848)

Zweck

Dividiert eine (64-Bit-) Gleitkommazahl doppelter Genauigkeit durch eine andere und gibt das Ergebnis in die spezifizierten Zielworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	/D(848)
	Ausführung einmal bei steigender Flanke	@/D(848)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

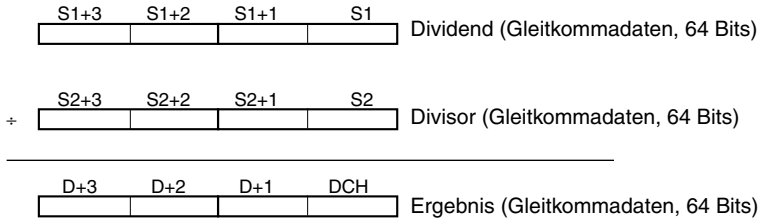
Operanden-Spezifikationen

Bereich	Dd	Dr	D
CIO-Bereich	CIO 0000 bis CIO 6140		
Arbeitsbereich	W000 bis W508		
Haftmerkerbereich	H000 bis H508		
Zusatz-Systembereich	A000 bis A956		A448 bis A956
Zeitgeberbereich	T0000 bis T4092		
Zählerbereich	C0000 bis C4092		
DM-Bereich	D00000 bis D32764		

Bereich	Dd	Dr	D
EM-Bereich ohne Bank	E00000 bis E32764		
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

/D(848) dividiert die (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten Dd bis Dd+3 durch die (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten Dr bis Dr+3 und schreibt das Ergebnis in die Worte D bis D+3. (Die Gleitkommadaten müssen im IEEE754-Format vorliegen.)



Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Die verschiedenen Kombinationen von Daten für Dividend und Divisor führen zu den in der folgenden Tabelle aufgeführten Ergebnissen.

	Dividend				
Divisor	**0**	**Zahl**	**+∞**	**–∞**	**NaN**
0	Siehe Hinweis 3	+/–∞	+∞	–∞	
Zahl	0	Siehe Hinweis 1	+/–∞	+/–∞	
+∞	0	Siehe Hinweis 2	Siehe Hinweis 3	Siehe Hinweis 3	
–∞	0	Siehe Hinweis 2	Siehe Hinweis 3	Siehe Hinweis 3	
NaN					Siehe Hinweis 3

Hinweis

1. Das Ergebnis kann Null (einschließlich Unterlauf), eine Zahl, +∞ oder –∞ sein.
2. Bei einem Unterlauf ist das Ergebnis Null.
3. Der Fehlermerker wird auf EIN gesetzt und der Befehl wird nicht ausgeführt.

Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn die Dividend- und Divisordaten nicht als Gleitkommadaten erkannt werden. EIN, wenn der Dividend oder der Divisor keine Zahl (NaN) ist. EIN, wenn Dividend und Divisor beide 0 sind. EIN, wenn Dividend und Divisor beide +∞ oder –∞ sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Daten von Dividend (Dd bis Dd+3) und Divisor (Dr bis Dr-3) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

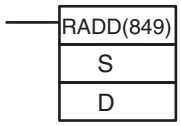
3-16-9 DOUBLE DEGREES TO RADIANS: RADD(849)

Zweck

Konvertiert eine (64-Bit-) Gleitkommazahl doppelter Genauigkeit von Grad zu Radianten und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Erstes Quellenwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	RADD(849)
	Ausführung einmal bei steigender Flanke	@RADD(849)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

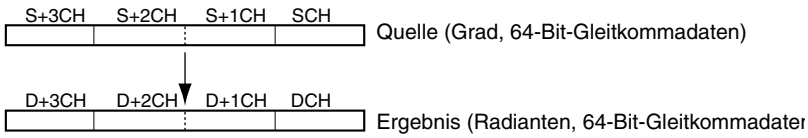
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	

Bereich	S	D
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl RADD(849) konvertiert die (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten S bis S+3 aus Grad zu Radianten und schreibt das Ergebnis in die Worte D bis D+3. (Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Die Konvertierung von Grad in Radianten erfolgt anhand der folgenden Formel:

Grad $\times$ π/180 = Radianten

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

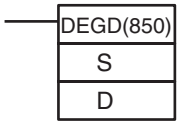
3-16-10 DOUBLE RADIANS TO DEGREES: DEGD(850)

Zweck

Konvertiert eine (64-Bit-) Gleitkommazahl doppelter Genauigkeit von Radianten zu Grad und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Erstes Quellenwort

D: Erstes Zielwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	DEGD(850)
	Ausführung einmal bei steigender Flanke	@DEGD(850)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

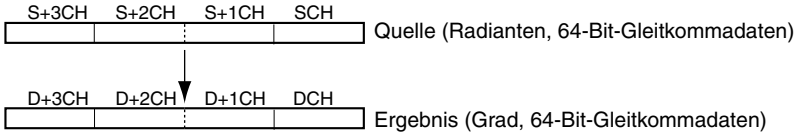
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	

Bereich	S	D
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl DEGD(850) konvertiert die (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten S bis S+3 aus Radianten zu Grad und schreibt das Ergebnis in die Worte D bis D+3. (Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Die Konvertierung von Radianten zu Grad erfolgt anhand der folgenden Formel:

Radianten $\times$ 180/π = Grad

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

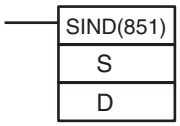
3-16-11 DOUBLE SINE: SIND(851)

Zweck

Berechnet den Sinus einer (64-Bit-) Gleitkommazahl doppelter Genauigkeit (in Radianten) und gibt das Ergebnis in die spezifizierten Zielworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Erstes Quellenwort

D: Erstes Zielwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SIND(851)
	Ausführung einmal bei steigender Flanke	@SIND(851)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

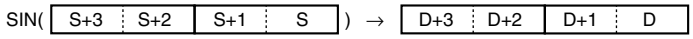
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

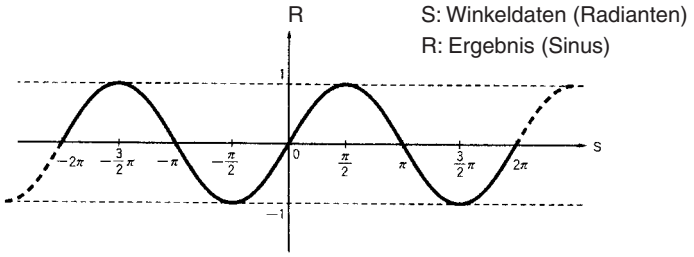
Beschreibung

SIND(851) berechnet den Sinus des Winkels (in Radianten), der als (64-Bit-) Gleitkommawert doppelter Genauigkeit ausgedrückt in den Worten S bis S+3 steht, und schreibt das Ergebnis in die Worte D bis D+3.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Spezifizieren Sie den gewünschten Winkel (–65.535 bis 65.535) in Radianten in den Worten S bis S+3. Wenn der Winkel außerhalb des Bereichs zwischen –65.535 und 65.535 liegt, tritt ein Fehler auf und der Befehl wird nicht ausgeführt. Informationen zur Konvertierung zwischen Grad und Radianten finden Sie unter *3-16-9 DOUBLE DEGREES TO RADIANS: RADD(849)* oder *3-16-10 DOUBLE RADIANS TO DEGREES: DEGD(850)*.

Die folgende Abbildung zeigt die Beziehung zwischen Winkel und Ergebnis.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 65.535 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	Unverändert
Unterlauf-Merker	UF	Unverändert
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

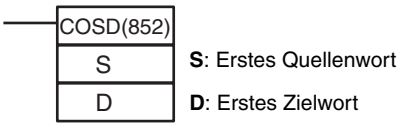
3-16-12 DOUBLE COSINE: COSD(852)

Zweck

Berechnet den Kosinus einer (64-Bit-) Gleitkommazahl doppelter Genauigkeit (in Radianten) und gibt das Ergebnis in die spezifizierten Zielworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	COSD(852)
	Ausführung einmal bei steigender Flanke	@COSD(852)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

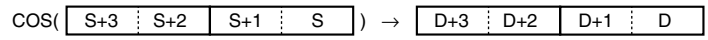
Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

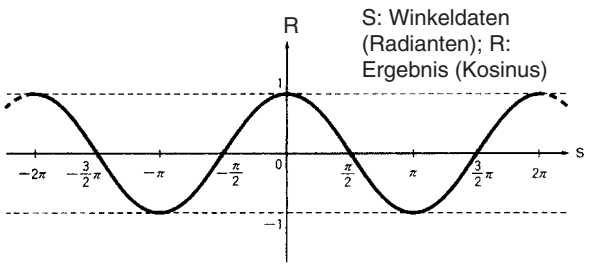
Beschreibung

COSD(852) berechnet den Kosinus des Winkels (in Radianten), der als (64-Bit-) Gleitkommawert doppelter Genauigkeit ausgedrückt in den Worten S bis S+3 steht, und schreibt das Ergebnis in die Worte D bis D+3.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Spezifizieren Sie den gewünschten Winkel (–65.535 bis 65.535) in Radianten in den Worten S bis S+3. Wenn der Winkel außerhalb des Bereichs zwischen –65.535 und 65.535 liegt, tritt ein Fehler auf und der Befehl wird nicht ausgeführt. Informationen zur Konvertierung zwischen Grad und Radianten finden Sie unter *3-16-9 DOUBLE DEGREES TO RADIANS: RADD(849)* oder *3-16-10 DOUBLE RADIANS TO DEGREES: DEGD(850)*.

Die folgende Abbildung zeigt die Beziehung zwischen Winkel und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 65.535 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	Unverändert
Unterlauf-Merker	UF	Unverändert
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

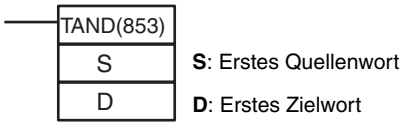
3-16-13 DOUBLE TANGENT: TAND(853)

Zweck

Berechnet den Tangens einer (64-Bit-) Gleitkommazahl doppelter Genauigkeit (in Radianten) und gibt das Ergebnis in die spezifizierten Zielworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TAND(853)
	Ausführung einmal bei steigender Flanke	@TAND(853)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

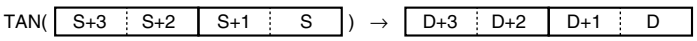
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	

Bereich	S	D
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

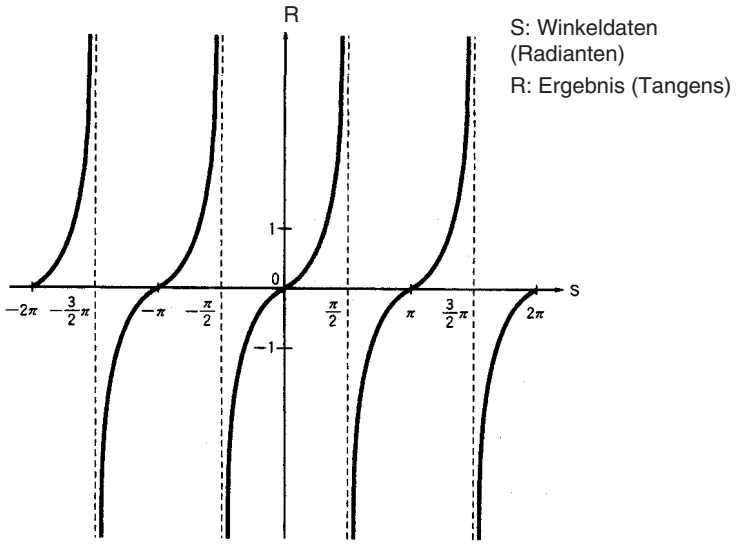
TAND(853) berechnet den Tangens des Winkels (in Radianten), der als (64-Bit-) Gleitkommawert doppelter Genauigkeit ausgedrückt in den Worten S bis S+3 steht, und schreibt das Ergebnis in die Worte D bis D+3.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Spezifizieren Sie den gewünschten Winkel (–65.535 bis 65.535) in Radianten in den Worten S bis S+3. Wenn der Winkel außerhalb des Bereichs zwischen –65.535 und 65.535 liegt, tritt ein Fehler auf und der Befehl wird nicht ausgeführt. Informationen zur Konvertierung zwischen Grad und Radianten finden Sie unter *3-16-9 DOUBLE DEGREES TO RADIANS: RADD(849)* oder *3-16-10 DOUBLE RADIANS TO DEGREES: DEGD(850)*.

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Winkel und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 65.535 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als (64-Bit-) Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	Unverändert
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

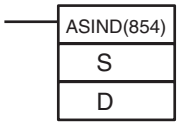
3-16-14 DOUBLE ARC SINE: ASIND(854)

Zweck

Berechnet den Arkussinus einer (64-Bit-) Gleitkommazahl doppelter Genauigkeit und gibt das Ergebnis in die spezifizierten Zielworte aus. (Die Arkussinusfunktion ist die Umkehrung der Sinusfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Sinuswert zwischen -1 und 1 entspricht.)

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Erstes Quellenwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ASIND(854)
	Ausführung einmal bei steigender Flanke	@ASIND(854)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

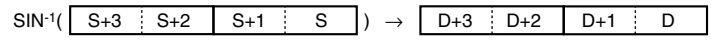
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	

Bereich	S	D
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

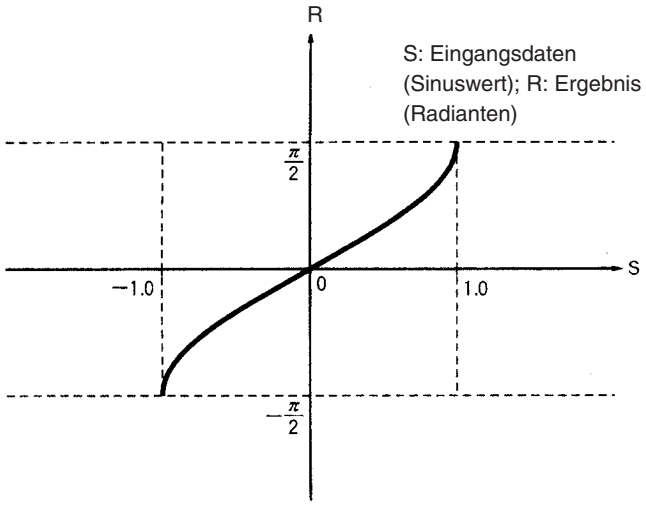
ASIND(854) berechnet den Winkel (in Radianten) eines Sinuswerts, der als (64-Bit-) Gleitkommazahl doppelter Genauigkeit ausgedrückt in den Worten S bis S+3 steht, und schreibt das Ergebnis in die Worte D bis D+3.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Die Quellendaten müssen im Bereich zwischen –1,0 und 1,0 liegen. Wenn der Absolutwert der Quellendaten größer als 1,0 ist, tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

Das Ergebnis wird als Winkel (in Radianten) im Bereich zwischen $-\pi/2$ und $\pi/2$ in die Worte D bis D+3 ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 1,0 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	Unverändert
Unterlauf-Merker	UF	Unverändert
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

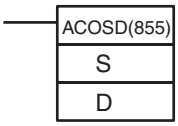
3-16-15 DOUBLE ARC COSINE: ACOSD(855)

Zweck

Berechnet den Arkuskossinus einer (64-Bit-) Gleitkommazahl doppelter Genauigkeit und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. (Die Arkuskosinusfunktion ist die Umkehrung der Kosinusfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Kosinuswert zwischen –1 und 1 entspricht.)

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Erstes Quellenwort

D: Erstes Zielwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	ACOSD(855)
	Ausführung einmal bei steigender Flanke	@ACOSD(855)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

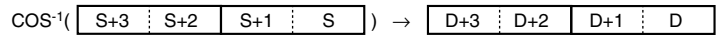
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	

Bereich	S	D
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

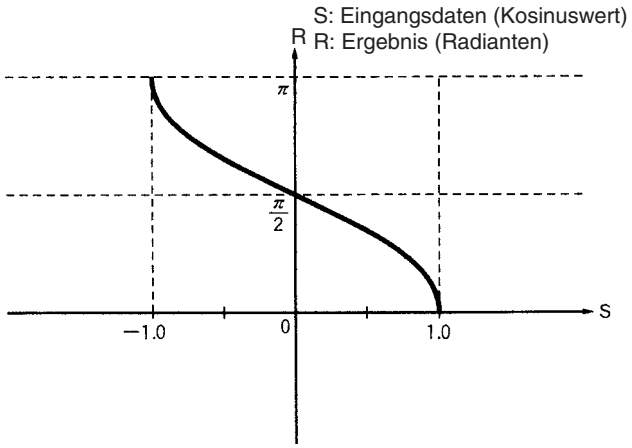
ACOSD(855) berechnet den Winkel (in Radianten) eines Kosinuswerts, der als (64-Bit-) Gleitkommazahl doppelter Genauigkeit ausgedrückt in den Worten S bis S+3 steht, und schreibt das Ergebnis in die Worte D bis D+3. (Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Die Quellendaten müssen im Bereich zwischen –1,0 und 1,0 liegen. Wenn der Absolutwert der Quellendaten größer als 1,0 ist, tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

Das Ergebnis wird als Winkel (in Radianten) im Bereich zwischen 0 und π in die Worte D bis D+3 ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. EIN, wenn der Absolutwert der Quellendaten größer als 1,0 ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	Unverändert
Unterlauf-Merker	UF	Unverändert
Negativ-Merker	N	Unverändert

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

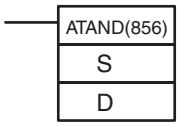
3-16-16 DOUBLE ARC TANGENT: ATAND(856)

Zweck

Berechnet den Arkustangens einer (64-Bit-) Gleitkommazahl doppelter Genauigkeit und gibt das Ergebnis in die spezifizierten Ergebnisworte aus. (Die Arkustangensfunktion ist die Umkehrung der Tangensfunktion. Sie ergibt wieder den Winkel, der einem gegebenen Tangenswert entspricht.)

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Erstes Quellenwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ATAND(856)
	Ausführung einmal bei steigender Flanke	@ATAND(856)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

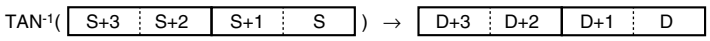
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	

Bereich	S	D
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

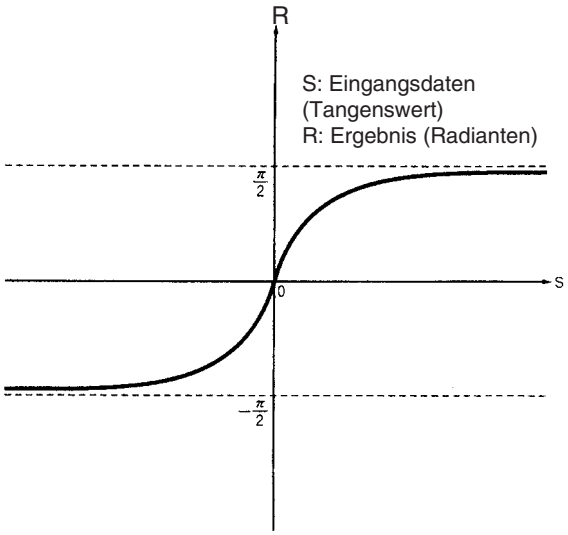
Beschreibung

ATAND(856) berechnet den Winkel (in Radianten) eines Tangenswerts, der als (64-Bit-) Gleitkommazahl doppelter Genauigkeit ausgedrückt in den Worten S bis S+3 steht, und schreibt das Ergebnis in D bis D+3.
(Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Das Ergebnis wird als Winkel (in Radianten) im Bereich zwischen $-\pi/2$ und $\pi/2$ in die Worte D bis D+3 ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	Unverändert
Unterlauf-Merker	UF	Unverändert
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

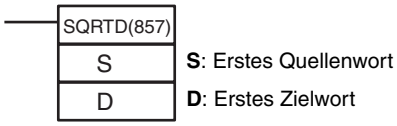
3-16-17 DOUBLE SQUARE ROOT: SQRTD(857)

Zweck

Berechnet die Quadratwurzel einer (64-Bit-) Gleitkommazahl doppelter Genauigkeit und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SQRTD(857)
	Ausführung einmal bei steigender Flanke	@SQRTD(857)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

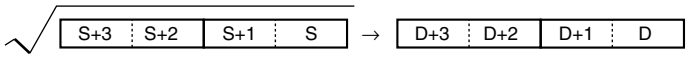
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	

Bereich	S	D
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

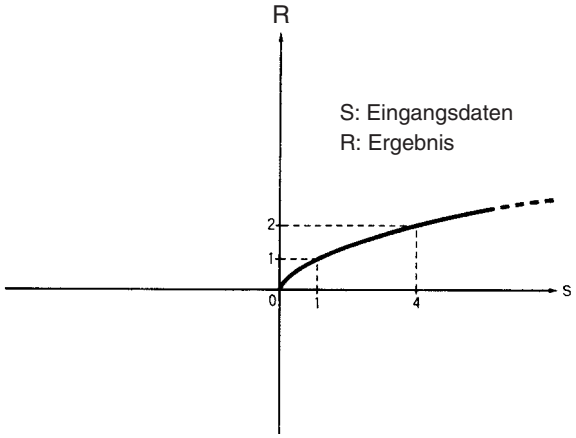
SQRTD(857) berechnet die Quadratwurzel der (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten S bis S+3 und schreibt das Ergebnis in die Worte D bis D+3. (Die Gleitkomma-Quellendaten müssen im IEEE754-Format vorliegen.)



Die Quellendaten müssen positiv sein. Bei negativen Quellendaten tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten negativ sind. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als (64-Bit-) Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	Unverändert
Negativ-Merker	N	Unverändert

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

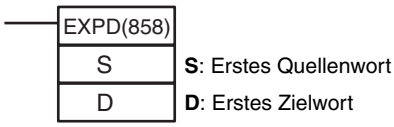
3-16-18 DOUBLE EXPONENT: EXPD(858)

Zweck

Berechnet die natürliche Exponentialgröße (Basis e) einer (64-Bit-) Gleitkommazahl doppelter Genauigkeit und gibt das Ergebnis in die spezifizierten Ergebnisworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	EXPD(858)
	Ausführung einmal bei steigender Flanke	@EXPD(858)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

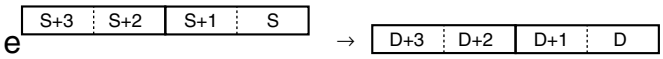
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	

Bereich	S	D
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

EXPD(858) berechnet die natürliche Exponentialgröße (Basis e) der (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten S bis S+3 und schreibt das Ergebnis in die Worte D bis D+3. Mit anderen Worten: EXPD(858) berechnet e^x (x = Quelle) und schreibt das Ergebnis in die Worte D bis D+3.

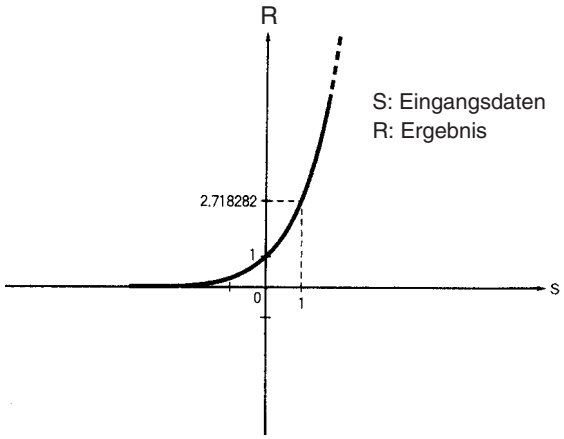


Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird ±∞ ausgegeben.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt und als Ergebnis wird 0 ausgegeben.

Hinweis Die Konstante e ist 2,718282.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als (64-Bit-) Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als (64-Bit-) Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Negativ-Merker	N	Unverändert

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

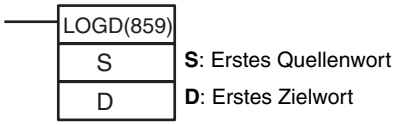
3-16-19 DOUBLE LOGARITHM: LOGD(859)

Zweck

Berechnet den natürlichen Logarithmus (Basis e) einer (64-Bit-) Gleitkommazahl doppelter Genauigkeit und gibt das Ergebnis in die spezifizierten Zielworte aus.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	LOGD(859)
	Ausführung einmal bei steigender Flanke	@LOGD(859)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	A448 bis A956
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	

Bereich	S	D
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

LOGD(859) berechnet den natürlichen Logarithmus (Basis e) der (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten S bis S+3 und schreibt das Ergebnis in die Worte D bis D+3.

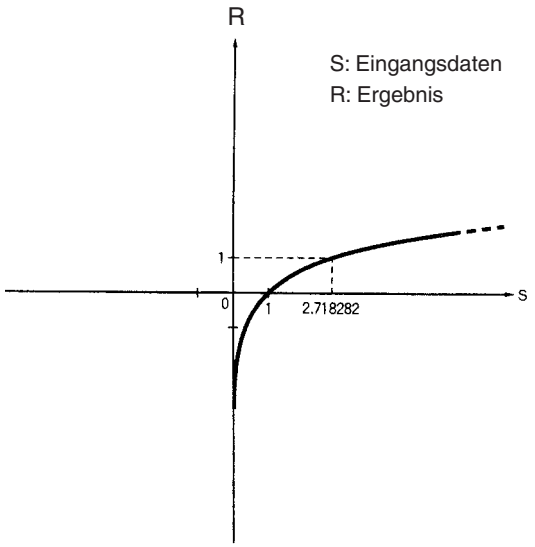
$\log_e$ [S+3 | S+2 | S+1 | S] → [D+3 | D+2 | D+1 | D]

Die Quellendaten müssen positiv sein. Bei negativen Quellendaten tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt und als Ergebnis wird $\pm\infty$ ausgegeben.

Hinweis Die Konstante e ist 2,718282.

Die folgende Abbildung zeigt die Beziehung zwischen Eingangsdaten und Ergebnis.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Quellendaten nicht als Gleitkommadaten erkannt werden. EIN, wenn die Quellendaten negativ sind. EIN, wenn die Quellendaten keine Zahl (NaN) sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als (64-Bit-) Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Unterlauf-Merker	UF	Unverändert
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Quellendaten in den Worten S bis S+3 müssen als Gleitkommadaten im IEEE754-Format vorliegen.

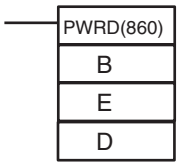
3-16-20 DOUBLE EXPONENTIAL POWER: PWRD(860)

Zweck

Erhebt eine (64-Bit-) Gleitkommazahl doppelter Genauigkeit in die Potenz einer anderen (64-Bit-) Gleitkommazahl doppelter Genauigkeit.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



B: Erstes Basiswort
E: Erstes Exponentenwort
D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	PWRD(860)
	Ausführung einmal bei steigender Flanke	@PWRD(860)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

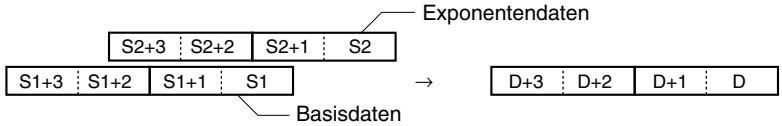
Operanden-Spezifikationen

Bereich	B	E	D
CIO-Bereich	CIO 0000 bis CIO 6140		
Arbeitsbereich	W000 bis W508		
Haftmerkerbereich	H000 bis H508		
Zusatz-Systembereich	A000 bis A956		A448 bis A956
Zeitgeberbereich	T0000 bis T4092		
Zählerbereich	C0000 bis C4092		
DM-Bereich	D00000 bis D32764		
EM-Bereich ohne Bank	E00000 bis E32764		

Bereich	B	E	D
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

PWRD(860) erhebt die (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten B bis B+3 in die Potenz der (64-Bit-) Gleitkommazahl doppelter Genauigkeit in den Worten E bis E+3. Mit anderen Worten: PWRD(860) berechnet X^Y (X = Inhalt von B bis B; Y = Inhalt von E bis E+3).



Wenn beispielsweise die Basisworte (B bis B+3) den Wert 3,1 enthalten und die Exponentenworte (E bis E+3) den Wert 3 enthalten, ist das Ergebnis $3{,}1^3$ bzw. 29,791.

Wenn der Absolutwert des Ergebnisses größer als der Maximalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Überlaufmerker auf EIN gesetzt.

Wenn der Absolutwert des Ergebnisses kleiner als der Minimalwert ist, der sich als Gleitkommadaten ausdrücken lässt, wird der Unterlaufmerker auf EIN gesetzt.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Basisdaten (B bis B+3) oder die Exponentendaten (E bis E+3) nicht als Gleitkommadaten erkannt werden. EIN, wenn die Basisdaten (B bis B+3) oder die Exponentendaten (E bis E+3) keine Zahl (NaN) sind. EIN, wenn die Basisdaten (B bis B+3) oder die Exponentendaten (E bis E+3) kleiner als 0 sind. (Division durch 0) EIN, wenn die Basis (B bis B+3) negativ ist und der Exponent (E bis E+3) keine Ganzzahl ist. (Wurzel einer negativen Zahl) In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn sowohl der Exponent als auch die Mantisse des Ergebnisses 0 sind. In allen anderen Fällen AUS.
Überlauf-Merker	OF	EIN, wenn der Absolutwert des Ergebnisses zu groß ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.

Name	Bezeich-nung	Funktion
Unterlauf-Merker	UF	EIN, wenn der Absolutwert des Ergebnisses zu klein ist, um ihn als Gleitkommawert doppelter Genauigkeit ausdrücken zu können.
Negativ-Merker	N	EIN, wenn das Ergebnis negativ ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Basisdaten (B bis B+3) und die Exponentendaten (E bis E+3) müssen als Gleitkommadaten im IEEE754-Format vorliegen.

3-16-21 Symbolvergleichsbefehle für Gleitkommadaten doppelter Genauigkeit

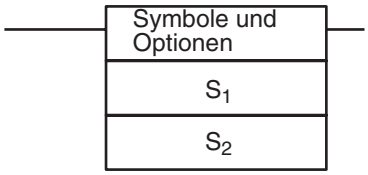
Zweck

Diese Symbolvergleichsbefehle vergleichen zwei Gleitkommawerte doppelter Genauigkeit (64-Bit-Gleitkommadaten im IEEE754-Format) und erzeugen eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist.

Diese Befehle werden nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Hinweis Weitere Informationen zu Binär-Symbolvergleichsbefehlen (mit/ohne Vorzeichen) finden Sie unter *3-7-1 Symbolvergleichsbefehle (300 bis 328)*. Weitere Informationen zu Symbolvergleichsbefehlen für Gleitkommadaten einfacher Genauigkeit finden Sie unter *3-15-21 Vergleichsbefehle für Gleitkommadaten einfacher Genauigkeit*.

Kontaktplansymbol



Variationen

Variationen	Erzeugt in jedem Zyklus ein EIN, in dem die Vergleichsbedingung wahr ist.	Symbol-vergleichsbefehl
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S_1	S_2
CIO-Bereich	CIO 0000 bis CIO 6140	
Arbeitsbereich	W000 bis W508	
Haftmerkerbereich	H000 bis H508	
Zusatz-Systembereich	A000 bis A956	
Zeitgeberbereich	T0000 bis T4092	
Zählerbereich	C0000 bis C4092	
DM-Bereich	D00000 bis D32764	
EM-Bereich ohne Bank	E00000 bis E32764	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	

Bereich	S_1	S_2
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

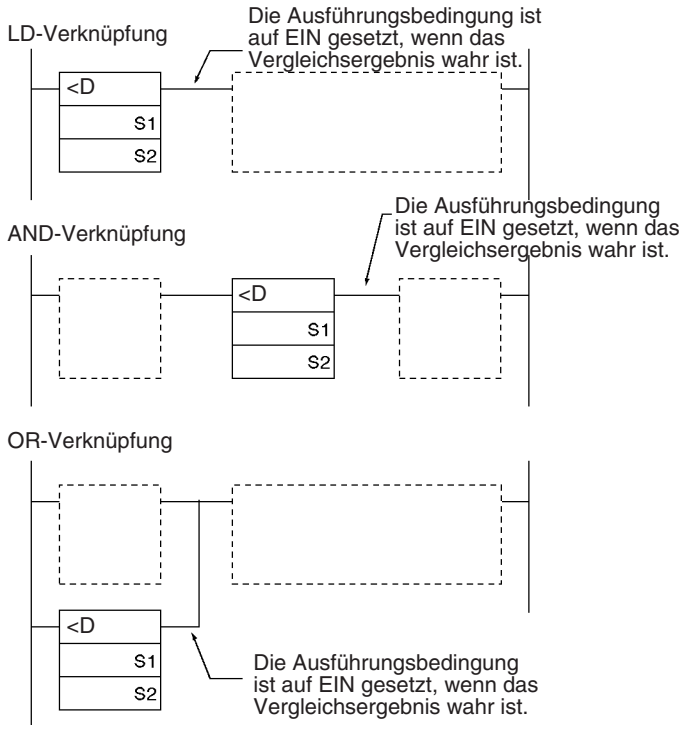
Beschreibung

Der Symbolvergleichsbefehl vergleicht die in S_1 und S_2 spezifizierten Gleitkommawerte doppelter Genauigkeit (64-Bit-Daten im IEEE754-Format) und erzeugt eine EIN-Ausführungsbedingung, wenn die Vergleichsbedingung wahr ist. Wenn die Daten in Worten gespeichert sind, spezifizieren S_1 und S_2 jeweils die ersten der vier Worte, die die 64-Bit-Daten enthalten. Die 64-Bit-Gleitkommadaten können nicht als Konstanten eingegeben werden.

Eingänge für die Befehle

Die Symbolvergleichsbefehle werden genau wie die Befehl LD, AND und OR behandelt, um die Ausführung nachfolgender Befehle zu steuern.

Eingangs-typ	Funktion
LD	Der Befehl kann direkt mit der linken Stromschiene verbunden werden.
AND	Der Befehl kann nicht direkt mit der linken Stromschiene verbunden werden.
OR	Der Befehl kann direkt mit der linken Stromschiene verbunden werden.



Optionen

Es gibt 18 verschiedene Kombinationsmöglichkeiten aus diesen drei Eingangstypen und sechs Symbolen.

Symbol		Option (Datenformat)
=	(Gleich)	D: Gleitkommadaten doppelter Genauigkeit
< >	(Ungleich)	
<	(Kleiner als)	
<=	(Kleiner als oder gleich)	
>	(Größer als)	
>=	(Größer als oder gleich)	

Übersicht über Symbolvergleichsbefehle

In der folgenden Tabelle sind Funktionscodes, AWL-Kürzel. Namen und Funktionen der Symbolvergleichsbefehle für Gleitkommadaten einfacher Genauigkeit aufgeführt. (C1 = S_1+3, S_1+2, S_1+1, S_1 und C2 = S_2+3, S_2+2, S_2+1, S_2.)

Code	AWL	Name	Funktion
335	LD=D	LOAD DOUBLE FLOATING EQUAL	Wahr, wenn C1 = C2
	AND=D	AND DOUBLE FLOATING EQUAL	
	OR=D	OR DOUBLE FLOATING EQUAL	
336	LD <>D	LOAD DOUBLE FLOATING NOT EQUAL	Wahr, wenn C1 ≠ C2
	AND <>D	AND DOUBLE FLOATING NOT EQUAL	
	OR <>D	OR DOUBLE FLOATING NOT EQUAL	
337	LD <D	LOAD DOUBLE FLOATING LESS THAN	Wahr, wenn C1 < C2
	AND <D	AND DOUBLE FLOATING LESS THAN	
	OR <D	OR DOUBLE FLOATING LESS THAN	
338	LD <=D	LOAD DOUBLE FLOATING LESS THAN OR EQUAL	Wahr, wenn C1 ≤ C2
	AND <=D	AND DOUBLE FLOATING LESS THAN OR EQUAL	
	OR <=D	OR DOUBLE FLOATING LESS THAN OR EQUAL	
339	LD >D	LOAD DOUBLE FLOATING GREATER THAN	Wahr, wenn C1 > C2
	AND >D	AND DOUBLE FLOATING GREATER THAN	
	OR >D	OR DOUBLE FLOATING GREATER THAN	
340	LD >=D	LOAD DOUBLE FLOATING GREATER THAN OR EQUAL	Wahr, wenn C1 ≥ C2
	AND >=D	AND DOUBLE FLOATING GREATER THAN OR EQUAL	
	OR >=D	OR DOUBLE FLOATING GREATER THAN OR EQUAL	

Merker

In dieser Tabelle steht C1 für den Inhalt von S1 bis S1+3 und C2 für den Inhalt von S2 bis S2+3.

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn C1 oder C2 keine gültige Gleitkommazahl (NaN) ist. EIN, wenn C1 oder C2 den Wert +∞ hat. EIN, wenn C1 oder C2 den Wert −∞ hat. In allen anderen Fällen AUS.
Größer-als-Merker	>	EIN, wenn C1 > C2. In allen anderen Fällen AUS.
Größer-oder-gleich-Merker	> =	EIN, wenn C1 ≥ C2. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn C1 = C2. In allen anderen Fällen AUS.
Ungleich-Merker	=	EIN, wenn C1 ≠ C2. In allen anderen Fällen AUS.
Kleiner-als-Merker	<	EIN, wenn C1 < C2. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Merker	< =	Ein, wenn C1 ≤ C2. In allen anderen Fällen AUS.
Negativ-Merker	N	Unverändert

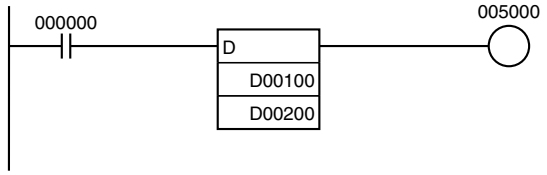
Vorsichtsmaßnahmen

Symbolvergleichsbefehle können nicht mit Verbindung zur rechten Stromschiene verwendet werden, d. h., zwischen ihnen und der rechten Stromschiene muss ein weiterer Befehl eingesetzt werden.

Beispiel

AND DOUBLE FLOATING LESS THAN: AND<D(331)

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, werden die Gleitkommadaten in den Worten D00100 bis D00103 mit den Gleitkommadaten in den Worten D00200 bis D00203 verglichen. Wenn der Inhalt von D00100 bis D00103 kleiner ist als der von D00200 bis D00203, wird die Programmausführung mit der nächsten Zeile fortgesetzt und CIO 005000 wird auf EIN gesetzt. Wenn der Inhalt von D00100 bis D00103 nicht kleiner ist als der von D00200 bis D00203, wird CIO 005000 auf AUS gesetzt.



DOUBLE FLOATING LESS THAN-Vergleich (<D)

	15															0
S1: D00100	1	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0
S1+1: D00101	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	0
S1+2: D00102	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1
S1+3: D00103	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Dezimalwert: 3,4580

	15															0
S1: D00100	0	1	1	1	1	0	0	1	0	0	1	1	1	1	1	0
S2+1: D00101	1	0	1	0	1	0	0	0	0	1	0	1	1	0	0	0
S2+2: D00102	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	1
S2+3: D00103	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1

Dezimalwert: -1,4876

34580>14876

Erzeugt keine EIN-Bedingung.

	15															0
S1: D00100	1	1	0	1	1	1	1	0	1	0	0	1	0	0	0	1
S1+1: D00101	1	0	1	0	1	0	0	1	1	0	1	1	0	1	1	0
S1+2: D00102	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0
S1+3: D00103	1	1	0	0	1	0	1	0	0	0	0	0	0	0	1	0

Dezimalwert: –3,4580E+48

	15															0
S1 :D00100	0	1	0	1	0	1	0	0	0	1	0	1	0	0	1	1
S2+1: D00101	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	1
S2+2: D00102	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
S2+3: D00103	0	1	0	0	1	0	0	1	1	1	1	1	0	0	0	0

Dezimalwert: 1,4876E+48

–3.4580E+48<1.4876E+48

Erzeugt eine EIN-Bedingung.

3-17 Tabellendatenverarbeitungsbefehle

In diesem Abschnitt werden Befehle beschrieben, die zur Handhabung von Tabellendaten, Stapeln und anderen Datenbereichen verwendet werden. Die untersten 5 Befehle in der Tabelle (mit einem Sternchen gekennzeichnet) werden nur von den CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Befehl	AWL	Funktionscode	Seite
SET STACK	SSET	630	666
PUSH ONTO STACK	PUSH	632	669
FIRST IN FIRST OUT	FIFO	633	672
LAST IN FIRST OUT	LIFO	634	675
DIMENSION RECORD TABLE	DIM	631	678
SET RECORD LOCATION	SETR	635	681
GET RECORD NUMBER	GETR	636	683
DATA SEARCH	SRCH	181	685
SWAP BYTES	SWAP	637	687

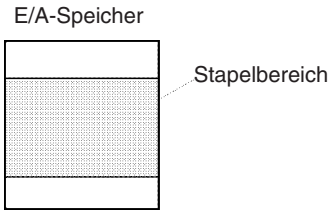
Befehl	AWL	Funktionscode	Seite
FIND MAXIMUM	MAX	182	689
FIND MINIMUM	MIN	183	693
SUM	SUM	184	697
FRAME CHECKSUM	FCS	180	700
STACK NUMBER OUTPUT	SNUM	638	704
STACK DATA READ	SREAD	639	707
STACK DATA OVERWRITE	SWRIT	640	710
STACK DATA INSERT	SINS	641	713
STACK DATA DELETE	SDEL	642	716

Alle diese Befehle definieren oder verarbeiten eine Gruppe von Worten. Die Gruppe von Worten in einem Stapel wird durch SSET(630) definiert und die Gruppe von Worten in einer Datensatztabelle werden durch DIM(631) definiert. Die Gruppe von Worten, die in einem Bereichsbefehl verwendet werden, wird unabhängig in jedem Befehl definiert.

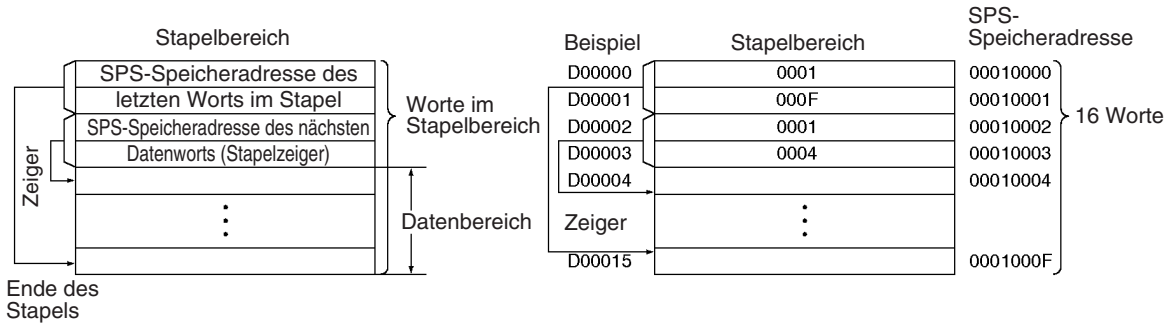
Gruppe	Zweck	Befehle
Stapel	Verarbeitung von FIFO- (zuerst geschriebene Daten werden auch zuerst ausgelesen) oder LIFO-Datentabellen (zuletzt geschriebene Daten werden zuerst ausgelesen)	SSET(630), PUSH(632), FIFO(633), LIFO(634), SREAD(639), SWRIT(640), SINS(641), SDEL(642) und SNUM(638)
Datensatz-tabelle	Verarbeitung von aus Datensätzen bestehenden Tabellen (Die Datensatzgröße ist anwenderdefiniert.)	DIM(631), SETR(635) und GETR(636)
Bereich	Verarbeitung eines Wortbereichs zum Suchen nach Werten wie der Prüfsumme, einem bestimmten Wert oder dem größten oder kleinsten Wert innerhalb des Bereichs.	FCS(180), SRCH(181), MAX(182), MIN(183), SUM(184) und SWAP(637)

Stapelbefehle

Stapelbefehle verarbeiten speziell definierte Datentabellen, die als Stapel bezeichnet werden. Die ersten zwei Worte im Stapel enthalten die SPS-Speicheradresse des letzten Worts im Stapel, die folgenden zwei Worte enthalten den Stapelzeiger (die SPS-Speicheradresse des Worts, das durch den nächsten PUSH(632)-Befehl überschrieben wird).



In der folgenden Abbildung ist die Grundstruktur eines Stapels dargestellt.

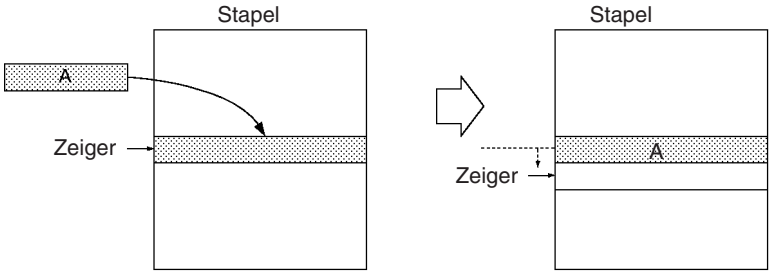


Die folgenden Befehle definieren Stapelbereiche oder arbeiten mit ihnen. Grundsätzlich speichert PUSH(632) Daten in dem nächsten freien Datenwort im Stapel. FIFO(633) und LIFO(634) lesen Daten aus dem Stapel. FIFO(633) liest das zuerst gespeicherte Wort, während LIFO(634) das zuletzt gespeicherte Wort liest.

Die letzten fünf Befehle werden nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt. SNUM(638) zählt die Anzahl der Datenelemente (Worte) indem spezifizierten Stapel. Dieser Befehl kann beispielsweise verwendet werden, um die Anzahl von Objekten in einer Förderanlage zu bestimmen. Die Befehle SREAD(639), SWRIT(640), SINS(641) und SDEL(642) werden zum Lesen, Überschreiben, Einfügen bzw. Löschen von Datenelementen in einem Stapel verwendet. Beispielsweise kann beim Transport von Objekten mit einer Förderanlage mit diesen Befehlen ein Datenelement in einem Stapel, das einem Objekt in der Förderanlage entspricht, hinzugefügt, entfernt oder geändert werden.

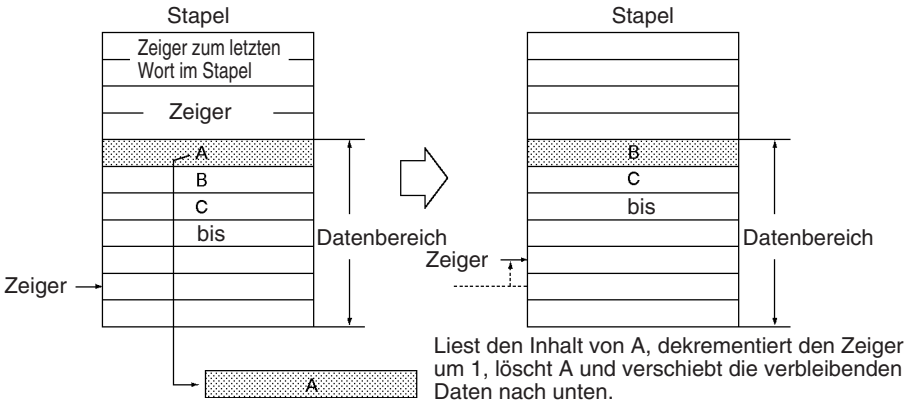
PUSH(632)

Speichert Daten in der durch den Stapelzeiger angezeigten Adresse und inkrementiert den Stapelzeiger um Eins.



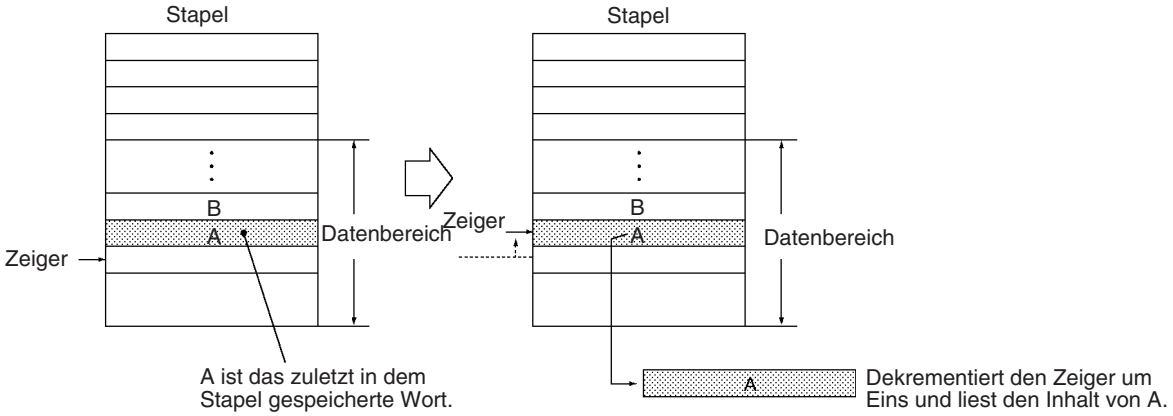
FIFO(633)

Liest das zuerst gespeicherte (älteste) Datenwort in dem Stapel, verschiebt die verbleibenden Daten um ein Wort nach unten und dekrementiert den Stapelzeiger um Eins.



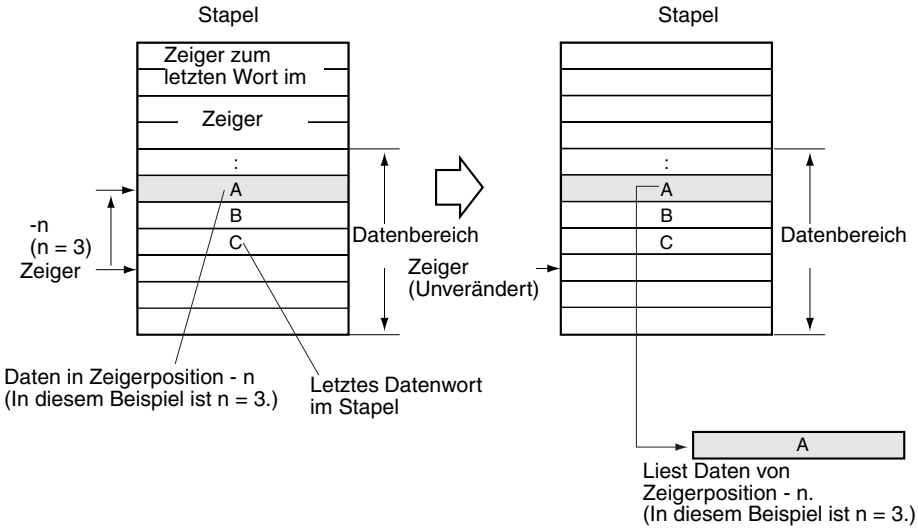
LIFO(634)

Liest das zuletzt (das neueste) in dem Stapel gespeicherte Datenwort. Dekrementiert den Zeiger um Eins und liest die an dieser Adresse (die zuletzt in dem Stapel gespeicherten Daten). Die gelesenen Daten werden nicht gelöscht.



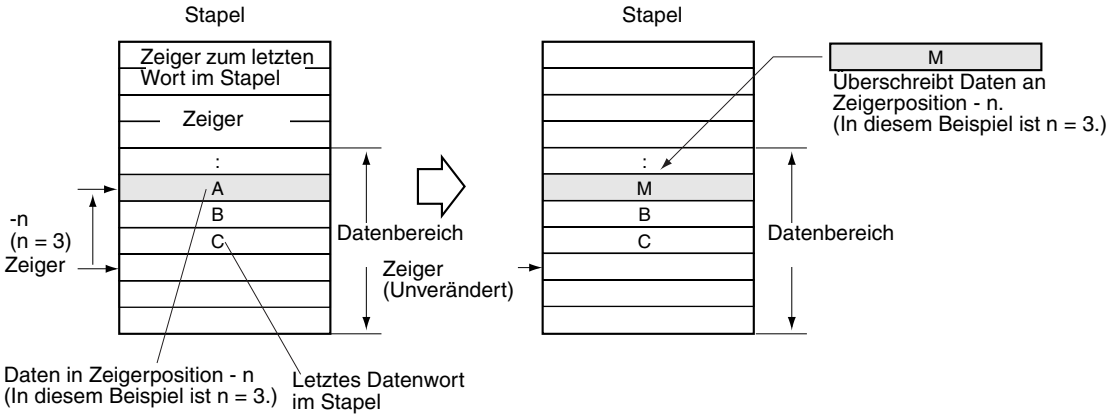
SREAD(639)

Liest die Daten ab dem spezifizierten Datenelement in dem Stapel. Der Offset-Wert bestimmt die Stelle des gewünschten Worts (die Anzahl der Worte vor der aktuellen Zeigerposition).



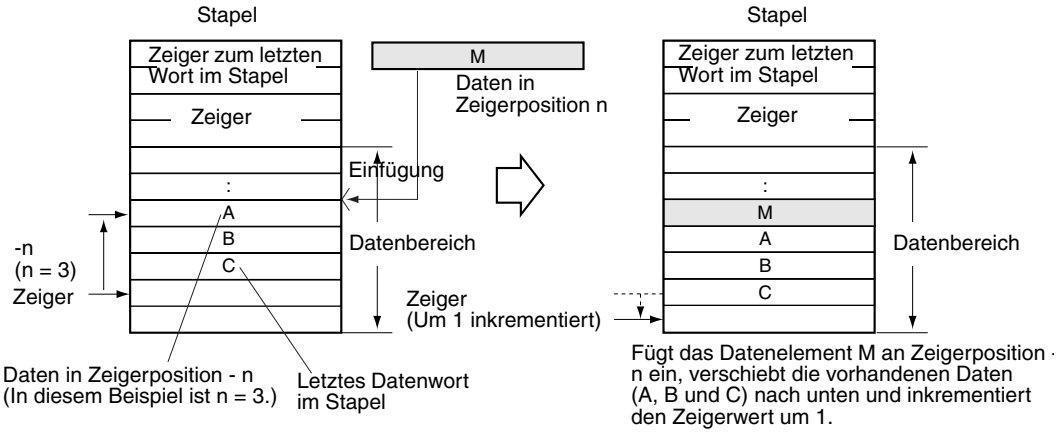
SWRIT(640)

Schreibt die Quellendaten in das spezifizierte Datenelement im Stapel (überschreibt vorhandene Daten). Der Offset-Wert bestimmt die Stelle des gewünschten Worts (die Anzahl der Worte vor der aktuellen Zeigerposition).



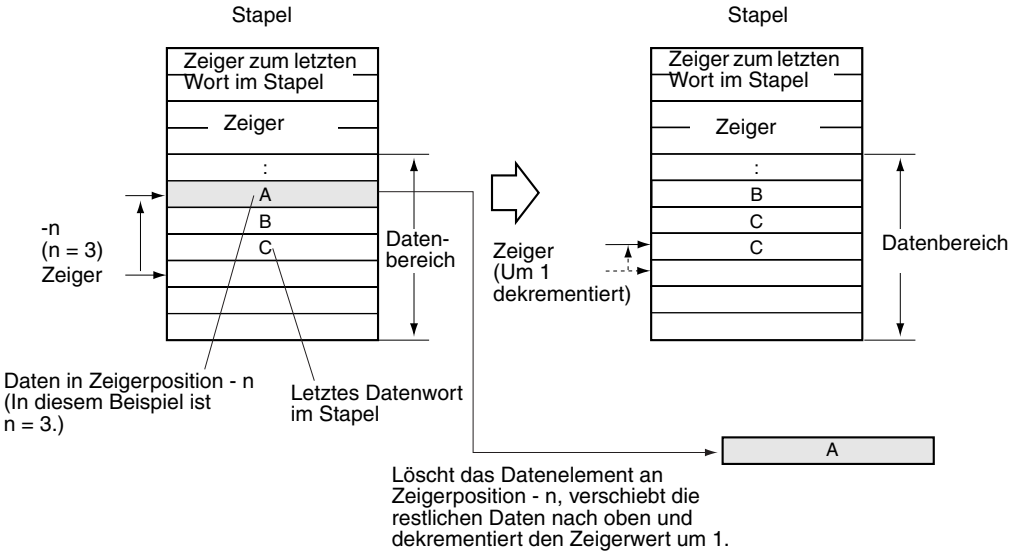
SINS(641)

Fügt die Quellendaten an der spezifizierten Stelle in den Stapel ein und verschiebt den Rest der Daten im Stapel nach unten. Der Offset-Wert bestimmt die Stelle des gewünschten Worts (die Anzahl der Worte vor der aktuellen Zeigerposition).



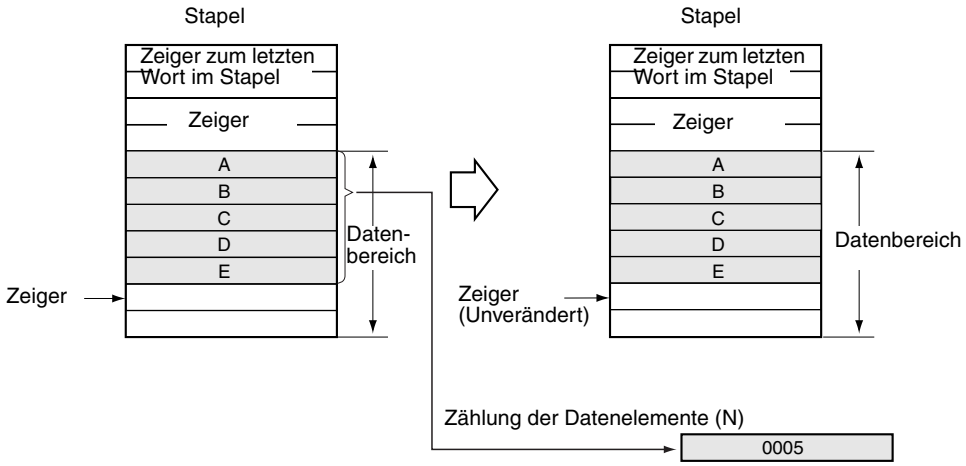
SDEL(642)

Löscht das Datenelement an der spezifizierten Stelle im Stapel und verschiebt den Rest der Daten im Stapel nach oben. Der Offset-Wert bestimmt die Stelle des gewünschten Worts (die Anzahl der Worte vor der aktuellen Zeigerposition).



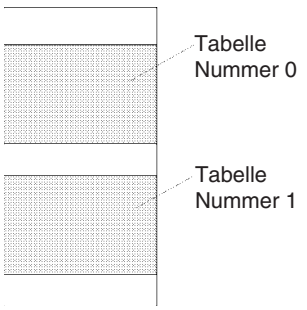
SNUM(638)

Zählt die Anzahl der Daten im Stapel (die Anzahl der Datenworte) vom Stapelzeiger bis zum Anfang des Datenbereichs.

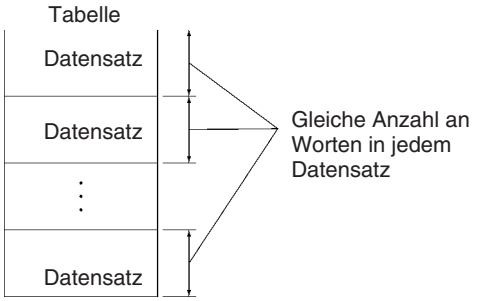


Datensatztabellen-Befehle

Eine Reihe von Daten, die aus mehr als einem Datensatz mit der gleichen Anzahl von Worten in jedem Datensatz bestehen, wird als Tabellendaten bezeichnet. Tabellendaten, die im spezifizierten E/A-Speicher gespeichert sind, können durch Verwendung des Befehls DIM als Tabellenbereich registriert werden. Es können bis zu 16 separate Tabellen mit den Tabellennummern 0 bis 15 definiert werden.



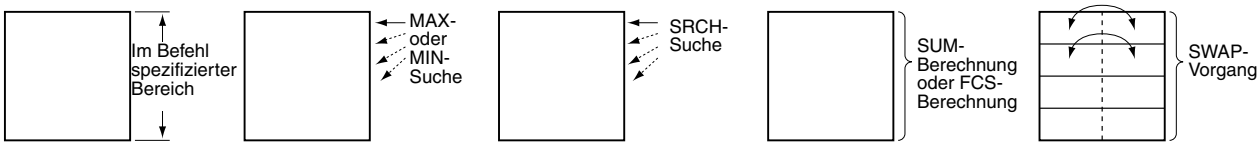
In der folgenden Abbildung ist die Grundstruktur einer Datensatztabelle dargestellt. Jeder Datensatz in der Tabelle besteht aus der gleichen Anzahl von Worten.



Zur indirekten Referenzierung von Tabellendaten können Indexregister (IR) verwendet werden. Die Berechnung der Adresse des Datensatzes kann auf einfache Weise mit den Befehlen SETR(635) (SET RECORD NUMBER) und GETR(636) (GET RECORD NUMBER) erfolgen.

Bereichsbefehle

Die hier einbezogenen Bereichsbefehle verarbeiten einen spezifizierten Bereich von Worten, um den Maximalwert (MAX(182)), den Minimalwert (MIN(183)) oder einen bestimmten Wert (SRCH(181)) zu suchen, die Summe (SUM(184)) oder die Frame-Prüfsumme (FCS(180)) zu berechnen oder den Inhalt der linken Bytes und den der rechten Bytes der Worte auszutauschen (SWAP(637)).

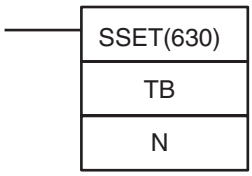


3-17-1 SET STACK: SSET(630)

Zweck

Definiert einen mit dem spezifizierten Wort beginnenden Stapel der spezifizierten Länge.

Kontaktplansymbol



TB: Erste Stapeladresse

N: Anzahl der Worte

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SSET(630)
	Ausführung einmal bei steigender Flanke	@SSET(630)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

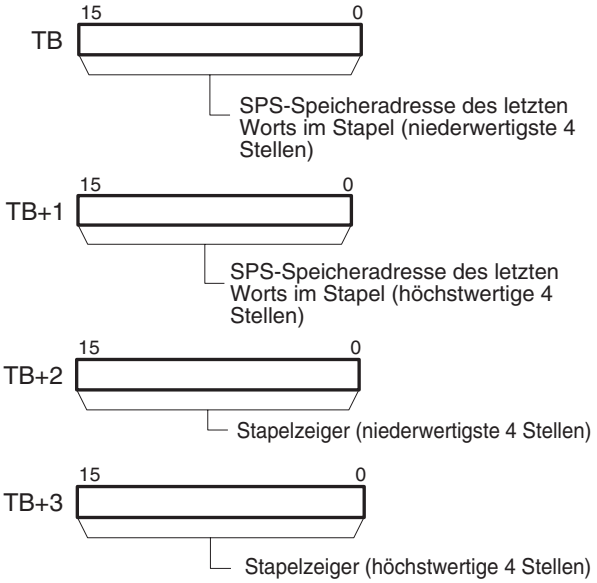
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

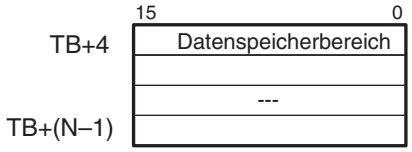
Operanden

TB bis TB+3: Stapel-Steuerworte

Die ersten vier Worte des Stapels enthalten die SPS-Speicheradresse des letzten Worts im Stapel und den Stapelzeiger (die SPS-Speicheradresse des Worts, das als nächstes durch den Befehl PUSH(632) überschrieben wird).



TB+4 bis TB+(N–1): Datenspeicherbereich
Der Rest des Stapels wird zum Speichern von Daten verwendet.



Hinweis

1. Der Anfangswert des Stapelzeigers ist stets die SPS-Speicheradresse von TB+4.
2. TB and TB+(N–1) müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

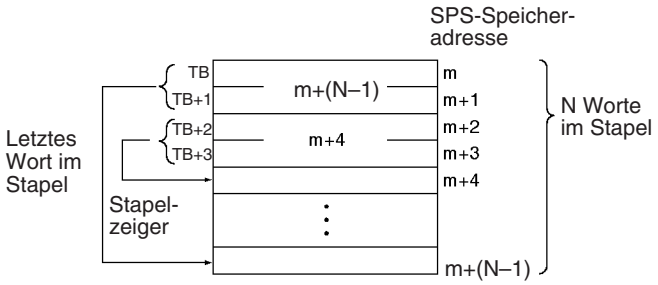
Bereich	TB	N
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W511	
Haftmerkerbereich	H000 bis H511	
Zusatz-Systembereich	A448 bis A959	A000 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	#0005 bis #FFFF (binär) oder &5 bis &65.535
Datenregister	---	DR0 bis DR15

Bereich	TB	N
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

SSET(630) speichert einen Stapel mit N Worten bei TB beginnend und bei TB+(N–1) endend. Die ersten zwei Worte des Stapels (TB+1 und TB) enthalten die SPS-Speicheradresse des letzten Worts im Stapel im 8-stelligen Hexadezimalformat. Die folgenden zwei Worte (TB+3 und TB+2) enthalten des Stapelzeiger. Der Stapelzeiger ist die SPS-Speicheradresse des nächsten Worts, das vom Befehl PUSH(632) überschrieben wird. Sein Anfangswert ist die Adresse von TB+4.

Der Befehl SSET(630) initialisiert den Datenbereich des Stapels (TB+4 bis TB+(N–1)) automatisch auf Nullen. In der folgenden Abbildung ist die Grundstruktur eines Stapels dargestellt.



Der Befehl SSET(630) dient einfach zur Einrichtung und Initialisierung eines Stapels. Verwenden Sie die folgenden Befehle, um Daten in dem Stapel zu speichern und Daten aus dem Stapel zu lesen.

1,2,3...

1. Der Befehl PUSH(632) speichert jeweils Daten in einem Wort des Stapels.
2. FIFO(633) und LIFO(634) lesen Daten aus dem Stapel. FIFO(633) liest das zuerst gespeicherte Wort, LIFO(634) das zuletzt gespeicherte Wort.
3. Der Stapelzeigerwert im Stapelsteuerwort wird bei Ausführung von PUSH(632), FIFO(633) oder LIFO(634) automatisch aktualisiert. Normalerweise muss sich der Anwender nicht um das Stapelsteuerwort kümmern. Wenn auf den Inhalt des Stapels für andere Vorgänge als die Ausführung der oben aufgeführten Befehle zugegriffen werden soll, können Sie den Stapelzeigerwert mittels des Indexregisters (IR) zur indirekten Referenzierung setzen.

Merker

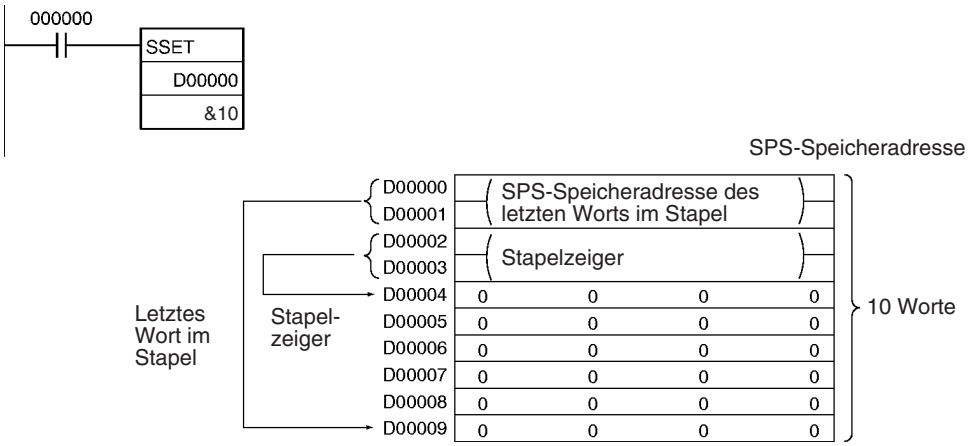
Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn N nicht innerhalb des spezifizierten Bereichs von 0005 bis FFFF liegt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Mindestwert für die Anzahl der Worte im Stapel (N) beträgt 5, da N auch die vier Worte umfasst, die den Zeiger zum letzten Wort im Stapel und den Stapelzeiger enthalten. Wenn N nicht innerhalb des Bereichs von 0005 bis FFFF liegt, tritt ein Fehler auf.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, erstellt SSET(630) einen 10-Worte-Stapel von D00000 bis D00009. D00000 und D00001 enthalten die SPS-Speicheradresse des letzten Worts im Stapel. D00002 und D00003 enthalten den Stapelzeiger. Der Stapel selbst beginnt mit D00004.

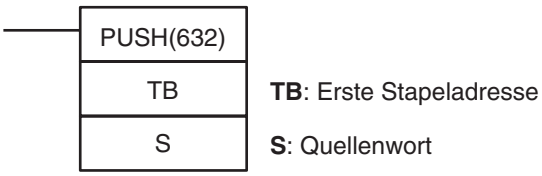


3-17-2 PUSH ONTO STACK: PUSH(632)

Zweck

Schreibt ein Datenwort in das spezifizierte Wort im Stapel.

Kontaktplansymbol



TB: Erste Stapeladresse

S: Quellenwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	PUSH(632)
	Ausführung einmal bei steigender Flanke	@PUSH(632)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

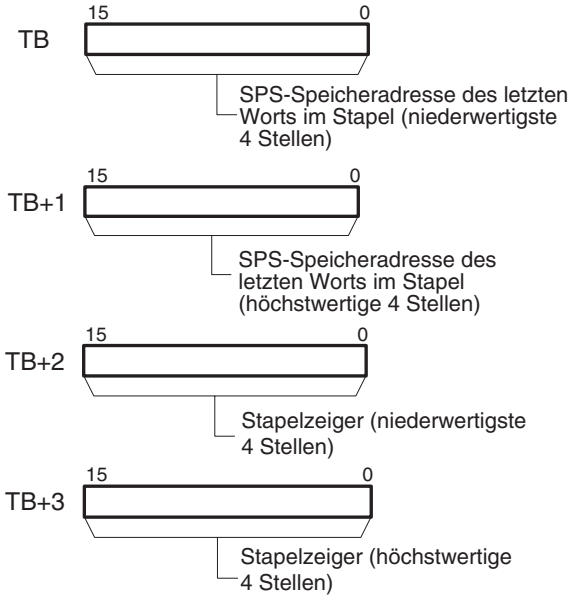
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

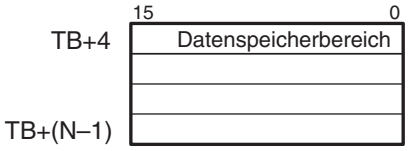
Operanden

TB bis TB+3: Stapel-Steuerworte

Die ersten vier Worte des Stapels enthalten die SPS-Speicheradresse des letzten Worts im Stapel und den Stapelzeiger (die SPS-Speicheradresse des Worts, das als nächstes durch den Befehl PUSH(632) überschrieben wird).



TB+4 bis TB+(N–1): Datenspeicherbereich
Der Rest des Stapels wird zum Speichern von Daten verwendet.



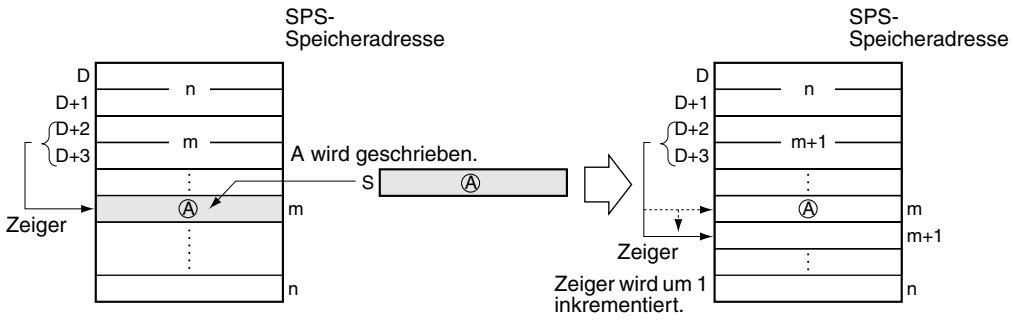
Operanden-Spezifikationen

Bereich	TB	S
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W511	
Haftmerkerbereich	H000 bis H511	
Zusatz-Systembereich	A448 bis A959	A000 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	#0000 bis #FFFF (binär)
Datenregister	---	DR0 bis DR15

Bereich	TB	S
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

PUSH(632) schreibt den Inhalt von S in die durch den Stapelzeiger (TB+3 und TB+2) angegebene Adresse und inkrementiert den Stapelzeiger um Eins.



Nachdem PUSH(632) verwendet wurde, um Daten in den Stapel zu schreiben, können FIFO(633) und LIFO(634) verwendet werden, um die Daten aus dem Stapel zu lesen.

Merker

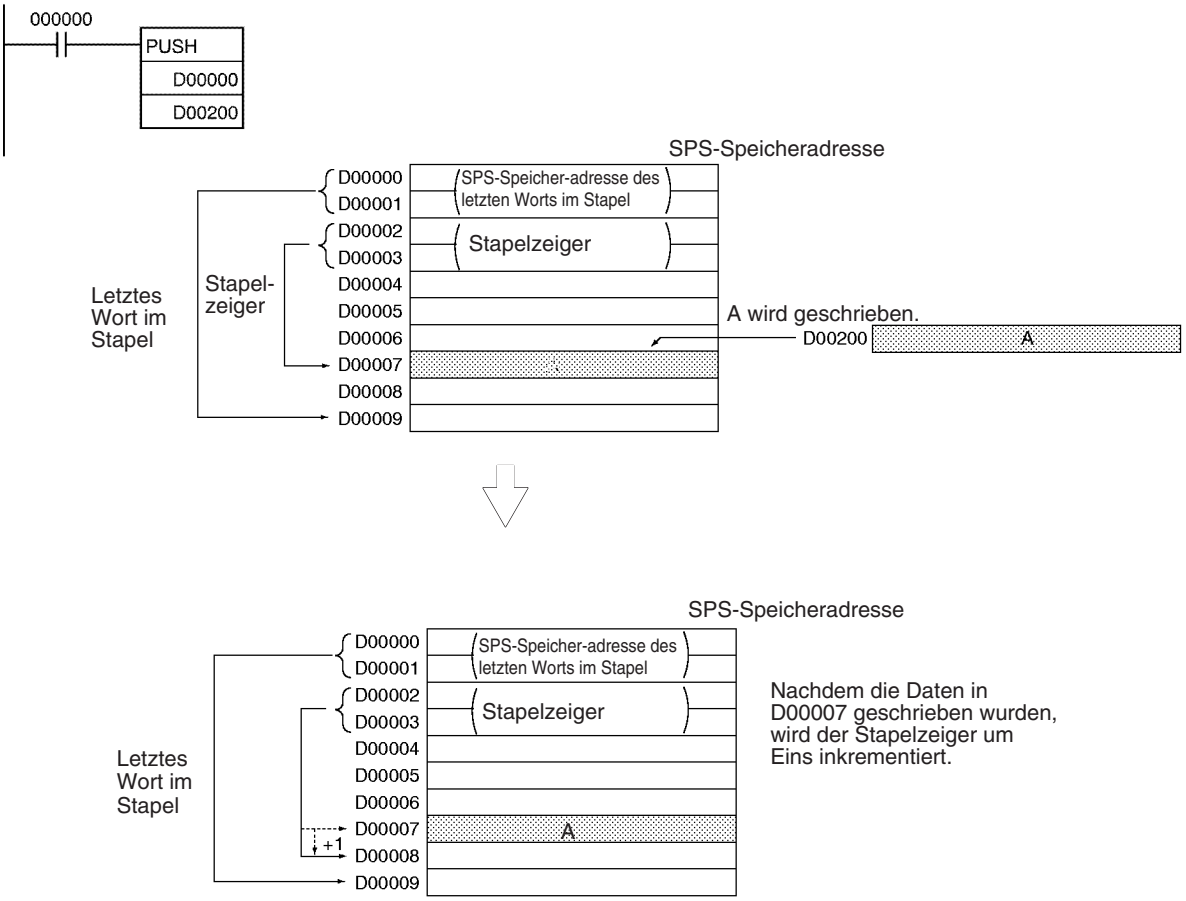
Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die durch den Stapelzeiger (TB+3 und TB+2) spezifizierte Adresse über das letzte Wort im Stapel hinaus geht. (Es handelt sich um einen Stapel-Überlauffehler.) In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Stapel muss zuvor durch den Befehl SSET(630) definiert werden.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, kopiert PUSH(632) den Inhalt von D00200 in den mit D00000 beginnenden Stapel. In diesem Fall wird durch den Stapelzeiger das Wort D00007 bestimmt.

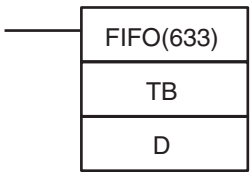


3-17-3 FIRST IN FIRST OUT: FIFO(633)

Zweck

Liest das zuerst in den spezifizierten Stapel geschriebene Datenwort (die ältesten Daten im Stapel)

Kontaktplansymbol



TB: Erste Stapeladresse

D: Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FIFO(633)
	Ausführung einmal bei steigender Flanke	@FIFO(633)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

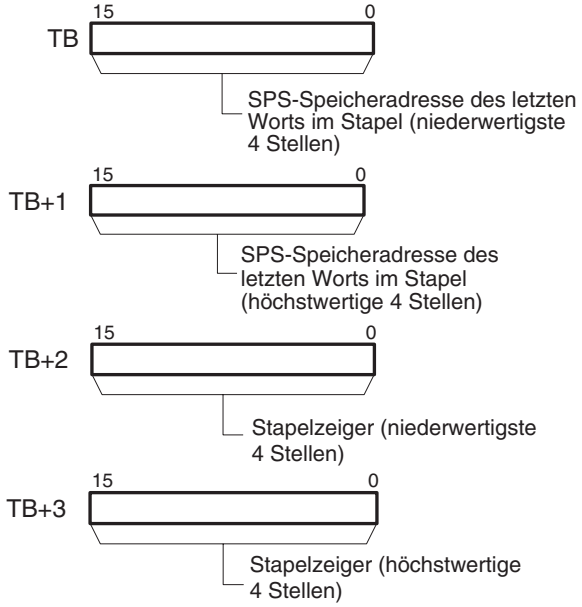
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

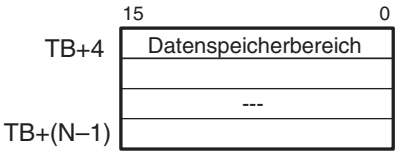
TB bis TB+3: Stapel-Steuerworte

Die ersten vier Worte des Stapels enthalten die SPS-Speicheradresse des letzten Worts im Stapel und den Stapelzeiger (die SPS-Speicheradresse des Worts, das als nächstes durch den Befehl PUSH(632) überschrieben wird).



TB+4 bis TB+(N–1): Datenspeicherbereich

Der Rest des Stapels wird zum Speichern von Daten verwendet.



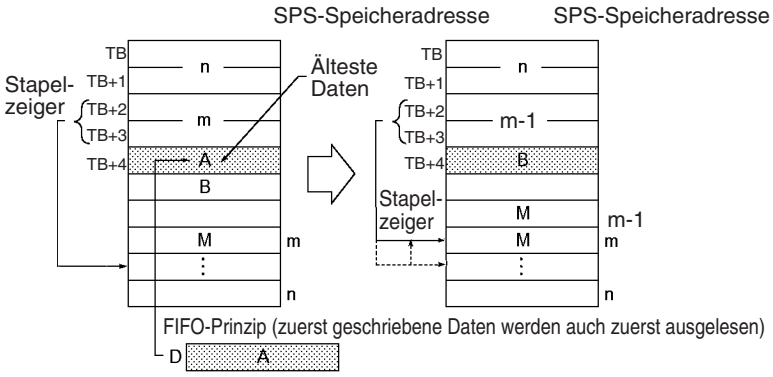
Operanden-Spezifikationen

Bereich	TB	D
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W511	
Haftmerkerbereich	H000 bis H511	
Zusatz-Systembereich	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	

Bereich	TB	D
Datenregister	---	DR0 bis DR15
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

FIFO(633) liest das älteste Datenwort im Stapel (TB+4) und gibt diese Daten nach D aus. Anschließend wird der Stapelzeiger (TB+3 und TB+2) um Eins dekrementiert, alle verbleibenden Daten im Stapel werden um ein Wort nach unten verschoben und die aus TB+4 gelesenen Daten werden gelöscht. Die Daten am Ende des Stapels (die Adresse, die durch den Stapelzeiger angegeben wurde) bleiben unverändert.



Verwenden Sie FIFO(633) in Kombination mit PUSH(632). Nachdem PUSH(632) verwendet wurde, um Daten in den Stapel zu schreiben, kann FIFO(633) verwendet werden, um die Daten aus dem Stapel nach dem First-In-First-Out-Prinzip zu lesen.

FIFO(633) liest die Anfangsdaten aus dem Stapel und löscht sie, um die nächsten Daten nach vorn zu verschieben.

Merker

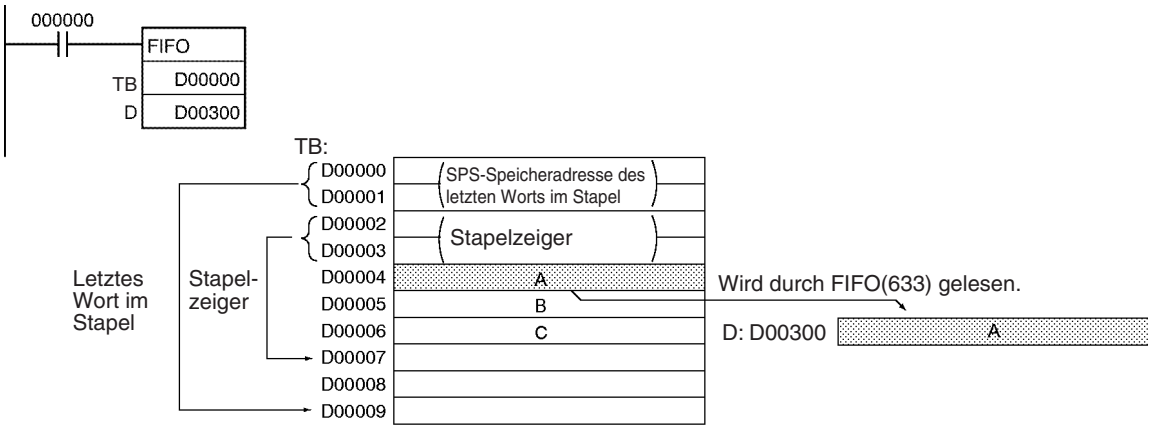
Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt des Stapelzeigers (TB+3 und TB+2) kleiner oder gleich der SPS-Speicheradresse des ersten Worts im Datenbereich des Stapels (TB+4) ist. (Es handelt sich um einen Stapel-Unterlauffehler.) In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

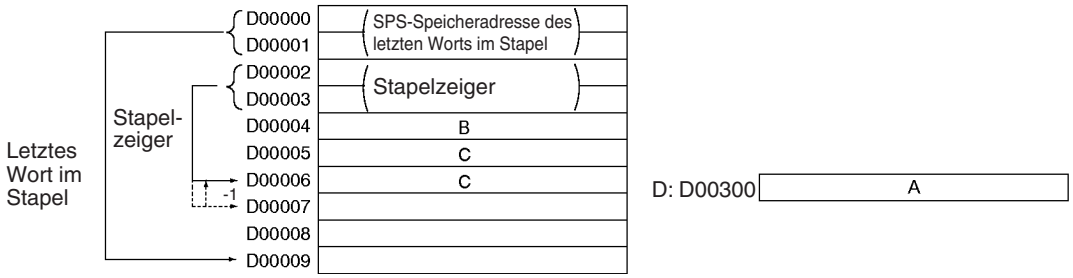
Der Stapel muss zuvor durch den Befehl SSET(630) definiert werden.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, liest FIFO(633) den Inhalt von D00004 (TB+4 bei einem mit D00000 beginnenden Stapel) und schreibt diese Daten in D00300.



Nachdem die Daten in D00300 geschrieben wurden, wird der Stapelzeiger um Eins dekrementiert und die verbleibenden Daten werden nach unten verschoben. (Der Inhalt von D00005 wird nach D00004 verschoben und der Inhalt von D00006 wird nach D00005 verschoben.)

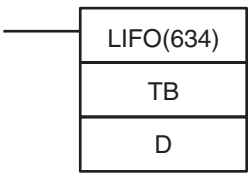


3-17-4 LAST IN FIRST OUT: LIFO(634)

Zweck

Liest das zuletzt in den spezifizierten Stapel geschriebene Datenwort (die neuesten Daten im Stapel)

Kontaktplansymbol



TB: Erste Stapeladresse

D: Zielwort

Variationen

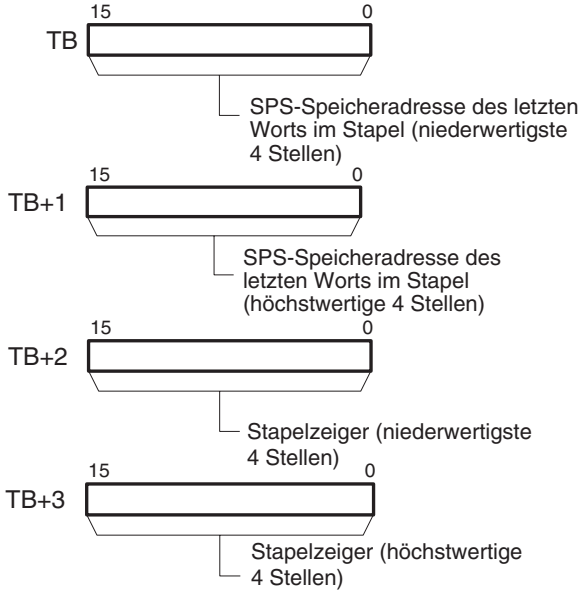
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	LIFO(634)
	Ausführung einmal bei steigender Flanke	@LIFO(634)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

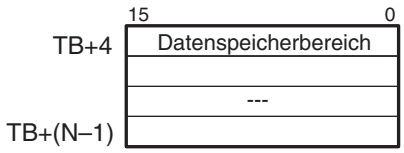
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

TB bis TB+3: Stapel-Steuerworte
Die ersten vier Worte des Stapels enthalten die SPS-Speicheradresse des letzten Worts im Stapel und den Stapelzeiger (die SPS-Speicheradresse des Worts, das als nächstes durch den Befehl PUSH(632) überschrieben wird).



TB+4 bis TB+(N–1): Datenspeicherbereich
Der Rest des Stapels wird zum Speichern von Daten verwendet.



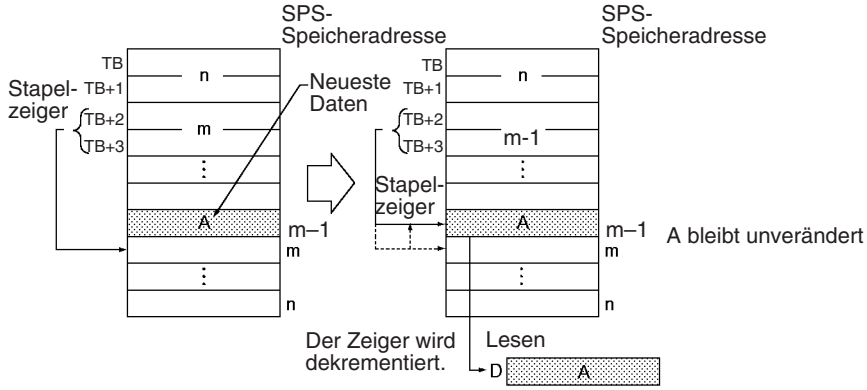
Operanden-Spezifikationen

Bereich	TB	D
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W511	
Haftmerkerbereich	H000 bis H511	
Zusatz-Systembereich	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zähler bereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	DR0 bis DR15

Bereich	TB	D
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

LIFO(634) liest die Daten aus der durch den Stapelzeiger angegebenen Adresse (das neueste Datenwort im Stapel), dekrementiert den Stapelzeiger um Eins und gibt die Daten nach D aus. Das gelesene Wort bleibt unverändert erhalten.



Verwenden Sie LIFO(634) in Kombination mit PUSH(632). Nachdem PUSH(632) verwendet wurde, um Daten in den Stapel zu schreiben, kann LIFO(634) verwendet werden, um die Daten aus dem Stapel nach dem Last-In-First-Out-Prinzip zu lesen. Nachdem Daten durch PUSH(632) gespeichert wurden, weist der Stapelzeiger auf die Adresse neben den letzten Daten.

Merker

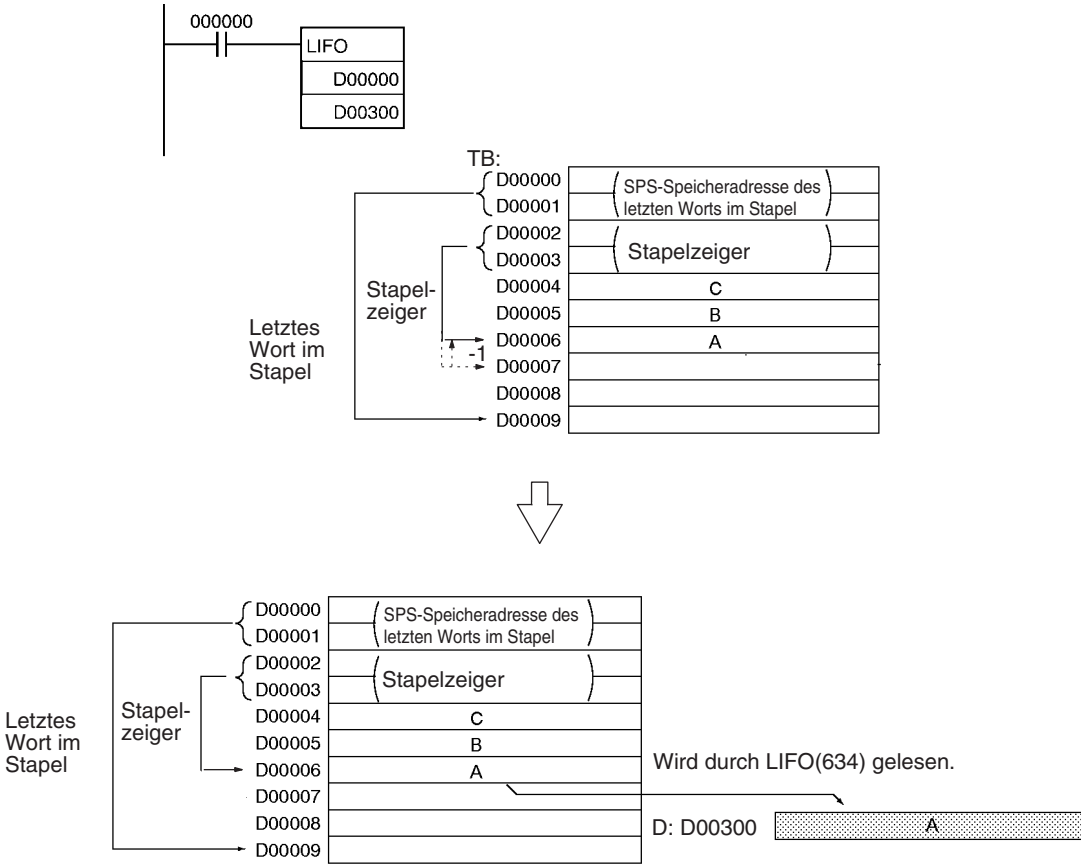
Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt des Stapelzeigers (TB+3 und TB+2) kleiner oder gleich der SPS-Speicheradresse des ersten Worts im Datenbereich des Stapels (TB+4) ist. (Es handelt sich um einen Stapel-Unterlauffehler.) In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Stapel muss zuvor durch den Befehl SSET(630) definiert werden.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, liest LIFO(634) den Inhalt des durch den Stapelzeiger angegebenen Worts (D00006) und schreibt diese Daten in D00300.



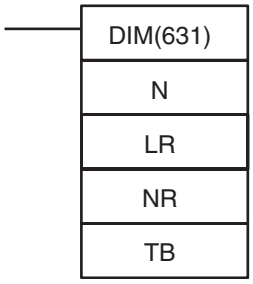
Nachdem die Daten in D00300 geschrieben wurden, wird der Stapelzeiger um Eins dekrementiert. Der Inhalt von D00006 bleibt unverändert erhalten.

3-17-5 DIMENSION RECORD TABLE: DIM(631)

Zweck

Definiert des spezifizierten E/A-Speicherbereich durch Deklaration der Länge der einzelnen Datensätze und der Anzahl der Datensätze als eine Datensatztabelle. Es können bis zu 16 Datensatztabellen definiert werden.

Kontaktplansymbol



N: Tabellennummer

LR: Länge jedes Datensatzes

NR: Anzahl der Datensätze

TB: Erstes Tabellenwort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	DIM(631)
	Ausführung einmal bei steigender Flanke	@DIM(631)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Tabellennummer
Spezifiziert die Tabellennummer. N muss zwischen 0 und 15 liegen.

LR: Länge jedes Datensatzes
Spezifiziert die Anzahl von Worten in jedem Datensatz. LR muss einen Wert zwischen 0001 und FFFF hexadezimal (1 bis 65.535 Worte) besitzen.

NR: Anzahl der Datensätze
Spezifiziert die Anzahl der Datensätze in der Tabelle. NR muss einen Wert zwischen 0001 und FFFF hexadezimal (1 bis 65.535 Worte) besitzen.

TB: Erstes Tabellenwort
Spezifiziert das erste Wort der Tabelle. Alle Worte der Tabelle müssen im gleichen Datenbereich liegen. Mit anderen Worten: TB und TB+LR×NR–1 müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	N	LR	NR	TB
CIO-Bereich	---	CIO 0000 bis CIO 6143		
Arbeitsbereich	---	W000 bis W511		
Haftmerkerbereich	---	H000 bis H511		
Zusatz-Systembereich	---	A000 bis A959		A448 bis A959
Zeitgeberbereich	---	T0000 bis T4095		
Zählerbereich	---	C0000 bis C4095		
DM-Bereich	---	D00000 bis D32767		
EM-Bereich ohne Bank	---	E00000 bis E32767		
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	0 bis 15	#0001 bis #FFFF (binär) oder &1 bis &65.535		---
Datenregister	---	DR0 bis DR15		---
Indexregister	---	---		
Indirektes Adressieren mittels Index-Registern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

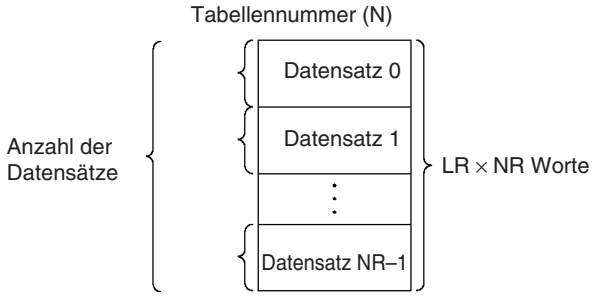
Beschreibung

DIM(631) registriert die Worte von TB bis TB+LR×NR–1 als Tabelle mit der Nummer N. Tabelle Nummer N umfasst NR Datensätze und jeder Datensatz besteht aus LR Worten. Die Bereichsdaten können nicht mehr geändert werden, sobald die Tabelle deklariert ist.

Verwenden Sie den Befehl DIM(631) in Kombination mit den Befehlen SETR(635) (SET RECORD NUMBER) oder GETR(636) (GET RECORD NUMBER) zur Vereinfachung der Berechnung von Adressen in Datentabellen. Verwenden Sie den Befehl DIM(631), um Daten in Datensätze aufzutei-

len, und anschließend den Befehl SETR(635), um die erste Adresse des gewünschten Datensatzes in einem Indexregister zu speichern. Das Indexregister kann dann als Zeiger in anderen Befehlen, wie beispielsweise Lese-, Schreib-, Such- oder Vergleichsbefehlen, verwendet werden.

Ein Beispiel: Wenn Temperaturen, Drücke oder andere Sollwerte als Datensätze gespeichert und die Datensätze verschiedener Modelle in einer Tabelle zusammengestellt werden, ist es ein Leichtes, die Sollwerte für die einzelnen Modelle für beliebige bestimmte Bedingungen zu lesen.



Die beiden mit DIM(631) in Verbindung stehenden Datensatzbefehle sind SETR(635) und GETR(636). SETR(635) setzt die erste SPS-Speicheradresse der spezifizierten Datensatznummer in dem spezifizierten Indexregister. GETR(636) gibt die Datensatznummer des Datensatzes aus, der den spezifizierten Indexregisterwert (SPS-Speicheradresse) enthält.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn LR oder NR den Wert 0000 haben. In allen anderen Fällen AUS.

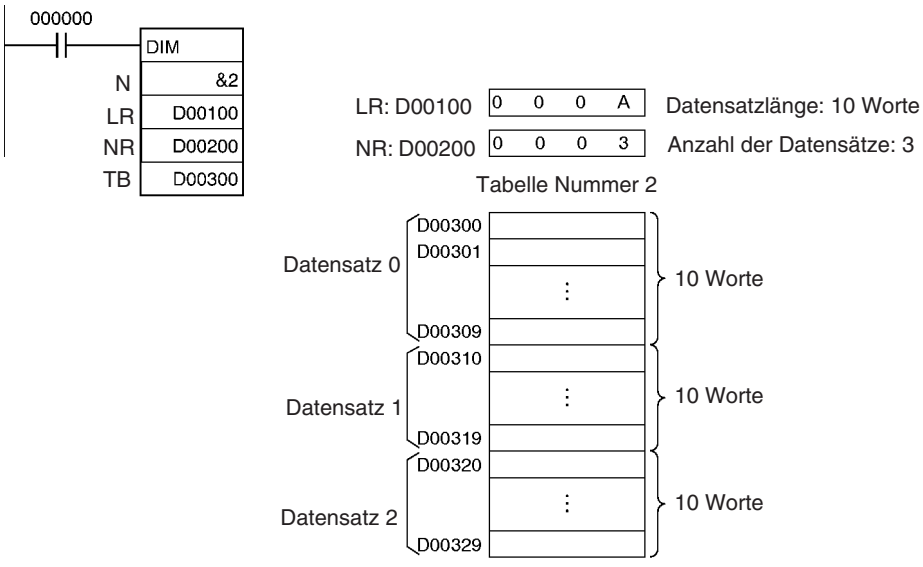
Vorsichtsmaßnahmen

Datensätze in einer registrierten Tabelle werden anhand ihrer Datensatznummer identifiziert, die im Bereich von 0 bis NR–1 liegt.

Je nach Einstellungen für die Datensatzlänge (LR) und die Anzahl der Datensätze (NR) ist es möglich, dass sich eine einzelne Tabelle (von TB und TB+LR×NR–1) über zwei Datenbereiche erstreckt. Vergewissern Sie sich, dass daraus keine Probleme entstehen, bevor Sie eine Tabelle spezifizieren, die sich über die Grenze eines Datenbereichs hinaus erstreckt.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, definiert DIM(631) die Datensatztabelle Nummer 2 mit drei Datensätzen, die aus jeweils 10 Worten bestehen. Die Tabelle beginnt mit D00300.

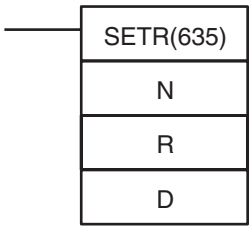


3-17-6 SET RECORD LOCATION: SETR(635)

Zweck

Schreibt die Position des spezifizierten Datensatzes (die SPS-Speicheradresse, an der der Datensatz beginnt) in das angegebene Indexregister.

Kontaktplansymbol



N: Tabellennummer

R: Datensatznummer

D: Ziel-Indexregister

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SETR(635)
	Ausführung einmal bei steigender Flanke	@SETR(635)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Tabellennummer
Spezifiziert die Tabellennummer. N muss eine Zahl zwischen 0 und 15 sein.

R: Datensatznummer
Gibt die Datensatznummer des gewünschten Datensatzes an. R muss einen Wert zwischen 0000 und FFFF hexadezimal (0 to 65.534) besitzen. Die Datensatznummern beginnen mit 0, demnach sind 0 bis NR–1 die gültigen Datensatznummern in einer Tabelle mit NR Datensätzen.

D: Ziel-Indexregister
Spezifiziert das gewünschte Indexregister. D muss ein Wert zwischen IR0 und IR15 sein.

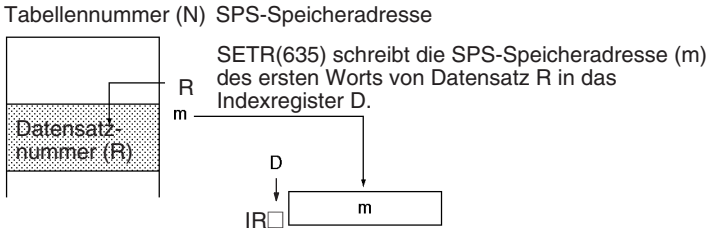
Operanden-Spezifikationen

Bereich	N	R	D
CIO-Bereich	---	CIO 0000 bis CIO 6143	---
Arbeitsbereich	---	W000 bis W511	---
Haftmerkerbereich	---	H000 bis H511	---
Zusatz-Systembereich	---	A000 bis A959	---
Zeitgeberbereich	---	T0000 bis T4095	---
Zählerbereich	---	C0000 bis C4095	---
DM-Bereich	---	D00000 bis D32767	---
EM-Bereich ohne Bank	---	E00000 bis E32767	---
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)	---
Indirekte DM/EM-Binäradresse	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767	---
Indirekte DM/EM-Adresse in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767	---
Konstanten	0 bis 15	#0000 bis #FFFE (binär) oder &0 bis 65534	---
Datenregister	---	DR0 bis DR15	---

Bereich	N	R	D
Indexregister	---		IR0 bis IR15
Indirektes Adressieren mittels Index-Registern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,– (– –)IR0 bis ,– (– –)IR15	---

Beschreibung

SETR(635) speichert die SPS-Speicheradresse des ersten Worts im spezifizierten Datensatz in das spezifizierte Indexregister. In der folgenden Abbildung ist die Grundfunktion des Befehls SETR(635) dargestellt.



Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die spezifizierte Tabellennummer (N) nicht durch den Befehl DIM(631) definiert wurde. EIN, wenn die spezifizierte Datensatznummer (R) größer ist als die höchste Datensatznummer in der Tabelle (NR–1). In allen anderen Fällen AUS.

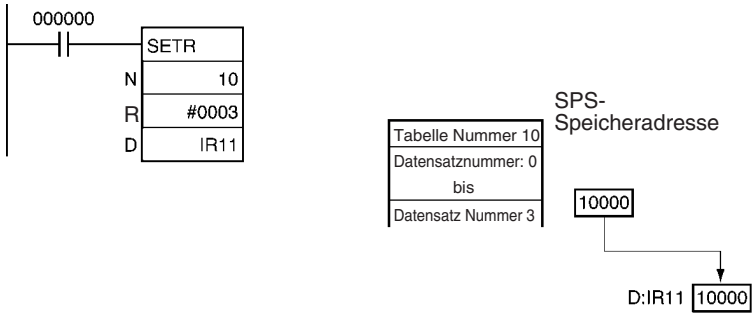
Vorsichtsmaßnahmen

Die Datensatztabelle muss zuvor durch den Befehl DIM(631) definiert werden.

Die gültigen Datensatznummern haben den Bereich von 0 bis NR–1, wobei NR die Anzahl der bei der Definition mittels DIM(631) spezifizierten Datensätze ist.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, sucht SETR(635) die SPS-Speicheradresse des ersten Worts von Datensatz 3 in Tabelle Nummer 10 und speichert diese Adresse im Indexregister IR11.

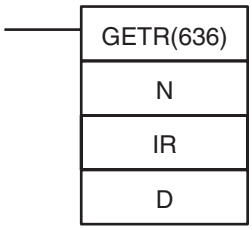


3-17-7 GET RECORD NUMBER: GETR(636)

Zweck

Gibt die Datensatznummer des Datensatzes an der SPS-Speicheradresse zurück, die im spezifizierten Indexregister enthalten ist.

Kontaktplansymbol



N: Tabellennummer

IR: Index Register

D: Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	GETR(636)
	Ausführung einmal bei steigender Flanke	@GETR(636)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Tabellennummer
Spezifiziert die Tabellennummer. N muss eine Zahl zwischen 0 und 15 sein.

IR: Indexregister
Spezifiziert das gewünschte Indexregister. D muss ein Wert zwischen IR0 und IR15 sein.

D: Zielwort
Spezifiziert das Wort, in das die Datensatznummer geschrieben wird.

Operanden-Spezifikationen

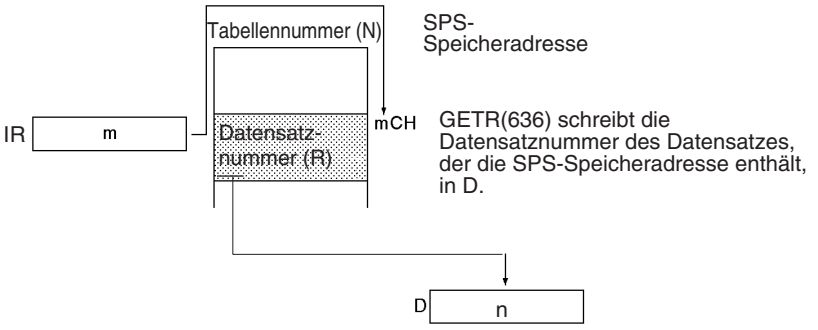
Bereich	N	IR	D
CIO-Bereich	---		CIO 0000 bis CIO 6143
Arbeitsbereich	---		W000 bis W511
Haftmerkerbereich	---		H000 bis H511
Zusatz-Systembereich	---		A448 bis A959
Zeitgeberbereich	---		T0000 bis T4095
Zählerbereich	---		C0000 bis C4095
DM-Bereich	---		D00000 bis D32767
EM-Bereich ohne Bank	---		E00000 bis E32767
EM-Bereich mit Bank	---		En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	---		@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adresse in BCD	---		*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	0 bis 15	---	---
Datenregister	---		DR0 bis DR15

Bereich	N	IR	D
Indexregister	---	IR0 bis IR15	---
Indirektes Adressieren mittels Index-Registern	---		,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

GETR(636) sucht nach dem Datensatz, der die im spezifizierten Indexregister enthaltene SPS-Speicheradresse umfasst und schreibt diese Datensatznummer in D. Die im Indexregister enthaltene SPS-Speicheradresse muss nicht das erste Wort des Datensatzes sein. Es kann ein beliebiges Wort des Datensatzes sein.

In der folgenden Abbildung ist die Grundfunktion des Befehls GETR(636) dargestellt.



Merker

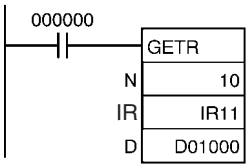
Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn die SPS-Speicheradresse in dem spezifizierten Indexregister nicht in der spezifizierten Tabelle (N) enthalten ist. EIN, wenn die spezifizierte Tabellennummer (N) nicht durch den Befehl DIM(631) definiert wurde. In allen anderen Fällen AUS.

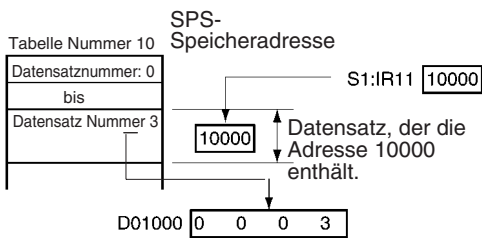
Vorsichtsmaßnahmen

Die Datensatztabelle muss zuvor durch den Befehl DIM(631) definiert werden und die SPS-Speicheradresse in dem spezifizierten Indexregister muss in der spezifizierten Tabelle liegen.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, sucht GETR(636) die Datensatznummer des Datensatzes, der die SPS-Speicheradresse in Indexregister IR11 enthält und schreibt diese Datensatznummer in D01000.



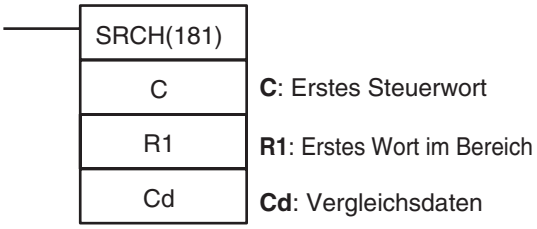


3-17-8 DATA SEARCH: SRCH(181)

Zweck

Sucht nach einem Datenwort innerhalb eines Wortbereichs.
In CPU-Baugruppen von CS1D-Einzel-CPU-Systemen und CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M kann dieser Befehl im Hintergrund ausgeführt werden. Detaillierte Informationen zur Hintergrundausführung finden Sie im *Speicherprogrammierbare Steuerungen CS/CJ-Serie Programmierhandbuch*.

Kontaktplansymbol



Variationen

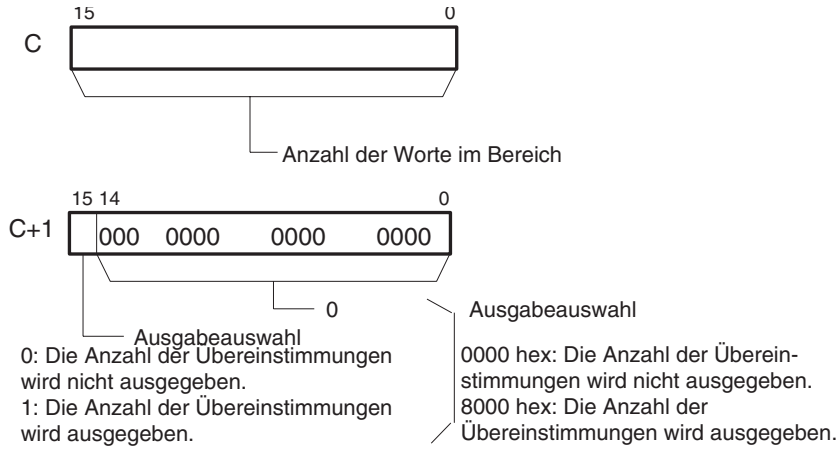
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SRCH(181)
	Ausführung einmal bei steigender Flanke	@SRCH(181)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

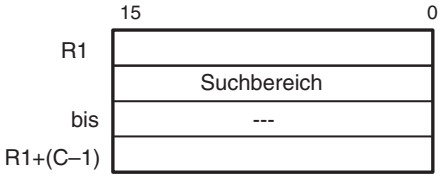
Operanden

C und C+1: Steuerworte
C spezifiziert die Anzahl der Worte in dem Bereich und Bit 15 von C+1 legt fest, ob die Anzahl der gefundenen Übereinstimmungen in DR00 ausgegeben wird oder nicht.



Hinweis C und C+1 müssen im gleichen Datenbereich liegen.

R1: Erstes Wort im Bereich
R1 spezifiziert das erste Wort des Suchbereichs. Die Worte zwischen R1 und R1+(C–1) werden nach den gewünschten Daten durchsucht. (C ist die Anzahl der in C eingestellten Worte.)



Hinweis R1 und R1+C–1 müssen im gleichen Datenbereich liegen.

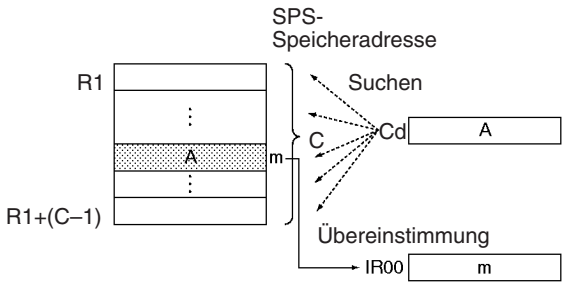
Operanden-Spezifikationen

Bereich	C	R1	Cd
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W510	W000 bis W511	
Haftmerkerbereich	H000 bis H510	H000 bis H511	
Zusatz-Systembereich	A000 bis A958	A000 bis A959	
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095	
Zählerbereich	C0000 bis C4094	C0000 bis C4095	
DM-Bereich	D00000 bis D32766	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	Nur spezifizierte Werte	---	#0000 bis #FFFF (binär)
Datenregister	---		DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

SRCH(181) durchsucht den Bereich des Speichers von R1 bis R1+C–1 nach Worten, die die spezifizierten Vergleichsdaten (Cd) enthalten. Wenn eine Übereinstimmung gefunden wird, schreibt SRCH(181) die SPS-Speicheradresse des Worts in IR00 und setzt den Gleich-Merker auf EIN.
(Wenn zwei oder mehr Übereinstimmungen vorhanden sind, wird nur die Adresse des ersten Worts, das die Vergleichsdaten enthält in IR00 geschrieben.)

Wenn Bit 15 von C+1 auf 1 gesetzt ist, schreibt SRCH(181) die Anzahl der gefundenen Übereinstimmungen in DR00. Wenn Bit 15 von C+1 auf 0 gesetzt ist, bleibt DR00 unverändert.



SRCH(181) durchsucht Tabellendaten, bei denen jeder Datensatz aus einem Wort besteht. Zum Durchsuchen von Tabellendaten, bei denen jeder Datensatz mehr als ein Wort enthält, müssen die Befehle DIM(631), SETR(635), GETR(636), FOR(512)–NEXT(513) oder BREAK(514) zusammen mit einem Indexregister (IR) verwendet werden.

Der Status des Gleich-Merkers kann unmittelbar nach Ausführung des Befehls geprüft werden, um zu ermitteln ob eine Übereinstimmung gefunden wurde.

Merker

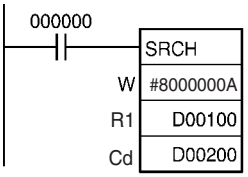
Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt von C nicht innerhalb des spezifizierten Bereichs von 0001 bis FFFF liegt. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn mindestens eines der Worte im Suchbereich die Vergleichsdaten enthält. EIN, wenn bei Spezifikation der Hintergrundausführung der Kommunikationsschnittstellen-Frei-Merker der als *Komm.-Port-Nummer* für *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

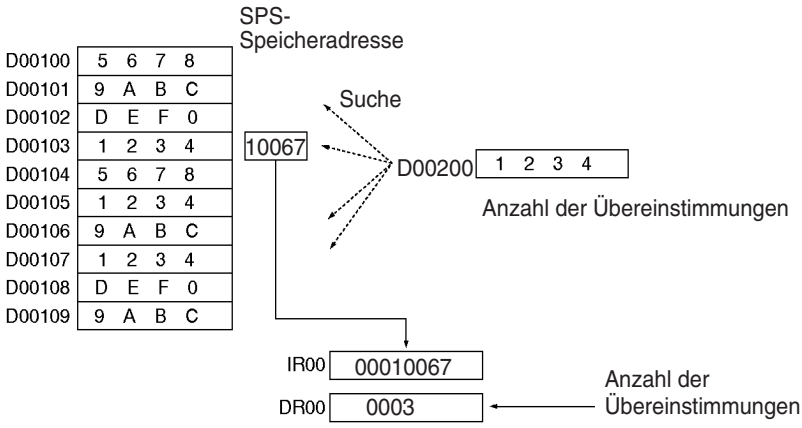
Vorsichtsmaßnahmen

Wenn keine Übereinstimmung gefunden wird, bleibt der Inhalt von IR00 und DR00 unverändert.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, durchsucht SRCH(181) den mit D00100 beginnenden und 10 Worte umfassenden Bereich nach Worten, die den gleichen Inhalt wie D00200 besitzen. Die SPS-Speicheradresse des ersten Worts, das eine Übereinstimmung enthält, wird in IR00 geschrieben und die Anzahl der gefundenen Übereinstimmungen wird in DR00 geschrieben.





Wenn die Tabellenlänge als &10 (10 dezimal) oder A hexadezimal spezifiziert ist, wird die Anzahl der gefundenen Übereinstimmungen nicht in das Datenregister DR00 ausgegeben.

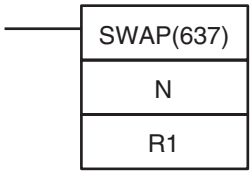
3-17-9 SWAP BYTES: SWAP(637)

Zweck

Vertauscht die Bytes höchster und niedrigster Wertigkeit in allen Worten innerhalb des Bereichs.

In CPU-Baugruppen von CS1D-Einzel-CPU-Systemen und CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M kann dieser Befehl im Hintergrund ausgeführt werden. Detaillierte Informationen zur Hintergrundausführung finden Sie im *Speicherprogrammierbare Steuerungen CS/CJ-Serie Programmierhandbuch*.

Kontaktplansymbol



N: Anzahl der Worte

R1: Erstes Wort im Bereich

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SWAP(637)
	Ausführung einmal bei steigender Flanke	@SWAP(637)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

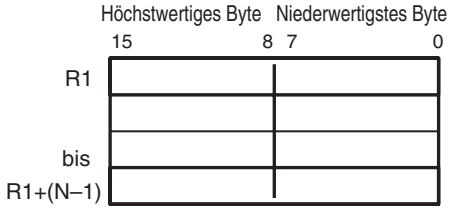
Operanden

N: Anzahl der Worte

N spezifiziert die Anzahl der Worte innerhalb des Bereichs und muss zwischen 0001 und FFFF hexadezimal (bzw. &1 und &65.535) liegen.

R1: Erstes Wort im Bereich

R1 spezifiziert das erste Wort des Bereichs. R1 und R1+(N–1) müssen im gleichen Datenbereich liegen.



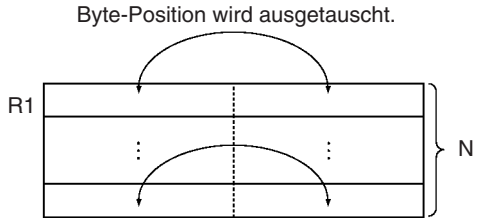
Operanden-Spezifikationen

Bereich	N	R1
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W511	
Haftmerkerbereich	H000 bis H511	
Zusatz-Systembereich	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0001 bis #FFFF (binär) oder &1 bis &65.535	---

Bereich	N	R1
Datenregister	DR00 bis DR15	---
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

SWAP(637) vertauscht die Position der beiden Bytes aller Worte im Speicherbereich zwischen R1 und R1+N–1. Dieser Befehl kann verwendet werden, um die Reihenfolge ASCII-codierter Zeichen in jedem Wort umzukehren.

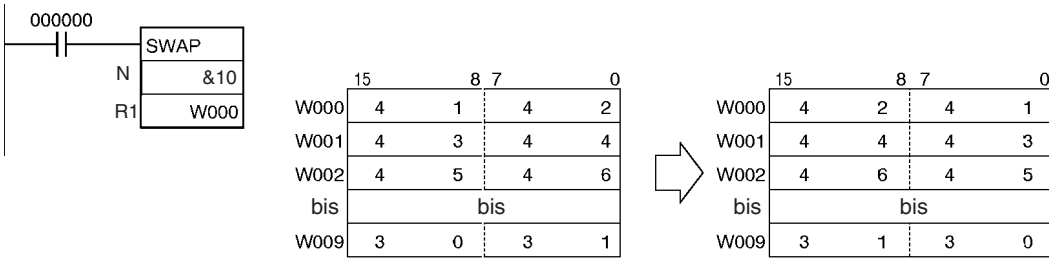


Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn N gleich 0000 ist. EIN, wenn bei Spezifikation der Hintergrundausführung der Kommunikationsschnittstellen-Frei-Merker der als *Komm.-Port-Nummer* für *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, vertauscht SWAP(637) die Daten des höchstwertigen Bytes mit denen im niederwertigsten Byte in allen Worten des 10 Worte umfassenden Bereichs von W000 bis W009.

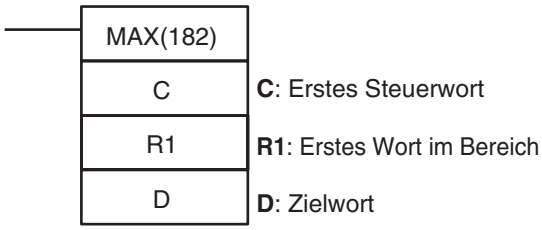


3-17-10 FIND MAXIMUM: MAX(182)

Zweck

Sucht den maximalen Wert innerhalb des Bereichs.
In CPU-Baugruppen von CS1D-Einzel-CPU-Systemen und CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M kann dieser Befehl im Hintergrund ausgeführt werden. Detaillierte Informationen zur Hintergrundausführung finden Sie im *Speicherprogrammierbare Steuerungen CS/CJ-Serie Programmierhandbuch*.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MAX(182)
	Ausführung einmal bei steigender Flanke	@MAX(182)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

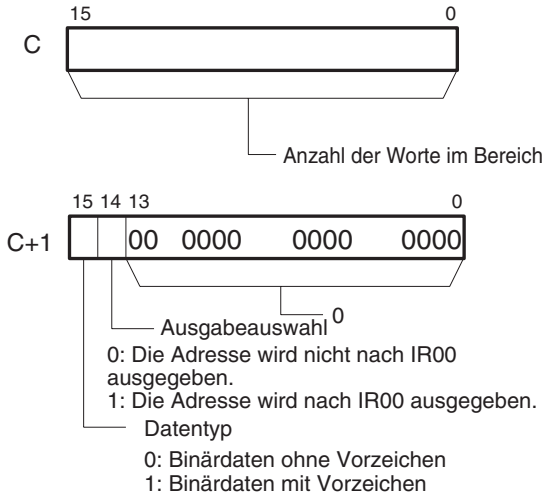
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C und C+1: Steuerworte

C spezifiziert die Anzahl der Worte im Bereich, Bit 15 von C+1 legt fest. ob die Daten als Binärdaten mit oder ohne Vorzeichen behandelt werden, und Bit 14 von C+1 legt fest, ob die SPS-Speicheradresse des Worts, dass den Maximalwert enthält, nach IR00 ausgegeben wird oder nicht.

Hinweis C und C+1 müssen im gleichen Datenbereich liegen.

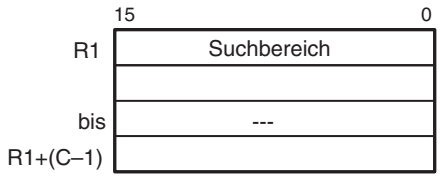


Die nachstehende Tabelle zeigt die möglichen Werte von C.

C+1	Datentyp	Indexregister-Ausgabe
0000	Binär ohne Vorzeichen	Nein
4000	Binär ohne Vorzeichen	Ja
8000	Binär mit Vorzeichen	Nein
C000	Binär mit Vorzeichen	Ja

R1: Erstes Wort im Bereich

R1 spezifiziert das erste Wort des Suchbereichs. Die Worte zwischen R1 und R1+(C–1) werden nach den Maximalwert durchsucht. (C ist die Anzahl der in C spezifizierten Worte.)



Hinweis R1 und R1+(C–1) müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

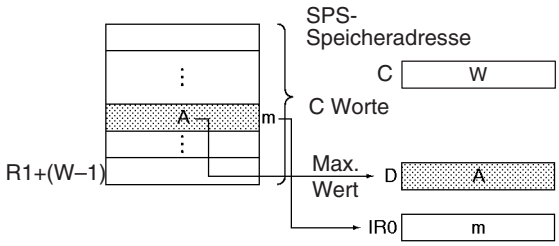
Bereich	C	R1	D
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W510	W000 bis W511	
Haftmerkerbereich	H000 bis H510	H000 bis H511	
Zusatz-Systembereich	A000 bis A958	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095	
Zählerbereich	C0000 bis C4094	C0000 bis C4095	
DM-Bereich	D00000 bis D32766	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	Nur spezifizierte Werte	---	
Datenregister	---		DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

MAX(182) durchsucht den Bereich des Speichers von R1 bis R1+C–1 nach dem Maximalwert innerhalb dieses Bereichs und gibt diesen Maximalwert nach D aus.

Wenn Bit 14 von C+1 auf 1 gesetzt ist, schreibt der Befehl MAX(182) die SPS-Speicheradresse des Worts, das den Maximalwert enthält, in IR00. (Wenn zwei oder mehr Worte innerhalb des Bereichs den Maximalwert enthalten, wird die Adresse des ersten Worts, das den Maximalwert enthält, in IR00 geschrieben.)

Wenn Bit 15 von C+1 auf 1 gesetzt ist, behandelt der Befehl MAX(182) die Daten innerhalb des Bereichs als Binärdaten mit Vorzeichen.



Merker

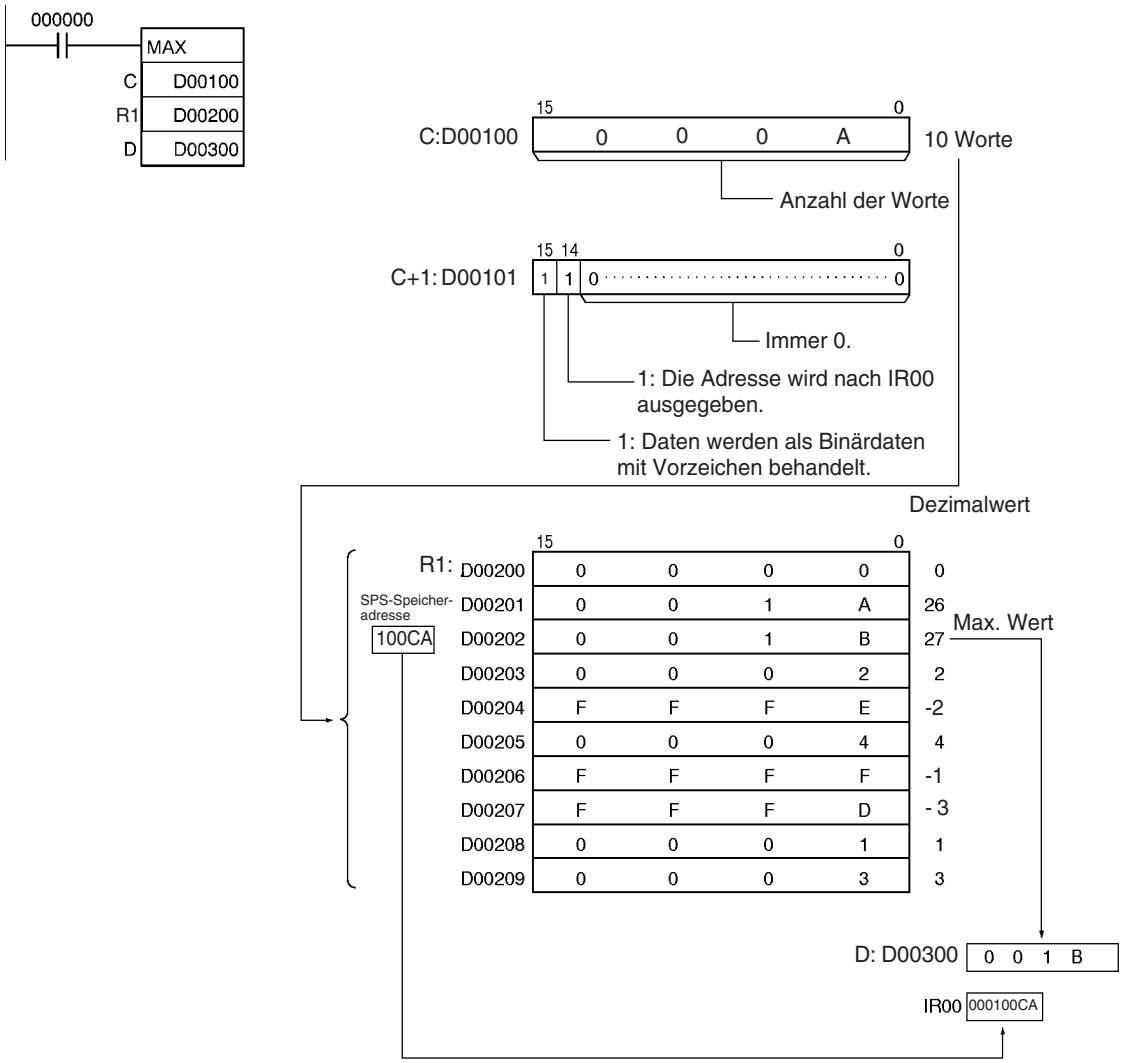
Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt von C nicht innerhalb des spezifizierten Bereichs von 0001 bis FFFF liegt. EIN, wenn bei Spezifikation der Hintergrundausführung der Kommunikationsschnittstellen-Frei-Merker der als *Komm.-Port-Nummer* für *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn der Maximalwert 0000 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn Bit 15 in dem Wort, das den Maximalwert enthält, auf EIN gesetzt ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn Bit 15 von C+1 auf 1 gesetzt ist, werden die Daten innerhalb des Bereichs als Binärdaten mit Vorzeichen behandelt und die Hexadezimalwerte 8000 bis FFFF als negativ betrachtet. Dementsprechend ist das Suchergebnis je nach Datentypeinstellung unterschiedlich.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt wird, durchsucht MAX(182) den 10 Worte umfassenden und mit D00200 beginnenden Bereich nach dem Maximalwert. Der Maximalwert wird in D00300 geschrieben und die SPS-Speicheradresse des Worts, das den Maximalwert enthält, wird in IR00 geschrieben.

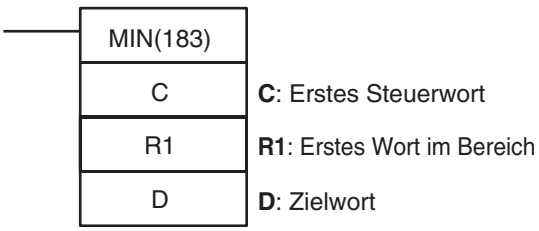


3-17-11 FIND MINIMUM: MIN(183)

Zweck

Sucht den geringsten Wert innerhalb des Bereichs.
In CPU-Baugruppen von CS1D-Einzel-CPU-Systemen und CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M kann dieser Befehl im Hintergrund ausgeführt werden. Detaillierte Informationen zur Hintergrundausführung finden Sie im *Speicherprogrammierbare Steuerungen CS/CJ-Serie Programmierhandbuch*.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MIN(183)
	Ausführung einmal bei steigender Flanke	@MIN(183)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

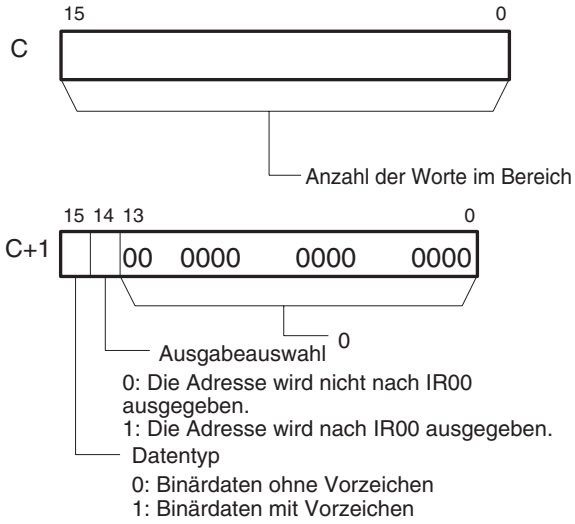
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C und C+1: Steuerworte

C spezifiziert die Anzahl der Worte im Bereich, Bit 15 von C+1 legt fest. ob die Daten als Binärdaten mit oder ohne Vorzeichen behandelt werden, und Bit 14 von C+1 legt fest, ob die SPS-Speicheradresse des Worts, dass den Minimalwert enthält, nach IR00 ausgegeben wird oder nicht.

Hinweis C und C+1 müssen im gleichen Datenbereich liegen.

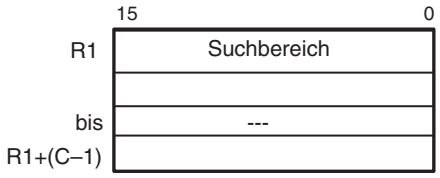


Die nachstehende Tabelle zeigt die möglichen Werte von C.

C+1	Datentyp	Indexregister-Ausgabe
0000	Binär ohne Vorzeichen	Nein
4000	Binär ohne Vorzeichen	Ja
8000	Binär mit Vorzeichen	Nein
C000	Binär mit Vorzeichen	Ja

R1: Erstes Wort im Bereich

R1 spezifiziert das erste Wort des Suchbereichs. Die Worte zwischen R1 und R1+(C–1) werden nach dem Minimalwert durchsucht. (C ist die Anzahl der in C spezifizierten Worte.)



Hinweis R1 und R1+C–1 müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	C	R1	D
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W510	W000 bis W511	
Haftmerkerbereich	H000 bis H510	H000 bis H511	

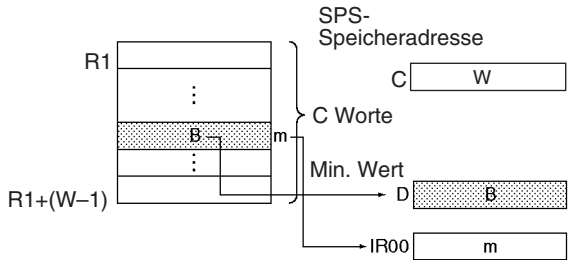
Bereich	C	R1	D
Zusatz-Systembereich	A000 bis A958	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095	
Zählerbereich	C0000 bis C4094	C0000 bis C4095	
DM-Bereich	D00000 bis D32766	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	Nur spezifizierte Werte	---	
Datenregister	---		DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

MIN(183) durchsucht den Bereich des Speichers von R1 bis R1+C–1 nach dem Minimalwert innerhalb dieses Bereichs und gibt diesen Minimalwert nach D aus.

Wenn Bit 14 von C+1 auf 1 gesetzt ist, schreibt der Befehl MIN(183) die SPS-Speicheradresse des Worts, das den Minimalwert enthält, in IR00. (Wenn zwei oder mehr Worte innerhalb des Bereichs den Minimalwert enthalten, wird die Adresse des ersten Worts, das den Minimalwert enthält, in IR00 geschrieben.)

Wenn Bit 15 von C+1 auf 1 gesetzt ist, behandelt der Befehl MIN(183) die Daten innerhalb des Bereichs als Binärdaten mit Vorzeichen.



Merker

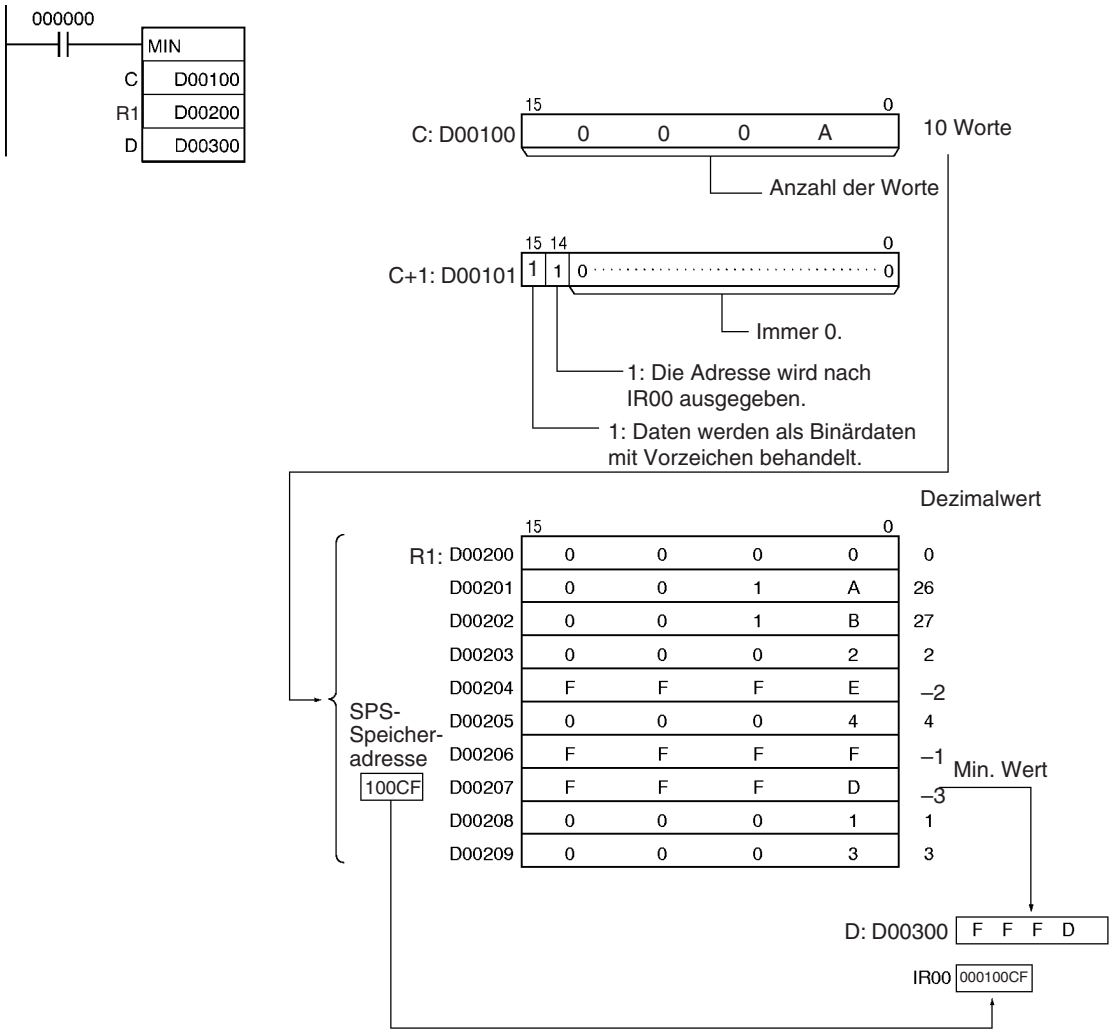
Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt von C nicht innerhalb des spezifizierten Bereichs von 0001 bis FFFF liegt. EIN, wenn bei Spezifikation der Hintergrundausführung der Kommunikationsschnittstellen-Frei-Merker der als *Komm.-Port-Nummer* für *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn der Minimalwert 0000 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn Bit 15 in dem Wort, das den Minimalwert enthält, auf EIN gesetzt ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn Bit 15 von C+1 auf 1 gesetzt ist, werden die Daten innerhalb des Bereichs als Binärdaten mit Vorzeichen behandelt und die Hexadezimalwerte 8000 bis FFFF als negativ betrachtet. Dementsprechend ist das Suchergebnis je nach Datentypeinstellung unterschiedlich.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt wird, durchsucht MIN(183) den 10 Worte umfassenden und mit D00200 beginnenden Bereich nach dem Minimalwert. Der Minimalwert wird in D00300 geschrieben und die SPS-Speicheradresse des Worts, das den Minimalwert enthält, wird in IR00 geschrieben.

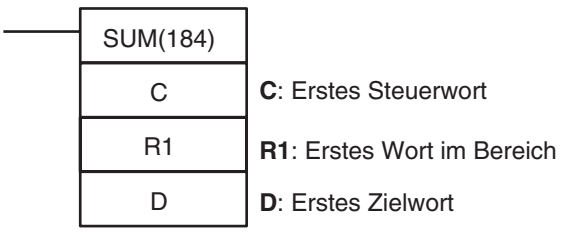


3-17-12 SUM: SUM(184)

Zweck

Addiert die Bytes oder Worte innerhalb des Bereichs und gibt das Ergebnis in zwei Worte aus.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SUM(184)
	Ausführung einmal bei steigender Flanke	@SUM(184)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

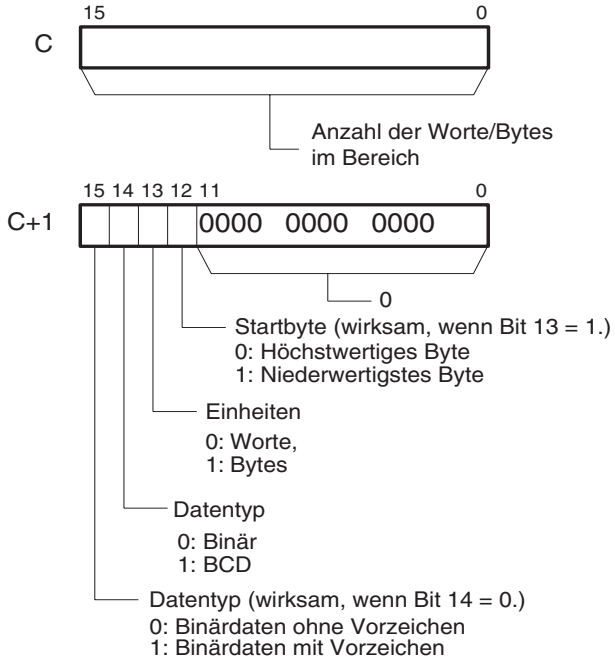
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C und C+1: Steuerworte

C spezifiziert die Anzahl der Einheiten (Bytes oder Worte) deren Summe gebildet werden soll. (Bit 13 von C+1 spezifiziert, ab Bytes oder Worte addiert werden.)

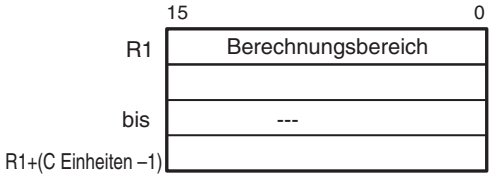
Bits 12 bis 15 von C+1 geben an, welche Art von Daten addiert wird (siehe folgende Abbildung).



Hinweis C und C+1 müssen im gleichen Datenbereich liegen.

R1: Erstes Wort im Bereich

R1 spezifiziert das erste Wort des Bereichs. Die Länge des Bereichs hängt sowohl von der Anzahl der Einheiten als auch von Startbyte ab, wenn Bytes addiert werden.



Hinweis Alle Worte des Berechnungsbereichs müssen im gleichen Datenbereich liegen.

D: Erstes Zielwort

Das Ergebnis der Berechnung wird nach D+1 und D ausgegeben. Die höchstwertigen vier Stellen werden in D+1, die niederwertigsten vier Stellen in D gespeichert.

Operanden-Spezifikationen

Bereich	C	R1	D
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142
Arbeitsbereich	W000 bis W510	W000 bis W511	W000 bis W510
Haftmerkerbereich	H000 bis H510	H000 bis H511	H000 bis H510
Zusatz-Systembereich	A000 bis A958	A000 bis A959	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095	T0000 bis T4094
Zählerbereich	C0000 bis C4094	C0000 bis C4095	C0000 bis C4094
DM-Bereich	D00000 bis D32766	D00000 bis D32767	D00000 bis D32766

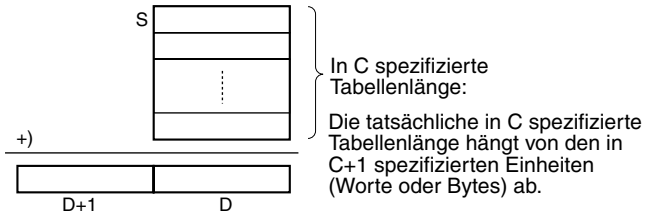
Bereich	C	R1	D
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767	E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	Nur spezifizierte Werte	---	
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl SUM(184) addiert C Dateneinheiten, mit den Daten in R1 beginnend, zueinander und gibt das Ergebnis nach D+1 und D aus. Die Einstellungen in C+1 bestimmen, ob es sich bei den Einheiten um Worte oder Bytes handelt, ob es sich bei den Daten um Binär- (mit oder ohne Vorzeichen) oder BCD-Daten handelt, und ob mit dem nieder- (rechten) oder höherwertigen (linken) Byte von R1 (falls Bytes addiert werden) begonnen wird.

Wenn Bit 14 von C+1 auf 0 gesetzt ist, behandelt der Befehl SUM(184) die Daten als Binärdaten. In diesem Fall bestimmt Bit 15, ob es sich um Daten mit Vorzeichen (Bit 15 = 1) oder ohne Vorzeichen (Bit 15 = 0) handelt.

Wenn Bit 13 von C+1 auf 1 gesetzt ist, addiert der Befehl SUM(184) die Datenbytes. In diesem Fall bestimmt Bit 12 ob die Berechnung mit dem niederwertigsten Byte von R1 (Bit 12 = 1) oder dem höchstwertigen Byte von R1 (Bit 12 = 0) beginnt.

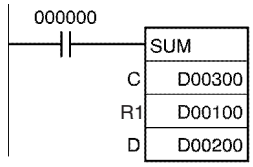


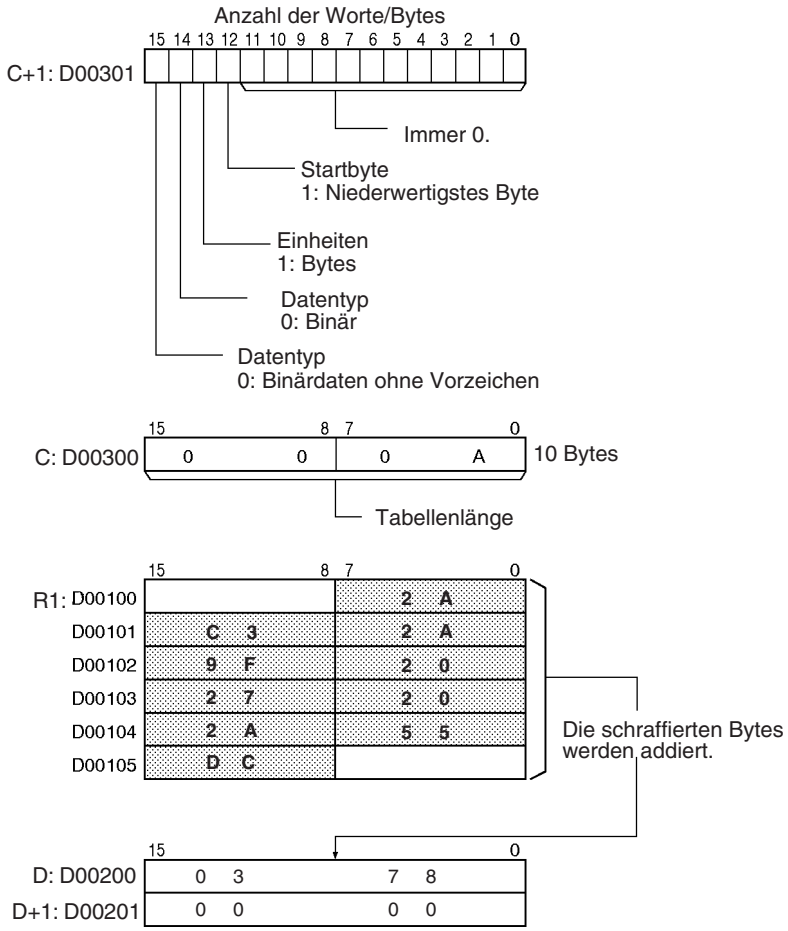
Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt von C nicht innerhalb des spezifizierten Bereichs von 0001 bis FFFF liegt. EIN, wenn BCD-Daten spezifiziert wurden, der Bereich jedoch Binärdaten enthält. EIN, wenn bei Spezifikation der Hintergrundausführung der Kommunikationsschnittstellen-Frei-Merker der als *Komm.-Port-Nummer* für *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn Bit 15 des Ergebnisses EIN ist. In allen anderen Fällen AUS.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, addiert der Befehl SUM(184) 10 Bytes vorzeichenloser Binärdaten, mit dem niederwertigen Byte von D00100 beginnend, und schreibt das Ergebnis in D00201 und D00200.



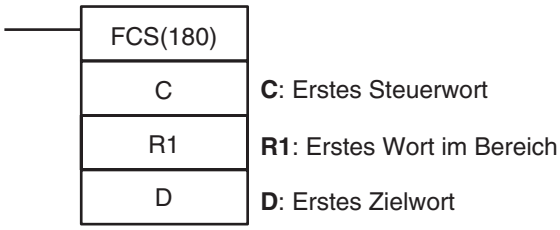


3-17-13 FRAME CHECKSUM: FCS(180)

Zweck

Berechnet den Rahmensummenwert (FCS) für den spezifizierten Bereich und gibt das Ergebnis im ASCII-Format aus.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FCS(180)
	Ausführung einmal bei steigender Flanke	@FCS(180)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

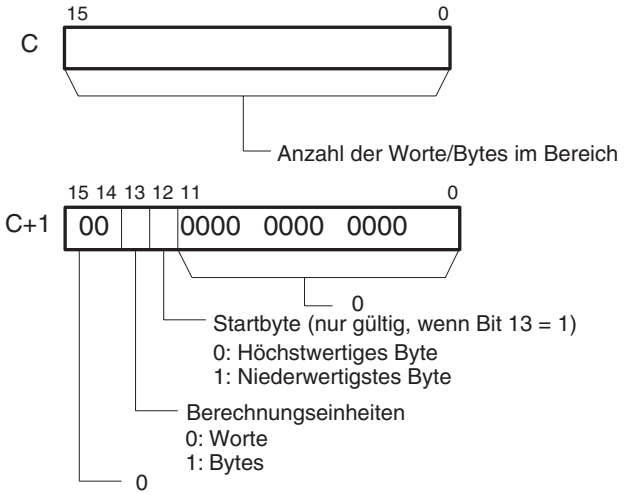
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C und C+1: Steuerworte

C spezifiziert die Anzahl der Einheiten (Bytes oder Worte), die für die Berechnung der Rahmenprüfsumme verwendet werden sollen. (Bit 13 von C+1 spezifiziert, ab Bytes oder Worte verwendet werden.)

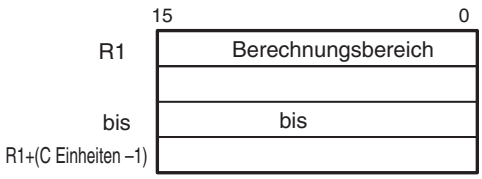
Wenn Bit 13 von C+1 auf 1 gesetzt ist, berechnet der Befehl FCS(180) den Rahmenprüfsummenwert (FCS) der Datenbytes. In diesem Fall bestimmt Bit 12 ob die Berechnung mit dem niederwertigsten Byte von R1 (Bit 12 = 1) oder dem höchstwertigen Byte von R1 (Bit 12 = 0) beginnt.



Hinweis C und C+1 müssen im gleichen Datenbereich liegen.

R1: Erstes Wort im Bereich

R1 spezifiziert das erste Wort des Bereichs. Die Länge des Bereichs hängt sowohl von der Anzahl der Einheiten als auch vom Startbyte ab, wenn Bytes für die Berechnung verwendet werden.



Hinweis Alle Worte des Berechnungsbereichs müssen im gleichen Datenbereich liegen.

D: Erstes Zielwort
Wenn Bytes zur Berechnung gewählt wurden, wird das Berechnungsergebnis in D ausgegeben.

Wenn Worte zur Berechnung gewählt wurden, wird das Berechnungsergebnis in D+1 und D ausgegeben. In diesem Fall werden die höchstwertigen vier Stellen in D+1, die niederwertigsten vier Stellen in D gespeichert.

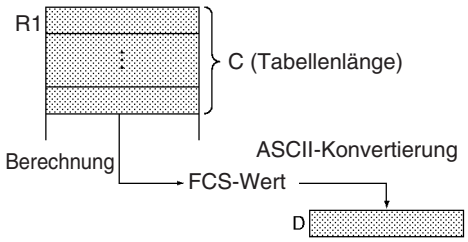
Operanden-Spezifikationen

Bereich	C	R1	D
CIO-Bereich	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W510	W000 bis W511	
Haftmerkerbereich	H000 bis H510	H000 bis H511	
Zusatz-Systembereich	A000 bis A958	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4094	T0000 bis T4095	
Zählerbereich	C0000 bis C4094	C0000 bis C4095	
DM-Bereich	D00000 bis D32766	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32766	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	Nur spezifizierte Werte	---	
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

FCS(180) berechnet den Rahmenprüfsummenwert für C Dateneinheiten mit den Daten in R1 beginnend, konvertiert den Wert in das ASCII-Format und gibt das Ergebnis nach D (bei Bytes) bzw. D+1 und D (bei Worten) aus. Die Einstellungen in C+1 bestimmen, ob es sich bei den Einheiten um Worte oder Bytes handelt, ob es sich bei den Daten um Binär- (mit oder ohne Vorzeichen) oder BCD-Daten handelt, und ob mit dem nieder- (rechten) oder höherwertigen (linken) Byte von R1 (falls Bytes addiert werden) begonnen wird.

Wenn Bit 13 von C+1 auf 1 gesetzt ist, verarbeitet der Befehl FCS(180) Datenbytes. In diesem Fall bestimmt Bit 12 ob die Berechnung mit dem niederwertigsten Byte von R1 (Bit 12 = 1) oder dem höchstwertigen Byte von R1 (Bit 12 = 0) beginnt.

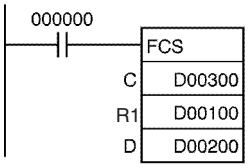


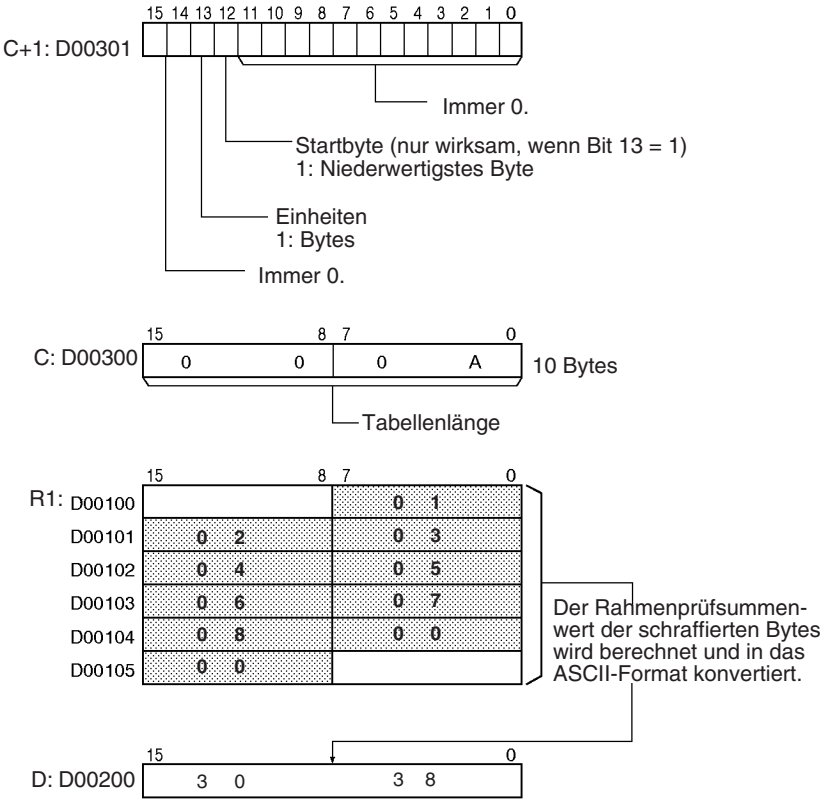
Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt von C nicht innerhalb des spezifizierten Bereichs von 0001 bis FFFF liegt. EIN, wenn bei Spezifikation der Hintergrundausführung der Kommunikationsschnittstellen-Frei-Merker der als *Komm.-Port-Nummer* für *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, berechnet FCS(180) den Rahmenprüfsummenwert von 10 Bytes Daten mit dem niederwertigen Byte von D00100 beginnend und schreibt das Ergebnis in D00200.





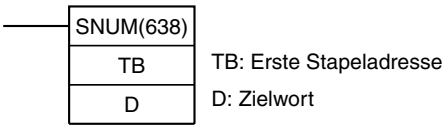
3-17-14 STACK SIZE READ: SNUM(638)

Zweck

Zählt die Menge der Stapeldaten (Anzahl der Worte) im spezifizierten Stapel.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SNUM(638)
	Ausführung einmal bei steigender Flanke	@SNUM(638)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

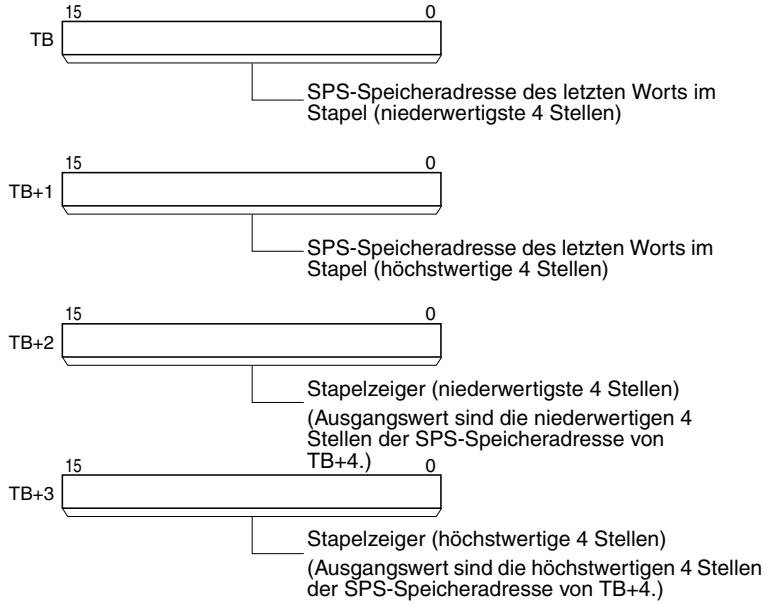
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

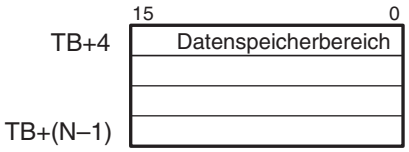
TB bis TB+3: Stapel-Steuerworte

Die ersten vier Worte des Stapels enthalten die SPS-Speicheradresse des letzten Worts im Stapel und den Stapelzeiger (die SPS-Speicheradresse des nächsten freien Worts im Stapel).



TB+4 bis TB+(N–1): Datenspeicherbereich

Der Rest des Stapels wird zum Speichern von Daten verwendet.



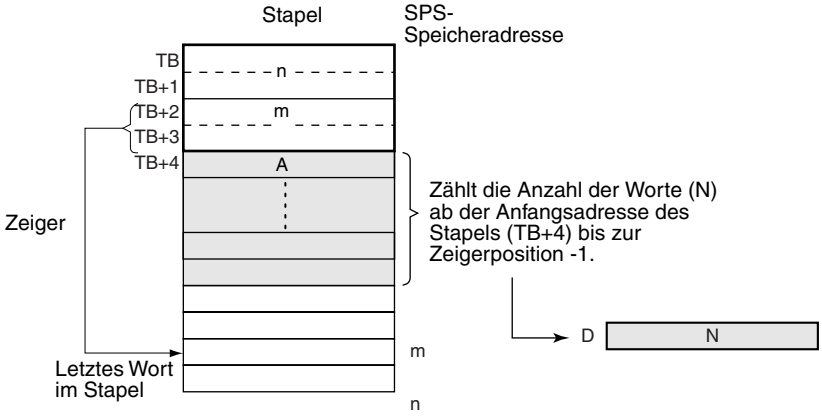
Operanden-Spezifikationen

Bereich	TB	D
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich	W000 bis W511	
Haftmerkerbereich	H000 bis H511	
Zusatz-Systembereich	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	

Bereich	TB	D
Datenregister	---	DR0 bis DR15
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

SNUM(638) zählt die Anzahl der Datenworte im spezifizierten Stapel ab dem Anfang des Datenbereichs bei TB+4 bis zu der Adresse unmittelbar vor der Adresse, die durch den Stapelzeiger (TB+3 und TB+2) angegeben wird. SNUM(638) verändert weder die Daten im Stapel noch die des Stapelzeigers.



Merker

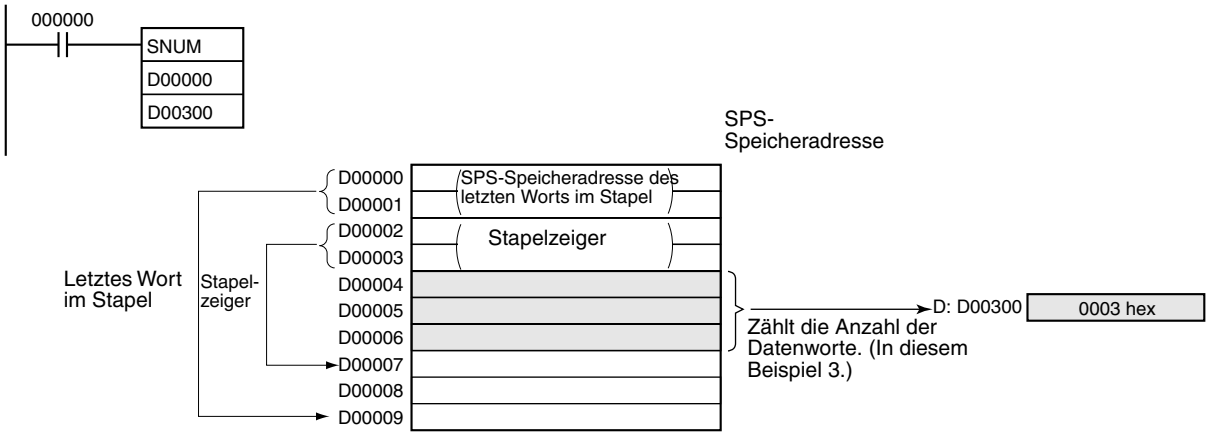
Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Anzahl von Datenworten im Stapel (der nach D ausgegebene Wert) 0 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Stapel muss zuvor durch den Befehl SSET(630) definiert werden.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, zählt der Befehl SNUM(638) die Anzahl der Worte ab dem Anfang des Datenbereichs an D00004 bis zur Zeigerposition - 1 (D00006) und gibt das Ergebnis nach D00300 aus. (In diesem Fall wird durch den Stapelzeiger D00007 bestimmt.)



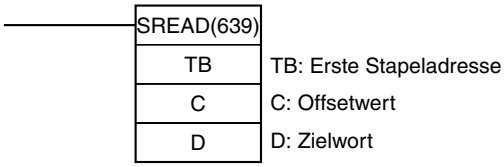
3-17-15 STACK DATA READ: SREAD(639)

Zweck

Liest die Daten ab dem spezifizierten Datenelement im Stapel. Der Offset-Wert bestimmt die Stelle des gewünschten Datenelements (Anzahl der Datenelemente vor der aktuellen Zeigerposition).

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SREAD(639)
	Ausführung einmal bei steigender Flanke	@SREAD(639)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

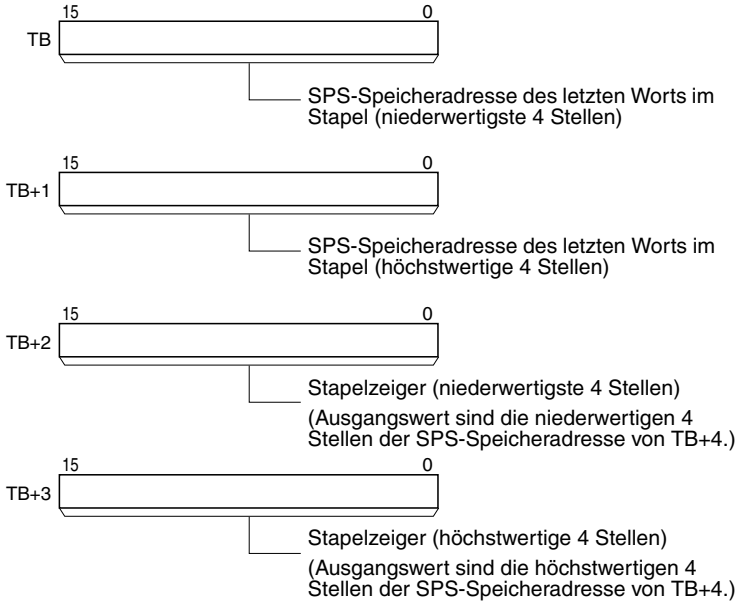
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

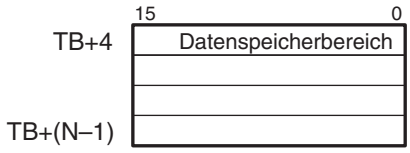
Operanden

TB bis TB+3: Stapel-Steuerworte

Die ersten vier Worte des Stapels enthalten die SPS-Speicheradresse des letzten Worts im Stapel und den Stapelzeiger (die SPS-Speicheradresse des nächsten freien Worts im Stapel).



TB+4 bis TB+(N–1): Datenspeicherbereich
Der Rest des Stapels wird zum Speichern von Daten verwendet.

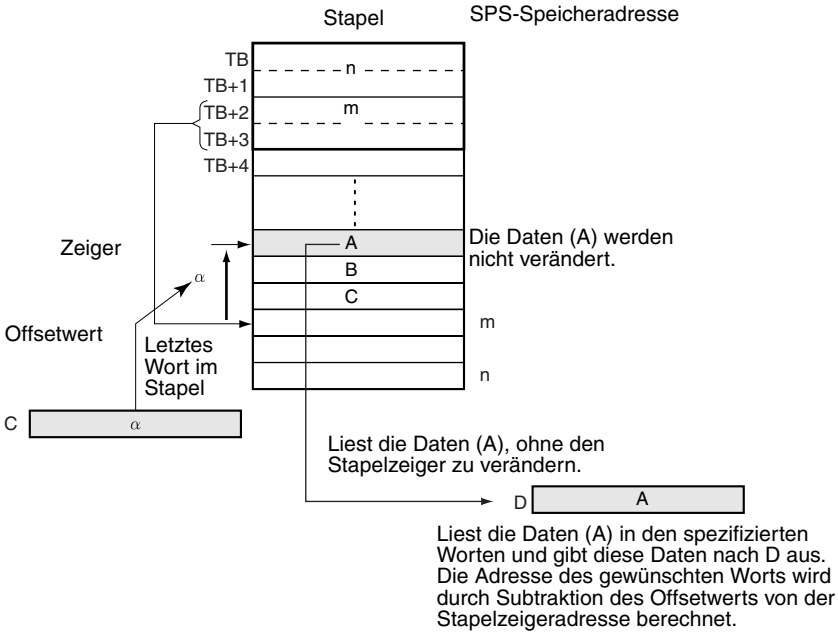


Operanden-Spezifikationen

Bereich	TB	C	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerkerbereich	H000 bis H511		
Zusatz-Systembereich	A448 bis A959	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0001 bis #FFFB (hexadezimal)	---
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

SREAD(639) liest die Daten aus der durch den Stapelzeiger (TB+3 und TB+2) spezifizierten Adresse minus des Offsetwerts in C. SREAD(639) verändert weder die Daten im Stapel noch die des Stapelzeigers.



SREAD(639) kann verwendet werden, um die Daten eines Objekts zu lesen, dass sich in einer Förderanlage befindet. Die Position des gewünschten Objekts ist einfach die Anzahl der Objekte (der Offsetwert) zurückgezählt, die vor dem neuesten Objekt in die Förderanlage gelangt sind.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die spezifizierte Leseposition nicht innerhalb des Stapelbereichs liegt. EIN, wenn der in C spezifizierte Offsetwert 0 ist oder größer als die Maximalgröße des Datenbereichs (FFFB hex). In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn die in D ausgegebenen Daten 0000 sind. In allen anderen Fällen AUS.

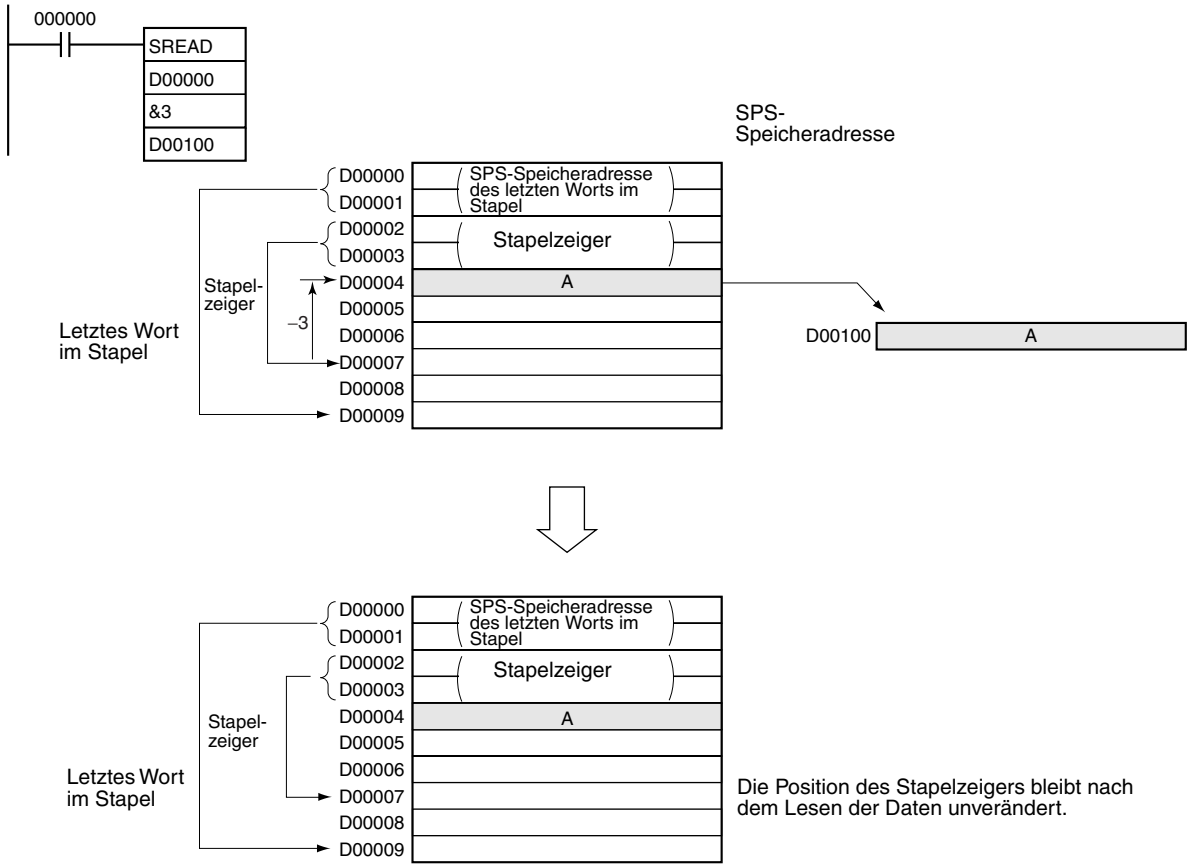
Vorsichtsmaßnahmen

Der Stapel muss zuvor durch den Befehl SSET(630) definiert werden.

Die vom Stapelzeiger spezifizierte Adresse muss größer sein als die SPS-Speicheradresse, die den Anfang des Datenbereichs bildet (TB+4). Wenn der Wert des Stapelzeigers kleiner ist als die SPS-Speicheradresse von TB+4, tritt ein Stapelunterlauffehler auf.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, liest der Befehl SREAD(639) die Daten in dem spezifizierten Wort des bei D00000 beginnenden Stapels und gibt diese Daten nach D00100 aus. In diesem Fall spezifiziert der Stapelzeiger die Adresse D00007 und der Offsetwert beträgt 3, sodass die Daten aus D00004 gelesen werden.



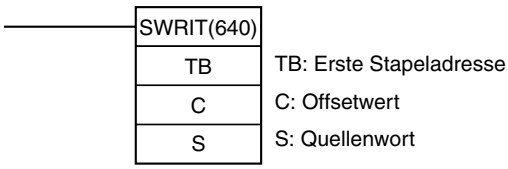
3-17-16 STACK DATA OVERWRITE: SWRIT(640)

Zweck

Schreibt die Eingangsdaten für das spezifizierte Datenelement in den Stapel (überschreibt vorhandene Daten). Der Offset-Wert bestimmt die Stelle des gewünschten Datenelements (Anzahl der Datenelemente vor der aktuellen Zeigerposition).

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SWRIT(640)
	Ausführung einmal bei steigender Flanke	@SWRIT(640)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

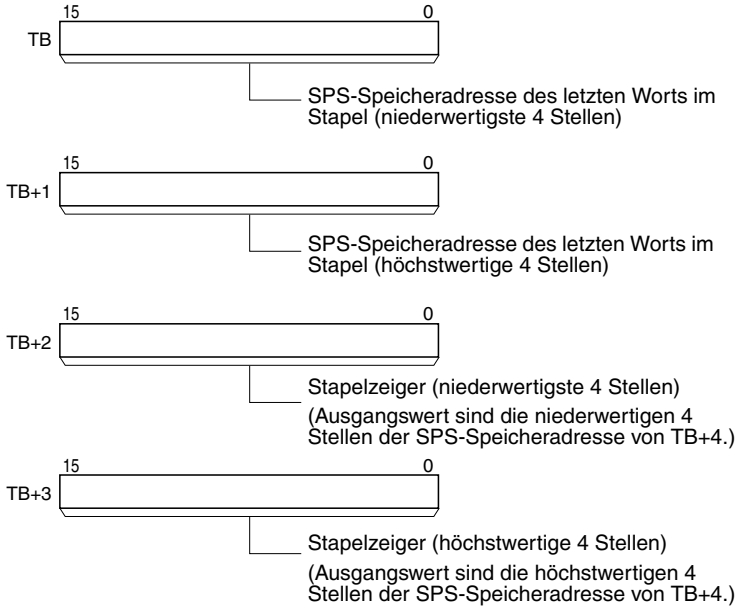
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

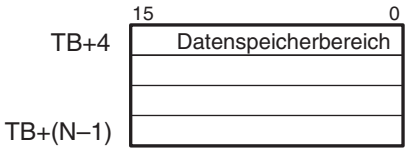
TB bis TB+3: Stapel-Steuerworte

Die ersten vier Worte des Stapels enthalten die SPS-Speicheradresse des letzten Worts im Stapel und den Stapelzeiger (die SPS-Speicheradresse des nächsten freien Worts im Stapel).



TB+4 bis TB+(N–1): Datenspeicherbereich

Der Rest des Stapels wird zum Speichern von Daten verwendet.



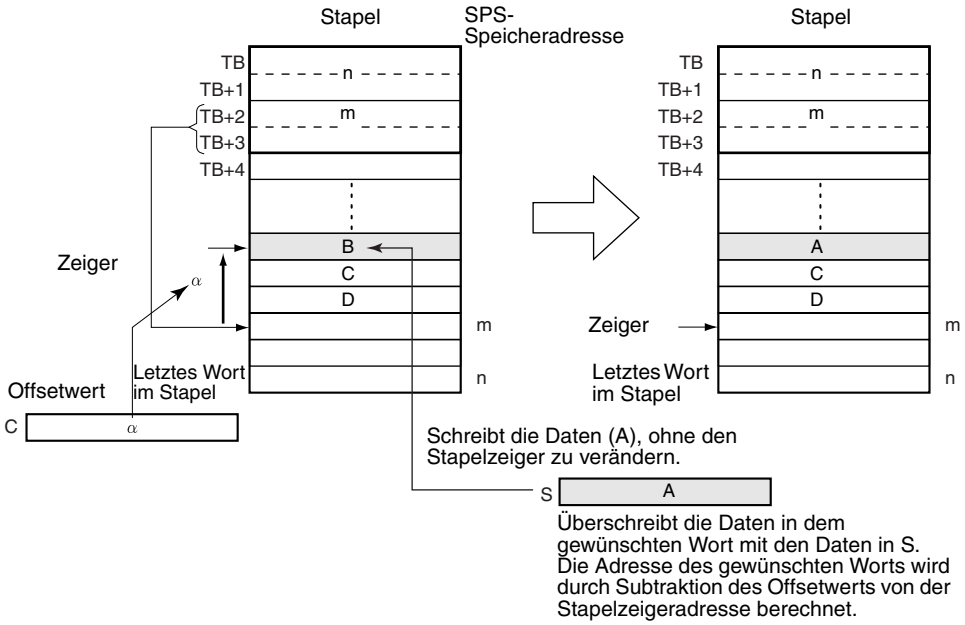
Operanden-Spezifikationen

Bereich	TB	C	S
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerkerbereich	H000 bis H511		
Zusatz-Systembereich	A448 bis A959	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		

Bereich	TB	C	S
Konstanten	---	#0001 bis #FFFB (hexadezimal)	#0000 bis #FFFF (hexadezimal)
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

SWRIT(640) überschreibt die Daten in dem gewünschten Wort mit den in S spezifizierten Daten. Die Adresse des gewünschten Worts wird durch Subtraktion des Offsetwerts in C von der Position des Stapelzeigers (TB+3 und TB+2) berechnet. SWRIT(640) verändert den Stapelzeiger nicht.



SWRIT(640) kann verwendet werden, um die Daten eines Objekts zu ändern, dass sich in einer Förderanlage befindet. Die Position des gewünschten Objekts ist einfach die Anzahl der Objekte (der Offsetwert) zurückgezählt, die vor dem neuesten Objekt in die Förderanlage gelangt sind.

Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn die spezifizierte Schreibposition nicht innerhalb des Stapelbereichs liegt. EIN, wenn der in C spezifizierte Offsetwert 0 ist oder größer als die Maximalgröße des Datenbereichs (FFFB hex). In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

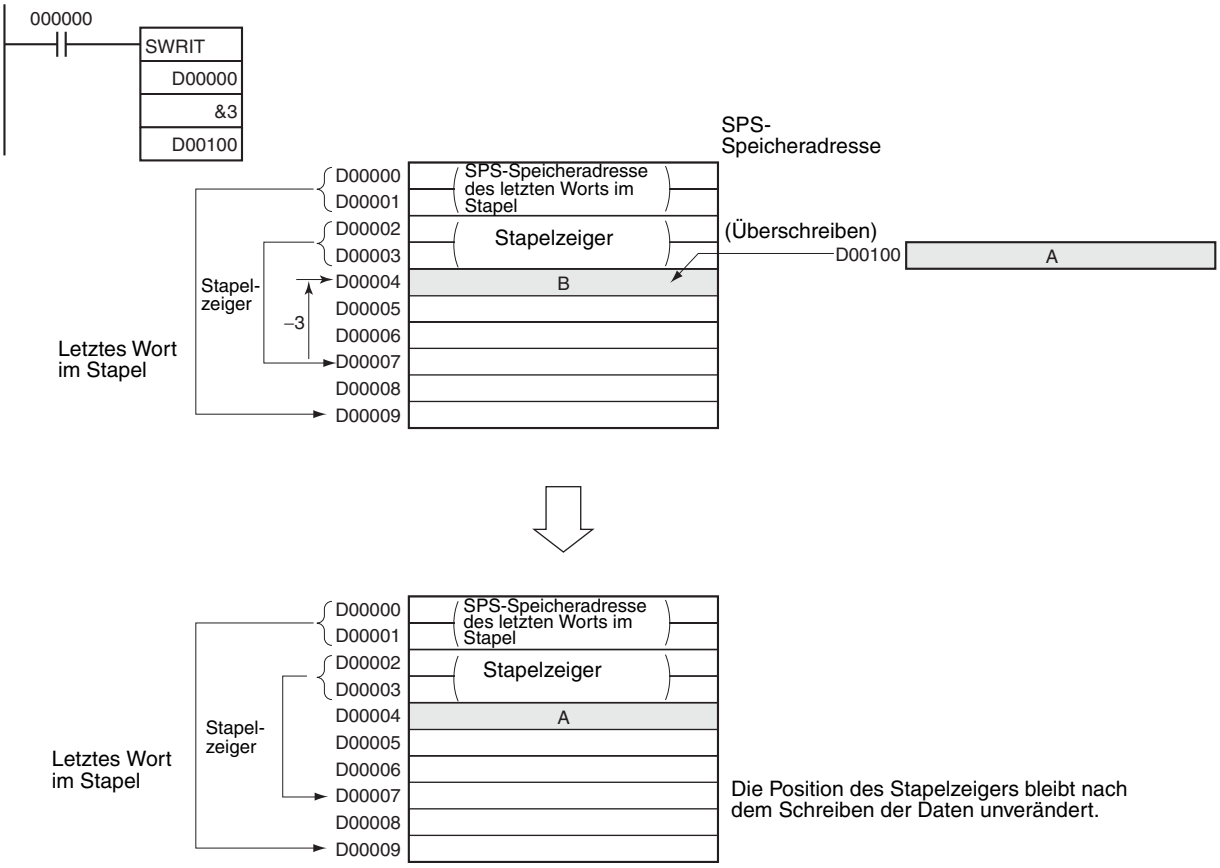
Der Stapel muss zuvor durch den Befehl SSET(630) definiert werden.

Die vom Stapelzeiger spezifizierte Adresse muss größer sein als die SPS-Speicheradresse, die den Anfang des Datenbereichs bildet (TB+4). Wenn der Wert des Stapelzeigers kleiner ist als die SPS-Speicheradresse von TB+4, tritt ein Stapelunterlauffehler auf.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, schreibt der Befehl SWRIT(640) die Daten in D00100 in das spezifizierte Wort des bei

D00000 beginnenden Stapels. In diesem Fall spezifiziert der Stapelzeiger die Adresse D00007 und der Offsetwert beträgt 3, sodass die Daten in D00004 überschrieben werden.



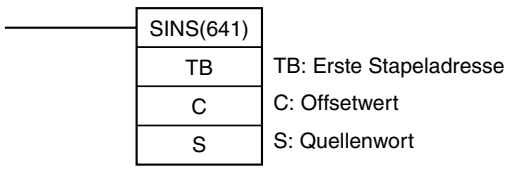
3-17-17 STACK DATA INSERT: SINS(641)

Zweck

Fügt die Quelldaten an der spezifizierten Stelle in den Stapel ein und verschiebt den Rest der Daten im Stapel nach unten. Der Offset-Wert bestimmt die Stelle des gewünschten Datenelements (Anzahl der Datenelemente vor der aktuellen Zeigerposition).

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SINS(641)
	Ausführung einmal bei steigender Flanke	@SINS(641)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

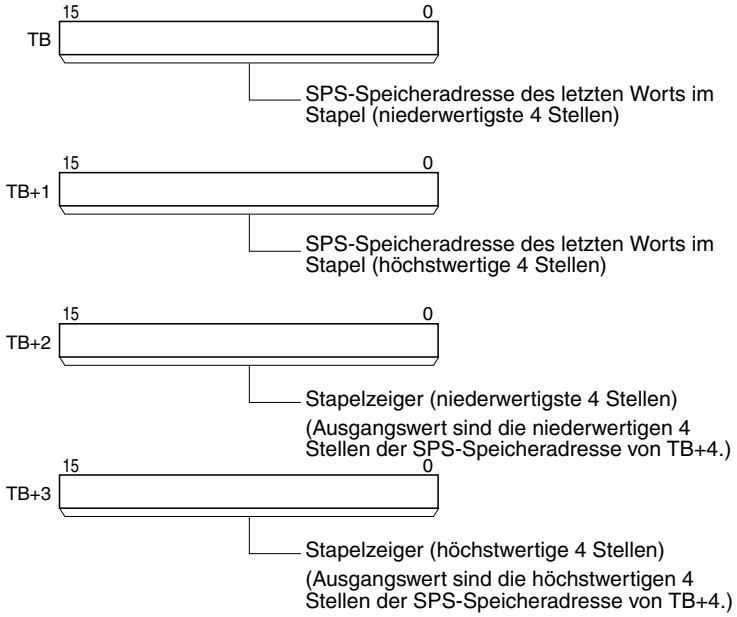
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

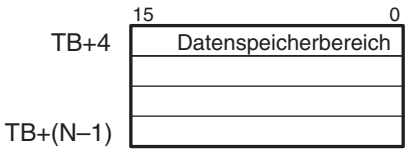
TB bis TB+3: Stapel-Steuerworte

Die ersten vier Worte des Stapels enthalten die SPS-Speicheradresse des letzten Worts im Stapel und den Stapelzeiger (die SPS-Speicheradresse des nächsten freien Worts im Stapel).



TB+4 bis TB+(N–1): Datenspeicherbereich

Der Rest des Stapels wird zum Speichern von Daten verwendet.



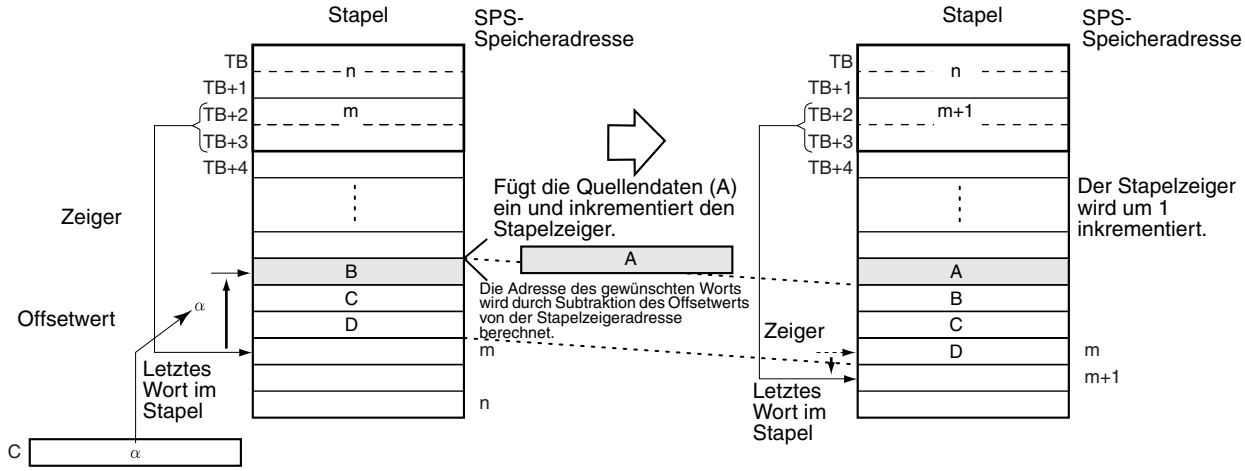
Operanden-Spezifikationen

Bereich	TB	C	S
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerkerbereich	H000 bis H511		
Zusatz-Systembereich	A448 bis A959	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		

Bereich	TB	C	S
Konstanten	---	#0001 bis #FFFB (hexadezimal)	#0000 bis #FFFF (hexadezimal)
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

SINS(641) fügt die Quellendaten an der gewünschten Adresse ein und verschiebt die vorhandenen Daten um ein Wort nach unten. Gleichzeitig inkrementiert SINS(641) den Stapelzeiger (TB+3 und TB+2) um 1. Die Position der gewünschten Adresse wird durch Subtraktion des Offsetwerts in C von der Position des Stapelzeigers berechnet.



SINS(641) kann verwendet werden, um Daten für ein Objekt einzufügen, das zwischen anderen Objekten in eine Förderanlage platziert wurde. Die Position der Einfügestelle ist einfach die Anzahl der Objekte (der Offsetwert) zurückgezählt, die vor dem neuesten Objekt in die Förderanlage gelangt sind.

Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn die durch den Spatelzeiger spezifizierte Adresse (TB+3 und TB+2) größer ist als die SPS-Speicheradresse des letzten Worts im Datenbereich des Stapels. (Es handelt sich um einen Stapel-Überlauffehler.) EIN, wenn der spezifizierte Offsetwert größer als die Maximalgröße des Datenbereichs -1 (FFFA hex) ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

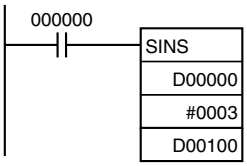
Der Stapel muss zuvor durch den Befehl SSET(630) definiert werden.

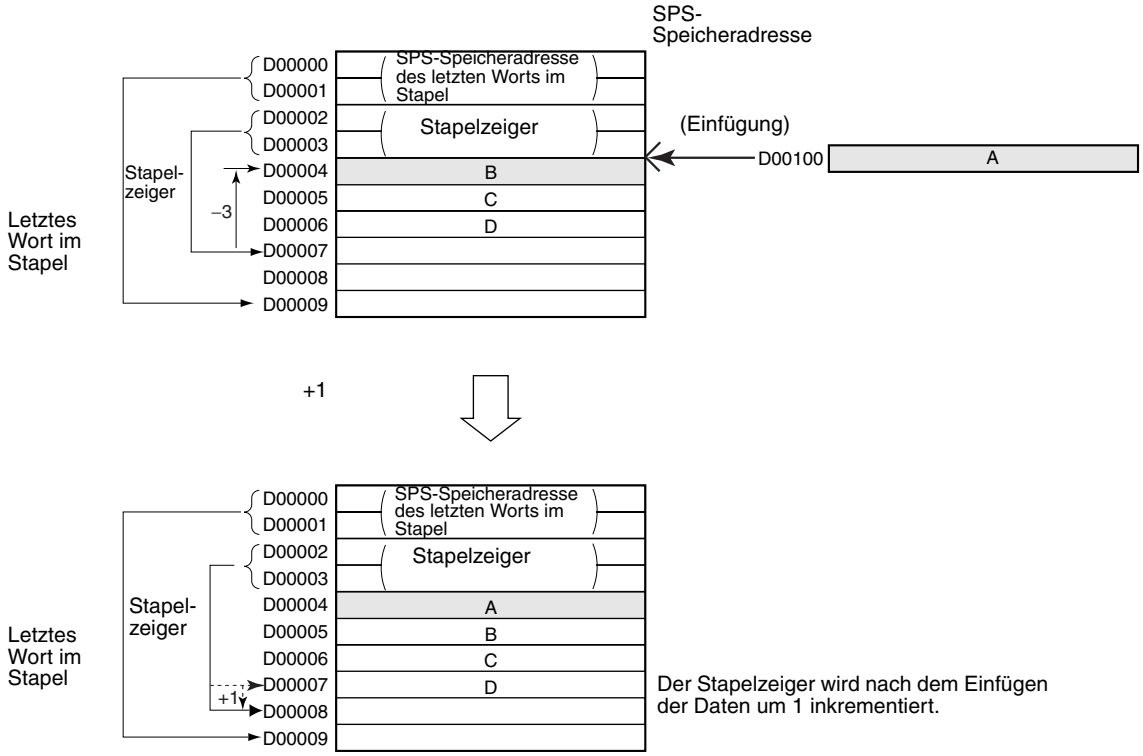
SINS(641) fügt ein Datenwort in den Stapel ein, daher muss mindestens ein freies Wort am Ende des Stapels vorhanden sein. Wenn der Stapel voll ist, tritt ein Fehler auf und die Quellendaten werden nicht eingefügt.

Wenn bei der Ausführung von SINS(641) die durch den Stapelzeiger spezifizierte Adresse (TB+3 und TB+2) bereits größer ist als die Adresse des letzten Worts im Stapel (TB+1 und TB), tritt ein Stapelüberlauffehler auf und die Quellendaten werden nicht eingefügt.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, fügt der Befehl SINS(641) die Daten in D00100 an der spezifizierten Adresse in den bei D00000 beginnenden Stapel ein. In diesem Fall spezifiziert der Stapelzeiger die Adresse D00007 und der Offsetwert beträgt 3, sodass die Quellendaten in D00004 eingefügt werden. Die vorhandenen Daten werden um ein Wort nach unten verschoben und die Daten in D00007 werden überschrieben. Gleichzeitig wird der Stapelzeiger von D00007 auf D00008 inkrementiert.





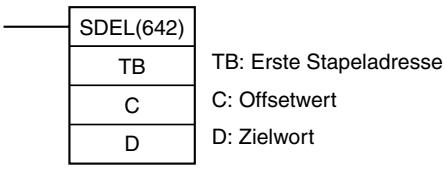
3-17-18 STACK DATA DELETE: SDEL(642)

Zweck

Löscht das Datenelement an der spezifizierten Stelle im Stapel, gibt diese Daten in das spezifizierte Zielwort aus und verschiebt die im Stapel verbleibenden Daten nach oben. Der Offset-Wert bestimmt die Stelle des gewünschten Datenelements (Anzahl der Datenelemente vor der aktuellen Zeigerposition).

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SDEL(642)
	Ausführung einmal bei steigender Flanke	@SDEL(642)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

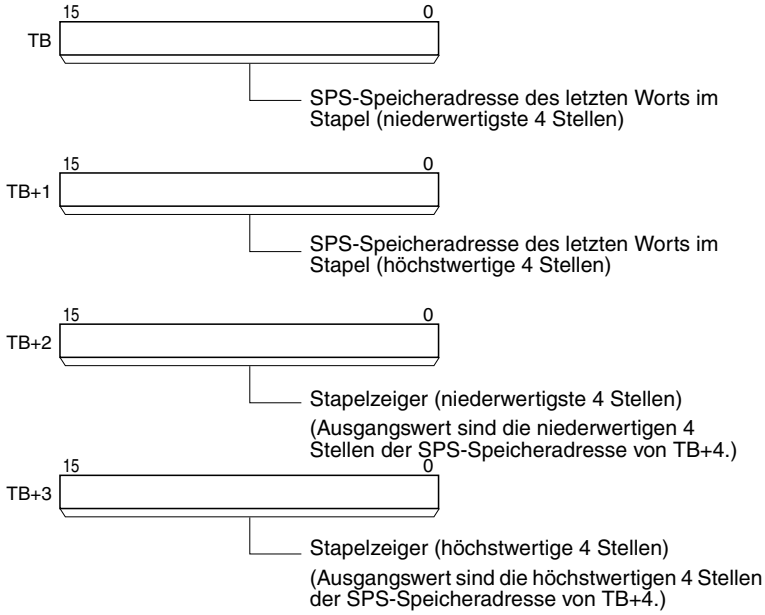
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

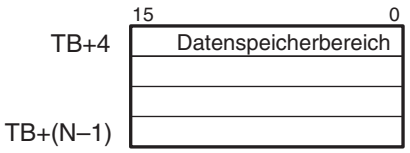
TB bis TB+3: Stapel-Steuerworte

Die ersten vier Worte des Stapels enthalten die SPS-Speicheradresse des letzten Worts im Stapel und den Stapelzeiger (die SPS-Speicheradresse des nächsten freien Worts im Stapel).



TB+4 bis TB+(N–1): Datenspeicherbereich

Der Rest des Stapels wird zum Speichern von Daten verwendet.



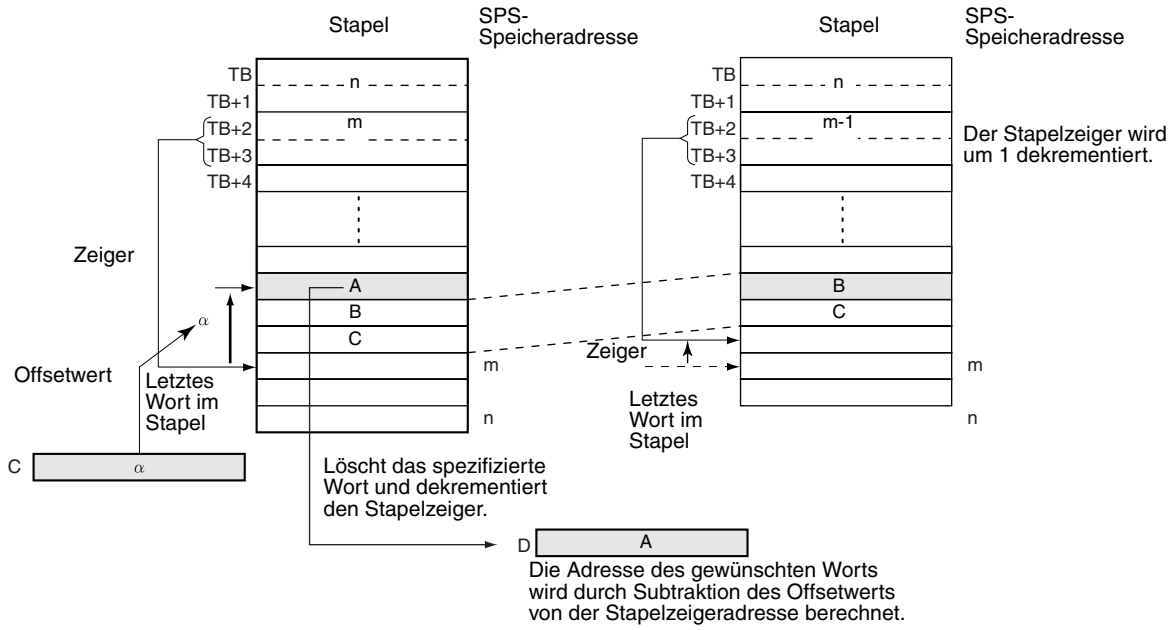
Operanden-Spezifikationen

Bereich	TB	C	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerkerbereich	H000 bis H511		
Zusatz-Systembereich	A448 bis A959	A000 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		

Bereich	TB	C	D
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0001 bis #FFFB (hexadezimal)	---
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

SDEL(642) löscht die Daten an der spezifizierten Stelle im Stapel, gibt diese Daten in das spezifizierte Zielwort aus und verschiebt die im Stapel verbleibenden Daten nach oben. Gleichzeitig dekrementiert SDEL(642) den Stapelzeiger (TB+3 und TB+2) um 1. Die Position der gewünschten Adresse wird durch Subtraktion des Offsetwerts in C von der Position des Stapelzeigers berechnet.



SDEL(642) kann verwendet werden, um Daten für ein Objekt zu löschen, das zwischen anderen Objekten aus einer Förderanlage entfernt wurde. Die Position der Löschstelle ist einfach die Anzahl der Objekte (der Offsetwert) zurückgezählt, die vor dem neuesten Objekt in die Förderanlage gelangt sind.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt des Stapelzeigers (TB+3 und TB+2) kleiner oder gleich der SPS-Speicheradresse des ersten Worts im Datenbereich des Stapels (TB+4) ist. (Es handelt sich um einen Stapel-Unterlauffehler.) EIN, wenn der in C spezifizierte Offsetwert 0 ist oder größer als die Maximalgröße des Datenbereichs (FFFB hex). In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn die in D ausgegebenen Daten 0000 sind. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

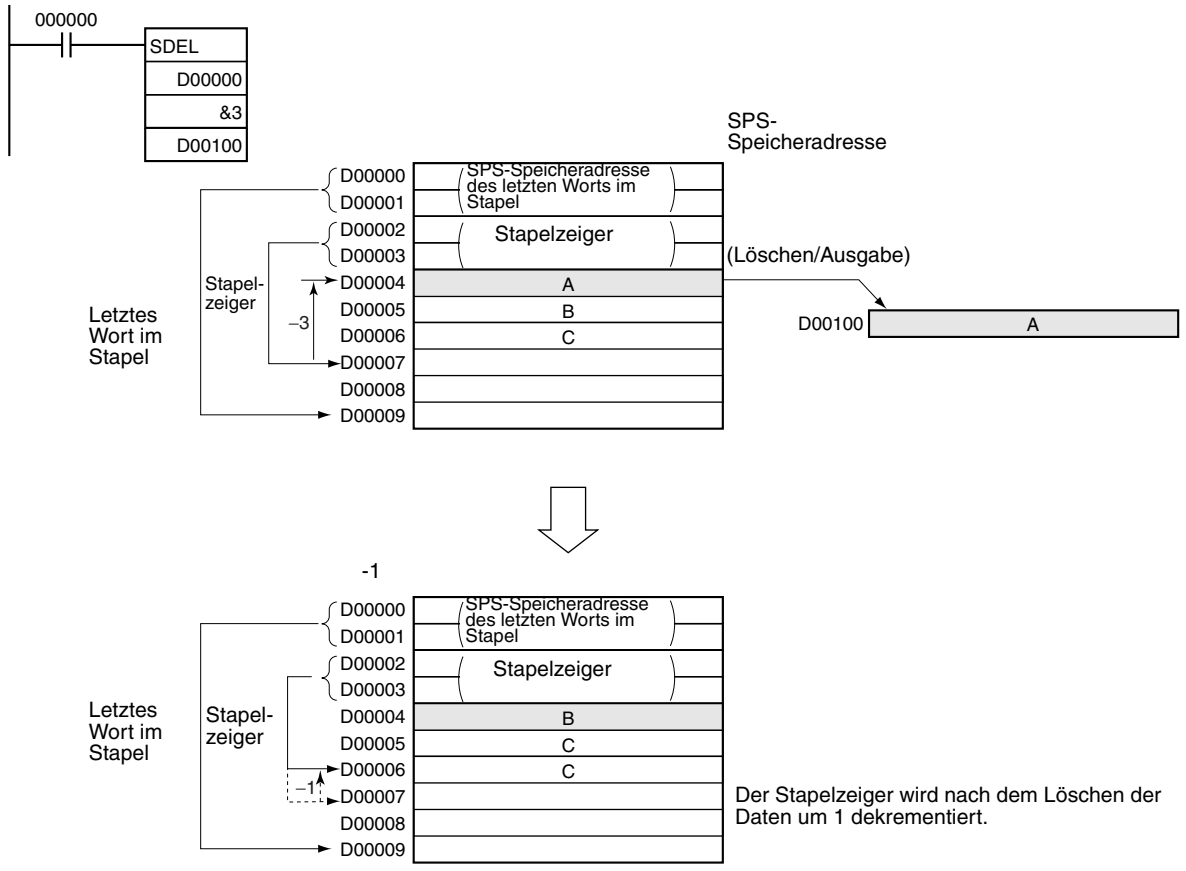
Der Stapel muss zuvor durch den Befehl SSET(630) definiert werden.

Die vom Stapelzeiger spezifizierte Adresse muss größer sein als die SPS-Speicheradresse, die den Anfang des Datenbereichs bildet (TB+4). Wenn der Wert des Stapelzeigers kleiner ist als die SPS-Speicheradresse von TB+4, tritt ein Stapelunterlauffehler auf.

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, löscht der Befehl SDEL(642) das Wort an der spezifizierten Adresse im mit D00000 beginnenden Stapel, gibt die gelöschten Daten in D00100 aus, verschiebt die verbleibenden Daten nach oben und dekrementiert den Stapelzeiger.

In diesem Fall spezifiziert der Stapelzeiger die Adresse D00007 und der Offsetwert beträgt 3, sodass die Daten aus D00004 gelöscht werden. Die verbleibenden Daten werden um ein Wort nach oben verschoben und der Stapelzeiger wird von D00007 auf D00006 dekrementiert.



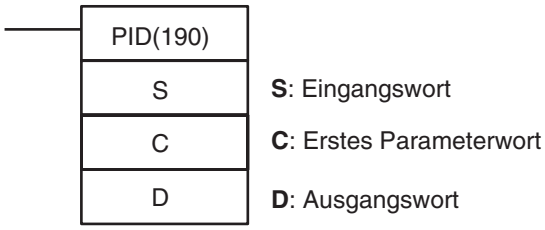
3-18 Datensteuerungsbefehle

3-18-1 PID CONTROL: PID(190)

Zweck

Führt eine PID-Regelung gemäß der spezifizierten Parameter aus.

Kontaktplansymbol



Variationen

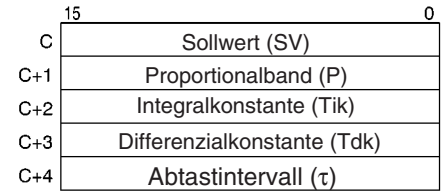
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	PID(190)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

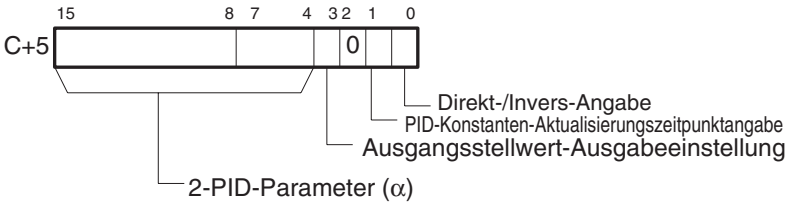
Anwendbarkeit auf Programmbereiche

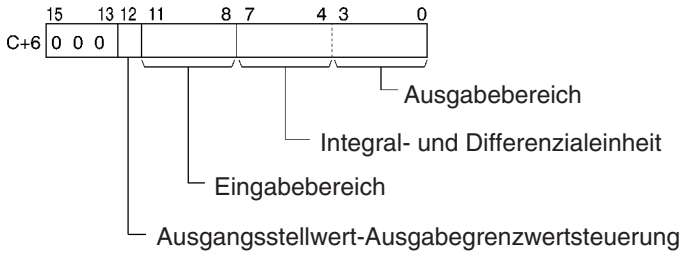
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Parameter

In den folgenden Abbildungen sind die Speicherorte der Parameterdaten dargestellt. Detaillierte Informationen zu den Parametern finden Sie unter *PID-Parametereinstellungen* in diesem Abschnitt.







Operanden-Spezifikationen

Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6105	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W473	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H473	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A921	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4057	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4057	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32729	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32729	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32729 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	DR0 bis DR15	---	DR0 bis DR15
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15		

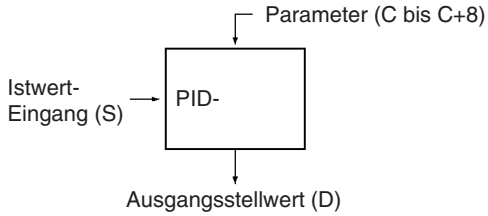
Beschreibung

Wenn die Ausführungsbedingung auf EIN gesetzt ist, führt der Befehl PID(190) eine zielwertgefilterte PID-Regelung mit zwei Freiheitsgraden gemäß der in C vorgegebenen Parameter (Sollwert, PID-Konstante usw.) aus. Dazu wird der spezifizierte Eingangsbereich von Binärdaten aus dem Eingangswort S gelesen und die PID-Verarbeitung gemäß der eingestellten Parameter durchgeführt. Das Ergebnis wird als Ausgangsstellwert im Ausgangswort D gespeichert.

Die Parameter werden gelesen, wenn der Status der Ausführungsbedingung von AUS zu EIN wechselt. Der Fehlermerker wird auf EIN gesetzt, wenn die Einstellungen außerhalb des zulässigen Bereichs liegen.

Wenn die Einstellungen innerhalb des zulässigen Bereich liegen, wird die PID-Verarbeitung unter Verwendung der Anfangswerte ausgeführt. Stoßfreier Betrieb erfolgt zu diesem Zeitpunkt nicht. Dieser wird auf Ausgangsstellwerte in der nachfolgenden Ausführung der PID-Verarbeitung angewandt. (Bei dem stoßfreien Betrieb handelt es sich um eine Verarbeitung, durch die der Ausgangsstellwert allmählich und kontinuierlich geändert wird, um die negativen Auswirkungen plötzlicher Änderungen zu vermeiden.)

Wenn die Ausführungsbedingung auf EIN gesetzt wird, wird der Istwert (PV) für das spezifizierte Abtastintervall eingegeben und die Verarbeitung wird durchgeführt.



Die Anzahl gültiger Eingangsdatenbits innerhalb der 16 Bits der Istwerteingabe (S) wird durch die Eingangsbereichseinstellung in C+6, Bits 08 bis 11 bestimmt. Wenn der Eingangsbereich z. B. auf 12 Bits (4 hex) festgelegt ist, kann der Istwert im Bereich zwischen 0000 hex und 0FFF hex liegen. (Werte, die größer als 0FFF hex sind, werden als 0FFF hex betrachtet.)

Der Sollwertbereich ist ebenfalls vom Eingangsbereich abhängig.

Bei Messwerten (Istwert, PV) und Sollwerten (SV) handelt es sich um vorzeichenlose Binärwerte zwischen 0000 hex und dem Maximalwert des Eingangsbereichs.

Die Anzahl gültiger Ausgangsdatenbits innerhalb der 16 Bits der Ausgangsstellwert-Ausgabe wird durch die Ausgangsbereichseinstellung in C+6, Bits 00 bis 03 bestimmt. Wenn der Ausgangsbereich z. B. auf 12 Bits (4 hex) festgelegt ist, kann der Ausgangsstellwert im Bereich zwischen 0000 hex und 0FFF hex ausgegeben werden.

Bei reinem Proportionalbetrieb kann die Ausgabe für den Ausgangsstellwert bei Gleichheit von Istwert (PV) und Sollwert (SV) wie folgt angegeben werden:

0: Ausgabe 0 %
1: Ausgabe 50 %

Als Regelrichtung für den Proportionalbetrieb kann entweder direkt oder invers festgelegt werden.

Für den ausgegebenen Ausgangsstellwert können Ober- und Untergrenzen festgelegt werden.

Das Abtastintervall kann in 10-ms-Einheiten (0,01 bis 99,99 s) festgelegt werden, das tatsächliche PID-Verhalten wird jedoch durch eine Kombination aus Abtastintervall und dem Zeitpunkt der Ausführung des Befehls PID(190) (in jedem Zyklus) bestimmt.

Für den Zeitpunkt, zu dem Änderungen an den PID-Konstanten möglich sind, gibt es zwei mögliche Einstellungen: 1) zu Beginn der Ausführung des PID-Befehls oder 2) zu Beginn der Ausführung des PID-Befehls und jedes Abtastintervall. Nur Proportionalband (P), Integralkonstante (Tik) und Differenzialkonstante (Tdk) können in jedem Abtastintervall (also während der Ausführung des PID-Befehls) geändert werden. Der Zeitpunkt wird in Bit 1 von C+5 spezifiziert.

Hinweis Die Einstellung in Bit 1 von C+5 wird nur von CPU-Baugruppen der SPS-Systeme CJ1, CS1-H und CJ1-H sowie von CS1-CPU-Baugruppen ab Lot-Nummer 001201□□□□ (Herstellungsdatum ab 1. Dezember 2000) unterstützt.

Von den PID-Parametern (C bis C+38) kann nur der Sollwert (SV) geändert werden, während die Ausführungsbedingung auf EIN gesetzt ist. Nach dem Ändern anderer Werte muss die Ausführungsbedingung von AUS zu EIN wechseln.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Daten für C nicht innerhalb des Bereichs liegen. EIN, wenn das tatsächliche Abtastintervall mehr als doppelt so lang wie das angegebene Abtastintervall ist. In allen anderen Fällen AUS.
Größer-als-Merker	>	EIN, wenn der Ausgangsstellwert nach PID-Verarbeitung die Obergrenze übersteigt. In allen anderen Fällen AUS.
Kleiner-als-Merker	<	EIN, wenn der Ausgangsstellwert nach PID-Verarbeitung unter der Untergrenze liegt. In allen anderen Fällen AUS.
Übertragsmerker	CY	EIN während der Ausführung der PID-Regelung. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

PID(190) wird ausgeführt, als wäre die Ausführungsbedingung ein STOP-RUN-Signal. PID-Berechnungen werden ausgeführt, wenn die Ausführungsbedingung während den nächsten Zyklus nach der Initialisierung von C+9 bis C+38 auf EIN gesetzt bleibt. Daher muss bei Verwendung des Immer-EIN-Merkers als Ausführungsbedingung für PID(190) ein separater Prozess eingerichtet werden, durch den C+9 bis C+38 zu Beginn des Betriebs initialisiert werden.

Wenn die Daten für C nicht innerhalb des Bereichs liegen, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Wenn das tatsächliche Abtastintervall mehr als doppelt so lang wie das angegebene Abtastintervall ist, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt. Dessen ungeachtet wird die PID-Regelung weiter ausgeführt.

Der Übertragsmerker wird während der Ausführung der PID-Regelung auf EIN gesetzt.

Der Größer-als-Merker wird auf EIN gesetzt, wenn der Ausgangsstellwert nach PID-Verarbeitung die Obergrenze übersteigt. In diesem Fall wird der obere Grenzwert als Ergebnis ausgegeben.

Der Kleiner-als-Merker wird auf EIN gesetzt, wenn der Ausgangsstellwert nach PID-Verarbeitung unter der Untergrenze liegt. In diesem Fall wird der untere Grenzwert als Ergebnis ausgegeben.

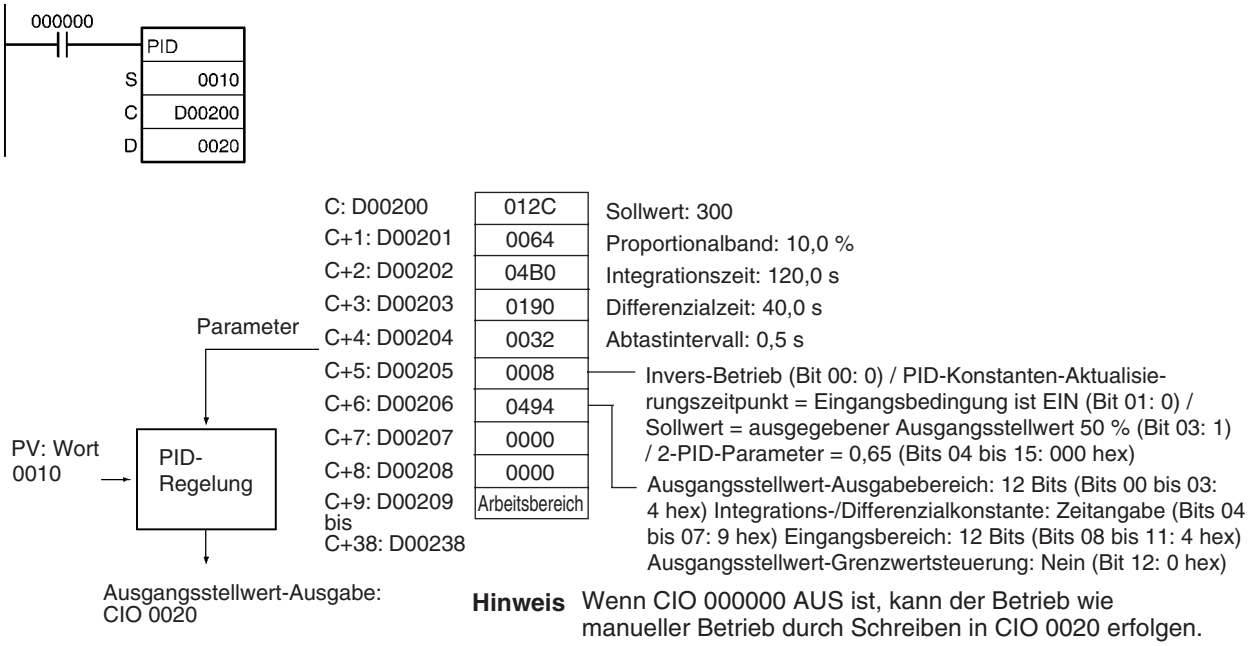
Von den PID-Parametern (C bis C+38) ist der Sollwert für C der einzige Wert, der geändert werden kann, während die Ausführungsbedingung auf EIN gesetzt ist. Bei Änderung beliebiger anderer Werte ist darauf zu achten, dass die Eingangsbedingung von AUS zu EIN wechselt, damit der neue Wert wirksam wird.

Beispiel

An der steigenden Flanke von CIO 000000 (AUS zu EIN) wird der Arbeitsbereich in D00209 bis D00238 entsprechend der in D00200 bis D00208 gesetzten Parameter (siehe unten) initialisiert. Nach der Initialisierung des Arbeitsbereichs wird die PID-Regelung ausgeführt und der Ausgangsstellwert wird in CIO 0020 ausgegeben.

Wenn CIO 000000 auf EIN gesetzt ist, wird die PID-Regelung in den Abtastintervallen entsprechend der in D00200 bis D00208 gesetzten Parameter ausgeführt. Der Ausgangsstellwert wird in CIO 0020 ausgegeben.

Die in den PID-Berechnungen verwendeten PID-Konstanten werden nicht geändert, wenn Proportionalband (P), Integralkonstante (Tik) oder Differenzialkonstante geändert werden, nachdem CIO 000000 auf EIN gesetzt ist.



Hinweis Wenn CIO 000000 AUS ist, kann der Betrieb wie manueller Betrieb durch Schreiben in CIO 0020 erfolgen.

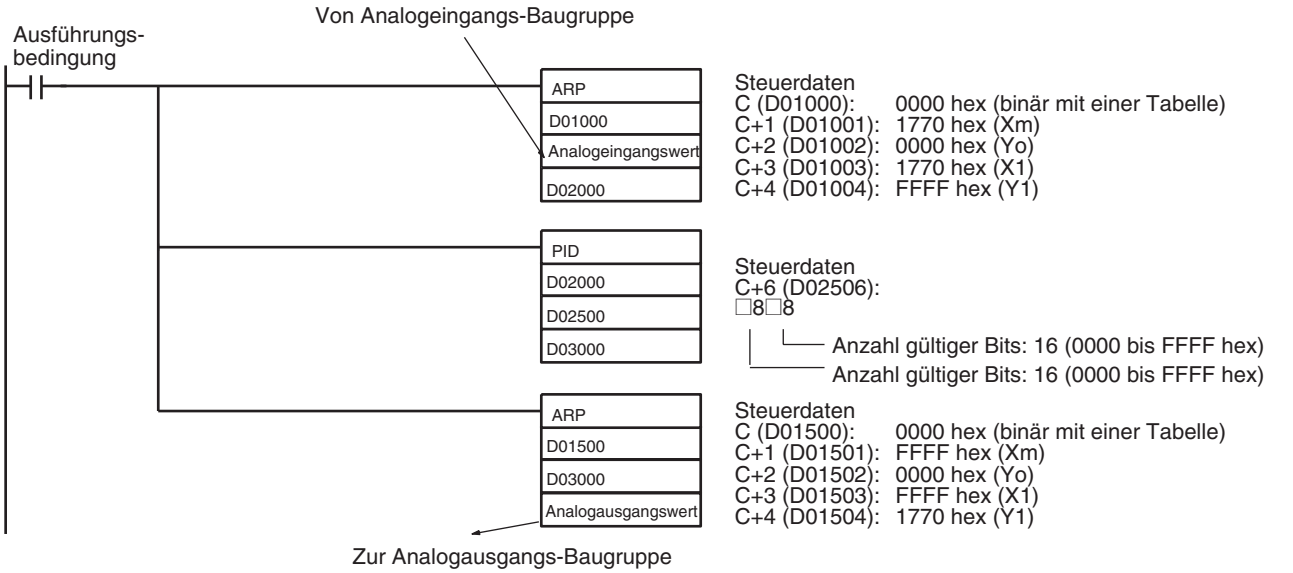
Bereiche für Eingangswerte und Ausgangsstellwert

Die Anzahl gültiger Eingangsdatenbits für den Messwert wird durch die Eingangsbereichseinstellung in C+6, Bits 08 bis 11, bestimmt. Die Anzahl gültiger Ausgangsdatenbits für den Ausgangsstellwert wird durch die Ausgangsbereichseinstellung in C+5, Bits 0 bis 3, bestimmt. Die Bereiche sind in der folgenden Tabelle aufgeführt.

C+6, Bits 08 bis 11 oder C+6, Bits 00 bis 03	Anzahl gültiger Bits	Bereich
0	8	0000 bis 00FF hex
1	9	0000 bis 01FF hex
2	10	0000 bis 03FF hex
3	11	0000 bis 07FF hex
4	12	0000 bis 0FFF hex
5	13	0000 bis 1FFF hex
6	14	0000 bis 3FFF hex
7	15	0000 bis 7FFF hex
8	16	0000 bis FFFF hex

Wenn der von einer Analogeingangsbaugruppe oder Analogausgangsbaugruppe verwendete Datenbereich durch Festlegung der Anzahl gültiger Bits nicht ordnungsgemäß eingestellt werden kann, kann der Befehl APR(069) (ARITHMETIC PROCESS) vor und nach PID(190) eingesetzt werden, um eine Konvertierung in die passenden Bereiche vorzunehmen.

Der folgende Programmabschnitt ist ein Beispiel für eine DRT1-AD04 Analogeingangsbaugruppe und eine DRT1-DA02 Analogausgangsbaugruppe, die als DeviceNet-Slaves arbeiten. Diese beiden Baugruppen nutzen einen Datenbereich von 0000 bis 1770 hex, der nicht durch bloßes Einstellen der Anzahl gültiger Stellen spezifiziert werden kann. Daher wird der Befehl APR(069) verwendet, um den Bereich von 0000 bis 1770 hex der Analogeingangsbaugruppe in 0000 bis FFFF hex als Eingang für PID(190) zu konvertieren. Dementsprechend wird der von PID(190) ausgegebene Ausgangsstellwert durch erneute Verwendung des Befehls APR(069) zurück in den Bereich von 0000 bis 1770 hex konvertiert, der als Ausgang der Analogausgangsbaugruppe dient.



Leistungsspezifikationen

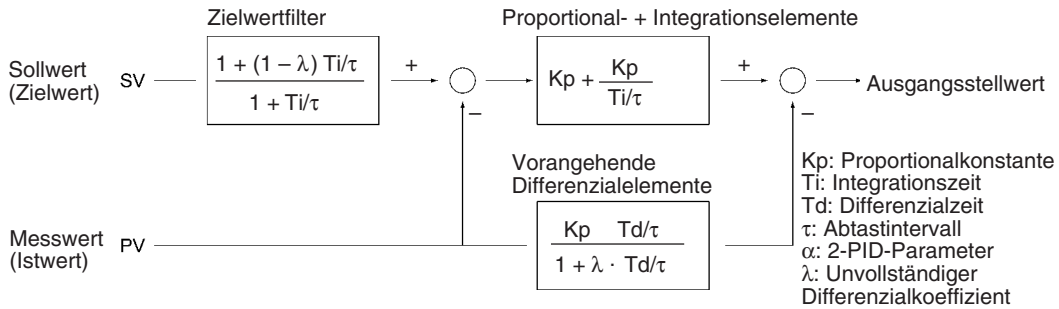
Eigenschaft			Beschreibung
PID-Regelungsart		---	PID-Regelung mit Zielwertfilterung und zwei Freiheitsgraden (direkt/invers)
Anzahl der PID-Regelkreise		---	Unbegrenzt (1 Regelkreis pro Befehl)
Abtastintervall		τ	0,01 bis 99,99 s
PID-Konstante	Proportionalband	P	0,1 bis 999,9 %
	Integralkonstante	Tik	1 bis 8191, 9999 (Keine Integrationsverhalten bei einem Vielfachen des Abtastintervalls, 9999.)
	Differenzialkonstante	Tdk	0 bis 8191 (Keine Differenzialverhalten bei einem Vielfachen des Abtastintervalls, 0.)
Sollwert		SV	0 bis 65535 (Gültig bis zum Maximalwert des Eingangsbereichs.)
Messwert		PV	0 bis 65535 (Gültig bis zum Maximalwert des Eingangsbereichs.)
Ausgangsstellwert		MV	0 bis 65535 (Gültig bis zum Maximalwert des Ausgangsbereichs.)

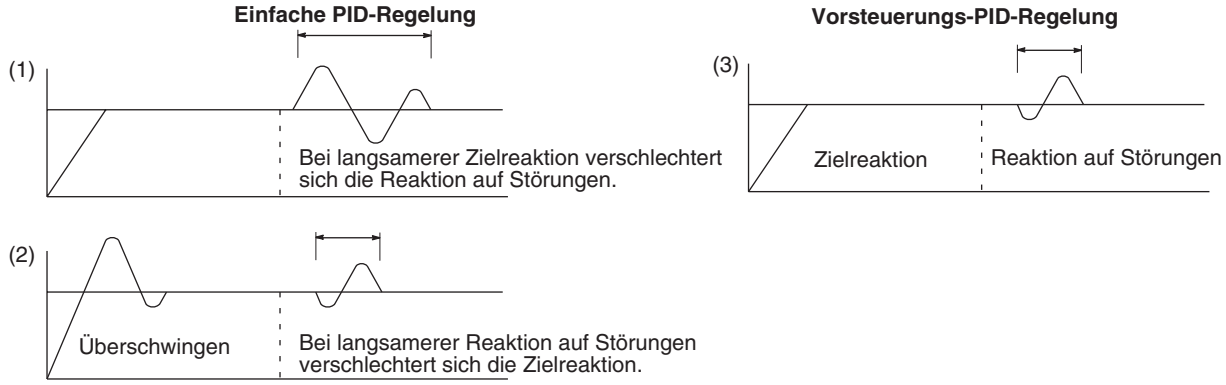
Berechnungsmethode Die Berechnungen bei der PID-Regelung erfolgen als Zielwertfilterungsregelung mit zwei Freiheitsgraden.

Blockdarstellung der Zielwert-PID-Regelung mit zwei Freiheitsgraden

Wenn ein Überschwingen durch einfache PID-Regelung vermieden wird, ist die Stabilisierung bei Störungen verlangsamt (1). Wenn die Stabilisierung bei Störungen beschleunigt wird, tritt hingegen ein Überschwingen auf und die Reaktion auf den Zielwert ist verlangsamt (2).

Wird hingegen die Zielwert-PID-Regelung mit zwei Freiheitsgraden verwendet, tritt kein Überschwingen auf und sowohl die Reaktion auf den Zielwert als auch die Stabilisierung bei Störungen können beschleunigt werden (3).





PID-Parameter-Einstellungen

Steuer-daten	Parameter	Inhalt	Einstellbereich	Änderung mit EIN-Eingangs-bedingung
C	Sollwert (SV)	Der Zielwert des geregelten Prozesses.	Binärdaten (die gleiche Anzahl von Bits, die auch für den Eingangsbereich spezifiziert ist)	Zulässig
C+1	Proportionalband	Der Parameter für P-Verhalten, Quotient aus Proportional-Regelbereich/Gesamtregel-bereich.	0001 bis 270F hex (1 bis 9999); (0,1 % bis 999,9 %, in Schritten von 0,1 %)	Kann mit EIN-Eingangs-bedingung geändert werden, wenn Bit 1 von C+5 auf 1 gesetzt ist.
C+2	Tik Integralkonstante	Eine Konstante als Maß für die Stärke des Integrations-verhaltens. Mit steigendem Wert nimmt die Integralstärke zu.	0001 bis 1FFF hex (1 bis 8191); (9999 = keine Ausführung der Integrationsfunktion) (siehe Hinweis 1)	
C+3	Tdk Differenzial-konstante	Eine Konstante als Maß für die Stärke des Differenzialverhaltens. Mit steigendem Wert nimmt die Differenzialstärke zu.	0001 bis 1FFF hex (1 bis 8191); (9999 = keine Ausführung der Differentialfunktion) (siehe Hinweis 1)	
C+4	Abtastintervall (τ)	Einstellung des Intervalls für die Ausführung der PID-Regelung.	0001 bis 270F hex (1 bis 9999); (0,01 bis 99,99 s, in Schritten von 10 ms)	Nicht zulässig
Bits 04 bis 15 von C+5	2-PID-Parameter (α)	Der Eingangsfilter-Koeffizient. Normalerweise wird eine Wert von 0,65 (d. h. eine Einstellung auf 000) verwendet. Die Effizienz des Filters steigt, wenn sich der Koeffizient 0 annähert.	000 hex: α = 0,65 Eine Einstellung zwischen 100 und 163 hex bedeutet eine Einstellung des Werts der niederwertigsten zwei Stellen zwischen α = 0,00 und α = 0,99. (siehe Hinweis 2)	
Bit 03 von C+5	Ausgangsstellwert-Ausgabegröße:	Legt den Ausgangsstellwert fest, der ausgegeben wird, wenn der Istwert dem Sollwert entspricht.	0: Ausgabe 0 % 1: Ausgabe 50 %	
Bit 01 von C+5	Einstellung für die Aktivierung von Änderungen der PID-Konstanten	Der Zeitpunkt für die Aktivierung von vorgenommenen Änderungen an Proportional-band (P), integralkonstante (Tik) und Differenzialkonstante (Tdk) zur Verwendung in PID-Berechnungen.	0: Bei Beginn der Ausführung des PID-Befehls 1: Bei Beginn der Ausführung des PID-Befehls und jedes Abtastintervall	Zulässig

Steuer-daten	Parameter	Inhalt	Einstellbereich	Änderung mit EIN-Eingangs-bedingung
Bit 00 von C+5	Direkt-/Invers-Spezifikation der PID-Regelung	Bestimmt die Richtung des Proportionalverhaltens.	0: Inverses Verhalten 1: Direktes Verhalten	Nicht zulässig
Bit 12 von C+6	Ausgangsstellwert-Ausgabegrenz-wertsteuerung	Legt fest, ob eine Grenzwert-steuerung auf die Ausgabe des Ausgangsstellwerts angewandt wird.	0: Deaktiviert (keine Grenzwertsteuerung) 1: Aktiviert (Grenzwertsteuerung)	
Bits 08 bis 11 von C+6	Eingangsbereich	Die Anzahl der Eingangsdatenbits	0: 8 Bits 5: 13 Bits 1: 9 Bits 6: 14 Bits 2: 10 Bits 7: 15 Bits 3: 11 Bits 8: 16 Bits 4: 12 Bits	
Bits 04 bis 07 von C+6	Integral- und Differenzialeinheit	Bestimmt die Einheit für die Angabe der Integral- und Differenzialkonstanten.	1: Vielfaches des Abtastintervalls 9: Zeit (Einheit: 100 ms)	
Bits 00 bis 03 von C+6	Ausgabebereich	Die Anzahl der Ausgangs-datenbits (Die Anzahl der Ausgangsdatenbits ist automatisch mit der der Eingangsdatenbits identisch.)	0: 8 Bits 5: 13 Bits 1: 9 Bits 6: 14 Bits 2: 10 Bits 7: 15 Bits 3: 11 Bits 8: 16 Bits 4: 12 Bits	
C+7	Ausgangsstellwert-Ausgabeunter-grenze	Die Untergrenze bei aktivierter Ausgangsstellwert-Ausgabebegrenzung.	0000 bis FFFF (binär) (siehe Hinweis 3)	
C +8	Ausgangsstellwert-Ausgabeobergrenze	Die Obergrenze bei aktivierter Ausgangsstellwert-Ausgabebegrenzung.	0000 bis FFFF (binär) (siehe Hinweis 3)	

Hinweis

1. Wird 1 für die Einheit spezifiziert, erstreckt sich der Bereich vom 1- bis 8.191-fachen des Intervalls. Wird 9 für die Einheit spezifiziert, erstreckt sich der Bereich vom 0,1 bis 819,1 s. Bei Spezifikation von 9 müssen Integrations- und Differenzialzeit auf einen Wert innerhalb des Bereichs vom 1- bis 8.191-fachen des Abtastintervalls eingestellt werden.
2. Die Einstellung des 2-PID-Parameters (α) auf 000 ergibt 0,65, den Normalwert.
3. Wenn die Ausgangsstellwert-Ausgabebegrenzung aktiviert ist (auf "1" gesetzt ist), müssen die Werte wie folgt eingestellt werden:

   0000 ≤ Ausgangsstellwert-Ausgabeuntergrenze ≤ Ausgangsstellwert-Ausgabeobergrenze ≤ Maximalwert des Ausgangsbereichs

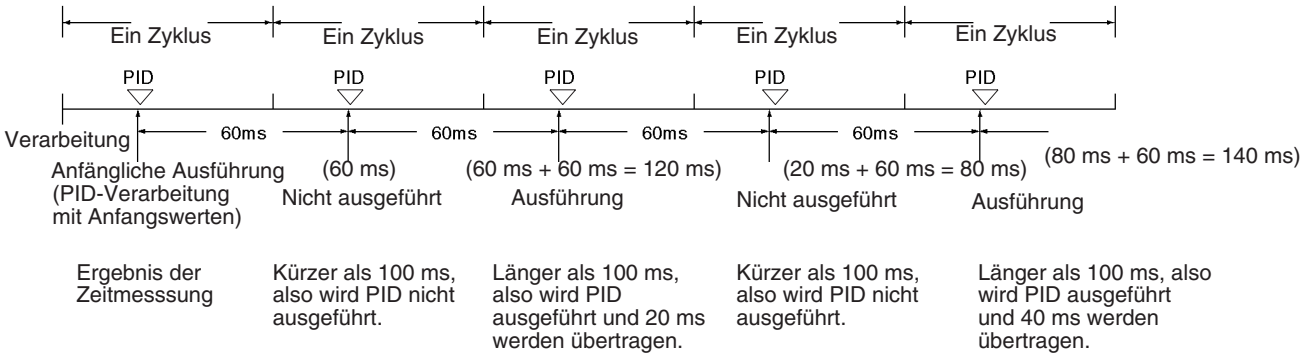
Abtastintervall und Zykluszeit

Das Abtastintervall kann in 10-ms-Einheiten (0,01 bis 99,99 s) festgelegt werden, das tatsächliche PID-Verhalten wird jedoch durch eine Kombination aus Abtastintervall und dem Zeitpunkt der Ausführung des PID-Befehls (in jedem Zyklus) bestimmt. Abtastintervall und Zykluszeit stehen in folgender Beziehung zueinander:

- Wenn das Abtastintervall kürzer als die Zykluszeit ist, wird die PID-Regelung in jedem Zyklus ausgeführt, und nicht mit jedem Abtastintervall.
- Wenn das Abtastintervall länger oder genauso lang wie die Zykluszeit ist, wird die PID-Regelung nicht in jedem Zyklus ausgeführt, der Befehl PID(190) wird jedoch ausgeführt, wenn der kumulierte Wert der Zykluszeit (die Zeit zwischen PID-Befehlen) länger oder genauso lang wie das Abtastintervall ist. Der Überschussanteil des kumulierten Zeitwerts (der kumulierte Wert der Zykluszeit abzüglich des Abtastintervalls) wird in den nächsten kumulierten Zeitwert übernommen.

  Ein Beispiel: Angenommen das Abtastintervall hat eine Länge von 100 ms und die Zykluszeit beträgt durchweg 60 ms. Im ersten Zyklus nach erstmaliger Ausführung wird PID(190) nicht ausgeführt, das 60 ms kürzer sind als 100 ms. Im zweiten Zyklus sind 60 ms + 60 ms länger als 100 ms, also wird PID(190) ausgeführt. Der Überschuss von 20 ms (120 ms – 100 ms = 20 ms) wird übernommen.

Im dritten Zyklus wird der Überschuss von 20 ms zu den 60 ms Zykluszeit addiert. Da die Summe von 80 ms kürzer als 100 ms ist, wird PID(190) nicht ausgeführt. Im vierten Zyklus werden die 80 ms zu den 60 ms Zykluszeit addiert. Da die Summe von 140 ms länger als 100 ms ist, wird PID(190) ausgeführt und der Überschuss von 40 ms (140 ms – 100 ms = 40 ms) wird übernommen. Dieses Verfahren wiederholt sich in allen folgenden Zyklen.



Regelverhalten

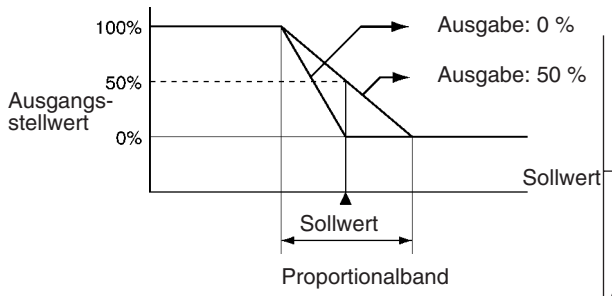
Proportionalverhalten (P)

Proportionalverhalten ist eine Funktionsweise, bei der ein Proportionalband im Verhältnis zum Sollwert eingeführt wird. Innerhalb dieses Bandes verhält sich der Ausgangsstellwert proportional zur Abweichung. In der folgenden Abbildung ist ein Beispiel für Inversfunktion dargestellt.

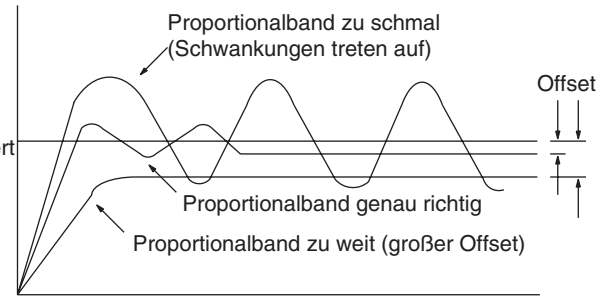
Wenn das Proportionalverhalten verwendet wird und der Istwert kleiner als das Proportionalband wird, hat der Ausgangsstellwert einen Wert von 100 % (den Maximalwert). Innerhalb des Proportionalbands ist die Größe des Ausgangsstellwerts proportional zur Abweichung (der Differenz zwischen Sollwert und Istwert) und wird allmählich verringert, bis Sollwert und Istwert übereinstimmen (also bis die Abweichung 0 beträgt). An diesem Punkt hat der Ausgangsstellwert seinen Minimalwert von 0 % (bzw. 50 %, je nach Einstellung des Parameters für die Ausgangsstellwert-Ausgabegröße). Der Ausgangsstellwert beträgt ebenfalls 0 %, wenn der Istwert über dem Sollwert liegt.

Die Größe des Proportionalbands wird als Prozentsatz des Gesamteingangsbereichs angegeben. Je kleiner das Proportionalband ist, umso größer ist die Proportionalkonstante und umso stärker fällt die Korrekturreaktion aus. Beim Proportionalverhalten tritt allgemein ein Offset (Restabweichung) auf, dieser Offset kann jedoch zur reduziert werden, indem das Proportionalband kleiner gewählt wird. Wird das Proportionalband zu klein, treten Schwankungen auf.

Proportionalverhalten (Inverses Verhalten)



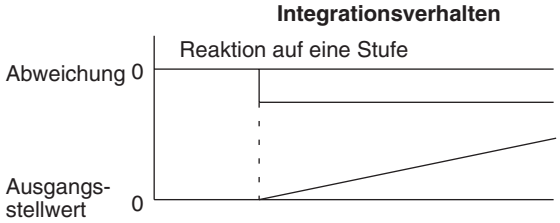
Anpassung des Proportionalbands

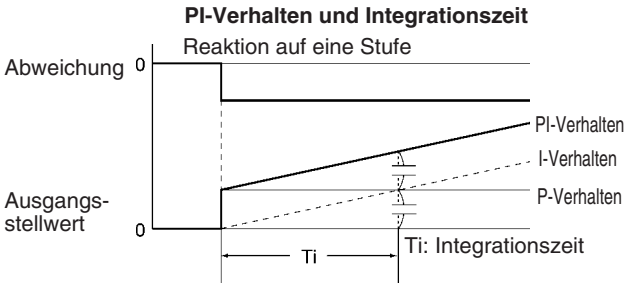


Integrationsverhalten (I)

Die Kombination des Integrationsverhaltens mit dem Proportionalverhalten führt zu Reduzierung des Offsets entsprechend der verstrichenen Zeit, sodass der Istwert mit dem Sollwert übereinstimmt. Die Stärke des Integrationsverhaltens wird durch die Integrationszeit bestimmt, wobei es sich um die Zeit handelt, die der Ausgangsstellwert des Integralverhaltens benötigt, um den gleichen Wert wie der Ausgangsstellwert des Proportionalverhaltens in

Beziehung auf die Stufenabweichung zu erreichen (siehe folgende Abbildung). Je kürzer die Integrationszeit ist, umso stärker ist die Korrektur durch das Integrationsverhalten. Wenn die Integrationszeit zu kurz ist, ist die Korrektur zu stark und es treten Schwankungen auf.

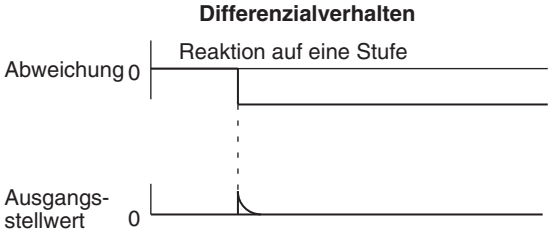


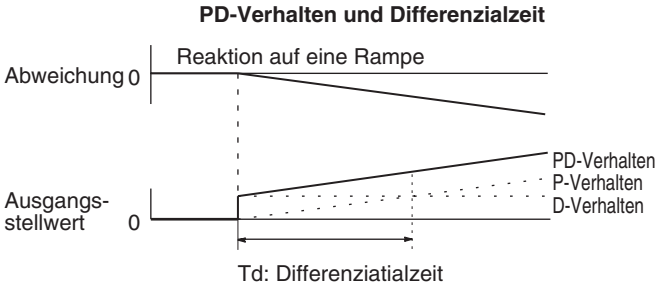


Differenzialverhalten (D)

Durch Proportionalverhalten und Integrationsverhalten erfolgen Änderungen in Bezug auf die Regelungsergebnisse, sodass es zwangsläufig zu einer Reaktionsverzögerung kommt. Das Differenzialverhalten kann dieses Manko kompensieren. Als Reaktion auf eine plötzliche Störung bietet es einen großen Ausgangsstellwert und kann so den ursprünglichen Status schnell wieder herstellen. Die Korrektur erfolgt, indem sich der Ausgangsstellwert proportional zur Steigung (Abweichungskoeffizient) verhält, die durch die Abweichung verursacht wird.

Die Stärke des Differenzialverhaltens wird durch die Differenzialzeit bestimmt, wobei es sich um die Zeit handelt, die der Ausgangsstellwert des Differenzialverhaltens benötigt, um den gleichen Wert wie der Ausgangsstellwert des Proportionalverhaltens in Beziehung auf die Stufenabweichung zu erreichen (siehe folgende Abbildung). Je länger die Integrationszeit ist, umso stärker ist die Korrektur durch das Differenzialverhalten.



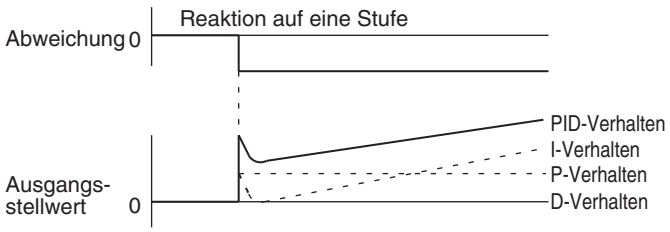


PID-Verhalten

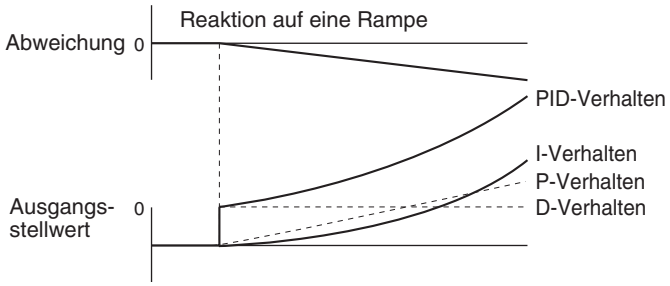
Das PID-Verhalten kombiniert Proportionalverhalten (P), Integrationsverhalten (I) und Differenzialverhalten (D). Es führt zu überlegenen Regelungsergebnissen, selbst bei Regelstrecken mit Totzeit. Es setzt das Proportional-

verhalten ein, um einen ruckfreie Regelung ohne Überschwingen zu leisten, das Integrationsverhalten, um Offsets automatische zu korrigieren, und das Differenzialverhalten, um eine schnelle Reaktion auf Störungen zu bieten.

Reaktion des PID-Regelausgangs auf eine Stufe



Reaktion des PID-Regelausgangs auf eine Rampe

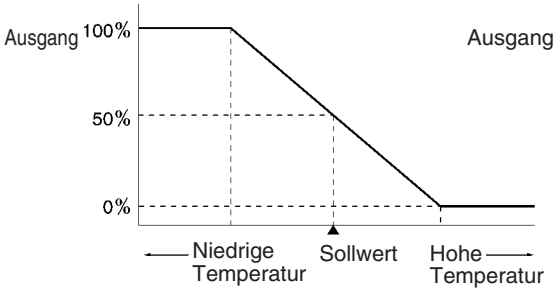


Richtung der Regelung

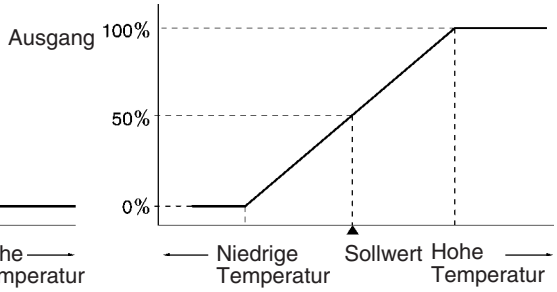
Bei Verwendung der PID-Regelung können Sie zwischen den zwei folgenden Regelrichtungen wählen. In beiden Richtungen wird der Ausgangsstellwert vergrößert, wenn die Differenz zwischen Sollwert und Istwert zunimmt.

- Direktes Verhalten: Der Ausgangsstellwert wird vergrößert, wenn der Istwert größer als der Sollwert ist.
- Inverses Verhalten: Der Ausgangsstellwert wird vergrößert, wenn der Istwert kleiner als der Sollwert ist.

Inverses Verhalten



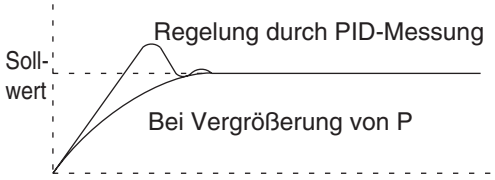
Direktes Verhalten



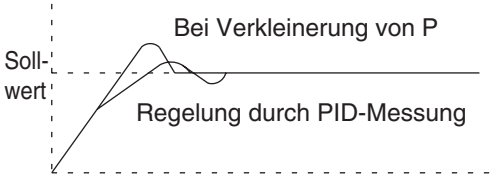
Anpassung der PID-Parameter

Nachfolgend ist die allgemeine Beziehung zwischen den PID-Parametern und dem Status der Regelung erläutert.

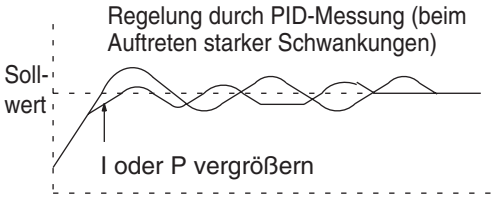
- Wenn es kein Problem darstellt, dass eine gewisse Zeit für die Stabilisierung (Einregelzeit) erforderlich ist, es hingegen wichtig ist, dass kein Überschwingen verursacht wird, muss das Proportionalband vergrößert werden.



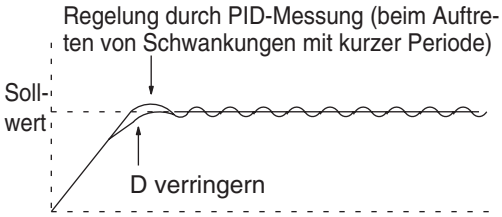
- Wenn ein Überschwingen nicht problematisch ist aber eine schnelle Stabilisierung der Regelung gewünscht wird, kann das Proportionalband verkleinert werden. Wenn das Proportionalband jedoch zu sehr verkleinert wird, kann dies zu Schwankungen führen.



- Wenn starke Schwankungen auftreten oder der Betrieb zwischen Überschwingen und Unterschwingen wechselt, kann die Ursache dafür ein zu starkes Integrationsverhalten sein. Die Schwankungen lassen sich reduzieren, indem die Integrationszeit verlängert und das Proportional-band erweitert wird.



- Wenn Schwankungen mit kurzer Periode auftreten, kann es sein, dass das Regelsystem zu stark reagiert, da das Differenzialverhalten zu stark ist. Stellen Sie das Differenzialverhalten in diesen Fall schwächer ein.



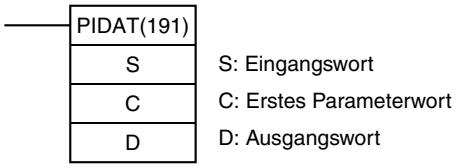
3-18-2 PID CONTROL WITH AUTOTUNING: PIDAT(191)

Zweck

Führt eine PID-Regelung gemäß der spezifizierten Parameter aus. Die PID-Konstanten können durch Autotuning eingestellt werden.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

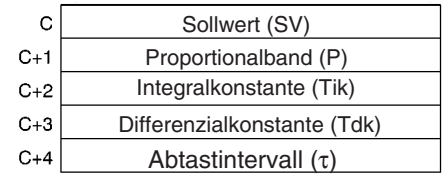
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	PIDAT(191)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

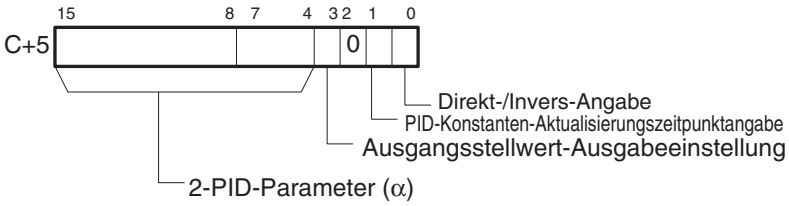
Anwendbarkeit auf Programmbereiche

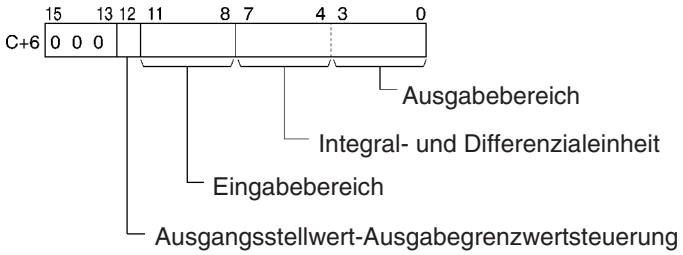
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

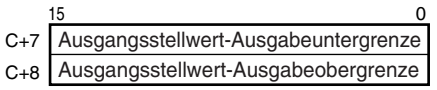
Parameter

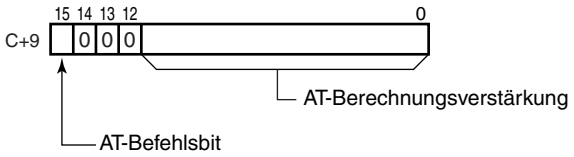
In den folgenden Abbildungen sind die Speicherorte der Parameterdaten dargestellt. Detaillierte Informationen zu den Parametern finden Sie unter *PID-Parametereinstellungen* in diesem Abschnitt.

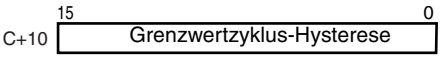


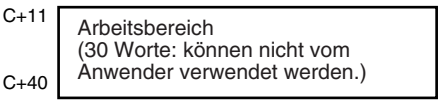












Operanden-Spezifikationen

Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6105	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W473	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H473	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A921	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4057	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4057	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32729	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32729	E00000 bis E32767

Bereich	S	C	D
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32729 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	DR0 bis DR15	---	DR0 bis DR15
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15		

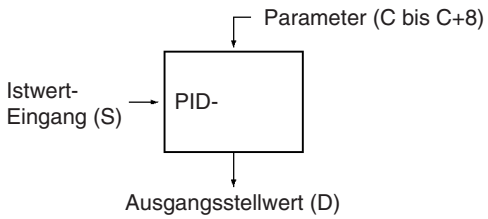
Beschreibung

Wenn die Ausführungsbedingung auf EIN gesetzt ist, führt der Befehl PIDAT(191) eine zielwertgefilterte PID-Regelung mit zwei Freiheitsgraden gemäß der in C vorgegebenen Parameter (Sollwert, PID-Konstante usw.) aus. Dazu wird der spezifizierte Eingangsbereich von Binärdaten aus dem Eingangswort S gelesen und die PID-Regelung gemäß der eingestellten Parameter durchgeführt. Das Ergebnis wird als Ausgangsstellwert im Ausgangswort D gespeichert.

Die Parametereinstellungen werden gelesen, wenn der Status der Ausführungsbedingung von AUS zu EIN wechselt. Der Fehlermerker wird auf EIN gesetzt, wenn die Einstellungen außerhalb des zulässigen Bereichs liegen.

Wenn die Einstellungen innerhalb des zulässigen Bereich liegen, wird die PID-Verarbeitung unter Verwendung der Anfangswerte ausgeführt. Stoßfreier Betrieb erfolgt zu diesem Zeitpunkt nicht. Dieser wird auf Ausgangsstellwerte in der nachfolgenden Ausführung der PID-Verarbeitung angewandt. (Bei dem stoßfreien Betrieb handelt es sich um eine Verarbeitung, durch die der Ausgangsstellwert allmählich und kontinuierlich geändert wird, um die negativen Auswirkungen plötzlicher Änderungen zu vermeiden.)

Wenn die Ausführungsbedingung auf EIN gesetzt wird, wird der Istwert (PV) für das spezifizierte Abtastintervall eingegeben und die Verarbeitung wird durchgeführt.



Autotuning

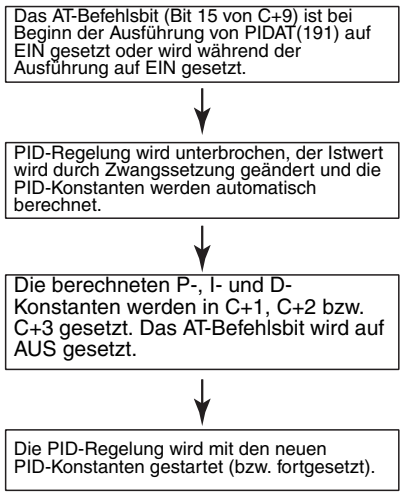
Der Status des AT-Befehlsbits (Bit 15 von C+9) wird in jedem Zyklus geprüft. Wenn dieses Steuerbit in einem Zyklus auf EIN gesetzt ist, beginnt der Befehl PIDAT(191) mit dem Autotuning der PID-Konstanten. (Während der Durchführung des Autotunings werden Änderungen des Sollwerts nicht berücksichtigt.)

Für das Autotuning wird die Grenzwertzyklus-Methode verwendet. PIDAT(191) führt zwangsweise Änderungen des Ausgangsstellwerts durch (max. Ausgangsstellwert↔ min. Ausgangsstellwert) und überwacht dabei die Eigenschaften der Regelstrecke. Die PID-Konstanten werden anhand der ermittelten Eigenschaften berechnet, und neue P-, I- und D-Konstanten wer-

den automatische in C+1, C+2 und C+3 gespeichert. Ist dies erfolgt, wird das AT-Befehlsbit (Bit 15 von C+9) auf AUS gesetzt und die PID-Regelung wird mit den neuen PID-Konstanten in C+1, C+2 und C+3 fortgesetzt.

- Wenn das AT-Befehlsbit bei Beginn der Ausführung von PIDAT(191) auf EIN gesetzt ist, wird zunächst ein Autotuning durchgeführt und anschließend wird die PID-Regelung mit den berechneten PID-Konstanten gestartet.
- Wenn das AT-Befehlsbit während der Ausführung von PIDAT(191) auf EIN gesetzt wird, unterbricht PIDAT(191) die mit den vom Anwender eingestellten PID-Konstanten ausgeführte PID-Regelung, führt ein Autotuning aus und die PID-Regelung mit den berechneten PID-Konstanten fort.

Das folgende Ablaufschema zeigt das Autotuning-Verfahren:



Hinweis

1. Wenn das Autotuning während der Ausführung unterbrochen wird, indem das AT-Befehlsbit auf AUS gesetzt wird, startet die PID-Regelung mit den vor Beginn des Autotunings verwendeten PID-Konstanten.
2. Auch beim Auftreten eines Fehlers bei der Autotuning-Ausführung startet die PID-Regelung mit den vor Beginn des Autotunings verwendeten PID-Konstanten.

Bei den beiden in den Hinweisen 2 und 2 beschriebenen Fällen werden jedoch die neuen die PID-Konstanten aktiviert, wenn sie zum Zeitpunkt der Unterbrechung bereits berechnet waren.

PID-Regelung

Die Anzahl gültiger Eingangsdatenbits innerhalb der 16 Bits der Istwerteingabe (S) wird durch die Eingangsbereichseinstellung in C+6, Bits 08 bis 11 bestimmt. Wenn der Eingangsbereich z. B. auf 12 Bits (4 hex) festgelegt ist, kann der Istwert im Bereich zwischen 0000 hex und 0FFF hex liegen. (Werte, die größer als 0FFF hex sind, werden als 0FFF hex betrachtet.)

Der Sollwertbereich ist ebenfalls vom Eingangsbereich abhängig.

Bei Messwerten (Istwert, PV) und Sollwerten (SV) handelt es sich um vorzeichenlose Binärwerte zwischen 0000 hex und dem Maximalwert des Eingangsbereichs.

Die Anzahl gültiger Ausgangsdatenbits innerhalb der 16 Bits der Ausgangsstellwert-Ausgabe wird durch die Ausgangsbereichseinstellung in C+6, Bits 00 bis 03 bestimmt. Wenn der Ausgangsbereich z. B. auf 12 Bits (4 hex) festgelegt ist, kann der Ausgangsstellwert im Bereich zwischen 0000 hex und 0FFF hex ausgegeben werden.

Bei reinem Proportionalbetrieb kann die Ausgabe für den Ausgangsstellwert bei Gleichheit von Istwert (PV) und Sollwert (SV) wie folgt spezifiziert werden:

0: Ausgabe 0 %
1: Ausgabe 50 %

Als Regelrichtung für den Proportionalbetrieb kann entweder direkt oder invers festgelegt werden.

Für den ausgegebenen Ausgangsstellwert können Ober- und Untergrenzen festgelegt werden.

Das Abtastintervall kann in 10-ms-Einheiten (0,01 bis 99,99 s) festgelegt werden, die tatsächliche PID-Funktion wird jedoch durch eine Kombination aus Abtastintervall und dem Zeitpunkt der Ausführung des Befehls PIDAT(191) (in jedem Zyklus) bestimmt.

Für den Zeitpunkt, zu dem Änderungen an den PID-Konstanten möglich sind, gibt es zwei mögliche Einstellungen: 1) zu Beginn der Ausführung des Befehls PIDAT(191) oder 2) zu Beginn der Ausführung des PID-Befehls und jedes Abtastintervall. Nur Proportionalband (P), Integralkonstante (Tik) und Differenzialkonstante (Tdk) können in jedem Abtastintervall (also während der Ausführung des PID-Befehls) geändert werden. Der Zeitpunkt wird in Bit 1 von C+5 spezifiziert.

Bei manueller Änderung der PID-Konstanten muss die Einstellung für die Aktivierung von Änderungen der PID-Konstanten (Bit 1 von C+5) auf 1 gesetzt werden, damit die Werte in C+1, C+2 und C+3 für die PID-Berechnung in jedem Abtastintervall aktualisiert werden. Diese Einstellung ermöglicht auch eine manuelle Anpassung der PID-Konstanten nach einem Autotuning.

Von den PID-Parametern (C bis C+38) können nur die folgenden Parameter geändert werden, während die Ausführungsbedingung auf EIN gesetzt ist. Bei Änderung beliebiger anderer Werte ist darauf zu achten, dass ein Wechsel der Ausführungsbedingung von AUS zu EIN stattfindet, damit die neuen Einstellungen wirksam werden.

- Sollwert in C
  (Kann nur während der PID-Regelung geändert werden. Eine Änderung des Sollwerts während des Autotunings wird nicht übernommen.)
- Einstellung für die Aktivierung von Änderungen der PID-Konstanten (Bit 1 von C+5)
- P-, I- und D-Konstanten in C+1, C+2 und C+3
  (Änderungen an diesen Konstanten werden nur in jedem Abtastintervall übernommen, wenn die Einstellung für die Aktivierung von Änderungen der PID-Konstanten (Bit 1 von C+5) auf 1 gesetzt ist.)
- AT-Befehlsbit (Bit 15 von C+9)
- AT-Berechnungsverstärkung (Bits 0 bis 14 von C+9) und Grenzwertzyklus-Hysterese (C+10) (Dise Werte werden beim Start des Autotunings gelesen.)

Hinweis Der Befehl PIDAT(191) ist mit dem Befehl PID(190) identisch, bietet jedoch eine zusätzliche Autotuning-Funktion (AT), d. h., die PID-Regelung selbst wird von beiden Befehle identisch ausgeführt. Im Abschnitt *3-18-1 PID CONTROL: PID(190)* finden Sie weitere Informationen und Beispiele zur PID-Regelung.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Daten für C nicht innerhalb des Bereichs liegen. EIN, wenn das tatsächliche Abtastintervall mehr als doppelt so lang wie das angegebene Abtastintervall ist. EIN, wenn während des Autotunings ein Fehler aufgetreten ist. In allen anderen Fällen AUS.
Größer-als-Merker	>	EIN, wenn der Ausgangsstellwert nach PID-Verarbeitung die Obergrenze übersteigt. In allen anderen Fällen AUS.
Kleiner-als-Merker	<	EIN, wenn der Ausgangsstellwert nach PID-Verarbeitung unter der Untergrenze liegt. In allen anderen Fällen AUS.
Übertragsmerker	CY	EIN während der Ausführung der PID-Regelung. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

PIDAT(191) wird ausgeführt, als wäre die Ausführungsbedingung ein STOP-RUN-Signal. PID-Berechnungen werden ausgeführt, wenn die Ausführungsbedingung während den nächsten Zyklus nach der Initialisierung von C+11 bis C+40 auf EIN gesetzt bleibt. Daher muss bei Verwendung des Immer-EIN-Merkers als Ausführungsbedingung für PIDAT(191) ein separater Prozess eingerichtet werden, durch den C+11 bis C+40 zu Beginn des Betriebs initialisiert werden.

Wenn die Daten für C nicht innerhalb des Bereichs liegen, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Wenn während des Autotunings ein Fehler auftritt, wird der Fehlermerker auf EIN gesetzt.

Wenn das tatsächliche Abtastintervall mehr als doppelt so lang wie das angegebene Abtastintervall ist, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt. Dessen ungeachtet wird die PID-Regelung weiter ausgeführt.

Der Übertragsmerker wird während der Ausführung der PID-Regelung auf EIN gesetzt.

Der Größer-als-Merker wird auf EIN gesetzt, wenn der Ausgangsstellwert nach PID-Verarbeitung die Obergrenze übersteigt. In diesem Fall wird der obere Grenzwert als Ergebnis ausgegeben.

Der Kleiner-als-Merker wird auf EIN gesetzt, wenn der Ausgangsstellwert nach PID-Verarbeitung unter der Untergrenze liegt. In diesem Fall wird der untere Grenzwert als Ergebnis ausgegeben.

PID-Parameter-Einstellungen

Steuer-daten	Parameter	Inhalt	Einstellbereich	Änderung mit EIN-Eingang-bedingung
C	Sollwert (SV)	Der Zielwert des geregelten Prozesses.	Binärdaten (die gleiche Anzahl von Bits, die auch für den Eingangsbereich spezifiziert ist)	Zulässig
C+1	Proportionalband	Der Parameter für P-Verhalten, Quotient aus Proportional-Regelbereich/ Gesamtregelbereich.	0001 bis 270F hex (1 bis 9999); (0,1 % bis 999,9 %, in Schritten von 0,1 %)	Kann mit EIN-Eingangsbedingung geändert werden, wenn Bit 1 von C+5 auf 1 gesetzt ist.
C+2	Tik Integralkonstante	Eine Konstante als Maß für die Wirkung der Integrationsfunktion. Mit steigendem Wert nimmt die Integralstärke zu.	0001 bis 1FFF hex (1 bis 8191); (9999 = keine Ausführung der Integrationsfunktion) (siehe Hinweis 1)	
C+3	Tdk Differenzialkonstante	Eine Konstante als Maß für die Wirkung der Differenzialfunktion. Mit steigendem Wert nimmt die Differenzialstärke zu.	0001 bis 1FFF hex (1 bis 8191); (9999 = keine Ausführung der Differentialfunktion) (siehe Hinweis 1)	
C+4	Abtastintervall (τ)	Einstellung des Intervalls für die Ausführung der PID-Regelung.	0001 bis 270F hex (1 bis 9999); (0,01 bis 99,99 s, in Schritten von 10 ms)	Nicht zulässig
Bits 04 bis 15 von C+5	2-PID-Parameter (α)	Der Eingangsfilter-Koeffizient. Normalerweise wird eine Wert von 0,65 (d. h. eine Einstellung auf 000) verwendet. Die Effizienz des Filters steigt, wenn sich der Koeffizient 0 annähert.	000 hex: α = 0,65 Eine Einstellung zwischen 100 und 163 hex bedeutet eine Einstellung des Werts der niederwertigsten zwei Stellen zwischen α = 0,00 und α = 0,99. (siehe Hinweis 2)	
Bit 03 von C+5	Ausgangsstellwert-Ausgabegröße:	Legt den Ausgangsstellwert fest, der ausgegeben wird, wenn der Istwert dem Sollwert entspricht.	0: Ausgabe 0 % 1: Ausgabe 50 %	

Steuer-daten	Parameter	Inhalt	Einstellbereich	Änderung mit EIN-Eingang-bedingung
Bit 01 von C+5	Einstellung für die Aktivierung von Änderungen der PID-Konstanten	Der Zeitpunkt für die Aktivierung von vorgenommenen Änderungen an Proportional-band (P), Integralkonstante (Tik) und Differenzialkonstante (Tdk) zur Verwendung in PID-Berechnungen.	0: Bei Beginn der Ausführung des PID-Befehls 1: Bei Beginn der Ausführung des PID-Befehls und jedes Abtastintervall	Zulässig
Bit 00 von C+5	Direkt-/Invers-Spezifikation der PID-Regelung	Bestimmt die Richtung der Proportionalfunktion.	0: Inverses Verhalten 1: Direktes Verhalten	Nicht zulässig
Bit 12 von C+6	Ausgangsstellwert-Ausgabegrenz-wertsteuerung	Legt fest, ob eine Grenzwert-steuerung auf die Ausgabe des Ausgangsstellwerts angewandt wird.	0: Deaktiviert (keine Grenzwertsteuerung) 1: Aktiviert (Grenzwertsteuerung)	
Bits 08 bis 11 von C+6	Eingangsbereich	Die Anzahl der Eingangsdatenbits	0: 8 Bits 5: 13 Bits 1: 9 Bits 6: 14 Bits 2: 10 Bits 7: 15 Bits 3: 11 Bits 8: 16 Bits 4: 12 Bits	
Bits 04 bis 07 von C+6	Integral- und Differenzialeinheit	Bestimmt die Einheit für die Angabe der Integral- und Differenzialkonstanten.	1: Vielfaches des Abtastintervalls 9: Zeit (Einheit: 100 ms)	
Bits 00 bis 03 von C+6	Ausgabebereich	Die Anzahl der Ausgangs-datenbits (Die Anzahl der Ausgangsdatenbits ist automatisch mit der der Eingangsdatenbits identisch.)	0: 8 Bits 5: 13 Bits 1: 9 Bits 6: 14 Bits 2: 10 Bits 7: 15 Bits 3: 11 Bits 8: 16 Bits 4: 12 Bits	
C +7	Ausgangsstellwert-Ausgabeunter-grenze	Die Untergrenze bei aktivierter Ausgangsstellwert-Ausgabebegrenzung.	0000 bis FFFF (binär) (siehe Hinweis 3)	
C +8	Ausgangsstellwert-Ausgabeobergrenze	Die Obergrenze bei aktivierter Ausgangsstellwert-Ausgabebegrenzung.	0000 bis FFFF (binär) (siehe Hinweis 3)	
Bit 15 von C+9	AT-Befehlsbit	Dieses Steuerbit startet das Autotuning. • Setzen Sie das AT-Befehlsbit zur Durchführung des Autotunings auf 1. (Das Autotuning kann während der Ausführung des Befehls PIDAT(191) gestartet werden.) • Dieses Bit wird automatisch auf AUS gesetzt, wenn das Autotuning abgeschlossen ist. Das Autotuning wird unterbrochen, wenn das AT-Befehlsbit manuell auf AUS gesetzt wird. In diesem Fall werden die neuen die PID-Konstanten jedoch aktiviert, wenn sie zum Zeitpunkt der Unterbrechung des Autotunings bereits berechnet waren.	Als Steuerbit: • 0 → 1: Ausführung des Autotunings • 1 → 0: Unterbrechung des Autotunings (PIDAT(191) setzt das Bit nach Abschluss des Autotunings automatisch auf AUS.) Als Merker: 0: Autotuning wird nicht ausgeführt. 1: Autotuning wird ausgeführt.	Zulässig

Steuer-daten	Parameter	Inhalt	Einstellbereich	Änderung mit EIN-Eingang-bedingung
Bits 00 bis 11 von C+9	AT-Berechnungs-verstärkung	Stellen Sie diesen Parameter ein, um die Beeinflussung der gespeicherten Werte durch die Ergebnisse der PID-Berechnungen anzupassen. Normalerweise kann dieser Parameter auf der Standardeinstellung (0000) belassen werden. • Erhöhen Sie den Wert zur Steigerung der Stabilität. • Verringern Sie den Wert zur Steigerung der Regelempfind-lichkeit.	0000 hex: 1,00 (Standardeinstellung) 0001 bis 03E8 hex (1 bis 1000); (0,91 bis 10,00 in Schritten von 0,01)	Zulässig (Diese Parameter werden beim Start des Autotunings gelesen.)
C+10	Grenzwertzyklus-Hysterese	Einstellung der Hysterese bei Erzeugung des Grenzwert-zyklus. Die Standardeinstellung bei Invers-Betrieb setzt den Ausgangsstellwert mit einer Hysterese von Sollwert – 20 % auf EIN. Erhöhen Sie diese Einstellung, wenn aufgrund eines instabilen Istwerts kein einwandfreier Grenzwertzyklus erzeugt werden kann. Beachten Sie jedoch, dass die Genauigkeit des Autotuning abnimmt, wenn die Grenzwert-zyklus-Hysterese größer als notwendig ist.	0000 hex: 0,20 % (Standardeinstellung) 0001 bis 03E8 hex: 0,01 bis 10,00 % in Schritten von 0,01 % FFFF hex: 0,00 % **Hinweis** Die Spezifikation erfolgt als Prozentsatz des Eingangsbereichs.	

Hinweis

1. Wird 1 für die Einheit spezifiziert, erstreckt sich der Bereich vom 1- bis 8.191-fachen des Intervalls. Wird 9 für die Einheit spezifiziert, erstreckt sich der Bereich vom 0,1 bis 819,1 s. Bei Spezifikation von 9 müssen Integrations- und Differenzialzeit auf einen Wert innerhalb des Bereichs vom 1- bis 8.191-fachen des Abtastintervalls eingestellt werden.
2. Die Einstellung des 2-PID-Parameters (α) auf 000 ergibt 0,65, den Normalwert.

   Wenn die Ausgangsstellwert-Ausgabebegrenzung aktiviert ist (auf "1" gesetzt ist), müssen die Werte wie folgt eingestellt werden:

   0000 ≤ Ausgangsstellwert-Ausgabeuntergrenze ≤ Ausgangsstellwert-Ausgabeobergrenze ≤ Maximalwert des Ausgangsbereichs

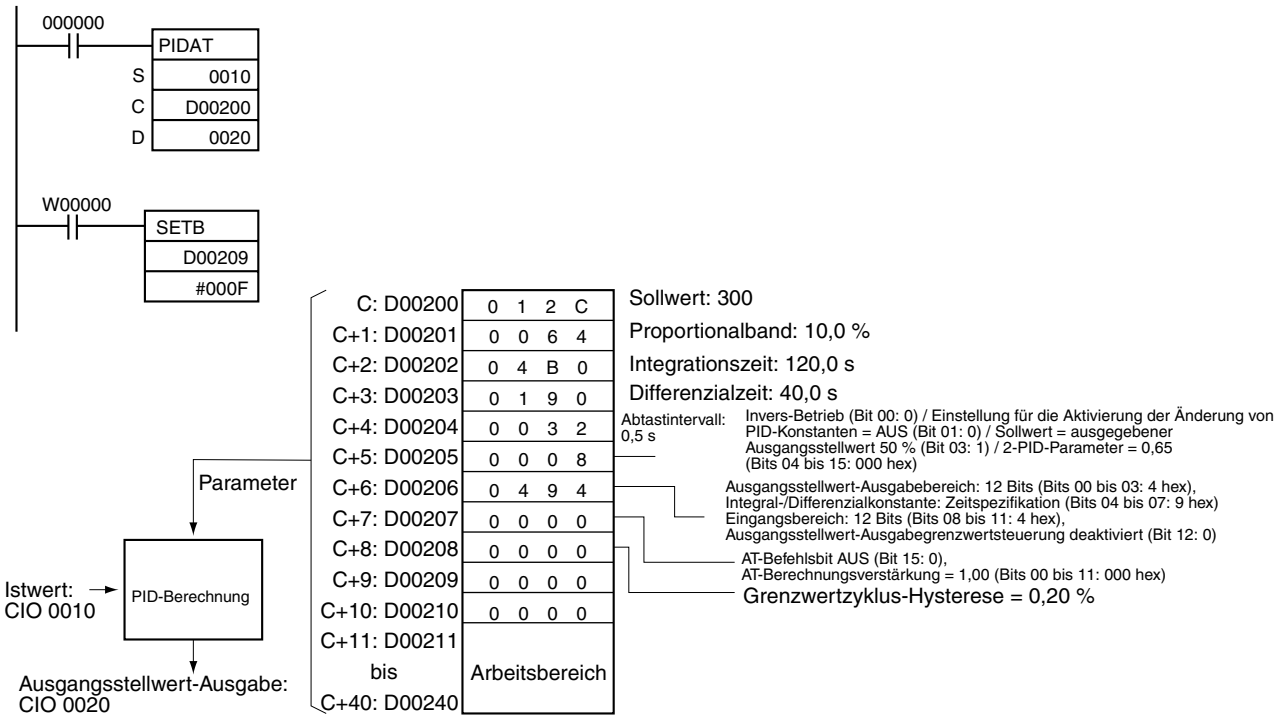
Beispiel 1: Unterbrechung der PID-Regelung zur Durchführung eines Autotunings

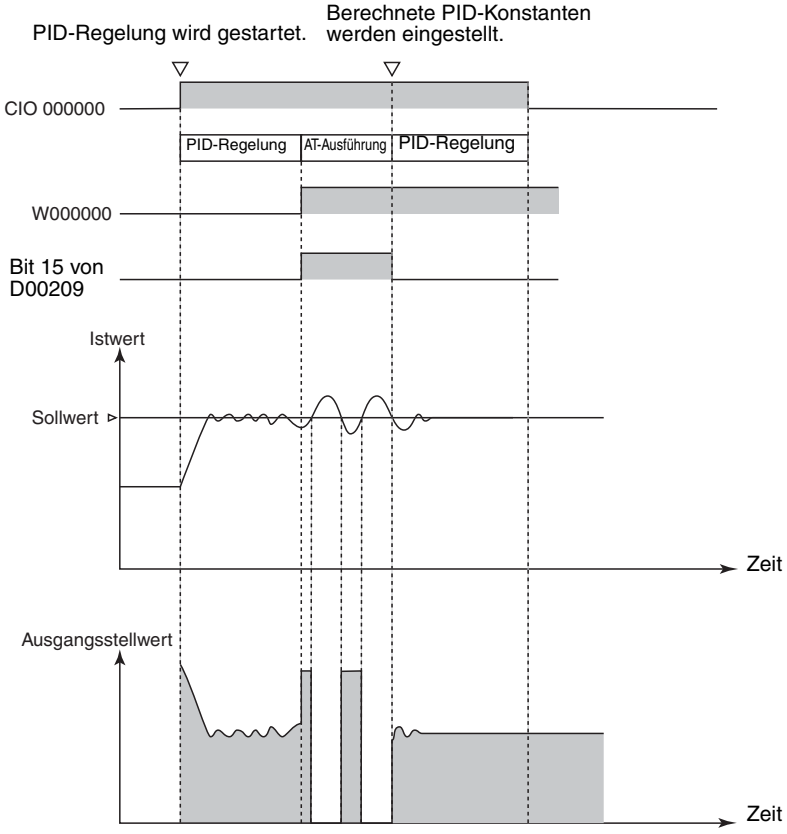
An der steigenden Flanke von CIO 000000 (AUS zu EIN) wird der Arbeitsbereich von D00211 bis D00240 entsprechend der in D00200 bis D00208 gesetzten Parameter (siehe unten) initialisiert. Nach der Initialisierung des Arbeitsbereichs wird die PID-Regelung ausgeführt und der Ausgangsstellwert wird in CIO 0020 ausgegeben.

Wenn CIO 000000 auf EIN gesetzt ist, wird die PID-Regelung in den Abtastintervallen entsprechend der in D00200 bis D00210 gesetzten Parameter ausgeführt. Der Ausgangsstellwert wird in CIO 0020 ausgegeben.

Die in den PID-Berechnungen verwendeten PID-Konstanten werden nicht geändert, selbst wenn Proportionalband (P), Integralkonstante (Tik) oder Differenzialkonstante geändert werden, nachdem CIO 000000 auf EIN gesetzt ist.

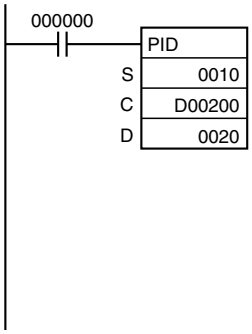
An der steigenden Flanke von W 000000 (AUS zu EIN), setzt der Befehl SETB(532) Bit 15 von D00209 (C+9) auf EIN und startet dadurch das Autotuning. Nach Abschluss des Autotunings werden die berechneten P-, I- und D-Konstanten in C+1, C+2 und C+3 geschrieben. Anschließend wird die PID-Regelung mit den neuen PID-Konstanten wieder gestartet.

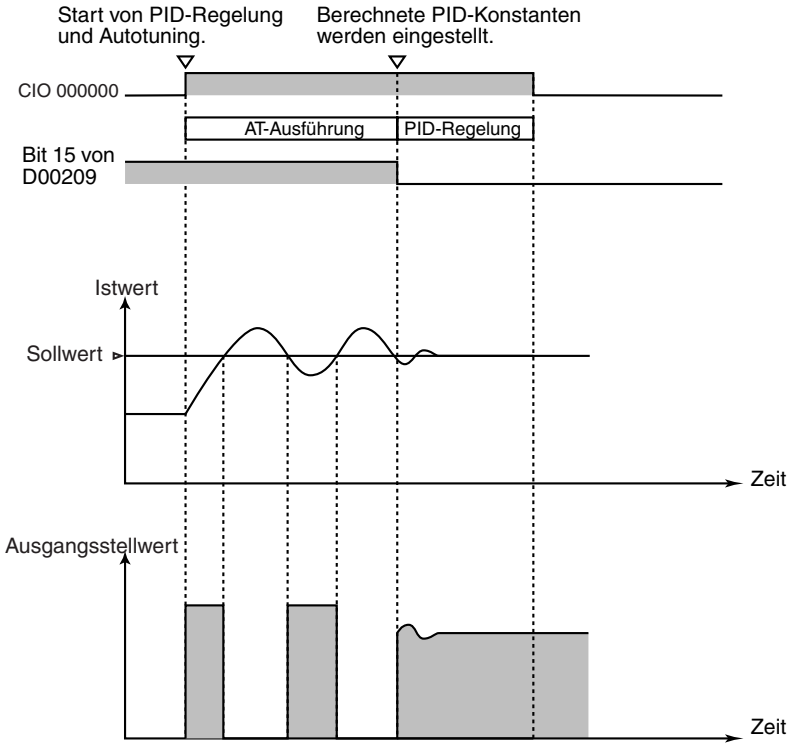

000000
PIDAT
S 0010
C D00200
D 0020
W00000
SETB
D00209
#000F
C: D00200 0 1 2 C
C+1: D00201 0 0 6 4
C+2: D00202 0 4 B 0
C+3: D00203 0 1 9 0
C+4: D00204 0 0 3 2
C+5: D00205 0 0 0 8
C+6: D00206 0 4 9 4
C+7: D00207 0 0 0 0
C+8: D00208 0 0 0 0
C+9: D00209 0 0 0 0
C+10: D00210 0 0 0 0
C+11: D00211
bis
C+40: D00240
Arbeitsbereich
Sollwert: 300
Proportionalband: 10,0 %
Integrationszeit: 120,0 s
Differenzialzeit: 40,0 s
Abtastintervall: 0,5 s
Invers-Betrieb (Bit 00: 0) / Einstellung für die Aktivierung der Änderung von PID-Konstanten = AUS (Bit 01: 0) / Sollwert = ausgegebener Ausgangsstellwert 50 % (Bit 03: 1) / 2-PID-Parameter = 0,65 (Bits 04 bis 15: 000 hex)
Ausgangsstellwert-Ausgabebereich: 12 Bits (Bits 00 bis 03: 4 hex), Integral-/Differenzialkonstante: Zeitspezifikation (Bits 04 bis 07: 9 hex) Eingangsbereich: 12 Bits (Bits 08 bis 11: 4 hex), Ausgangsstellwert-Ausgabegrenzwertsteuerung deaktiviert (Bit 12: 0)
AT-Befehlsbit AUS (Bit 15: 0), AT-Berechnungsverstärkung = 1,00 (Bits 00 bis 11: 000 hex)
Grenzwertzyklus-Hysterese = 0,20 %
Parameter
Istwert: CIO 0010
PID-Berechnung
Ausgangsstellwert-Ausgabe: CIO 0020


PID-Regelung wird gestartet.
Berechnete PID-Konstanten werden eingestellt.
CIO 000000
PID-Regelung
AT-Ausführung
PID-Regelung
W000000
Bit 15 von D00209
Istwert
Sollwert
Zeit
Ausgangsstellwert
Zeit

Beispiel 2:
Starten von PIDAT(191) mit Autotuning

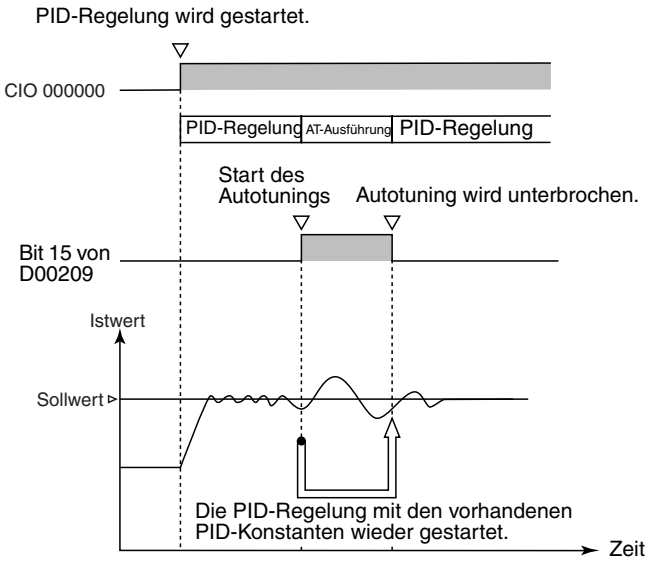
An der steigenden Flanke von CIO 000000 (AUS zu EIN) wird das Autotuning erst ausgeführt, wenn Bit 15 von D00209 (C+9) auf EIN gesetzt ist. Nach Abschluss des Autotunings werden die berechneten P-, I- und D-Konstanten in C+1, C+2 und C+3 geschrieben. Anschließend wird die PID-Regelung mit den berechneten PID-Konstanten gestartet.





Beispiel 3:
Unterbrechung des Autotunings vor dem Abschluss

Das Autotuning kann unterbrochen werden, indem Bit 15 von D00209 (C+9) von EIN auf AUS gesetzt wird. Die PID-Regelung wird mit den P-, I- und D-Konstanten neu gestartet, die vor Beginn des Autotunings eingestellt waren.

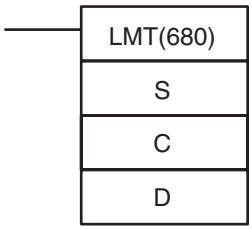


3-18-3 LIMIT CONTROL: LMT(680)

Zweck

Steuert Ausgabedaten in Abhängigkeit davon, ob sich Eingangsdaten innerhalb des Bereichs zwischen oberem und unterem Grenzwert befinden.

Kontaktplansymbol



S: Eingangswort
C: Erstes Grenzwertwort
D: Ausgangswort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	LMT(680)
	Ausführung einmal bei steigender Flanke	@LMT(680)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W510	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H510	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A958	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	
Datenregister	DR0 bis DR15	---	DR0 bis DR15

Bereich	S	C	D
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Wenn die Ausführungsbedingung auf EIN gesetzt ist, steuert der Befehl LMT(680) die Ausgangsdaten in Abhängigkeit davon, ob die spezifizierten Eingangsdaten (16-Bit-Binärdaten mit Vorzeichen) innerhalb des durch den oberen und unteren Grenzwert gebildeten Bereichs liegen. Die Worte C und C+1 haben folgenden Inhalt:

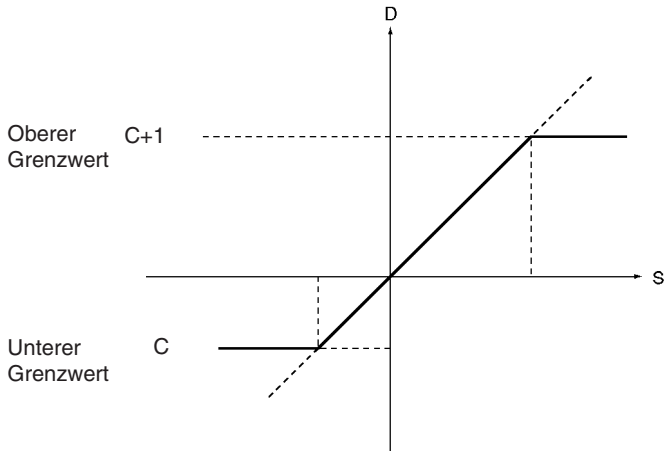
C	Untergrenzen-Daten (Minimal-Ausgabedaten)
C+1	Obergrenzen-Daten (Maximal-Ausgabedaten)

C und C+1 müssen eine identische Bereichsklassifizierung besitzen.

Wenn die Eingangsdaten (S) kleiner sind als der untere Grenzwert (C), werden die Untergrenzen-Daten nach D ausgegeben und der Kleiner-als-Merker wird auf EIN gesetzt.

Wenn die Eingangsdaten (S) größer sind als der obere Grenzwert (C+1), werden die Obergrenzen-Daten nach D ausgegeben und der Größer-als-Merker wird auf EIN gesetzt.

Wenn die Eingangsdaten (S) größer als oder gleich der Untergrenze (C) und kleiner als oder gleich der Obergrenze (C+1) sind, werden die Eingangsdaten (S) nach D ausgegeben.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der obere Grenzwert kleiner als der untere Grenzwert ist. In allen anderen Fällen AUS.
Größer-als-Merker	>	EIN, wenn die Eingangsdaten (S) größer als der obere Grenzwert sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Kleiner-als-Merker	<	EIN, wenn die Eingangsdaten (S) kleiner als der untere Grenzwert sind. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn der obere Grenzwert kleiner als der untere Grenzwert ist, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Wenn die Eingangsdaten (S) größer als der obere Grenzwert sind, wird der Größer-als-Merker auf EIN gesetzt.

Wenn das Ausgangswort D den Inhalt 0000 hex hat, wird der Gleich-Merker auf EIN gesetzt.

Wenn die Eingangsdaten (S) kleiner als der untere Grenzwert sind, wird der Kleiner-als-Merker auf EIN gesetzt.

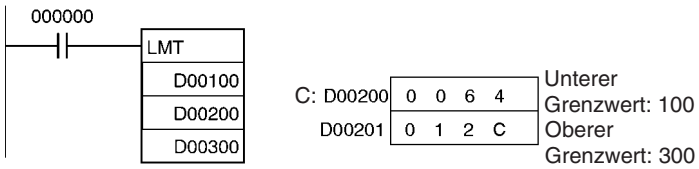
Wenn des höchstwertige Bit des Ausgangsworts D den Status 1 hat, wird der Negativ-Merker auf EIN gesetzt.

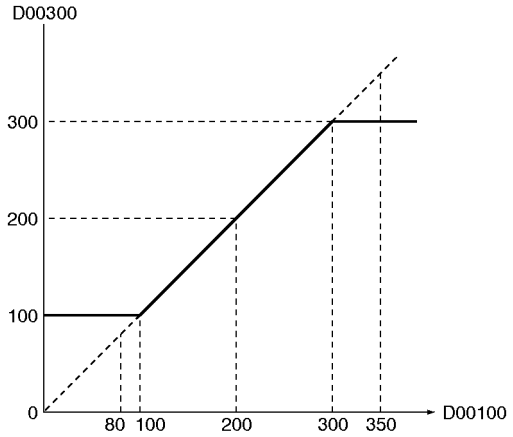
Beispiel

Wenn D00100 den Inhalt 0050 hex (80) hat, wird 0064 hex (100) nach D00300 ausgegeben, da 80 kleiner ist als der untere Grenzwert von 100.

Wenn D00100 den Inhalt 00C8 hex (200) hat, wird 00C8 hex (200) nach D00300 ausgegeben, da 200 zwischen oberem und unterem Grenzwert liegt.

Wenn D00100 den Inhalt 015E hex (350) hat, wird 012C hex (300) nach D00300 ausgegeben, da 350 größer ist als der obere Grenzwert von 300.



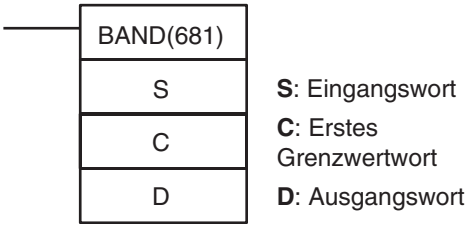


3-18-4 DEAD BAND CONTROL: BAND(681)

Zweck

Steuert die Ausgabedaten in Abhängigkeit davon, ob die Eingangsdaten innerhalb des Bereichs (Totbandbereich) liegen, der durch unteren und oberen Grenzwert definiert ist.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	BAND(681)
	Ausführung einmal bei steigender Flanke	@BAND(681)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W510	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H510	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A958	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Wenn die Ausführungsbedingung auf EIN gesetzt ist, steuert der Befehl BAND(681) die Ausgangsdaten in Abhängigkeit davon, ob die spezifizierten Eingangsdaten (16-Bit-Binärdaten mit Vorzeichen) innerhalb des durch den oberen und unteren Grenzwert gebildeten Bereichs (Totband) liegen. Die Worte C und C+1 haben folgenden Inhalt:

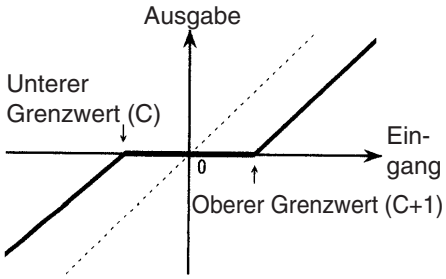
C	Untergrenzen-Daten (Unterer Grenzwert des Totbandes)
C+1	Obergrenzen-Daten (Oberer Grenzwert des Totbandes)

C und C+1 müssen eine identische Bereichsklassifizierung besitzen.

Wenn die Eingangsdaten (S) größer als oder gleich der Untergrenze (C) und kleiner als oder gleich der Obergrenze (C+1) sind, wird 0000 (hex) nach D ausgegeben und der Gleich-Merker wird auf EIN gesetzt.

Wenn die Eingangsdaten (S) kleiner sind als der untere Grenzwert (C), wird die Differenz zwischen Eingangsdaten und Untergrenzen-Daten nach D ausgegeben und der Kleiner-als-Merker wird auf EIN gesetzt.

Wenn die Eingangsdaten (S) größer sind als der obere Grenzwert (C+1), wird die Differenz zwischen Eingangsdaten und Obergrenzen-Daten nach D ausgegeben und der Größer-als-Merker wird auf EIN gesetzt.



Wenn die Ausgangsdaten kleiner als 8000 (hex) oder größer als 7FFF (hex) sind, wird das Vorzeichen umgekehrt. Ein Beispiel: Bei einem unteren Grenzwert von 0100 (hex) und Eingangsdaten von 8000 (hex) ergeben sich folgende Ausgangsdaten:
8000 (hex) [–32768] – 0100 (hex) [256] = 7F00 (hex) [32512]

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der obere Grenzwert kleiner als der untere Grenzwert ist. In allen anderen Fällen AUS.
Größer-als-Merker	>	EIN, wenn die Eingangsdaten (S) größer als der obere Grenzwert sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Kleiner-als-Merker	<	EIN, wenn die Eingangsdaten (S) kleiner als der untere Grenzwert sind. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn der obere Grenzwert kleiner als der untere Grenzwert ist, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Wenn die Eingangsdaten (S) größer als der obere Grenzwert sind, wird der Größer-als-Merker auf EIN gesetzt.

Wenn das Ausgangswort D den Inhalt 0000 hex hat, wird der Gleich-Merker auf EIN gesetzt.

Wenn die Eingangsdaten (S) kleiner als der untere Grenzwert sind, wird der Kleiner-als-Merker auf EIN gesetzt.

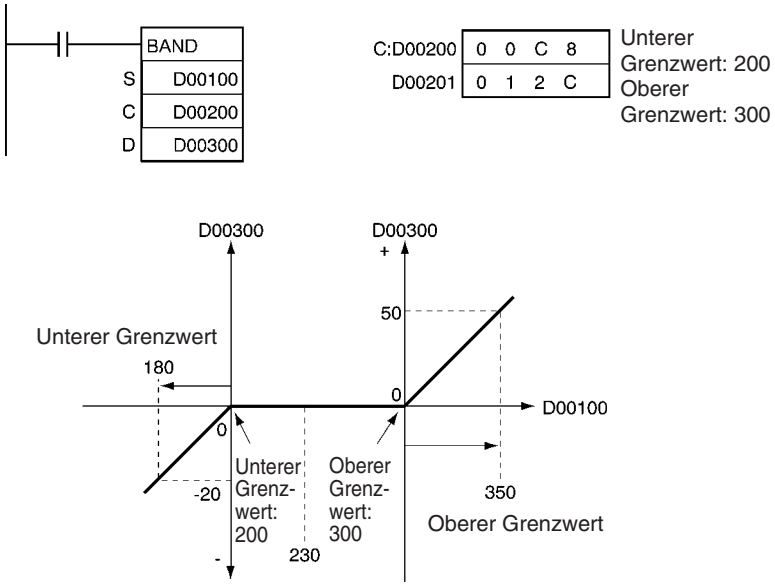
Wenn des höchstwertige Bit des Ausgangsworts D den Status 1 hat, wird der Negativ-Merker auf EIN gesetzt.

Beispiel

Wenn D00100 den Inhalt 00B4 hex (180) hat, wird 180 – 200 = FFEC hex (–20) nach D00300 ausgegeben, da 180 kleiner ist als der untere Grenzwert von 200.

Wenn D00100 den Inhalt 00E6 hex (230) hat, wird 0 nach D00300 ausgegeben, da 230 zwischen oberem und unterem Grenzwert liegt.

Wenn D00100 den Inhalt 015E hex (350) hat, wird 350 – 300 = 0032 hex (50) nach D00300 ausgegeben, da 350 größer ist als der obere Grenzwert von 300.

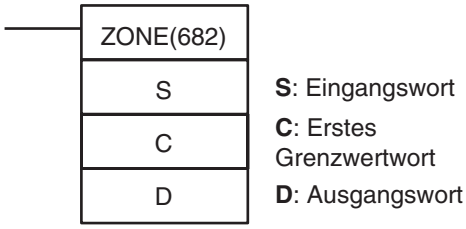


3-18-5 DEAD ZONE CONTROL: ZONE(682)

Zweck

Addiert den spezifizierten Offset zu den Eingangsdaten und gibt das Ergebnis aus.

Kontaktplansymbol



S: Eingangswort

C: Erstes Grenzwertwort

D: Ausgangswort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	ZONE(682)
	Ausführung einmal bei steigender Flanke	@ZONE(682)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden-Spezifikationen

Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W510	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H510	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A958	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	E00000 bis E32767

Bereich	S	C	D
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Wenn die Ausführungsbedingung auf EIN gesetzt ist, addiert der Befehl ZONE(682) den spezifizierten Offset zu den spezifizierten Eingangsdaten (16-Bit-Binärdaten mit Vorzeichen) und schreibt das Ergebnis in das spezifizierte Ausgangswort. Die Worte C und C+1 haben folgenden Inhalt:

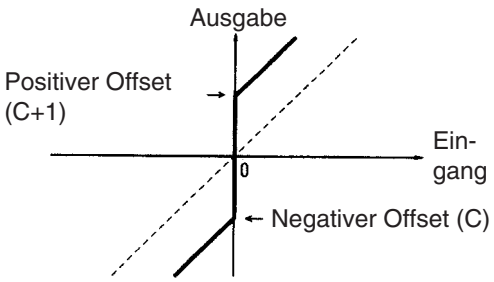
C	Negativer Offset
C+1	Positiver Offset

C und C+1 müssen eine identische Bereichsklassifizierung besitzen.

Wenn die Eingangsdaten (S) kleiner als Null sind, wird der negative Offset zu den Eingangsdaten addiert, das Ergebnis wird nach D ausgegeben und der Kleiner-als-Merker wird auf EIN gesetzt.

Wenn die Eingangsdaten (S) größer als Null sind, wird der positive Offset zu den Eingangsdaten addiert, das Ergebnis wird nach D ausgegeben und der Größer-als-Merker wird auf EIN gesetzt.

Wenn die Eingangsdaten (S) gleich Null sind, wird 0000 nach D ausgegeben und der Gleich-Merker wird auf EIN gesetzt.



Wenn die Ausgangsdaten kleiner als 8000 (hex) oder größer als 7FFF (hex) sind, wird das Vorzeichen umgekehrt. Ein Beispiel: Bei einem negativen Offsetwert von FF00 (hex) und Eingangsdaten von 8000 (hex) ergeben sich folgende Ausgangsdaten:
8000 (hex) [–32768] – FF00 (hex) [–256] = 7F00 (hex) [32512]

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der obere Grenzwert kleiner als der untere Grenzwert ist. In allen anderen Fällen AUS.
Größer-als-Merker	>	EIN, wenn die Eingangsdaten (S) größer als Null sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn die Eingangsdaten gleich 0 sind. In allen anderen Fällen AUS.
Kleiner-als-Merker	<	EIN, wenn die Eingangsdaten (S) kleiner als Null sind. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit des Ergebnisses 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn der obere Grenzwert kleiner als der untere Grenzwert ist, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Wenn die Eingangsdaten (S) größer als Null sind, wird der Größer-als-Merker auf EIN gesetzt.

Wenn das Ausgangswort D den Inhalt 0000 hex hat, wird der Gleich-Merker auf EIN gesetzt.

Wenn die Eingangsdaten (S) kleiner als Null sind, wird der Kleiner-als-Merker auf EIN gesetzt.

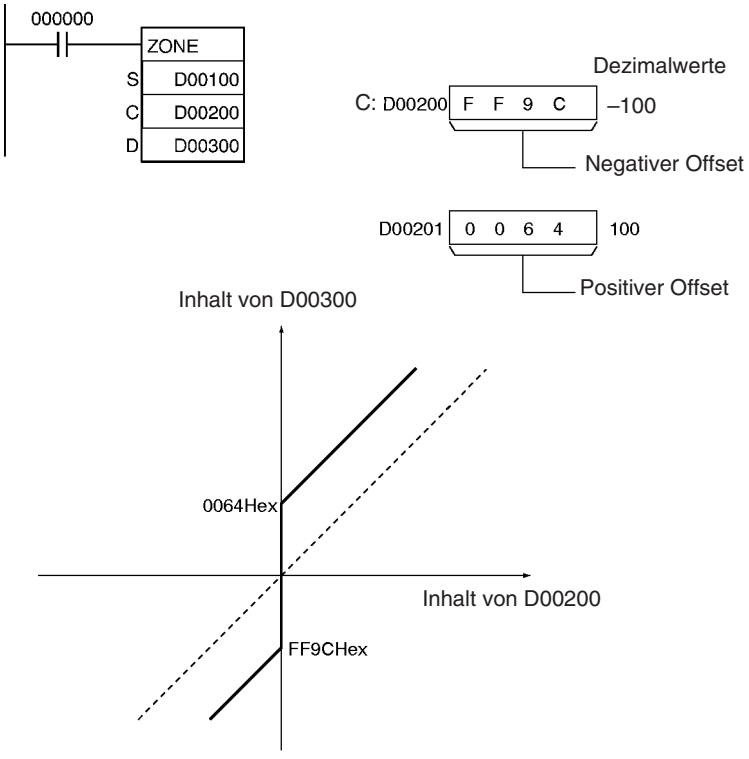
Wenn des höchstwertige Bit des Ausgangsworts D den Status 1 hat, wird der Negativ-Merker auf EIN gesetzt.

Beispiel

Wenn CIO 000000 auf EIN gesetzt ist, wird ein Offset von –100 zu dem Wert in D00100 addiert, wenn dieser Wert kleiner als Null ist, und der Ergebniswert wird in D00300 gespeichert.

Wenn der Inhalt von D00100 den Wert 0 hat, wird 0000 hex in D00300 gespeichert.

Wenn der Wert von D00100 größer als 0 ist, wird ein Offset von +100 zum Eingangswert addiert und der Ergebniswert wird in D00300 gespeichert.



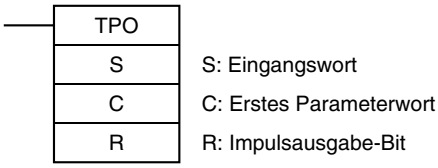
3-18-6 TIME-PROPORTIONAL OUTPUT: TPO(685)

Zweck

Verwendet das Tastverhältnis oder den Ausgangsstellwert aus dem spezifizierten Wort als Eingabe, konvertiert das Tastverhältnis in eine zeitproportionale Ausgabe auf der Basis der spezifizierten Parameter und gibt das Ergebnis über den spezifizierten Ausgang aus.

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TPO(685)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Operanden

S: **Eingangswort**

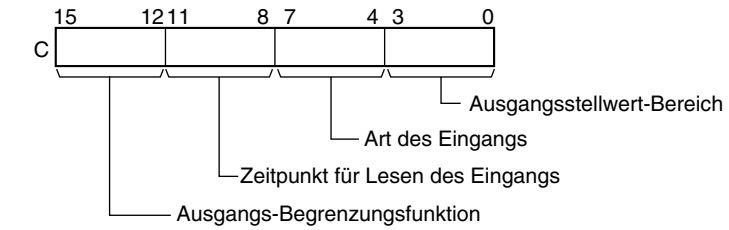
Spezifiziert das Eingangswort, das das Tastverhältnis oder den Ausgangsstellwert enthält, die als Eingangsdaten für diesen Befehl dienen. Bits 04 bis 07 von C spezifizieren die Art des Eingang, also ob das Eingangswort ein Tastverhältnis oder einen Ausgangsstellwert enthält. (Setzen Sie diese Bits auf 0 hex, um ein Tastverhältnis zu spezifizieren, oder auf 1 hex, um einen Ausgangsstellwert zu spezifizieren.)

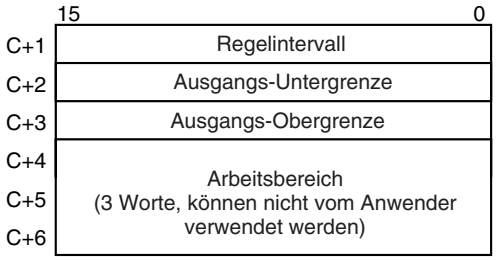
- Tastverhältnis als Eingang: 0000 bis 2710 hex (0,00 % bis 100,00 %)
- Ausgangsstellwert als Eingang (siehe Hinweis): 0000 bis FFFF hex (0 bis max. 65.535) (Bits 00 bis 03 von C spezifizieren den Bereich des Ausgangsstellwerts, d. h., die Anzahl gültiger Bits im Ausgangsstellwert. Spezifizieren Sie die gleich Anzahl von Bits, die auch für die Ausgangsbereichseinstellung in PID(190) spezifiziert wurde.)

Hinweis Wenn S ein Ausgangsstellwert ist, müssen Sie das Wort spezifizieren, das den von einem PID(190)- oder PIDAT(191)-Befehl ausgegebenen Ausgangsstellwert enthält.

C bis C+6: Parameter

In der folgenden Abbildung sind die Speicherorte der Parameterdaten dargestellt. Detaillierte Informationen zu den Parametern finden Sie unter *Parametereinstellungen* in diesem Abschnitt.





Hinweis: Detaillierte Informationen finden Sie in den Beschreibungen der einzelnen Parameter.

R: Impulsausgabe-Bit
Spezifiziert das Ziel-Ausgangsbit für die Impulsausgabe.

Normalerweise wird dafür ein Ausgangsbit spezifiziert, das einer Transistor-Ausgangsbaugruppe zugewiesen ist, und ein Halbleiterrelais an die Transistor-Ausgangsbaugruppe angeschlossen.

Operanden-Spezifikationen

Bereich	S	C	R
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6137	CIO 000000 bis CIO 614315
Arbeitsbereich	W000 bis W511	W000 bis W505	W00000 bis W51115
Haftmerkerbereich	H000 bis H511	H000 bis H505	H00000 bis H51115
Zusatz-Systembereich	A000 bis A959	A000 bis A953	A44800 bis A95915
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4089	---
Zählerbereich	C0000 bis C4095	C0000 bis C4089	---
DM-Bereich	D00000 bis D32767	D00000 bis D32761	---
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32761	---
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32761 (n = 0 bis C)	---
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		---
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		---
Konstanten	#0000 bis #FFFF (binär)	---	---
Datenregister	DR0 bis DR15	---	---

Bereich	S	C	R
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Verwendet ein Tastverhältnis oder einen Ausgangsstellwert aus der durch S spezifizierten Wortadresse als Eingang, konvertiert das Tastverhältnis auf Basis der in den Worten C bis C+3 spezifizierten Parameter in eine zeitproportionale Ausgabe (siehe Hinweis) und gibt eine Impulsausgabe an das in R spezifizierte Bit aus.

Hinweis

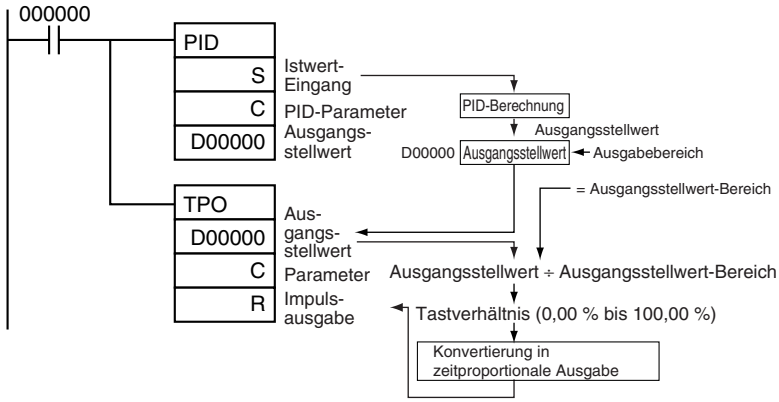
Die zeitproportionale Ausgabe ändert sich proportional zum EIN/AUS-Verhältnis in Eingangswort S. Das Intervall, in dem sich der EIN- und AUS-Status ändert, wird als Regelintervall bezeichnet und in Parameterwort C+1 eingestellt.

Beispiel: Bei einem Regelintervall von 1 s und einem Eingangswert von 50 % ist das Bit 0,5 s lang EIN und 0,5 s lang AUS. Bei einem Regelintervall von 1 s und einem Eingangswert von 80 % ist das Bit 0,8 s lang EIN und 0,2 s lang AUS.

Im Allgemeinen wird der Befehl TPO(685) zusammen mit dem Befehl PID(190) oder PIDAT(191) verwendet und das Ergebniswort (D) des PID-Befehls, das den Ausgangsstellwert enthält, wird als Eingangswort (S) für den Befehl TPO(685) spezifiziert. Zudem wird als R im Allgemeinen ein einer Transistor-Ausgangsbaugruppe zugeordnetes Ausgangsbit spezifiziert und an den Ausgang der Transistor-Ausgangsbaugruppe wird ein Halbleiterrelais angeschlossen, um die zeitproportionale Regelung einer Heizung durchzuführen (Proportionalregelung des EIN/AUS-Verhältnisses).

Kombination von TPO(685) mit einem PID-Regelungs-Befehl

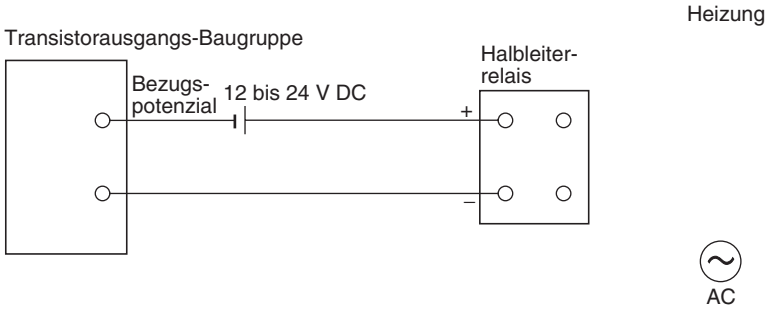
Bei Kombination von TPO(685) mit einem PID-Regelungs-Befehl wird der als Eingang verwendete Ausgangsstellwert durch den Ausgangsstellwertbereich geteilt, um das Tastverhältnis zu berechnen. Dieses Tastverhältnis wird dann in eine zeitproportionale Ausgabe konvertiert und in Form von Impulsen ausgegeben.



In diesem Fall müssen für den Ausgangsbereich des PID-Regelungs-Befehls und den Ausgangsstellwertbereich des Befehls TPO(685) die gleichen Werte eingestellt werden. Ein Beispiel: Wenn sowohl der Ausgangsbereich des PID-Regelungs-Befehls als auch der Ausgangsstellwertbereich des Befehls TPO(685) auf 12 Bits (0000 bis 0FFF hex) eingestellt ist, wird das Tastverhältnis berechnet, indem der Ausgangsstellwert des PID-Regelungs-Befehls durch 0FFF hex dividiert wird. Der Befehl TPO(685) konvertiert dieses Tastverhältnis dann in eine zeitproportionale Ausgabe.

Beispiel für externe Beschaltung

Schließen Sie ein Halbleiterrelais wie in der folgenden Abbildung gezeigt an die Transistor-Ausgangsbaugruppe an.



Parametereinstellungen

Steuerdaten		Parameter	Inhalt	Einstellbereich	Änderung mit EIN-Eingang-bedingung
Wort	Bits				
C	00 bis 03	Ausgangsstell-wert-Bereich	Spezifikation der Anzahl von Eingangsdatenbits	0 hex: 8 Bits 5 hex: 13 Bits 1 hex: 9 Bits 6 hex: 14 Bits 2 hex: 10 Bits 7 hex: 15 Bits 3 hex: 11 Bits 8 hex: 16 Bits 4 hex: 12 Bits	Zulässig
	04 bis 07	Art des Eingangs	Spezifikation, ob S ein Tastverhältnis oder einen Ausgangsstellwert enthält.	0 hex: Tastverhältnis Einstellbereich für S: 0000 bis 2710 hex (0,00 % bis 100,00 %) 1 hex: Ausgangsstellwert Einstellbereich für S: 0000 bis FFFF hex (0 bis 65.535) (Der Maximalwert der Einstellung ist von Ausgangsstellwert-Bereich abhängig, der in Bits 00 bis 03 von C spezifiziert ist.)	Zulässig
	08 bis 11	Zeitpunkt für Lesen des Eingangs	Spezifikation des Zeitpunkts für das Lesen des Eingangs	0 hex: Wert zu Beginn des Regelintervalls verwenden 1 hex: Kleineren Wert verwenden 2 hex: Höheren Wert verwenden 3 hex: Kontinuierliche Anpassung	Zulässig
	12 bis 15	Ausgabegrenz-wertsteuerung	Spezifikation, ob die Ausgabegrenzwert-Funktion aktiviert oder deaktiviert ist	0 hex: Deaktiviert 1 hex: Aktiviert (siehe Hinweis)	Zulässig
C+1	00 bis 15	Regelintervall	Regelintervall (Das Zeitintervall, in dem die EIN/AUS-Änderungen erfolgen.)	0000 bis 270F hex (1,00 bis 99,99 s) Hinweis: Beispielsweise wird 1,00 s als 0064 hex eingestellt, nicht als 0001 hex.	Zulässig
C +2	00 bis 15	Ausgangs-Untergrenze	Spezifikation des unteren Grenzwerts bei aktiviertem Ausgabegrenzwert.	0000 bis 2710 hex (0,00 % bis 100,00 %)	Zulässig
C +3	00 bis 15	Ausgangs-Obergrenze	Spezifikation des oberen Grenzwerts bei aktiviertem Ausgabegrenzwert.	0000 bis 2710 hex (0,00 % bis 100,00 %)	Zulässig
C+4	00 bis 15	Arbeitsbereich	Dieser Arbeitsbereich wird vom System verwendet. Er kann nicht vom Anwender verwendet werden.	Kann nicht verwendet werden.	---
C+5	00 bis 15				
C+6	00 bis 15				

Hinweis Wenn die Funktion zur Ausgabegrenzwertsteuerung aktiviert ist, müssen unterer und oberer Grenzwert wie folgt eingestellt werden: 0000 hex ≤ unterer Grenzwert ≤ oberer Grenzwert ≤ 2710 hex.

Ausführung

- Der Befehl wird ausgeführt, wenn die Eingangsbedingung auf EIN gesetzt ist.

- Während der Ausführung wird das Ausgangsbit (R) entsprechend des Tastverhältnisses auf EIN/AUS gesetzt.
- Die Parameter (in C bis C+3) werden jedes Mal bei Ausführung des Befehls in Echtzeit gelesen. Wenn Sie die Parameter ändern, sollten Sie alle Parameter gleichzeitig ändern, damit nicht verschiedene Parametersätze gemischt werden.
- Der Ausgang (R) wird bei Ausführung des Befehls auf EIN/AUS gesetzt. Die Genauigkeit der EIN/AUS-Zeiten des Ausgangs beträgt max. 10 ms.
- Die Ausführung des Befehls wird angehalten, wenn die Eingangsbedingung auf AUS gesetzt wird. Zu diesem Zeitpunkt wird der Wert für die verstrichene Zeit zurückgesetzt und das Regelintervall wird initialisiert.
- Die Einstellung für die Art des Eingangs (Bits 04 bis 07 von C) spezifiziert, ob das Eingangswort (S) ein Tastverhältnis oder einen Ausgangsstellwert enthält. Wenn S einen Ausgangsstellwert enthält, wird das Tastverhältnis berechnet, indem der eingegebenen Ausgangsstellwert durch den Ausgangsstellwert-Bereich (Bits 00 bis 03 von C) dividiert wird.

Einstellung für den Lesezeitpunkt des Eingangs (Bits 08 bis 11 von C)

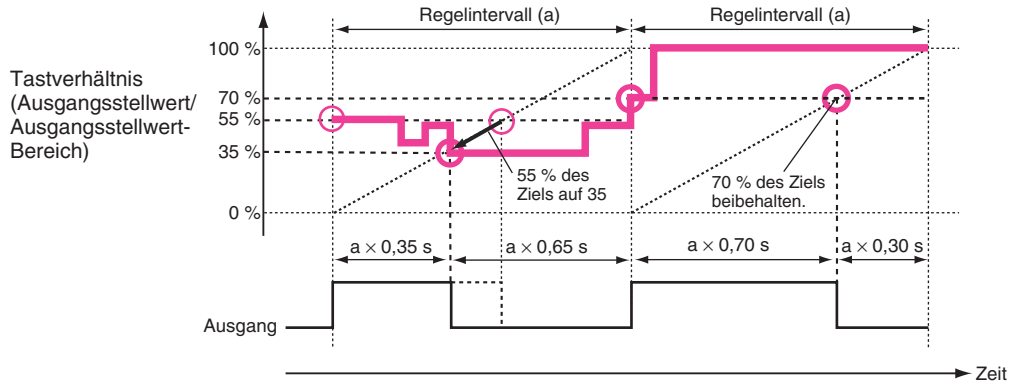
Die Einstellung für den Lesezeitpunkt des Eingangs (Bits 08 bis 11 von C) spezifiziert, wann das Eingangswort (S) gelesen wird. Die möglichen Einstellungen sind in der folgenden Tabelle erläutert.

Zeitpunkt für Lesen des Eingangs	Beschreibung
0: Wert zu Beginn des Regelintervalls verwenden	Das Eingangs-Tastverhältnis wird zu Begin des Regelintervalls gelesen. Das Tastverhältnis kann nicht während des Regelintervalls geändert werden.
1: Kleineren Wert verwenden	Wenn das Eingangs-Tastverhältnis unter das Tastverhältnis zu Beginn des Regelintervalls sinkt, hat der niedrigere Wert Priorität und die EIN-Zeit des Ausgangs wird entsprechend verkürzt.
2: Höheren Wert verwenden	Wenn das Eingangs-Tastverhältnis über das Tastverhältnis zu Beginn des Regelintervalls steigt, hat der höhere Wert Priorität und die EIN-Zeit des Ausgangs wird entsprechend verlängert.
3: Kontinuierliche Anpassung	Das Tastverhältnis wird bei jeder Ausführung des Befehls in Echtzeit gelesen und die EIN/AUS-Funktion wird innerhalb des Regelintervalls wiederholt.

Die folgenden Abbildungen zeigen die Funktionen der einzelnen Einstellungen für den Zeitpunkt zum Lesen des Eingangs.

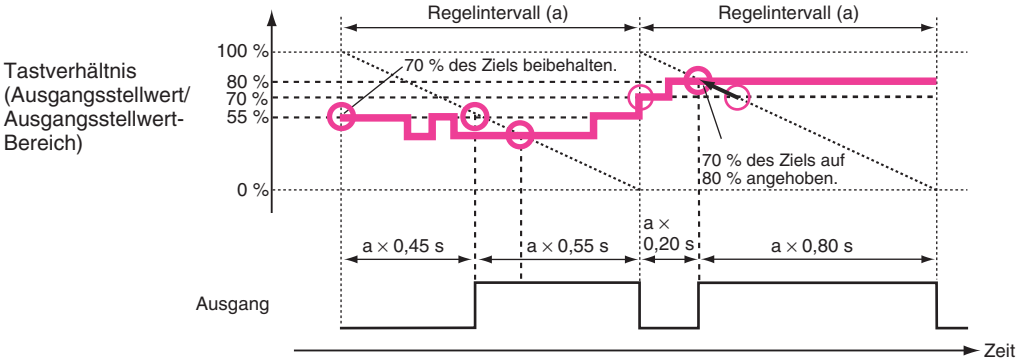
- Eingangszeit-Einstellung = 0 (Wert zu Beginn des Regelintervalls verwenden)

• Eingangszeit-Einstellung = 1 (Kleineren Wert verwenden.)



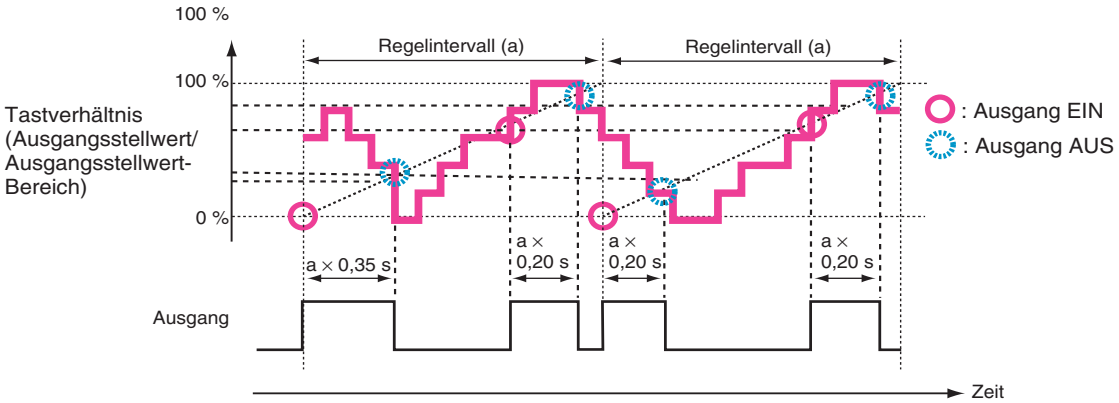
Wenn das Tastverhältnis früh genug unter das anfängliche Tastverhältnis fällt, wird das Tastverhältnis angepasst und der Ausgang wird früher auf AUS geschaltet. Diese Einstellung ist für Anwendungen geeignet, bei denen das Überschwingen vermieden werden soll, z. B. bei der zeitproportionalen Regelung einer Heizung, wobei ein relativ langes Regelintervall verwendet wird.

• Eingangszeit-Einstellung = 2 (Höheren Wert verwenden.)



Wenn das Tastverhältnis früh genug über das anfängliche Tastverhältnis steigt, wird das Tastverhältnis angepasst und der Ausgang wird früher auf EIN geschaltet. (Bei dieser Einstellung ist die EIN/AUS-Reihenfolge des Ausgangs umgekehrt, der Ausgang wechselt im Regelintervall von AUS zu EIN.) Diese Einstellung ist für Anwendungen geeignet, bei denen das Unterschwingen vermieden werden soll, z. B. bei der zeitproportionalen Regelung einer Kühlung, wobei ein relativ langes Regelintervall verwendet wird.

• Eingangszeit-Einstellung = 3 (Kontinuierliche Anpassung)



Änderungen des Tastverhältnisses werden in Echtzeit überwacht. Wenn das Tastverhältnis früh genug unter das anfängliche Tastverhältnis fällt, wird das Tastverhältnis angepasst und der Ausgang wird früher auf AUS geschaltet. Wenn das Tastverhältnis anschließend wieder steigt, erfolgt eine erneute Anpassung und der Ausgang wird wieder auf EIN geschaltet. Dieser Vorgang wird kontinuierlich wiederholt.
Diese Einstellung eignet sich zur Steigerung des Reaktionsvermögens, wenn das Regelintervall relativ lang ist und das Tastverhältnis sich rasch ändert. Diese Einstellung eignet sich auf für Beleuchtungs- und Leistungsanwendungen, bei denen eine präzise Regelung erforderlich ist.

Merker

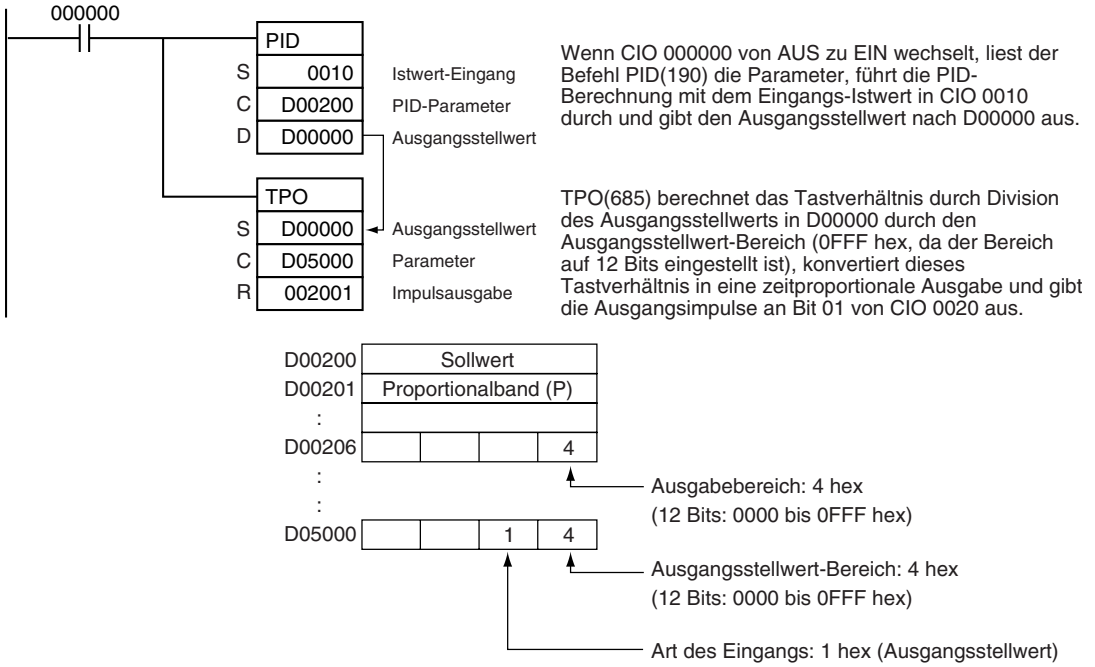
Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn die Eingangsdaten in S nicht innerhalb des gültigen Bereichs liegen. (Der gültige Eingangsdatenbereich hängt von der Einstellung für die Art des Eingangs ab.) EIN, wenn die Daten für C nicht innerhalb des gültigen Bereichs liegen. (Der Ausgangsstellwert-Bereich führt nur zu einem Fehler, wenn Ausgangsstellwert für die Art des Eingangs eingestellt ist.) EIN, wenn das Regelintervall in C+1 nicht innerhalb des gültigen Bereichs liegt. EIN, wenn die Ausgangsgrenzwert-Funktion aktiviert ist, der untere Grenzwert (C+2) oder der obere Grenzwert (C+3) jedoch außerhalb des gültigen Bereichs liegt. EIN, wenn die Ausgangsgrenzwert-Funktion aktiviert ist, der untere Grenzwert (C+2) jedoch größer oder gleich dem oberen Grenzwert (C+3) ist. In allen anderen Fällen AUS.

Beispiel

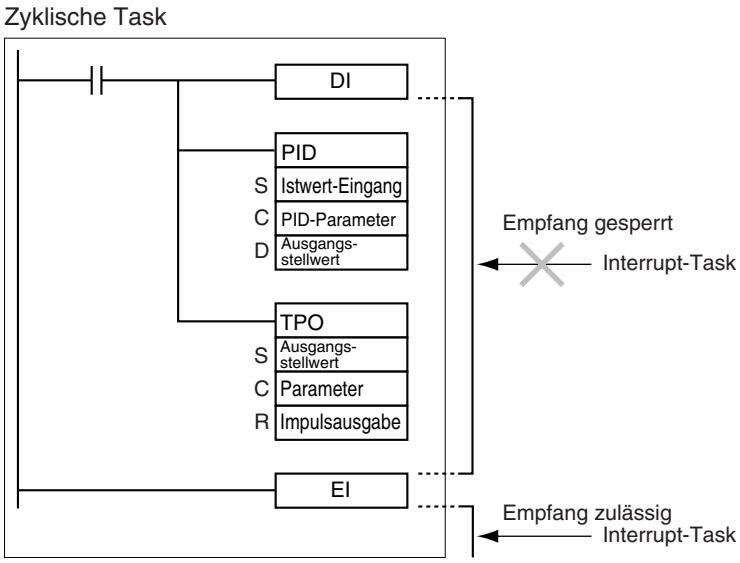
Beispiel 1: Kombination von TPO(685) mit PID(190)

Wenn CIO 000000 auf EIN gesetzt ist, liest der Befehl TPO(685) den vom Befehl PID(190) ausgegebenen Ausgangsstellwert (in D00000 enthalten), berechnet aus diesem Ausgangsstellwert ei Tastverhältnis (Tastverhältnis = Ausgangsstellwert ÷ Ausgangsstellwert-Bereich), konvertiert dieses Tastverhältnis in eine zeitproportionale Ausgabe und gibt die Impulse in CIO 002001 aus.

In diesem Fall ist CIO 0020 einer Transistor-Ausgangsbaugruppe zugeordnet und an das Bit CIO 002001 ist ein Halbleiterrelais zur Heizungsregelung angeschlossen.



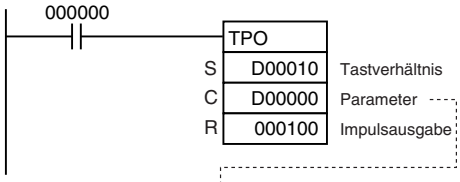
Hinweis Bei Verwendung von TPO(685) in Kombination mit PID(190) in einer zyklischen Tast, wenn zudem auch eine Interrupt Task verwendet wird, sollten Sie die Interrupts durch Ausführung des Befehls DI(693) (DISABLE INTERRUPTS) vor PID(190) und TPO(685) vorübergehend deaktivieren. Wenn die Interrupts nicht deaktiviert sind und ein Interrupt zwischen PID(190) und TPO(685) auftritt, kann das Regelintervall verschoben werden.



Beispiel 2: Alleinige Verwendung von TPO(685)

Wenn CIO 000000 auf EIN gesetzt ist, liest der Befehl TPO(685) das Tastverhältnis in D00010, konvertiert dieses Tastverhältnis in eine zeitproportionale Ausgabe und gibt die Impulse nach CIO 000100 aus.

In diesem Fall, hat das Regelintervall eine Länge von 1 s und die Ausgangsgrenzwert-Funktion ist aktiviert mit einem unteren Grenzwert von 20,00 % und einem oberen Grenzwert von 80,00 %.



TPO(685) liest das Tastverhältnis in D00010, konvertiert dieses Tastverhältnis in eine zeitproportionale Ausgabe und gibt die Ausgangsimpulse an Bit 00 von CIO 0001 aus.

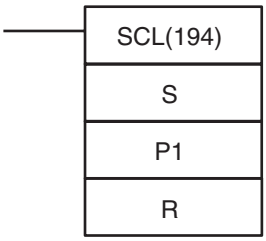
D00000	1	1	0	0	Tastverhältnis-Eingang, Wert zu Beginn lesen, Ausgangsgrenzwert-Funktion aktiviert
D00001	0	0	6	4	Regelintervall = 1,00 s
D00002	0	7	D	0	Ausgangs-Untergrenze = 20,00 %
D00003	1	F	4	0	Ausgangs-Obergrenze = 80,00 %
D00004	Nicht einstellen.				
D00005	Nicht einstellen.				
D00006	Nicht einstellen.				
:					
:					
D00010	0 bis 2710 hex				0 bis 100,00 %

3-18-7 SCALING: SCL(194)

Zweck

Konvertiert Binärdaten ohne Vorzeichen entsprechend der spezifizierten Linearfunktion in BCD-Daten ohne Vorzeichen.

Kontaktplansymbol



S: Quellenwort

P1: Erstes Parameterwort

R: Ergebniswort

Variationen

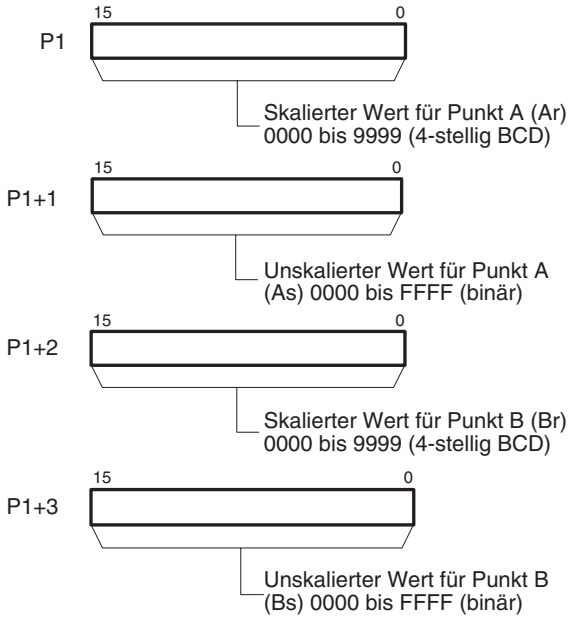
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SCL(194)
	Ausführung einmal bei steigender Flanke	@SCL(194)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Die Inhalte der vier Worte, die mit dem ersten Parameterwort (P1) beginnen, sind in den folgenden Abbildungen dargestellt.



Hinweis P1 bis P1+3 müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	S	P1	R
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6140	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W508	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H508	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A956	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4092	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4092	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32764	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32764	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32764 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	DR0 bis DR15	---	DR0 bis DR15

Bereich	S	P1	R
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl SCL(194) wird verwendet, um die im Quellenwort S enthaltenen Binärdaten ohne Vorzeichen in BCD-Daten ohne Vorzeichen zu konvertieren und das Ergebnis entsprechend der durch zwei Punkte (As, Ad) und (Bs, Bd) definierten Linearfunktion im Ergebniswort R zu speichern. Die Adresse des ersten Worts, das die Koordinaten der Punkte (As, Ar) und (Bs, Br) enthält, wird durch das erste Parameterwort P1 spezifiziert. Diese Punkte werden durch zwei Werte (As und Bs) vor Skalierung und 2 Werte (Ar und Br) nach Skalierung definiert.

Zur Konvertierung werden die folgenden Gleichungen eingesetzt.

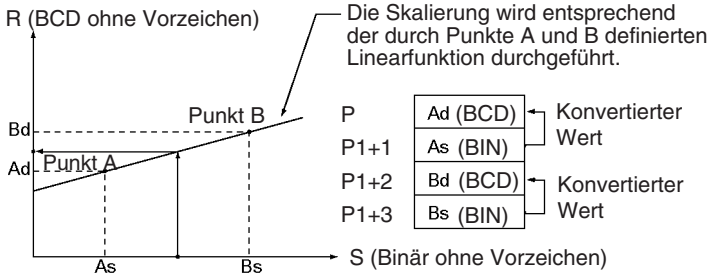
$$R = Bd - \frac{(Bd - Ad)}{\text{BCD-Konvertierung von (Bs – As)}} \times \text{BCD-Konvertierung von (Bs - S)}$$

Die Steigung der Gerade ergibt sich wie folgt:

$$R = Bd - \frac{(Bd - Ad)}{\text{BCD-Konvertierung von (Bs – As)}}$$

Die Punkte A und B können eine Gerade mit positiver oder negativer Steigung definieren. Die Verwendung einer negativen Steigung führt zu einer invertierten Skalierung.

Das Ergebnis wird auf die nächste Ganzzahl gerundet. Wenn das Ergebnis kleiner als 0000 ist, wird 0000 als Ergebnis ausgegeben. Wenn das Ergebnis größer als 9999 ist, wird 9999 ausgegeben.



Der Befehl SCL(194) kann verwendet werden, um die Ergebnisse der Analogsignal-Konvertierung von Werten der Analogeingangs-Baugruppen anhand anwenderdefinierter Skalierungsparameter zu skalieren. Ein Beispiel: Wenn ein Analogsignal im Bereich von 1 bis 5 V, das den Eingang einer Analogeingangs-Baugruppe bildet, als Hexadezimalwert zwischen 0000 und 0FA0 in den Speicher geschrieben wird, kann dieser gespeicherte Wert durch den Befehl SCL(194) auf 50 bis 200°C skaliert werden.

SCL(194) konvertiert vorzeichenlose Binärdaten in vorzeichenlose BCD-Daten. Zur Konvertierung eines negativen Werts ist es erforderlich, im Programm zunächst den maximalen Negativwert zu den Werten zu addieren, bevor der Befehl SCL(194) verwendet wird (siehe Beispiel).

Der Befehl SCL(194) kann keine negativen Werte in das Ergebniswort R ausgeben. Wenn das Ergebnis der Skalierung ein negativer Wert ist, wird 0000 nach R ausgegeben.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt von C (Ar) oder C+1 (Br) kein BCD-Wert ist. EIN, wenn die Inhalte von C+1 (As) und C+3 (Bs) gleich sind. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

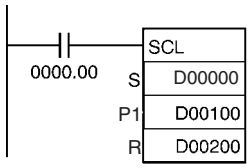
Wenn die Werte für Ar (C) und Br (C+2) keine BCD-Werte sind oder wenn die Werte für As (C+1) und Bs (C+3) gleich sind, tritt ein Fehler aus und der Fehlermerker wird auf EIN gesetzt.

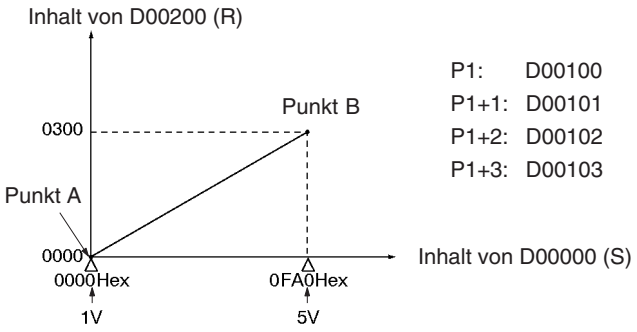
Der Gleich-Merker wird auf EIN gesetzt, wenn der Inhalt des Ergebnisworts D 0000 ist.

Beispiele

In dem folgenden Beispiel wird angenommen, dass ein Analogsignal im Bereich von 1 bis 5 V konvertiert wird und als Hexadezimalwert im Bereich von 0000 bis 0FA0 in D00000 eingegeben wird. Der Befehl SCL(194) wird zur Konvertierung (Skalierung) des Werts in CIO 0200 zu einem BCD-Wert zwischen 0000 und 0300 verwendet.

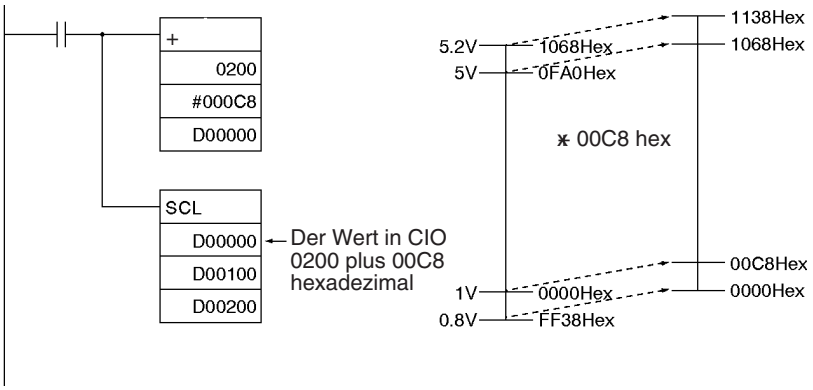
Wenn CIO 000000 auf EIN gesetzt ist, wird der Inhalt von D00000 anhand der durch die Punkte A (0000, 0000) und B (0FA0, 0300) definierten Linearfunktion skaliert. Die Koordinaten dieser Punkte sind in D00100 bis D00103 enthalten und das Ergebnis wird in D00200 ausgegeben.

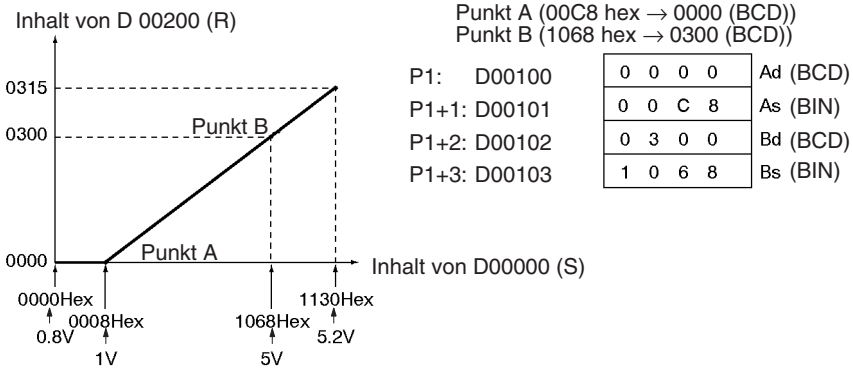




Negative Werte

Tatsächlich gibt eine Analogeingangs-Baugruppe für ein analoges Spannungssignal von 0,8 bis 5,2 V Hexadezimalwerte von FF38 bis 1068 aus. Der Befehl SCL(194) kann jedoch nur vorzeichenlose Binärwerte von 0000 bis FFFF hexadezimal verarbeiten, sodass es nicht möglich ist, den Befehl SCL(194) direkt zu verwenden, um Binärwerte mit Vorzeichen, die einer Spannung von weniger als 1 V entsprechen (0000 hexadezimal) zu verarbeiten, d. h. Hexadezimalwerte zwischen FF38 bis FFFF. In einer tatsächlichen Anwendung ist es daher erforderlich, den Wert 00C8 hexadezimal zu allen Werte zu addieren, sodass der niedrigste Wert, im Beispiel FF38 hexadezimal, als 0000 hexadezimal dargestellt wird, bevor der Befehl SCL(194) verwendet wird (siehe Beispiel).

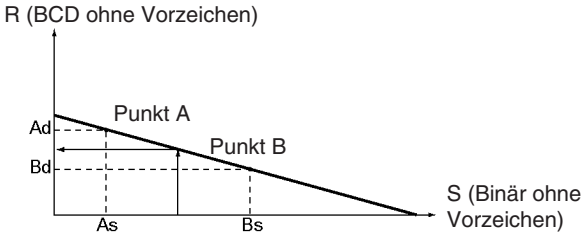




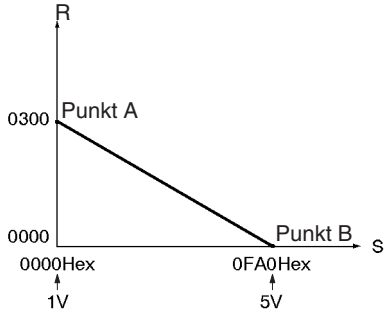
In diesem Beispiel werden Werte zwischen 0000 und 00C8 hexadezimal zu negativen Werten konvertiert. SCL(194) kann jedoch nur vorzeichenlose BCD-Werte zwischen 0000 und 9999 ausgeben, sodass 0000 BCD ausgegeben wird, wenn der Inhalt von D00000 ein Wert zwischen 0000 und 00C8 hexadezimal ist.

Invertierte Skalierung

Es kann auch eine invertierte Skalierung erfolgen, indem As < Bs und Ar > Br eingestellt wird. Daraus ergibt sich die folgende Beziehung.



Die invertierte Skalierung kann beispielsweise verwendet werden, um eine Konvertierung (invertierte Skalierung) von 1 bis 5 V (0000 bis 0FA0 hexadezimal) zu 0300 bis 0000 durchzuführen (siehe folgende Abbildung).

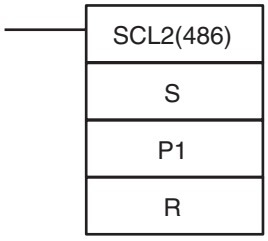


3-18-8 SCALING 2: SCL2(486)

Zweck

Konvertiert Binärdaten mit Vorzeichen entsprechend der spezifizierten Linearfunktion in BCD-Daten mit Vorzeichen. Ein Offset kann bei der Definition der Linearfunktion eingegeben werden.

Kontaktplansymbol



S: Quellenwort

P1: Erstes Parameterwort

R: Ergebniswort

Variationen

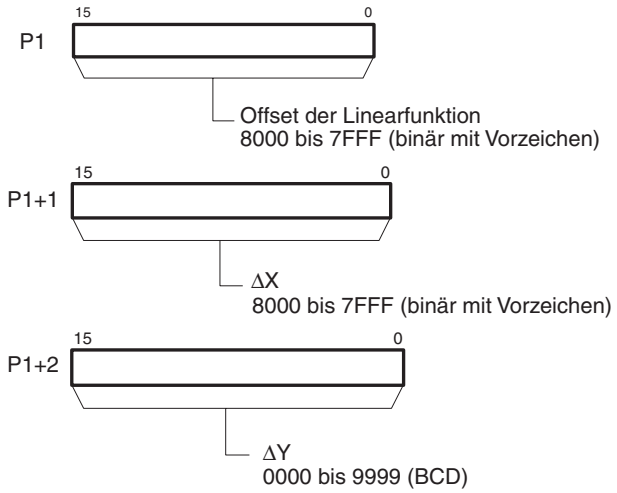
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SCL2(486)
	Ausführung einmal bei steigender Flanke	@SCL2(486)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Die Inhalte der drei Worte, die mit dem ersten Parameterwort (P1) beginnen, sind in den folgenden Abbildungen dargestellt.



Hinweis P1 bis P1+2 müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	S	P1	R
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6141	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W509	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H509	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A957	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4093	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4093	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32765	D00000 bis D32767

Bereich	S	P1	R
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32765	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32765 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl SCL2(486) wird zur Konvertierung der im Quellenwort S enthaltenen Binärdaten mit Vorzeichen in BCD-Daten mit Vorzeichen (die BCD-Daten enthalten den Absolutwert und der Übertragsmerker das Vorzeichen) verwendet. Dabei wird die durch die Steigung (ΔX, ΔY) und ein Offset definierte Linearfunktion angewendet und das Ergebnis wird in das Ergebniswort R geschrieben. Die Adresse des ersten der Worte, die ΔX, ΔY und den Offsetwert enthalten, ist durch das erste Parameterwort P1 spezifiziert. Das Vorzeichen des Ergebnisses wird durch den Status des Übertragsmerkers angegeben (EIN: negativ, AUS: positiv).

Zur Konvertierung werden die folgenden Gleichungen eingesetzt.

$$R = \frac{\Delta Y}{\text{BCD-Konvertierung von } \Delta X} \times ((\text{BCD-Konvertierung von S}) - (\text{BCD-Konvertierung des Offsets})$$

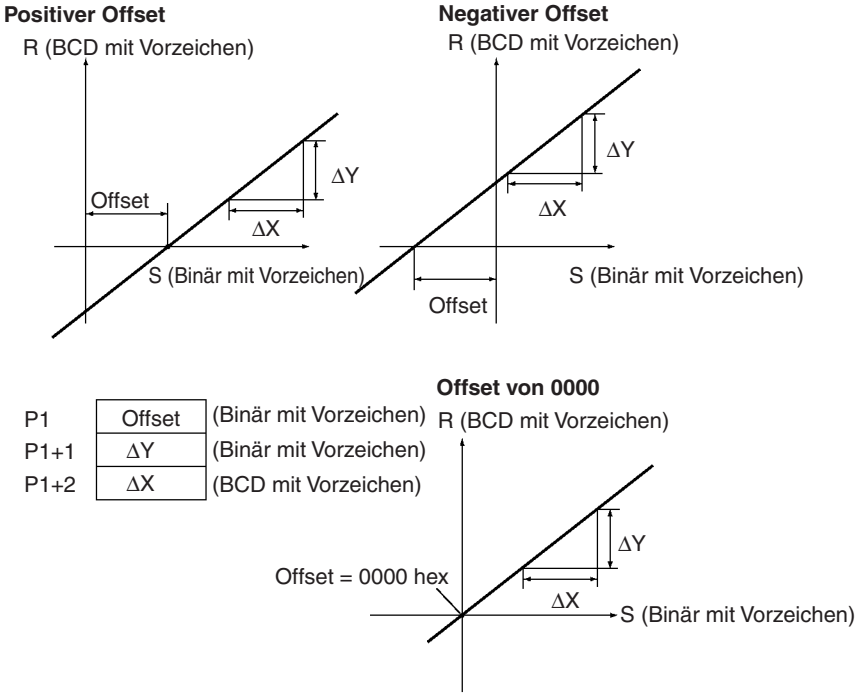
Die Gerade hat die Steigung $\Delta Y/\Delta X$.

Offset und Steigung können einen positiven Wert, 0 oder einen negativen Wert haben. Die Verwendung einer negativen Steigung führt zu einer invertierten Skalierung.

Das Ergebnis wird auf die nächste Ganzzahl gerundet.

Das in R enthaltene Ergebnis ist der Absolutwert der BCD-Konvertierung. Das Vorzeichen wird durch den Übertragsmerker angegeben. Das Ergebnis kann also zwischen –9999 und 9999 liegen.

Wenn das Ergebnis kleiner als –9999 ist, wird –9999 als Ergebnis ausgegeben. Wenn das Ergebnis größer als 9999 ist, wird 9999 ausgegeben.



Der Befehl SCL2(486) kann verwendet werden, um die Ergebnisse der Analogsignal-Konvertierung von Werten der Analogeingangs-Baugruppen anhand anwenderdefinierter Skalierungsparameter zu skalieren. Ein Beispiel: Wenn ein Analogsignal im Bereich von 1 bis 5 V, das den Eingang einer Analogeingangs-Baugruppe bildet, als Hexadezimalwert zwischen 0000 und 0FA0 in den Speicher geschrieben wird, kann dieser gespeicherte Wert durch den Befehl SCL2(486) auf -100 bis 200°C skaliert werden.

SCL2(486) konvertiert Binärwerte mit Vorzeichen zu BCD-Werten mit Vorzeichen. Daher können auch negative Werte für S direkt verarbeitet werden. Das Ergebnis in R und der Übertragsmerker können auch verwendet werden, um negative Werte als Ergebnis der Skalierung auszugeben.

Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn der Inhalt von C+1 (ΔX) 0000 ist. EIN, wenn der Inhalt von C+2 (ΔY) kein BCD-Wert ist. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Übertrags-merker	CY	EIN, wenn das Ergebnis negativ ist. AUS, wenn der Ergebnis Null oder positiv ist.

Vorsichtsmaßnahmen

Wenn ΔX (C+1) den Wert 0000 hat oder wenn ΔY (C+2) kein BCD-Wert ist, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Der Gleich-Merker wird auf EIN gesetzt, wenn der Inhalt des Ergebnisworts D 0000 ist.

Der Übertragsmerker wird auf EIN gesetzt, wenn der im Ergebniswort gespeicherte Wert negativ ist.

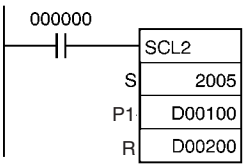
Beispiele

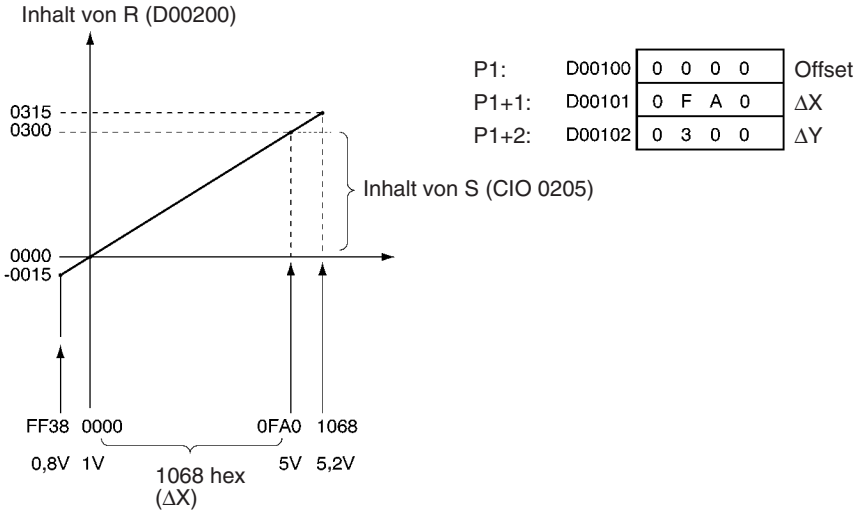
Skalierung des Analogeingangs von 1 bis 5 V auf 0 to 300

In dem folgenden Beispiel wird angenommen, dass ein Analogsignal im Bereich von 1 bis 5 V konvertiert wird und als Hexadezimalwert im Bereich von 0000 bis 0FA0 in CIO 0205 eingegeben wird. Der Befehl SCL2(486) wird zur Konvertierung (Skalierung) des Werts in CIO 0205 zu einem BCD-Wert zwischen 0000 und 0300 verwendet.

Wenn CIO 000000 auf EIN gesetzt ist, wird der Inhalt von CIO 0205 unter Verwendung der durch ΔX (0FA0), ΔY (0300) und den Offset (0) definierten

Linearfunktion skaliert. Diese Werte sind in D00100 bis D00102 enthalten. Das Ergebnis wird in D00200 ausgegeben.

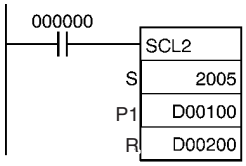


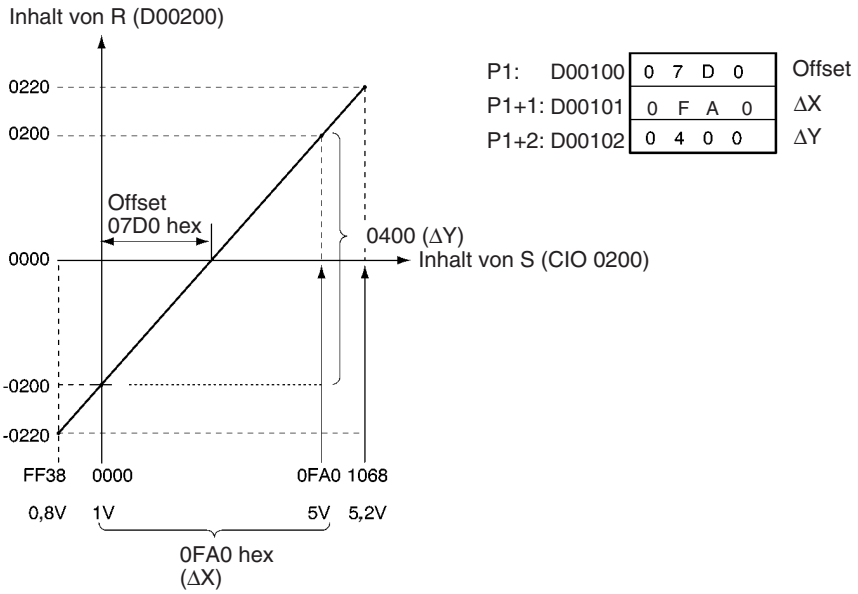


Skalierung des Analogeingangs von 1 bis 5 V auf -200 to 200

In dem folgenden Beispiel wird angenommen, dass ein Analogsignal im Bereich von 1 bis 5 V konvertiert wird und als Hexadezimalwert im Bereich von 0000 bis 0FA0 in CIO 2005 eingegeben wird. Der Befehl SCL2(486) wird zur Konvertierung (Skalierung) des Werts in CIO 0205 zu einem BCD-Wert zwischen –0200 und 0200 verwendet.

Wenn CIO 000000 auf EIN gesetzt ist, wird der Inhalt von CIO 2005 unter Verwendung der durch ΔX (0FA0), ΔY (0400) und den Offset (07D0) definierten Linearfunktion skaliert. Diese Werte sind in D00100 bis D00102 enthalten. Das Ergebnis wird in D00200 ausgegeben.



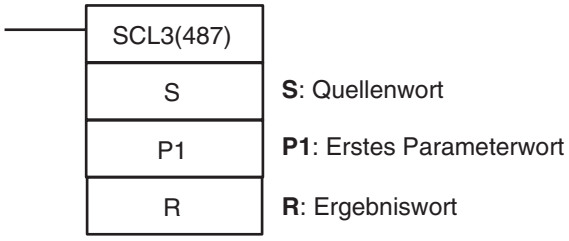


3-18-9 SCALING 3: SCL3(487)

Zweck

Konvertiert BCD-Daten mit Vorzeichen entsprechend der spezifizierten Linearfunktion in Binärdaten mit Vorzeichen. Ein Offset kann bei der Definition der Linearfunktion eingegeben werden.

Kontaktplansymbol



Variationen

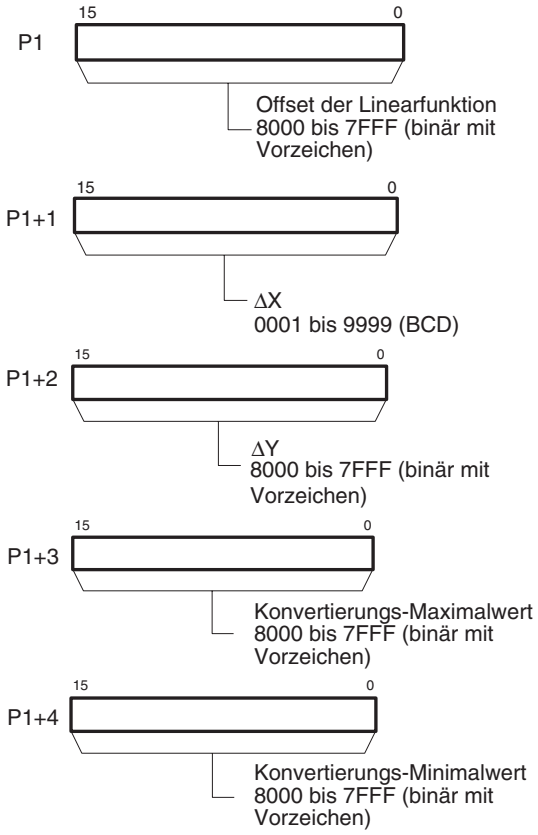
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SCL3(487)
	Ausführung einmal bei steigender Flanke	@SCL3(487)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Die Inhalte der fünf Worte, die mit dem ersten Parameterwort (P1) beginnen, sind in den folgenden Abbildungen dargestellt.



Hinweis P1 bis P1+4 müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	S	P1	R
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6139	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 to W507	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 to H507	H000 bis H511
Zusatz-Systembereich	A000 bis A447 A448 bis A959	A000 bis A443 A448 bis A955	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4091	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4091	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32763	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32763	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32763 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl SCL3(487) wird zur Konvertierung der im Quellenwort S enthaltenen BCD-Daten mit Vorzeichen (die BCD-Daten enthalten den Absolutwert und der Übertragsmerker das Vorzeichen) in Binärdaten mit Vorzeichen verwendet. Dabei wird die durch die Steigung (ΔX, ΔY) und ein Offset definierte Linearfunktion angewendet und das Ergebnis wird in das Ergebniswort R geschrieben. Zudem werden Konvertierungs-Maximalwert und -Minimalwert spezifiziert. Die Adresse des ersten der Worte, die ΔX, ΔY, den Offsetwert, den Konvertierungs-Maximalwert und den Konvertierungs-Minimalwert enthalten, ist durch das erste Parameterwort P1 spezifiziert.

Das Vorzeichen des Ergebnisses wird durch den Status des Übertragsmerkers angegeben (EIN: negativ, AUS: positiv). Verwenden Sie die Befehle STC(040) und CLC(041), um den Übertragsmerker auf EIN bzw. AUS zu setzen.

Zur Konvertierung werden die folgenden Gleichungen eingesetzt.

$$R = \frac{\Delta Y}{\text{Binär-Konvertierung von } \Delta X} \times ((\text{Binär-Konvertierung von S}) + (\text{Offset}))$$

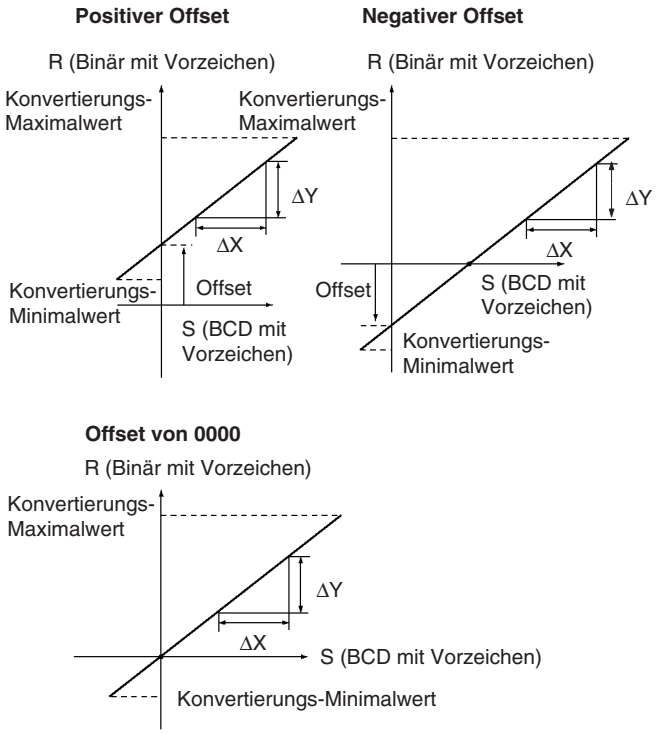
Die Gerade hat die Steigung $\Delta Y/\Delta X$.

Offset und Steigung können einen positiven Wert, 0 oder einen negativen Wert haben. Die Verwendung einer negativen Steigung führt zu einer invertierten Skalierung.

Das Ergebnis wird auf die nächste Ganzzahl gerundet.

Der Quellenwert in S wird als BCD-Absolutwert behandelt, das Vorzeichen wird durch den Übertragsmerker angegeben. Das Quellenwert kann also zwischen –9999 und 9999 liegen.

Wenn das Ergebnis kleiner ist als der Konvertierungs-Minimalwert, wird der Konvertierungs-Minimalwert als Ergebnis ausgegeben. Wenn das Ergebnis größer ist als der Konvertierungs-Maximalwert, wird der Konvertierungs-Maximalwert ausgegeben.



SCL3(487) wird verwendet, um Daten anhand einer anwenderdefinierten Skalierungsfunktion in vorzeichenbehaftete Binärdaten zur Ausgabe an Analogausgangs-Baugruppen zu konvertieren. Beispielsweise kann der Befehl SCL3(487) Daten im Bereich von 0 bis 200°C in Hexadezimaldaten im Bereich von 0000 bis 0FA0 konvertieren, um ein analoges Spannungssignal von 1 bis 5 V über die Analogausgangs-Baugruppe auszugeben.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der Inhalt von S kein BCD-Wert ist. EIN, wenn der Inhalt von C+1 (ΔX) nicht zwischen 0001 und 9999 BCD liegt. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn das Ergebnis 0 ist. In allen anderen Fällen AUS.
Negativ-Merker	N	EIN, wenn das höchstwertige Bit von R (des Ergebnisses) 1 ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn der Inhalt von S kein BCD-Wert ist oder wenn der Wert von ΔX (C+1) nicht zwischen 0001 und 9999 BCD liegt, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

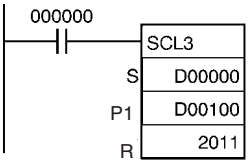
Der Gleich-Merker wird auf EIN gesetzt, wenn der Inhalt des Ergebnisworts D 0000 ist.

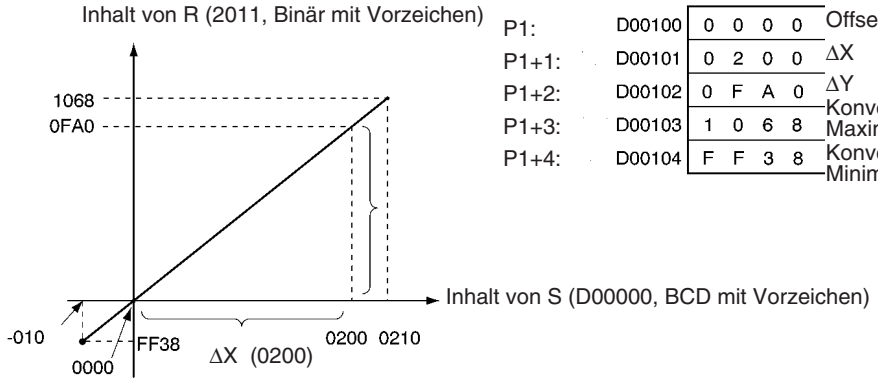
der Negativ-Merker wird auf EIN gesetzt, wenn das höchstwertige Bit des Ergebnisses in R auf 1 gesetzt ist, das Ergebnis also negativ ist.

Beispiele

Wenn ein Wert von 0 bis 200 in ein Analogsignal (z. B. von 1 bis 5 V) skaliert wird, erfolgt eine Konvertierung (Skalierung) eines vorzeichenbehafteten

BCD-Werts von 0000 bis 0200 in einen vorzeichenbehafteten Binärwert von 0000 bis 0FA0 hex zur Ausgabe an die Analogausgangs-Baugruppe. Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird der Inhalt von D00000 unter Verwendung der durch ΔX (0200), ΔY (0FA0) und den Offset (0) definierten Linearfunktion skaliert. Diese Werte sind in D00100 bis D00102 enthalten. Das Vorzeichen des BCD-Werts in D00000 wird durch den Übertragsmerker angegeben. Das Ergebnis wird nach CIO 2011 ausgegeben.



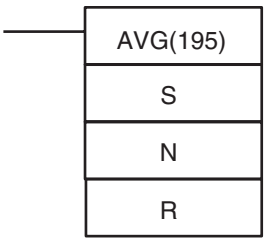


3-18-10 AVERAGE: AVG(195)

Zweck

Berechnet den Mittelwert eines Eingangsworts über die spezifizierte Anzahl von Zyklen.

Kontaktplansymbol



S: Quellenwort

N: Anzahl der Zyklen

R: Ergebniswort
R+1: Erstes Arbeitsbereichswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	AVG(195)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

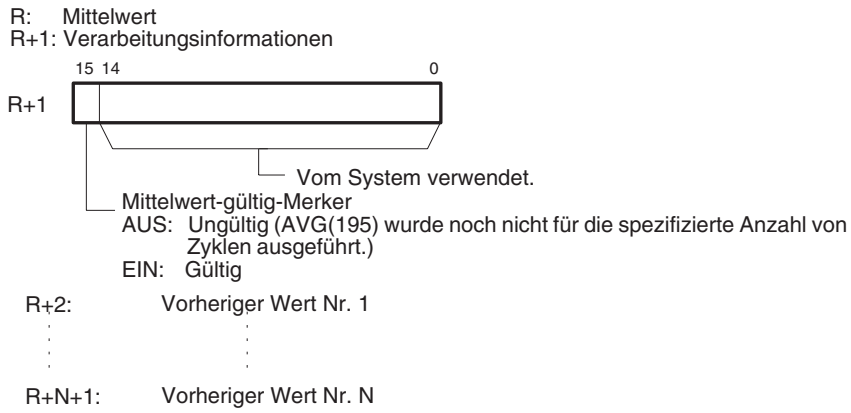
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
Nicht zulässig	OK	OK	OK

Operanden

N: **Anzahl der Zyklen**
Die Anzahl der Zyklen muss zwischen 0001 und 0040 hexadezimal (0 bis 64 Zyklen) liegen.

R: erstes Ergebniswort; R+1: erstes Arbeitsbereichswort

R enthält den Mittelwert nach der spezifizierten Anzahl von Zyklen. R+1 enthält Informationen zur Mittelwertbildung. R+2 bis R+N+1 enthalten die vorherigen Werte von S (siehe folgende Abbildung).



Hinweis R bis R+N+1 müssen im gleichen Datenbereich liegen.

Operanden-Spezifikationen

Bereich	S	N	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerkerbereich	H000 bis H511		
Zusatz-Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	#0001 bis #0040 (binär)	---
Datenregister	DR0 bis DR15		---
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15		

Beschreibung

Während der ersten N+1 Zyklen, in denen die Ausführungsbedingung auf EIN gesetzt ist, schreibt der Befehl AVG(195) die Werte von S der Reihenfolge nach in die mit R+2 beginnenden Worte. Der Vorwertzeiger (Bits 00 bis 07 von R+1) wird jedes Mal inkrementiert, wenn ein Wert geschrieben wurde. Bis der N-te Wert geschrieben ist, wird der Inhalt von S unverändert in R ausgegeben und der Mittelwert-gültig-Merker (Bit 15 von R+1) bleibt auf AUS gesetzt.

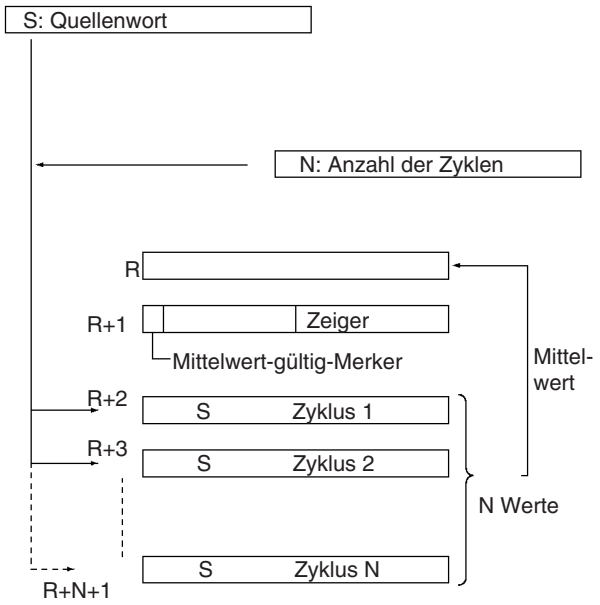
Wenn der N-te Wert in R+N+1 geschrieben ist, wird der Mittelwert aus allen gespeicherten Werten errechnet, dieser Mittelwert wird als Binärwert ohne Vorzeichen nach R ausgegeben und der Mittelwert-gültig-Merker (Bit 15 von

R+1) wird auf EIN gesetzt. In allen folgenden Zyklen wird der Wert in R mit den aktuellsten N Werten von S aktualisiert.

Der Maximalwert für N ist 64.

Der Vorwertzeiger wird auf 0 zurückgesetzt, nachdem N–1 Werte geschrieben wurden.

Der nach R ausgegebene Mittelwert wird auf die nächste Ganzzahl gerundet.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn N den Inhalt 0 hat. In allen anderen Fällen AUS.

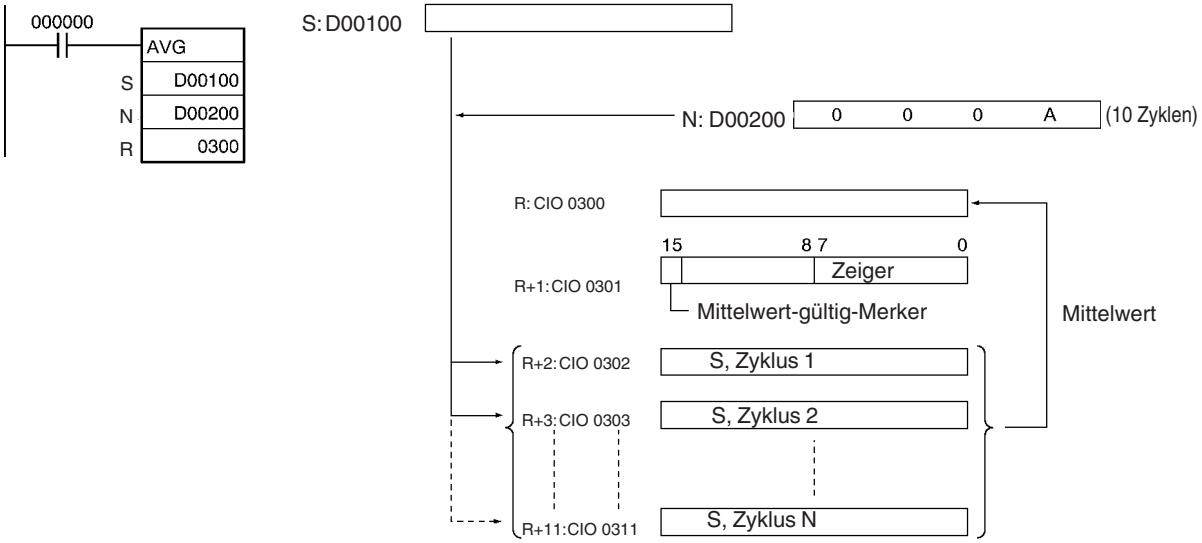
Vorsichtsmaßnahmen

Jedes Mal, wenn die Ausführungsbedingung von AUS zu EIN wechselt, wird der Inhalt des ersten Arbeitsbereichsworts (D+1) auf 0000 gelöscht.

Der Inhalt des ersten Arbeitsbereichsworts (D+1) wird bei der erstmaligen Programmausführung bei Beginn des Betriebs nicht auf 0000 gelöscht. Wenn AVG(195) im ersten Programmzyklus ausgeführt werden soll, muss das erste Arbeitsbereichswort durch das Programm gelöscht werden.

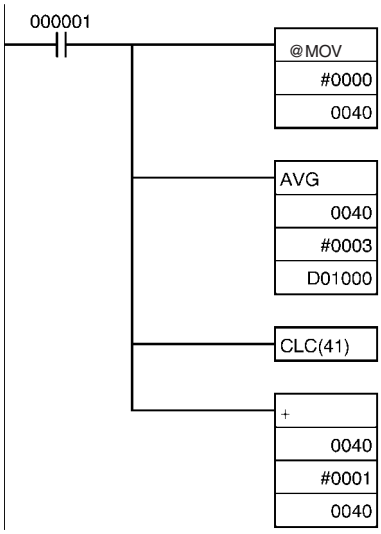
Wenn N (Anzahl der Zyklen) 0000 enthält, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird der Inhalt von D00100 einmal pro Zyklus für die in D00200 spezifizierte Anzahl von Zyklen gespeichert. Der Inhalt wird der Reihenfolge nach in den zehn Worten von CIO 0302 bis CIO 0311 gespeichert. Der Mittelwert des Inhalts dieser zehn Worte wird in CIO 0300 geschrieben und anschließend wird Bit 15 von CIO 0301 auf EIN gesetzt.



Beispiele

In dem folgenden Beispiel ist der Inhalt von CIO 0040 auf #0000 gesetzt und wird dann in jedem Zyklus um 1 inkrementiert. In den ersten beiden Zyklen überträgt AVG(195) den Inhalt von CIO 0040 nach D01002 und D01003. Der Inhalt von D01001 wird ebenfalls geändert (diese Tatsache kann zur Prüfung verwendet werden, ob sich das Ergebnis von AVG(195) geändert hat). Im dritten und den folgenden Zyklen berechnet AVG(195) den Mittelwert aus dem Inhalt von D01002 bis D01004 und schreibt diesen Mittelwert in D01000.



	1. Zyklus	2. Zyklus	3. Zyklus	4. Zyklus
CIO 0040	0000	0001	0002	0003

D01000	0000	0001	0001	0002	Mittelwert
D01001	0001	0002	8000	8001	Zeiger
D01002	0000	0000	0000	0003	3 vorherige Werte von IR 40
D01003	---	0001	0001	0001	
D01004	---	---	0002	0002	

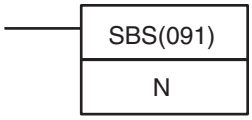
3-19 Unterprogramme

3-19-1 SUBROUTINE CALL: SBS(091)

Zweck

Ruft das Unterprogramm mit der spezifizierten Unterprogrammnummer auf und führt das Programm aus.

Kontaktplansymbol



N: Unterprogramm-Nummer

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SBS(091)
	Ausführung einmal bei steigender Flanke	@SBS(091)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Unterprogrammnummer
Spezifiziert die Unterprogrammnummer zwischen 0 und 1023 dezimal.

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 muss die Unterprogrammnummer im Bereich zwischen &0 und &255 dezimal liegen.

Operanden-Spezifikationen

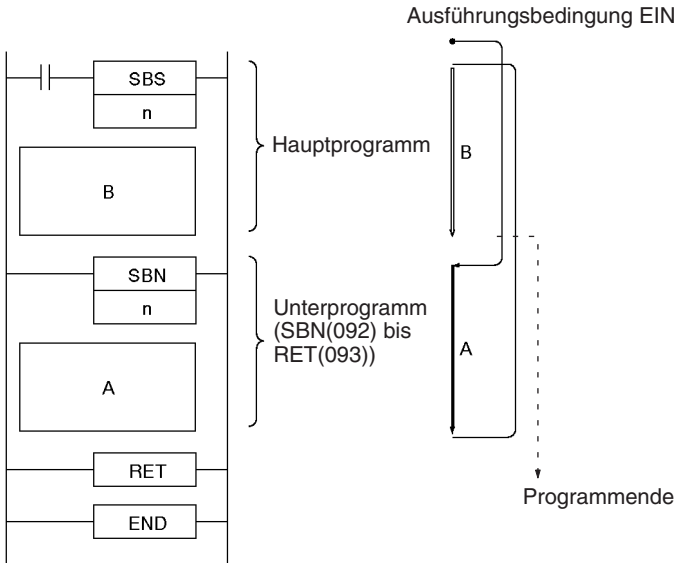
Bereich	N
CIO-Bereich	---
Arbeitsbereich	---
Haftmerkerbereich	---
Zusatz-Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradresse	---
Indirekte DM/EM-Adresse in BCD	---
Konstanten	0 bis 1023 (dezimal) (siehe Hinweis)
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Index-Registern	---

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 liegt der gültige Bereich zwischen &0 und &255 dezimal.

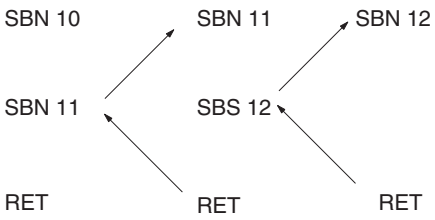
Beschreibung

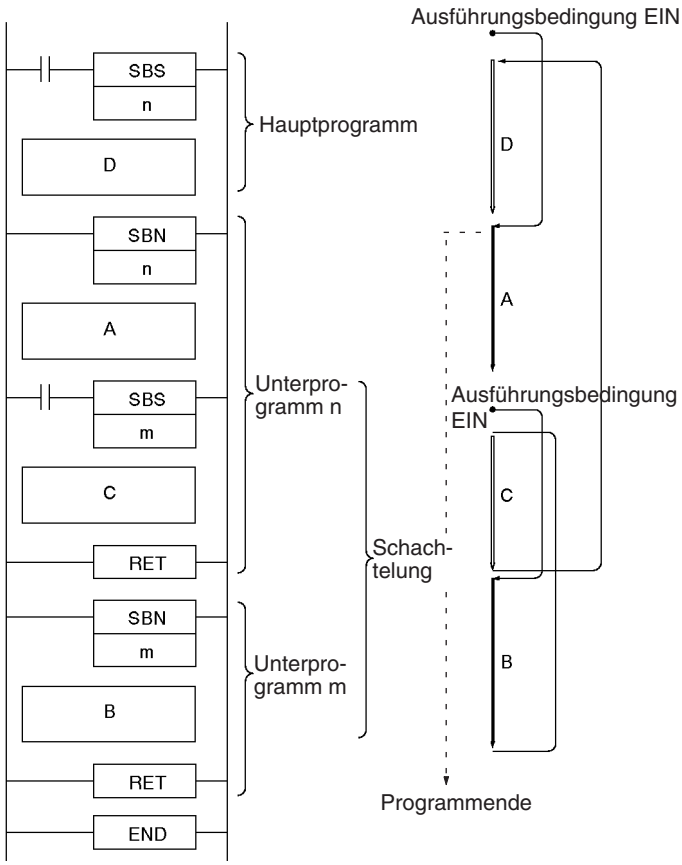
SBS(091) ruft das Unterprogramm mit der spezifizierten Unterprogrammnummer auf. Das Unterprogramm ist der Programmabschnitt zwischen SBN(092)

und RET(093). Wenn die Ausführung des Unterprogramms abgeschlossen ist, fährt die Programmausführung mit dem nächsten Befehl nach SBS(091) fort.



Unterprogramme können in bis zu 16 Ebenen verschachtelt werden. Unter Schachtelung versteht man den Aufruf eines Unterprogramms aus einem anderen Unterprogramm heraus. In der folgenden Abbildung ist ein Beispiel für eine Schachtelung mit drei Ebenen dargestellt.



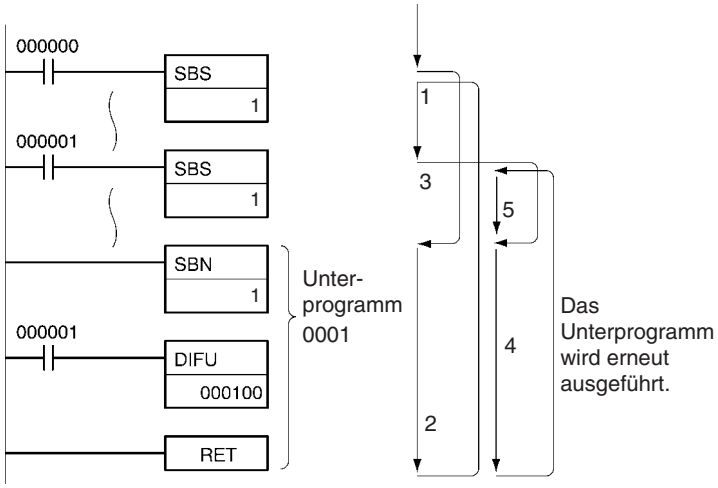


Hinweis Ein Unterprogramm kann mehrmals in einem Programm aufgerufen werden.

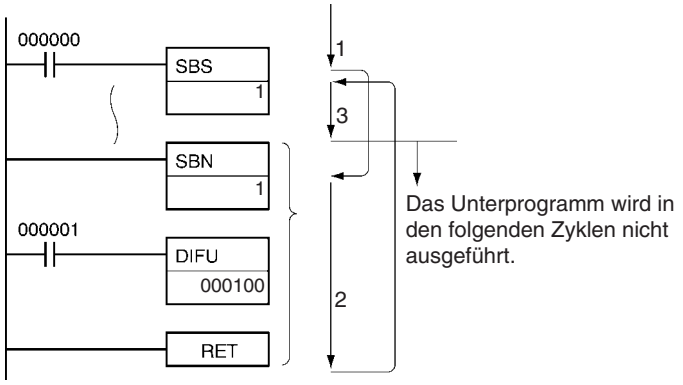
Unterprogramme und Flankenauswertung

Beachten Sie bei Verwendung von Flankenauswertungsbefehlen (DIFU(013), DIFU(014) oder Befehle mit Ausführung an der steigenden/fallenden Flanke) in Unterprogrammen die folgenden Vorsichtsmaßnahmen.

Die Funktionsweise eines Befehls mit Flankenauswertung ist unvorhersehbar, wenn ein Unterprogramm mehr als einmal im gleichen Zyklus ausgeführt wird. Im folgenden Beispiel wird das Unterprogramm 0001 ausgeführt, wenn CIO 000000 auf EIN gesetzt ist und CIO 000100 durch den Befehl DIFU(013) auf EIN gesetzt wird, nachdem ein Wechsel des Zustands von CIO 000001 von AUS zu EIN erfolgt ist. Wenn CIO 000001 im gleichen Zyklus auf EIN gesetzt ist, wird Unterprogramm 0001 erneut ausgeführt. Diesmal setzt der Befehl DIFU(013) CIO 000100 jedoch auf AUS, ohne dass der Status von CIO 000001 geprüft wird.



Umgekehrt bliebe der Ausgang eines Flankenauswertungsbefehls (DIFU(013) oder DIFD(014)) auf EIN gesetzt, wenn der Befehl ausgeführt wurde, der Ausgang auf EIN gesetzt wurde und das gleiche Unterprogramm kein zweites Mal aufgerufen wird.



Im folgenden Beispiel wird das Unterprogramm 0001 ausgeführt, wenn CIO 000000 auf EIN gesetzt ist. Der Ausgang CIO 000100 wird durch DIFU(013) auf EIN gesetzt, wenn der Status von CIO 000001 von AUS zu EIN gewechselt ist. Wenn CIO 000000 im folgenden Zyklus auf AUS gesetzt ist, wird das Unterprogramm 0001 nicht erneut ausgeführt und der Ausgang CIO 000100 bleibt auf EIN gesetzt.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn eine Schachtelung in mehr als 16 Ebenen vorhanden ist. EIN, wenn die spezifizierte Unterprogrammnummer nicht vorhanden ist. EIN, wenn das Unterprogramm sich selbst aufruft. EIN, wenn ein Unterprogramm aufgerufen wird, das ausgeführt wird. EIN, wenn das spezifizierte Unterprogramm nicht in der aktuellen Task definiert ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Befehl SBS(091) und der zugehörige Befehl SBN(092) müssen in der gleichen Task programmiert sein. Wenn der zugehörige Befehl SBN(092) nicht in der gleichen Task vorhanden ist, tritt ein Fehler auf.

Der Befehl SBS(091) wird als NOP(000) behandelt, wenn er in einem durch IL(002) und ILC(003) gesperrten Programmabschnitt liegt.

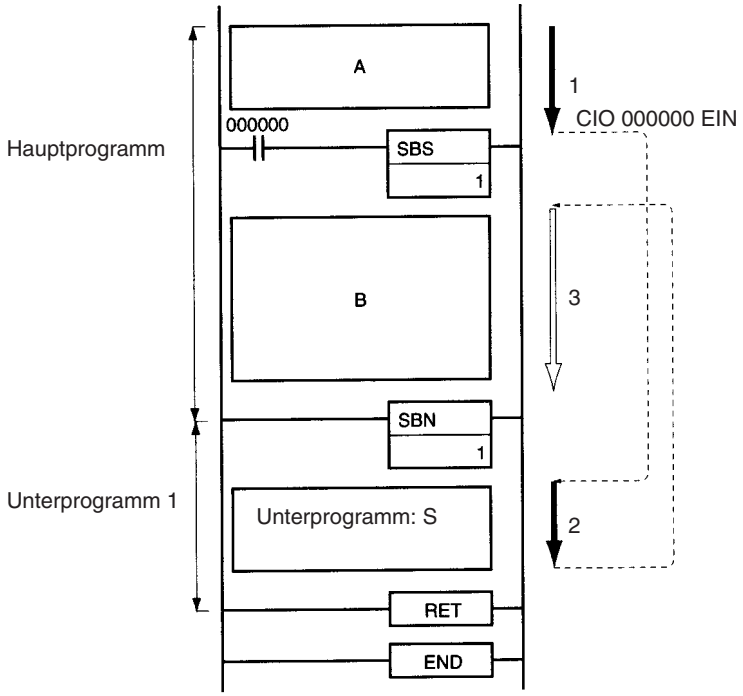
Wenn der Befehl SBS(091) in den folgenden Fällen ausgeführt wird, wird das Unterprogramm tatsächlich nicht aufgerufen und der Fehlermerker wird auf EIN gesetzt:

1,2,3...

1. Das spezifizierte Unterprogramm ist in der aktuellen Task nicht definiert.
2. Das Unterprogramm ruft sich selbst auf.
3. Die Unterprogramme sind in mehr als 16 Ebenen verschachtelt.
4. Das spezifizierte Unterprogramm wird ausgeführt.

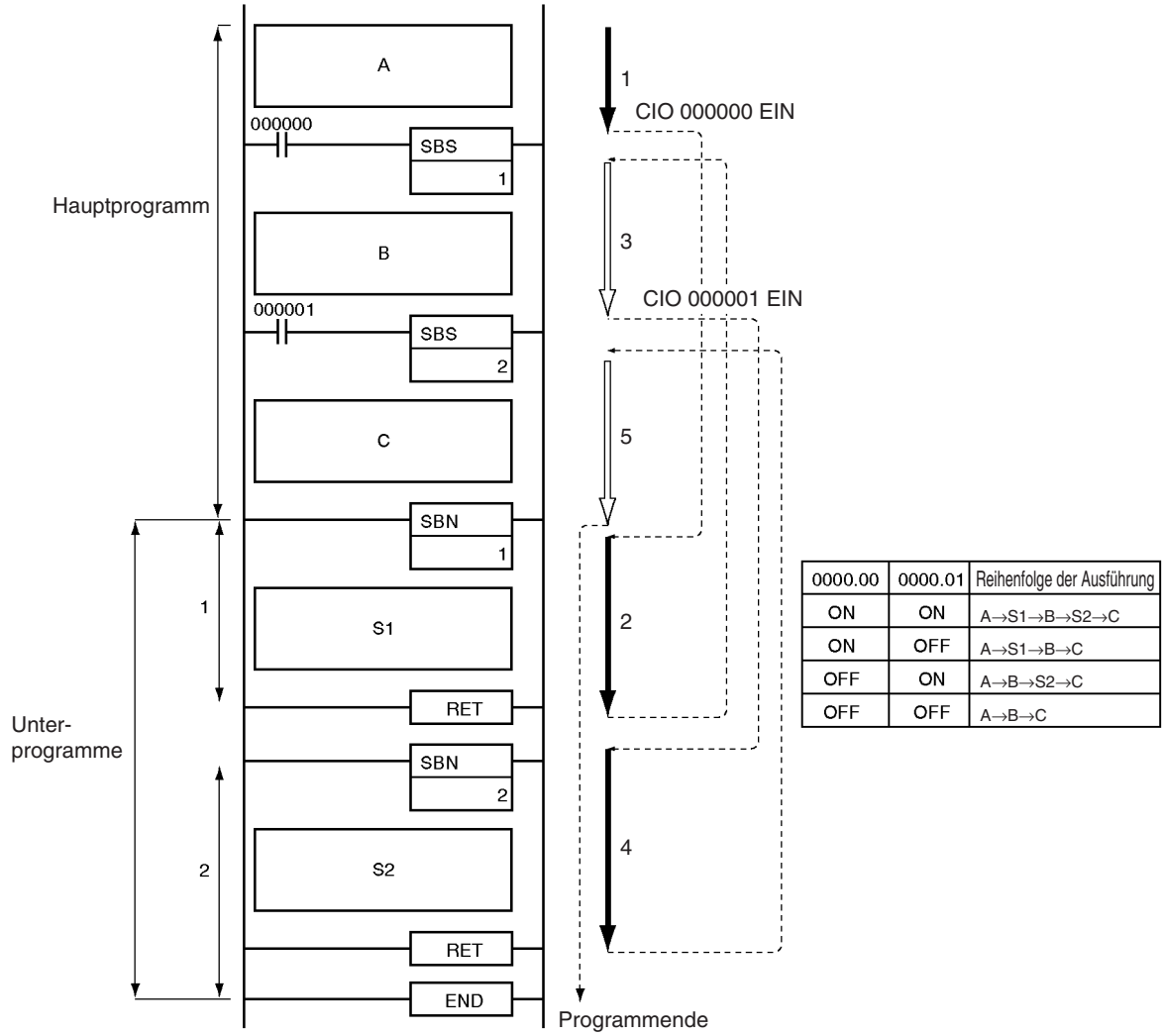
Beispiele

Beispiel 1: Sequentielle (nicht verschachtelte) Unterprogramme
Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird Unterprogramm 1 ausgeführt und die Programmausführung kehrt anschließend zum nächsten Befehl nach SBS(091) zurück. Anschließend wird der Rest des Hauptprogramms (bis zum Befehl unmittelbar vor SBN(092) 1) ausgeführt.



0000.00	Reihenfolge der Ausführung
ON	A→S→B
OFF	A→B

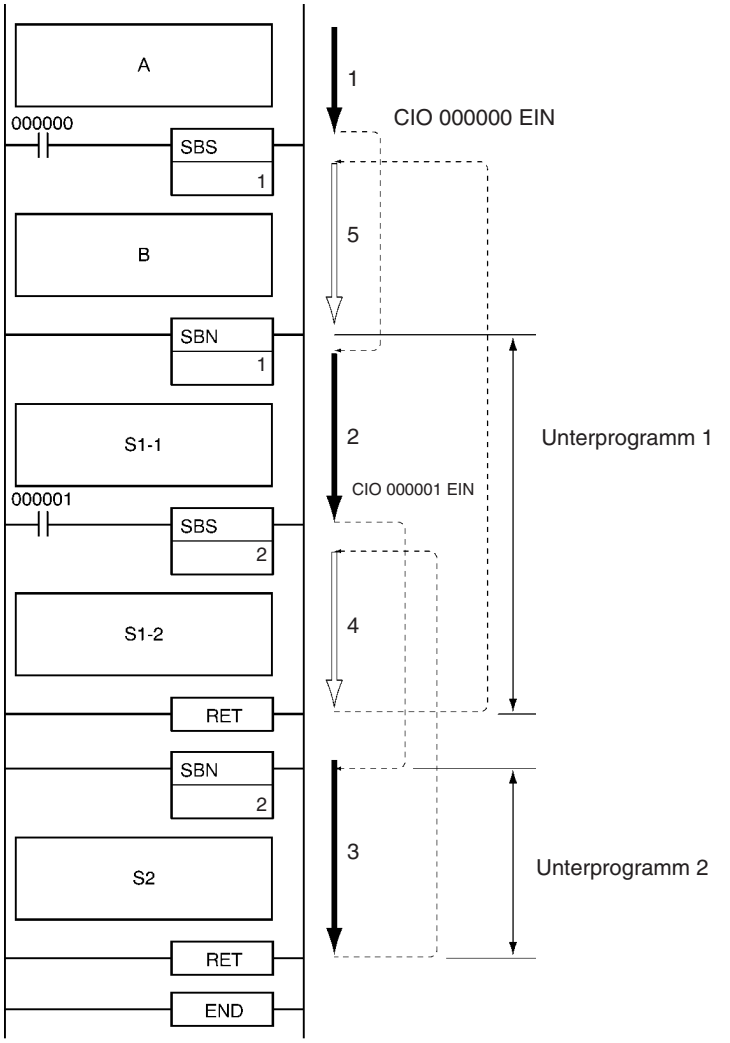
Beispiel 2: Sequentielle (nicht verschachtelte) Unterprogramme
Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird Unterprogramm 1 ausgeführt und die Programmausführung kehrt anschließend zum nächsten Befehl nach SBS(091) 1 zurück. Wenn CIO 000001 auf EIN gesetzt ist, wird Unterprogramm 2 ausgeführt und die Programmausführung kehrt anschließend zum nächsten Befehl nach SBS(091) 2 zurück.



0000.00	0000.01	Reihenfolge der Ausführung
ON	ON	A→S1→B→S2→C
ON	OFF	A→S1→B→C
OFF	ON	A→B→S2→C
OFF	OFF	A→B→C

Beispiel 3: Verschachtelte Unterprogramme

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird Unterprogramm 1 ausgeführt. Wenn CIO 000001 auf EIN gesetzt ist, wird Unterprogramm 2 aus Unterprogramm 1 heraus ausgeführt und die Programmausführung kehrt zum nächsten Befehl nach SBS(091) 2 zurück, wenn die Ausführung von Unterprogramm 2 abgeschlossen ist. Anschließend wird die Ausführung von Unterprogramm 1 fortgesetzt und die Programmausführung kehrt zum nächsten Befehl nach SBS(091) 1 zurück, wenn die Ausführung von Unterprogramm 1 abgeschlossen ist.



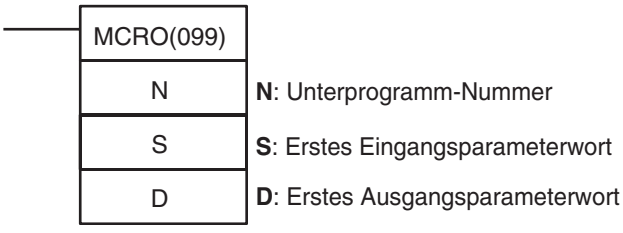
0000.00	0000.01	Reihenfolge der Ausführung
ON	ON	A→S1-1→S2→S1-2→B
ON	OFF	A→S1-1→S1-2→B
OFF	ON	A→B
OFF	OFF	A→B

3-19-2 MACRO: MCRO(099)

Zweck

Ruft das Unterprogramm mit der spezifizierten Unterprogramm-Nummer auf und führt das Programm unter Verwendung der Eingangsparameter in S bis S+3 und der Ausgangsparameter in D bis D+3 aus.

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	MCRO(099)
	Ausführung einmal bei steigender Flanke	@MCRO(099)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Unterprogrammnummer
Spezifiziert die Unterprogrammnummer zwischen 0 und 1023 dezimal.

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 muss die Unterprogrammnummer im Bereich zwischen 0 und 255 dezimal liegen.

Operanden-Spezifikationen

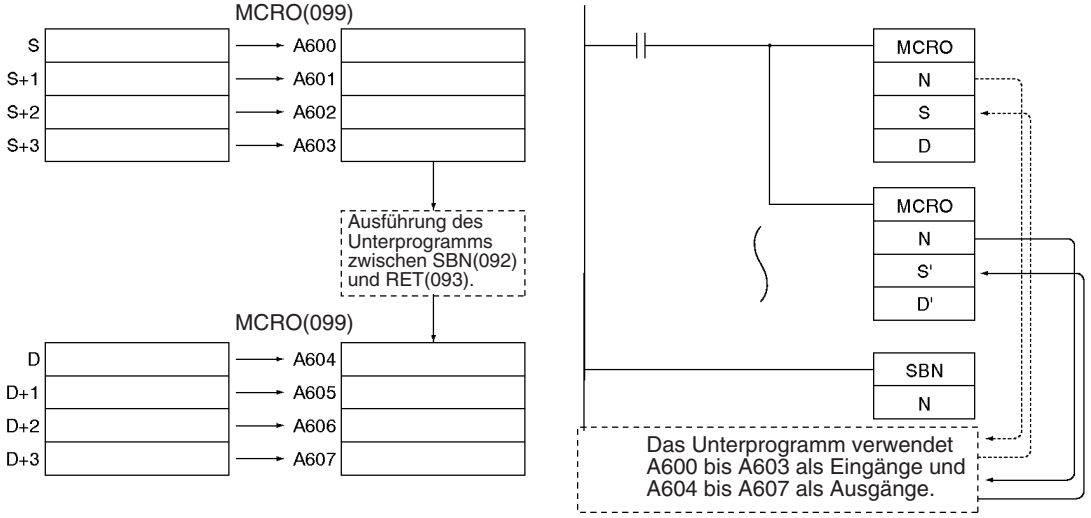
Bereich	N	S	D
CIO-Bereich	---	CIO 0000 bis CIO 6140	
Arbeitsbereich	---	W000 bis W508	
Haftmerkerbereich	---	H000 bis H508	
Zusatz-Systembereich	---	A000 bis A444 A448 bis A956	A448 bis A956
Zeitgeberbereich	---	T0000 bis T4092	
Zählerbereich	---	C0000 bis C4092	
DM-Bereich	---	D00000 bis D32764	
EM-Bereich ohne Bank	---	E00000 bis E32764	
EM-Bereich mit Bank	---	En_00000 bis En_32764 (n = 0 bis C)	
Indirekte DM/EM-Binäradresse	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	0 bis 1023 (dezimal) (siehe Hinweis)	---	
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15, IR0+(++) bis IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 liegt der gültige Bereich zwischen 0 und 255 dezimal.

Beschreibung

MCRO(099) ruft das Unterprogramm mit der spezifizierten Unterprogrammnummer genauso wie der Befehl SBS(091) auf. Im Gegensatz zu SBS(091) können beim Befehl MCRO(099) die Operanden S und D verwendet werden, um Bit- oder Wortadressen in dem Unterprogramm zu ändern, ohne dabei die Struktur des Unterprogramms zu verändern.

Wenn MCRO(099) ausgeführt wird, werden die Inhalte von S bis S+3 in A600 bis A603 (Makrobereich-Eingänge) kopiert, und das spezifizierte Unterprogramm wird ausgeführt. Wenn die Ausführung des Unterprogramms abgeschlossen ist, werden die Inhalte von A604 bis A607 (Makrobereich-Ausgänge) in D bis D+3 kopiert, und die Programmausführung wird mit dem nächsten Befehl nach MCRO(099) fortgesetzt.



Der Befehl MCRO(099) kann verwendet werden, um zwei oder mehr Unterprogramme in einem einzigen Unterprogramm zusammenzulegen, die zwar die gleiche Struktur, jedoch unterschiedliche Eingangs- und Ausgangsadressen haben. Bei der Ausführung von MCRO(099) werden die spezifizierten Eingangs- und Ausgangsdaten in das spezifizierte Unterprogramm übertragen.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn eine Schachtelung in mehr als 16 Ebenen vorhanden ist. EIN, wenn die spezifizierte Unterprogrammnummer nicht vorhanden ist. EIN, wenn das Unterprogramm sich selbst aufruft. EIN, wenn ein Unterprogramm aufgerufen wird, das ausgeführt wird. EIN, wenn das spezifizierte Unterprogramm nicht in der aktuellen Task definiert ist. In allen anderen Fällen AUS.

Die nachfolgende Tabelle zeigt die relevanten Worte im Zusatz-Systembereich.

Name	Adresse	Funktion
Makrobereich-Eingangsworte	A600 bis A603	Bei der Ausführung von MCRO(099) wird der Inhalt der vier Worte von S bis S+3 in A600 bis A603 kopiert. Diese Eingangsworte werden an das Unterprogramm übergeben.
Makrobereich-Eingangsworte	A604 bis A607	Nach Ausführung des in MCRO(099) spezifizierten Unterprogramms werden die Ausgangsdaten in diesen Ausgangsworten nach D bis D+3 kopiert.

Vorsichtsmaßnahmen

Die vier Worte mit Eingangsdaten (Worte oder Bits) in A600 bis A603 und die vier Worte mit Ausgangsdaten (Worte oder Bits) in A604 bis A607 müssen in dem durch MCRO(099) aufgerufenen Unterprogramm verwendet werden. Ein Übergeben von mehr als vier Datenworten ist nicht möglich.

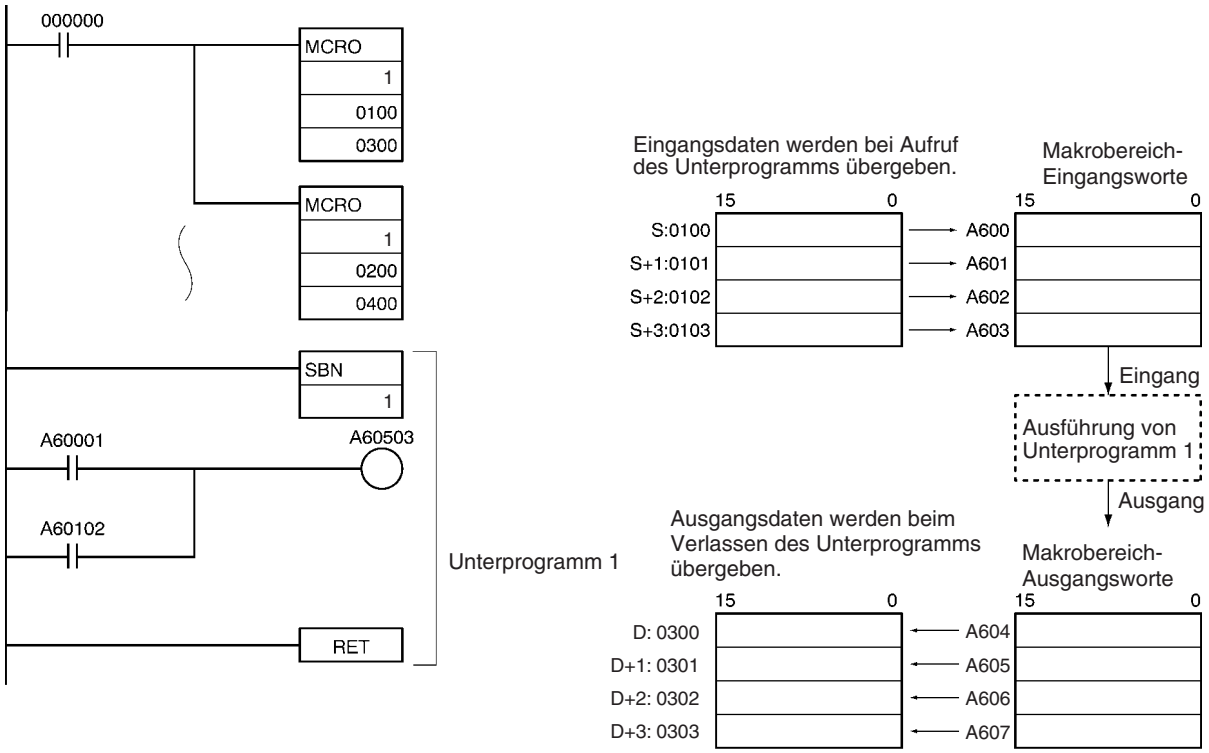
Eine Schachtelung von MCRO(099)-Befehlen ist möglich, jedoch müssen die Daten in den Makrobereichs-Eingangsworten und -Ausgangsworten (A600 bis A607) gespeichert werden, bevor ein anderes Unterprogramm aufgerufen wird, da alle MCRO(099)-Befehle die gleichen 8 Worte verwenden.

Beispiel

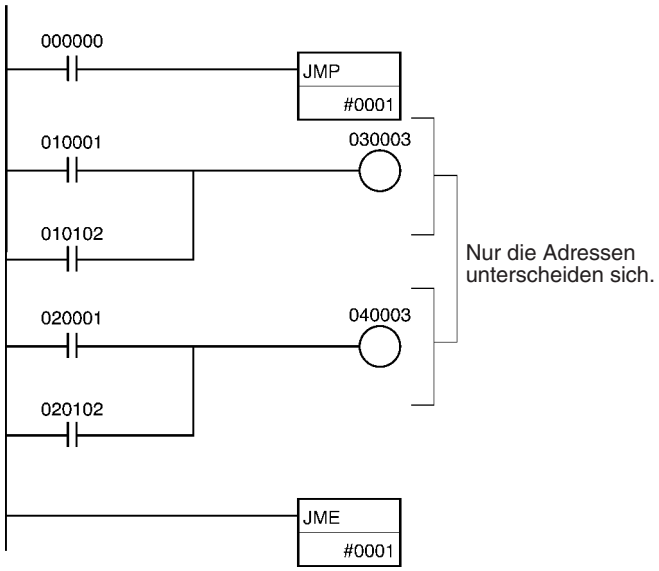
Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, übergeben zwei MCRO(099)-Befehle verschiedenen Eingangs- und Ausgangsdaten an Unterprogramm 1.

1,2,3... 1. Der erste MCRO(099)-Befehl übergibt die Eingangsdaten in CIO 0100 bis CIO 0103 und führt das Unterprogramm aus. Nach Abschluss der Ausführung des Unterprogramms werden die Ausgangsdaten in CIO 0300 bis CIO 0303 gespeichert.

2. Der zweite MCRO(099)-Befehl übergibt die Eingangsdaten in CIO 0200 bis CIO 0203 und führt das Unterprogramm aus. Nach Abschluss der Ausführung des Unterprogramms werden die Ausgangsdaten in CIO 0400 bis CIO 0403 gespeichert.



Der zweite MCRO(099)-Befehl funktioniert auf die gleiche Weise, jedoch werden die Eingangsdaten in CIO 0200 bis CIO 0203 in A600 bis A603 übergeben, und die Ausgangsdaten in A604 bis A607 werden in CIO 0400 bis CIO 0403 übergeben.

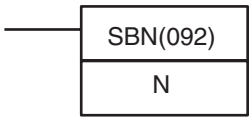


3-19-3 SUBROUTINE ENTRY: SBN(092)

Zweck

Kennzeichnet den Beginn des Unterprogramms mit der spezifizierten Unterprogramm-Nummer. Der Befehl wird in Kombination mit dem Befehl RET(093) zur Definition eines Unterprogramm-Bereichs verwendet.

Kontaktplansymbol



N: Unterprogramm-Nummer

Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	SBN(092)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	Nicht zulässig	OK	OK

Operanden

N: Unterprogrammnummer
Spezifiziert die Unterprogrammnummer zwischen 0 und 1023 dezimal.

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 muss die Unterprogrammnummer im Bereich zwischen 0 und 255 dezimal liegen.

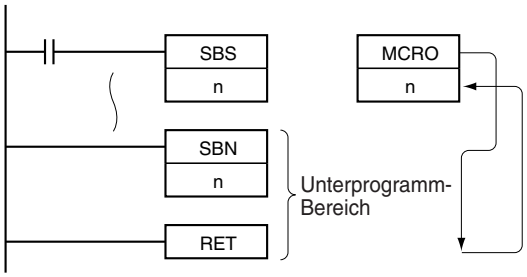
Operanden-Spezifikationen

Bereich	N
CIO-Bereich	---
Arbeitsbereich	---
Haftmerkerbereich	---
Zusatz-Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradresse	---
Indirekte DM/EM-Adresse in BCD	---
Konstanten	0 bis 1023 (dezimal)
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Index-Registern	---

Beschreibung

Der Befehl SBN(092) kennzeichnet den Beginn des Unterprogramms mit der spezifizierten Unterprogramm-Nummer. Das Ende des Unterprogramms wird durch den Befehl RET(093) gekennzeichnet.

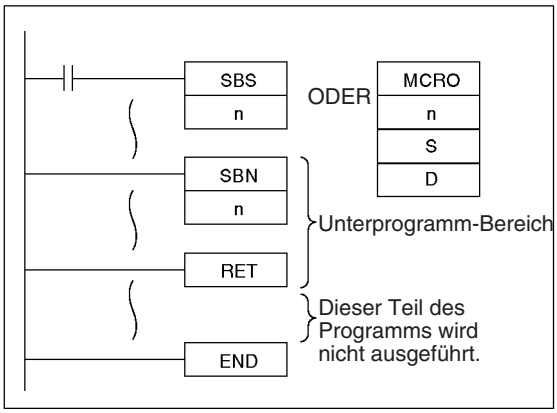
Der Bereich des Programms, der mit dem ersten SBN(092)-Befehl beginnt, ist der Unterprogramm-Bereich. Ein Unterprogramm wird nur ausgeführt, wenn es durch die Befehle SBS(091) oder MCRO(099) aufgerufen wird.



Vorsichtsmaßnahmen

Wenn das Unterprogramm nicht aufgerufen wird, werden die enthaltenen Befehle als NOP(000) behandelt.

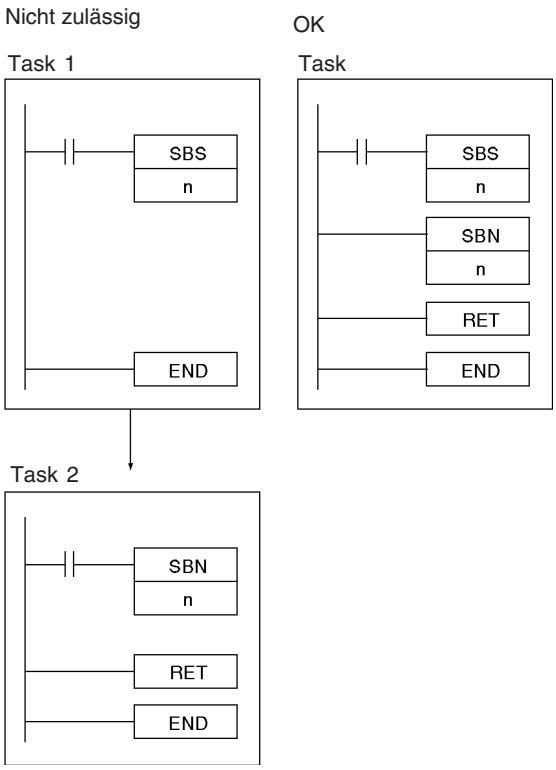
Platzieren Sie die Unterprogramme im Programm jeder Task hinter dem Hauptprogramm und unmittelbar vor dem END(001)-Befehl. Wenn ein Teil des Hauptprogramms hinter dem Unterprogramm-Bereich platziert wird, wird dieser Teil des Programms ignoriert.



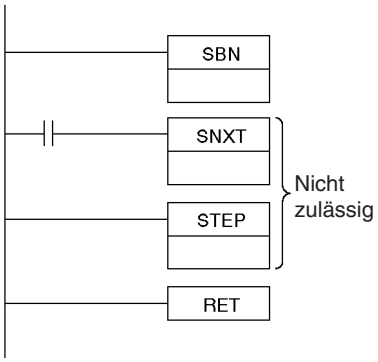
Hinweis

Die Methode zur Eingabe der Unterprogramm-Nummer N unterscheidet sich zwischen CX-Programmer und einer Programmierkonsole. Geben Sie bei CX-Programmer #0 bis #1023 ein und bei einer Programmierkonsole 0000 bis 1023.

Achten Sie darauf, alle Unterprogramme im gleichen Programm (Task) zu platzieren, in dem sich der zugehörigen Befehl SBS(091) bzw. MCRO(099) befindet. Ein Unterprogramm kann nicht aus einer anderen Task heraus aufgerufen werden. Es ist möglich, ein Unterprogramm in einer Interrupt-Task zu programmieren.

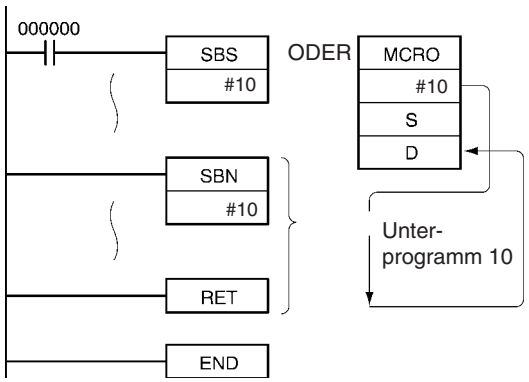


Die Schrittbefehle, STEP(008) und SNXT(009), können nicht in Unterprogrammen verwendet werden.



Beispiel

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird Unterprogramm 10 ausgeführt und die Programmausführung kehrt anschließend zum nächsten Befehl nach dem Befehl SBS(091) bzw. MCRO(099) zurück, der das Unterprogramm aufgerufen hat.



3-19-4 SUBROUTINE RETURN: RET(093)

Zweck

Kennzeichnet das Ende eines Unterprogramms. Der Befehl wird in Kombination mit dem Befehl SBN(092) zur Definition eines Unterprogramm-Bereichs verwendet.

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	RET(093)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	Nicht zulässig	OK	OK

Beschreibung

Der Befehl RET(093) kennzeichnet das Ende eines Unterprogramms, dessen Anfang durch den Befehl SBN(092) gekennzeichnet ist. Weitere Informationen zu den Funktionen von Unterprogrammen finden Sie unter *3-19-3 SUBROUTINE ENTRY: SBN(092)*.

Wenn die Programmausführung den Befehl RET(093) erreicht, kehrt sie automatisch zu dem nächsten Befehl nach dem Befehl SBS(091) bzw. MCRO(099) zurück, der das Unterprogramm aufgerufen hat. Wenn das Unterprogramm durch den Befehl MCRO(099) aufgerufen wurde, werden die Ausgangsdaten in A604 bis A607 in D bis D+3 geschrieben, bevor die Programmausführung zum Hauptprogramm zurückkehrt.

Vorsichtsmaßnahmen

Wenn das Unterprogramm nicht aufgerufen wird, werden die enthaltenen Befehle als NOP(000) behandelt.

Beispiel

Beispiele für die Funktionsweise des Befehls RET(093) finden Sie unter *3-19-3 SUBROUTINE ENTRY: SBN(092)*.

3-19-5 GLOBAL SUBROUTINE CALL: GSBS(750)

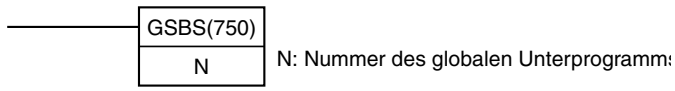
Zweck

Ruft das globale Unterprogramm mit der spezifizierten Unterprogrammnummer auf und führt das Programm aus. Das gleiche globale Unterprogramm kann aus mehreren verschiedenen Tasks aufgerufen werden.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Der Befehl GSBS(750) wird in Kombination mit den Befehlen GSBN(751) und GRET(752) verwendet, den Befehlen GLOBAL SUBROUTINE ENTRY (Globales Unterprogramm - Anfang) und GLOBAL SUBROUTINE RETURN (Globales Unterprogramm verlassen).

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	GSBS(750)
	Ausführung einmal bei steigender Flanke	@GSBS(750)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Nummer des globalen Unterprogramms
Spezifiziert die Nummer des globalen Unterprogramms zwischen 0 und 1023 dezimal.

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 muss die Unterprogrammnummer im Bereich zwischen 0 und 255 dezimal liegen.

Operanden-Spezifikationen

Bereich	N
CIO-Bereich	---
Arbeitsbereich	---
Haftmerkerbereich	---
Zusatz-Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradresse	---
Indirekte DM/EM-Adresse in BCD	---
Konstanten	0 bis 1023 (dezimal) (siehe Hinweis)
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Index-Registern	---

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 liegt der gültige Bereich zwischen 0 und 255 dezimal.

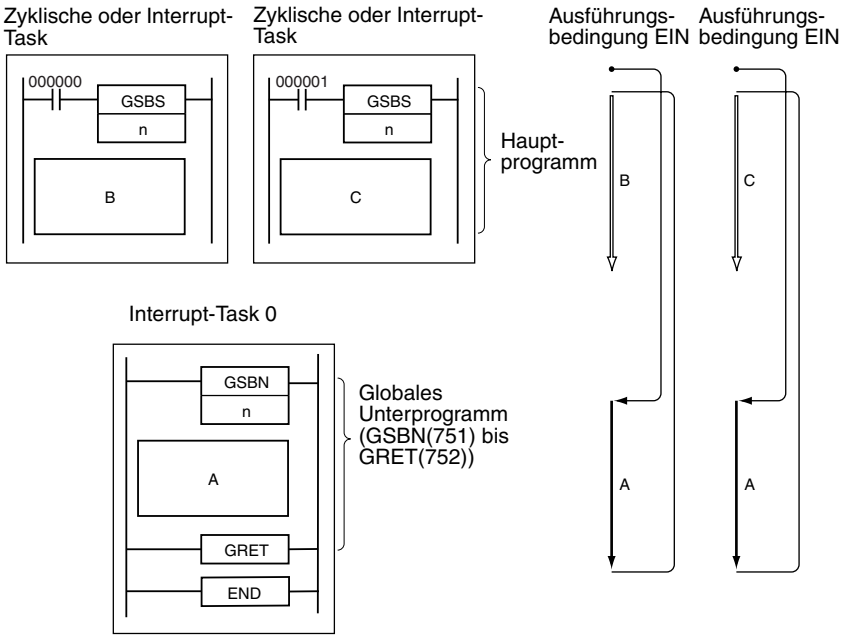
Beschreibung

Der Befehl GSBS(750) ruft das globale Unterprogramm mit der spezifizierten globalen Unterprogrammnummer auf. Das globale Unterprogramm ist der Programmabschnitt zwischen GSBN(751) und GRET(752). Wenn die Ausführung des globalen Unterprogramms abgeschlossen ist, fährt die Programmausführung mit dem nächsten Befehl nach GSBS(750) fort.

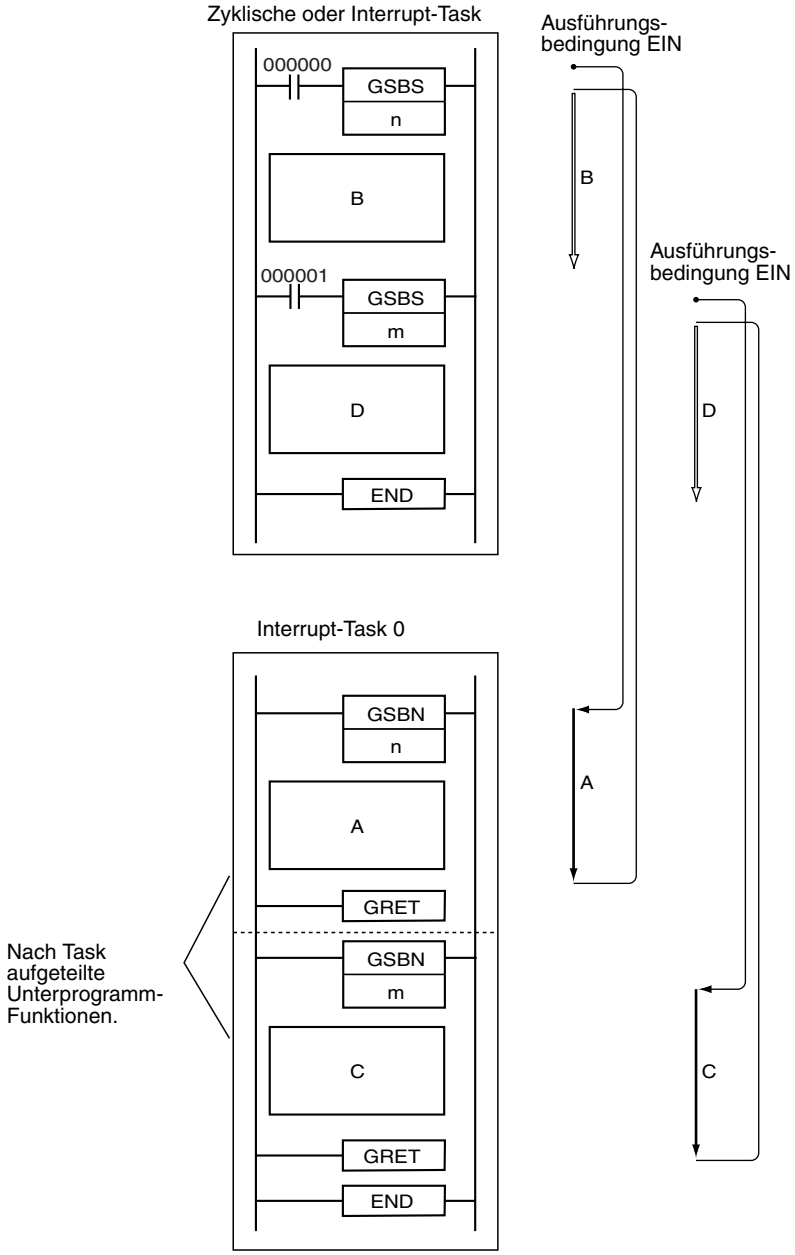
Dieser Befehl kann mit der gleichen globalen Unterprogrammnummer in mehreren verschiedenen Tasks programmiert werden, um dieses globale Unterprogramm aus verschiedenen Tasks heraus aufzurufen. Das Programm kann modularisiert werden, wenn globale Unterprogramme als Standard-Unterprogramme verwendet werden, die in vielen Tasks gemeinsam vorkommen.

Der Programmbereich des globalen Unterprogramms (zwischen GSBN(751) und GRET(752)) muss in Interrupt-Task 0 definiert werden. Wenn dieser Bereich in einer anderen Task definiert wird, tritt bei der Ausführung des Befehls GSBS(750) ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

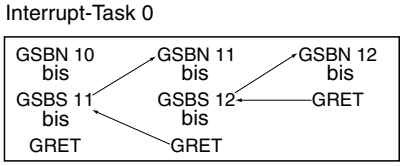
Der Befehl GSBS(750) kann sowohl in zyklischen Tasks (einschließlich zyklischer Zusatztasks) als auch in Interrupt-Tasks programmiert werden.



In Interrupt-Task 0 können mehrere Programmbereiche für globale Unterprogramme (zwischen GSBN(751) und GRET(752)) definiert werden.



Der Befehl SBS(091) oder GSBS(750) kann innerhalb eines Unterprogramm-Bereichs (SBN(092) bis RET(093)) oder globalen Unterprogramm-Bereichs (GSBN(751) bis GRET(752)) programmiert werden, um Unterprogramme zu „verschachteln“. Unterprogramme können in bis zu 16 Ebenen verschachtelt werden.

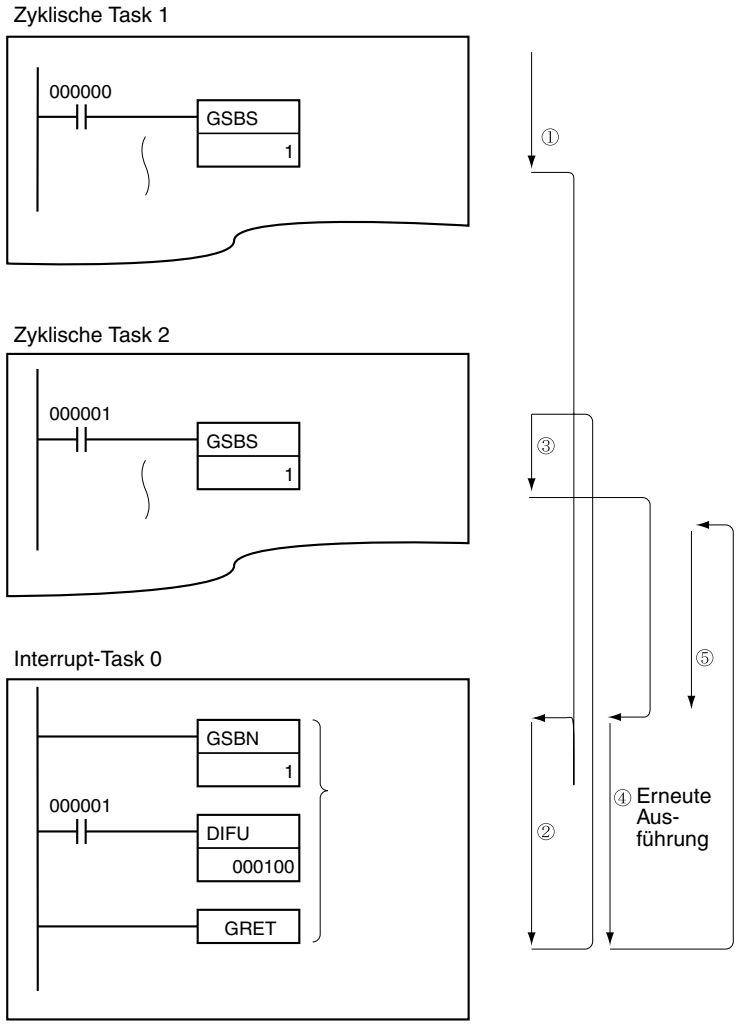


Globale Unterprogramme und Flankenauswertung

Beachten Sie bei Verwendung von Flankenauswertungsbefehlen (DIFU(013), DIFU(014) oder Befehle mit Ausführung an der steigenden/fallenden Flanke) in Unterprogrammen die folgenden Vorsichtsmaßnahmen.

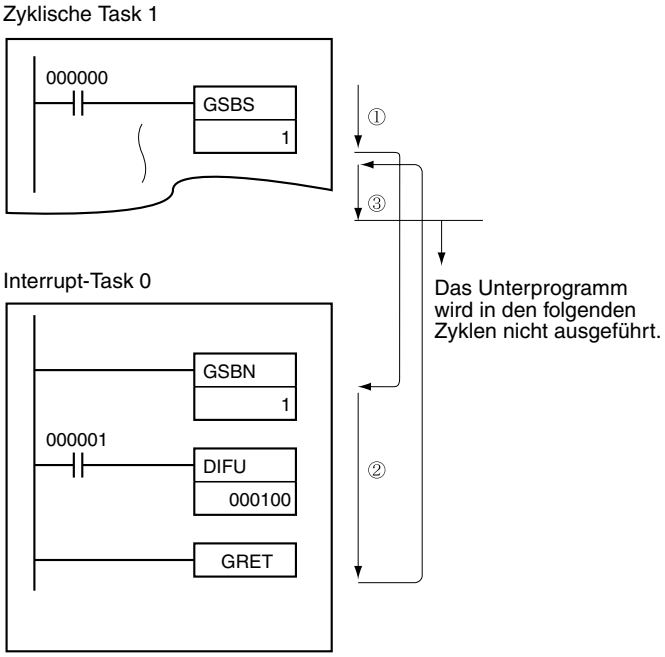
Die Funktionsweise eines Befehls mit Flankenauswertung ist unvorhersehbar, wenn ein globales Unterprogramm mehr als einmal im gleichen Zyklus ausgeführt wird. Im folgenden Beispiel wird das globale Unterprogramm 0001 ausgeführt, wenn CIO 000000 auf EIN gesetzt ist und CIO 000100 durch den

Befehl DIFU(013) auf EIN gesetzt wird, nachdem ein Wechsel des Zustands von CIO 000001 von AUS zu EIN erfolgt ist. Wenn CIO 000001 im gleichen Zyklus auf EIN gesetzt ist, wird das globale Unterprogramm 0001 erneut ausgeführt, jedoch erkennt der Befehl DIFU(013) keine steigende Flanke von CIO 000001, und daher wird CIO 000100 auf AUS gesetzt.



Umgekehrt bliebe der Ausgang eines Flankenauswertungsbefehls (DIFU(013) oder DIFD(014)) auf EIN gesetzt, wenn der Befehl ausgeführt wurde, der Ausgang auf EIN gesetzt wurde und das gleiche globale Unterprogramm kein zweites Mal aufgerufen wird.

Im folgenden Beispiel wird das globale Unterprogramm 0001 ausgeführt, wenn CIO 000000 auf EIN gesetzt ist. Der Ausgang CIO 000100 wird durch DIFU(013) auf EIN gesetzt, wenn der Status von CIO 000001 von AUS zu EIN gewechselt ist. Wenn CIO 000000 im folgenden Zyklus auf AUS gesetzt ist, wird das Unterprogramm 0001 nicht erneut ausgeführt und der Ausgang CIO 000100 bleibt auf EIN gesetzt.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn eine Schachtelung in mehr als 16 Ebenen (normale und globale Unterprogramme gezählt) vorhanden ist. EIN, wenn das spezifizierte globale Unterprogramm nicht vorhanden ist. EIN, wenn das globale Unterprogramm sich selbst aufruft. EIN, wenn ein globales Unterprogramm aufgerufen wird, das ausgeführt wird. EIN, wenn das spezifizierte globale Unterprogramm nicht in Interrupt-Task 0 definiert ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Befehl GLOBAL SUBROUTINE ENTRY, GSBN(751), und der zugehörige Befehl GLOBAL SUBROUTINE RETURN, GRET(752), müssen in Interrupt-Task 0 programmiert sein. Wenn der Bereich des globalen Unterprogramms nicht in Interrupt-Task 0 programmiert ist, tritt bei der Ausführung des Befehls GSBS(750) ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Der Befehl SUBROUTINE CALL, SBS(091), zum Aufruf eines normalen Unterprogramms, keinen Bereich eines globalen Unterprogramms (GSBN(751) bis GRET(752)) aufrufen.

Der Befehl GSBS(750) wird nicht ausgeführt, wenn er innerhalb eines durch IL(002) und ILC(003) gesperrten Programmabschnitts liegt. Daher sind derartige Ausführungssperren in Bereichen von globalen Unterprogrammen nicht zulässig.

Der gleiche Bereich eines globalen Unterprogramms (GSBN(751) bis GRET(752)) kann auch mehrmals aufgerufen werden.

Wenn der Befehl GSBS(750) in den folgenden Fällen ausgeführt wird, wird das globale Unterprogramm tatsächlich nicht aufgerufen und der Fehlermerker wird auf EIN gesetzt:

1,2,3...

1. Das spezifizierte globale Unterprogramm ist nicht definiert.
2. Die Schachtelung umfasst mehr als 16 Ebenen (normale und globale Unterprogramme gezählt).
3. Das globale Unterprogramm ruft sich selbst auf.

4. Das spezifizierte globale Unterprogramm wird ausgeführt.
5. Das spezifizierte globale Unterprogramm ist nicht in Interrupt-Task 0 definiert.

Beispiele

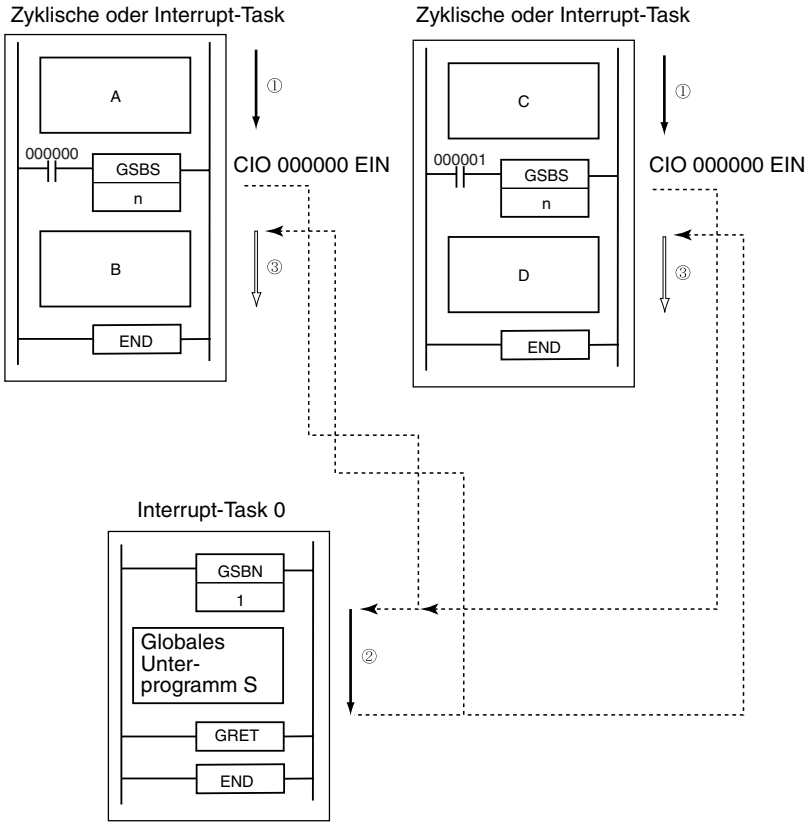
Beispiel 1

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird das globale Unterprogramm 1 ausgeführt und die Programmausführung kehrt anschließend zum nächsten Befehl nach GSBS(750) zurück.

Status von CIO 000000	Reihenfolge der Programmausführung
EIN	A → S → B
AUS	A → B

Wenn CIO 000001 im folgenden Beispiel auf EIN gesetzt ist, wird das globale Unterprogramm 1 ausgeführt und die Programmausführung kehrt anschließend zum nächsten Befehl nach GSBS(750) zurück.

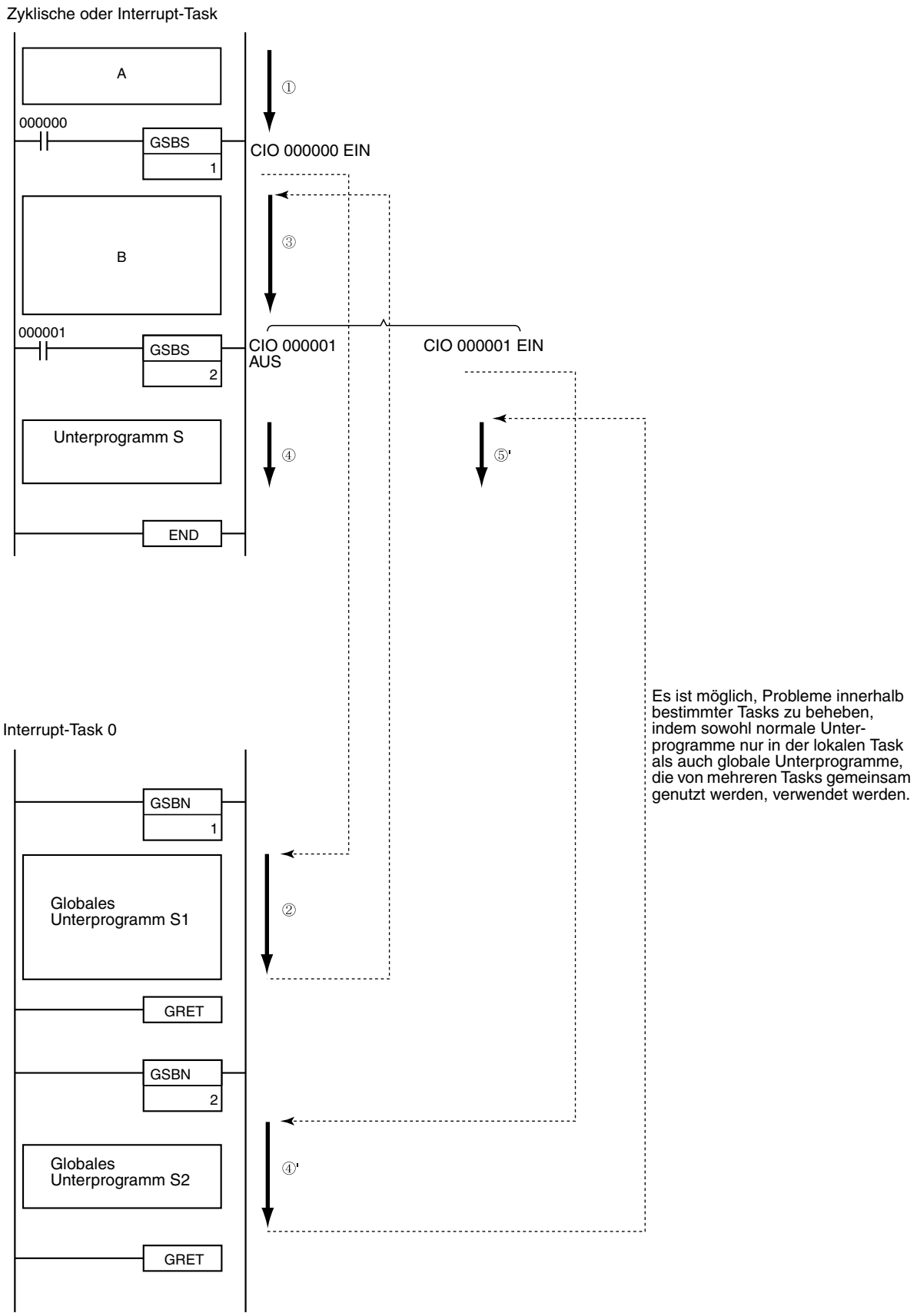
Status von CIO 000001	Reihenfolge der Programmausführung
EIN	C → S → D
AUS	C → D



Beispiel 2

In Interrupt-Task 0 können zwei oder mehr globale Unterprogramme programmiert werden. In diesem Fall kann Interrupt-Task 0 aufgeteilt und als Task der Unterprogrammfunktion verwendet werden.

Wenn CIO 000000 auf EIN gesetzt ist, wird das globale Unterprogramm 1 ausgeführt.
Wenn CIO 000001 auf EIN gesetzt ist, wird das globale Unterprogramm 2 ausgeführt.



3-19-6 GLOBAL SUBROUTINE ENTRY: GSBN(751)

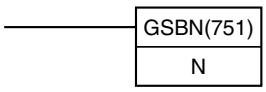
Zweck

Kennzeichnet den Beginn des globalen Unterprogramms mit der spezifizierten Unterprogrammnummer. Der Befehl wird in Kombination mit dem Befehl GRET(752) zur Definition des Bereichs eines globalen Unterprogramms verwendet.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Der Befehl GSBN(751) wird in Kombination mit den Befehlen GSBS(750) und GRET(752) verwendet, den Befehlen GLOBAL SUBROUTINE CALL (Globales Unterprogramm aufrufen) und GLOBAL SUBROUTINE RETURN (Globales Unterprogramm verlassen).

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	GSBN(751)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	Nicht zulässig	---	OK

Operanden

N: Nummer des globalen Unterprogramms
Spezifiziert die Nummer des globalen Unterprogramms zwischen 0 und 1023 dezimal.

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 muss die Unterprogrammnummer im Bereich zwischen 0 und 255 dezimal liegen.

Operanden-Spezifikationen

Bereich	N
CIO-Bereich	---
Arbeitsbereich	---
Haftmerkerbereich	---
Zusatz-Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradresse	---
Indirekte DM/EM-Adresse in BCD	---
Konstanten	0 bis 1023 (dezimal) (siehe Hinweis)
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Index-Registern	---

Hinweis Bei den CPU-Baugruppen CJ1M-CPU11 und CJ1M-CPU21 liegt der gültige Bereich zwischen 0 und 255 dezimal.

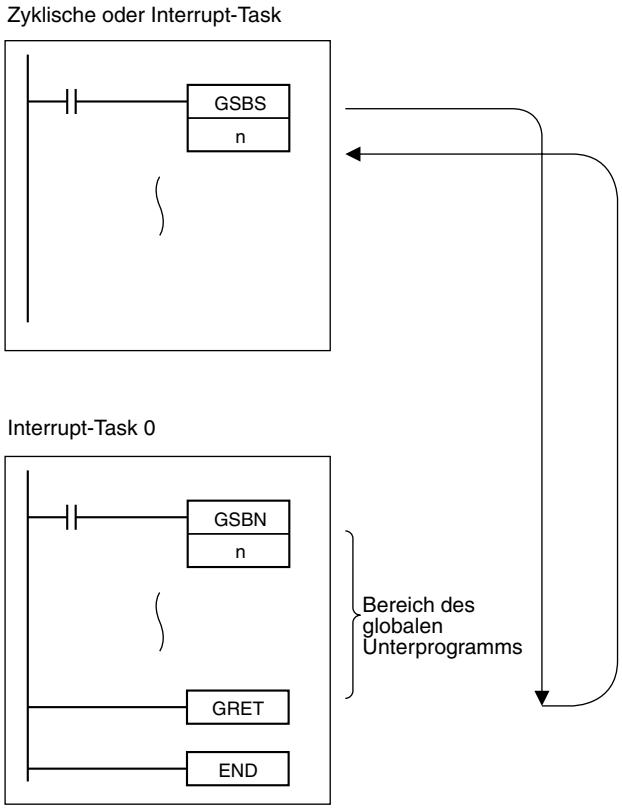
Beschreibung

Der Befehl GSBN(751) kennzeichnet den Beginn des globalen Unterprogramms mit der spezifizierten Unterprogrammnummer. Das Ende des Unterprogramms wird durch den Befehl GRET(752) gekennzeichnet.

Der Bereich des Programms, der mit dem ersten GSBN(751)-Befehl beginnt, ist der Unterprogramm-Bereich. Ein globales Unterprogramm wird nur ausgeführt, wenn es durch den Befehl GSBS(750) aufgerufen wird.

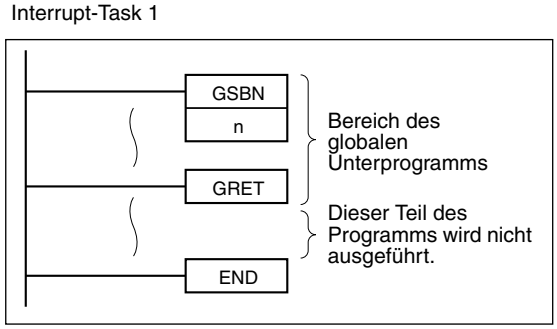
Der Programmbereich des globalen Unterprogramms (zwischen GSBN(751) und GRET(752)) muss in Interrupt-Task 0 definiert werden. Wenn dieser Bereich in einer anderen Task definiert wird, tritt bei der Ausführung des Befehls GSBS(750) ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Der Befehl GSBS(750) kann sowohl in zyklischen Tasks (einschließlich zyklischer Zusatztasks) als auch in Interrupt-Tasks programmiert werden.

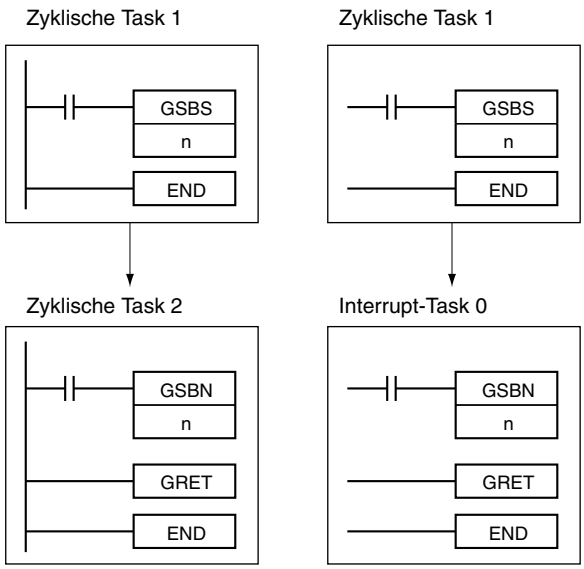


Vorsichtsmaßnahmen

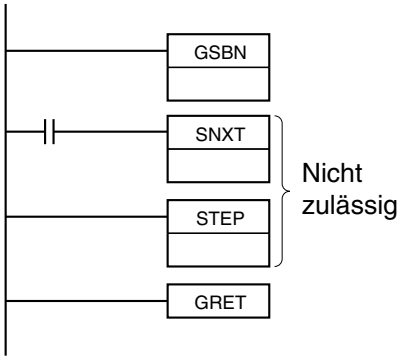
- Wenn das Unterprogramm nicht aufgerufen wird, werden die enthaltenen Befehle als NOP(000) behandelt.
- Platzieren Sie den Bereich des globalen Unterprogramms (von GSBN(751) bis GRET(752)) in Interrupt-Task 0 unmittelbar vor dem Befehl END(001). Wenn zwei oder mehr globale Unterprogramme verwendet werden, müssen diese als Gruppe gemeinsam in Interrupt-Task 0 hinter dem Ende des Hauptprogramms platziert werden. Wenn ein Teil des Hauptprogramms hinter dem Bereich des globalen Unterprogramms platziert wird, wird dieser Teil des Programms ignoriert.



- Die Methode zur Eingabe der Nummer N des globalen Unterprogramms unterscheidet sich zwischen CX-Programmer und einer Programmierkonsole. Geben Sie bei CX-Programmer #0 bis #1023 ein und bei einer Programmierkonsole 0000 bis 1023.
- Globale Unterprogramme müssen stets in Interrupt-Task 0 programmiert werden. Wenn ein globales Unterprogramm aufgerufen wird, und dieses Unterprogramm nicht in Interrupt-Task 0 enthalten ist, tritt ein Fehler auf.

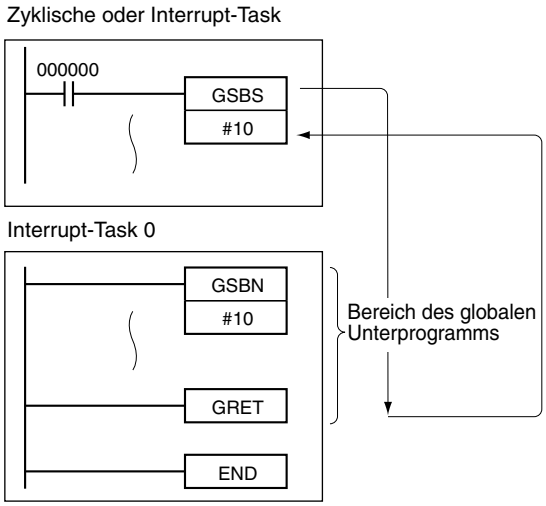


- Die Schrittbefehle, STEP(008) und SNXT(009), können nicht in globalen Unterprogrammen verwendet werden.



Beispiel

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, wird das globale Unterprogramm 10 ausgeführt und die Programmausführung kehrt anschließend zum nächsten Befehl nach dem Befehl GSBS(750) zurück, der das Unterprogramm aufgerufen hat.



3-19-7 GLOBAL SUBROUTINE RETURN: GRET(752)

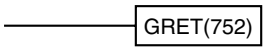
Zweck

Kennzeichnet das Ende eines Unterprogramms. Der Befehl wird in Kombination mit dem Befehl GSBN(751) zur Definition des Bereichs eines globalen Unterprogramms verwendet.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Der Befehl GRET(752) wird in Kombination mit den Befehlen GSBS(750) und GSBN(751) verwendet, den Befehlen GLOBAL SUBROUTINE CALL (Globales Unterprogramm aufrufen) und GLOBAL SUBROUTINE ENTRY (Globales Unterprogramm - Anfang).

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	GRET(752)
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	Nicht zulässig	Nicht zulässig	OK

Beschreibung

Der Befehl GRET(752) kennzeichnet das Ende eines globalen Unterprogramms, dessen Anfang durch den Befehl GSBN(751) gekennzeichnet ist. Weitere Informationen zu den Funktionen von globalen Unterprogrammen finden Sie unter *3-19-6 GLOBAL SUBROUTINE ENTRY: GSBN(751)*.

Wenn die Programmausführung den Befehl GRET(752) erreicht, kehrt sie automatisch zu dem nächsten Befehl nach dem Befehl GSBS(750) zurück, der das globale Unterprogramm aufgerufen hat.

Vorsichtsmaßnahmen

Wenn das Unterprogramm nicht aufgerufen wird, werden die enthaltenen Befehle als NOP(000) behandelt.

Beispiel

Beispiele für die Funktionsweise des Befehls GRET(752) finden Sie unter *3-19-6 GLOBAL SUBROUTINE ENTRY: GSBN(751)*.

3-20 Interrupt-Steuerungsbefehle

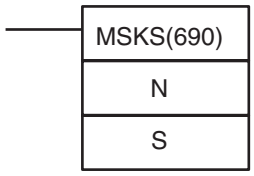
3-20-1 SET INTERRUPT MASK: MSKS(690)

Zweck

Beide Arten von Interrupt-Tasks (E/A und zeitgesteuert) werden maskiert (deaktiviert), wenn die SPS in die RUN-Betriebsart wechselt. MSKS(690) kann zum Maskieren oder Demaskieren von E/A-Interrupts verwendet werden sowie zum Einstellen der Intervallzeit für zeitgesteuerte Interrupts.

Der Befehl MSKS(690) wird von CS1D CPU-Baugruppen nicht unterstützt.

Kontaktplansymbol



N: Interrupt-Kennzeichner

S: Interrupt-Daten

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MSKS(690)
	Ausführung einmal bei steigender Flanke	@MSKS(690)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

CS1W-INT01/CJ1W-INT01

Spezifikation von E/A-Interrupt-Verarbeitung und Masken-Verarbeitung

Operand	Inhalt
N	Spezifikation der Baugruppennummer der Interrupt-Eingangsbaugruppe. 0: Baugruppennummer 0 1: Baugruppennummer 1
S	Interrupt-Maske Einstellung auf 0000 bis FFFF hex (16 Bits pro Baugruppe) Die einzelnen Bits spezifizieren Folgendes: 0: Interrupt aktiviert 1: Interrupt maskiert

Hinweis

1. Die Baugruppen CS1W-INT01 und C200HS-INT01 können nicht gleichzeitig verwendet werden.
2. Alle Interrupt-Eingänge, die zuvor erkannt wurden, werden gelöscht, wenn die Interrupt-Maskierung aufgehoben wird.
3. Die Interrupt-Eingangsbaugruppe CJ1W-INT01 kann nicht zusammen mit einer CJ1 CPU-Baugruppe verwendet werden. Zudem können E/A-Interrupt-Task nicht ausgeführt werden.

Der folgenden Tabelle können Sie die Beziehung zwischen Baugruppennummern der Interrupt-Eingangsbaugruppe und den Nummern der Interrupt-Tasks entnehmen.

Baugruppen-nummer	Nummern der Interrupt-Task	
0	100 bis 115	Bits 00 bis 15 von S entsprechen den Eingangs-Interrupt-Tasks.
1	116 bis 131	

Spezifikation der E/A-Interrupt-Verarbeitung, Spezifikation der steigenden/fallenden Flanke

Operand	Inhalt
N	Spezifikation der Baugruppennummer der Interrupt-Eingangsbaugruppe. 2: Baugruppennummer 0 3: Baugruppennummer 1
S	Spezifikation entweder der steigenden oder der fallenden Flanke des Interrupt-Eingangssignals. Einstellung auf 0000 bis FFFF hex (16 Bits pro Baugruppe) Die einzelnen Bits spezifizieren Folgendes: 0: Steigende Flanke 1: Fallende Flanke

Der folgenden Tabelle können Sie die Beziehung zwischen Baugruppennummern der Interrupt-Eingangsbaugruppe und Interrupt-Task-Nummern entnehmen.

Baugruppen-nummer	Nummer der Interrupt-Task	
2	100 bis 115	Bits 00 bis 15 von S entsprechen den Eingangs-Interrupt-Tasks.
3	116 bis 131	

Hinweis Alle Interrupt-Eingänge, die zuvor erkannt wurden, werden gelöscht, wenn die Spezifikation von steigender/fallender Flanke geändert wird.

C200HS-INT01

Spezifikation von E/A-Interrupt-Verarbeitung und Masken-Verarbeitung

Operand	Inhalt
N	Spezifikation der Baugruppennummer der Interrupt-Eingangsbaugruppe. 0: Baugruppennummer 0 1: Baugruppennummer 1 2: Baugruppennummer 2 3: Baugruppennummer 3
S	Interrupt-Maske Einstellung auf 0000 bis 00FF hex (8 Bits pro Baugruppe) Die einzelnen Bits spezifizieren Folgendes: 0: Interrupt aktiviert 1: Interrupt maskiert

Hinweis
1. Die Baugruppen CS1W-INT01 und C200HS-INT01 können nicht gleichzeitig verwendet werden.
2. Alle Interrupt-Eingänge, die zuvor erkannt wurden, werden gelöscht, wenn die Interrupt-Maskierung aufgehoben wird.

Der folgenden Tabelle können Sie die Beziehung zwischen Baugruppennummern der Interrupt-Eingangsbaugruppe und Interrupt-Task-Nummern entnehmen.

Baugruppen-nummer	Nummer der Interrupt-Task	
0	100 bis 107	Bits 00 bis 07 von S entsprechen den Eingangs-Interrupt-Tasks.
1	108 bis 115	
2	116 bis 123	
3	124 bis 131	

Hinweis Alle Interrupt-Eingänge, die zuvor erkannt wurden, werden gelöscht, wenn die Spezifikation von steigender/fallender Flanke geändert wird.

CJ1M CPU-Baugruppe mit integrierten Interrupt-Eingängen

Spezifikation von E/A-Interrupt-Verarbeitung und Masken-Verarbeitung

Operand	Inhalt
N	Spezifikation der Interrupt-Eingangsnummer 6: Interrupt-Eingang 0 7: Interrupt-Eingang 1 8: Interrupt-Eingang 2 9: Interrupt-Eingang 3
S	Interrupt-Maske 0000 hex: Interrupt aktiviert (Direktmodus) 0001 hex: Interrupt maskiert (Direktmodus) 0002 hex: Dekrementierungszähler gestartet und Interrupts aktiviert (Zählermodus) 0003 hex: Inkrementierungszähler gestartet und Interrupts aktiviert (Zählermodus)

Hinweis Alle Interrupt-Eingänge, die zuvor erkannt wurden, werden gelöscht, wenn die Interrupt-Maskierung aufgehoben wird.

Der folgenden Tabelle können Sie die Beziehung zwischen Interrupt-Eingangsnummern und Interrupt-Task-Nummern entnehmen.

Interrupt-Eingangsnummer	Nummer der Interrupt-Task	
Interrupt-Eingang 0	140	CIO 296000
Interrupt-Eingang 1	141	CIO 296001
Interrupt-Eingang 2	142	CIO 296002
Interrupt-Eingang 3	143	CIO 296003

Spezifikation der E/A-Interrupt-Verarbeitung, Spezifikation der steigenden/fallenden Flanke

Operand	Inhalt
N	Spezifikation der Interrupt-Eingangsnummer 10: Interrupt-Eingang 0 11: Interrupt-Eingang 1 12: Interrupt-Eingang 2 13: Interrupt-Eingang 3
S	Spezifikation entweder der steigenden oder der fallenden Flanke des Interrupt-Eingangssignals. 0000 hex: Steigende Flanke 0001 hex: Fallende Flanke

Der folgenden Tabelle können Sie die Beziehung zwischen Interrupt-Eingangsnummern und Interrupt-Task-Nummern entnehmen.

Interrupt-Eingangsnummer	Nummer der Interrupt-Task	
Interrupt-Eingang 0	140	CIO 296000
Interrupt-Eingang 1	141	CIO 296001
Interrupt-Eingang 2	142	CIO 296002
Interrupt-Eingang 3	143	CIO 296003

Hinweis Alle Interrupt-Eingänge, die zuvor erkannt wurden, werden gelöscht, wenn die Spezifikation von steigender/fallender Flanke geändert wird.

Spezifikation zeitgesteuerter Interrupts

Operand	Inhalt
N	Spezifikation der Nummer des zeitgesteuerten Interrupts 4: Interrupt-Task 2 5: Interrupt-Task 3
S	0000: Zeitgesteuerte Interrupts deaktiviert 0001 bis 270F hex: Intervall des zeitgesteuerten Interrupts (1 bis 9999) **Hinweis** Als Zeiteinheit für die Einstellung des Intervalls für zeitgesteuerte Interrupts kann in den Interrupt-Einstellungen im SPS-Setup zwischen 10 ms und 1,0 ms gewählt werden. Bei den CJ1M CPU-Baugruppen ist zudem eine Zeiteinheit von 0,1 ms wählbar. In diesem Fall erstreckt sich der Einstellungsbereich von 0005 bis 270F hex (5 bis 9999).

Zurücksetzen und Starten zeitgesteuerter Interrupts (nur CJ1M)

Operand	Inhalt		
N	Spezifikation der Nummer des zeitgesteuerten Interrupts 14: Zeitgesteuerter Interrupt 0 (Interrupt-Task 2) 15: Zeitgesteuerter Interrupt 1 (Interrupt-Task 3)		
S	Zeitgesteuerte Interrupts deaktiviert		0000 hex
	Einstellung des Intervalls zeitgesteuerter Interrupts und Starten des zeitgesteuerten Interrupts	Bereich von 10 bis 99.990 ms bzw. 1 bis 9.999 ms (bei einer Zeiteinheit von 10 ms bzw. 1 ms)	0001 bis 270F hex
		Bereich von 0,5 bis 999,9 ms (bei einer Zeiteinheit von 0,1 ms)	0005 bis 270F hex (Die Einstellungen 0001 bis 0004 hex können nicht verwendet werden. Andernfalls tritt ein Befehlsfehler auf.)

Operanden-Spezifikationen

Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich	---	W000 bis W511
Haftmerkerbereich	---	H000 bis H511
Zusatz-Systembereich	---	A000 bis A447 A448 bis A959
Zeitgeberbereich	---	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	---	@ D00000 bis @ 32767 @ E00000 bis @ 32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adresse in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	Nur spezifizierte Werte	
Datenregister	---	DR0 bis DR15

Bereich	N	S
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl MSKS(690) steuert E/A-Interrupts und zeitgesteuerte Interrupts. Mit dem Wert für N wird der Interrupt spezifiziert.

Baugruppe	N	Bedeutung
CS1W-INT01 oder CJ1W-IN01	0 oder 1	N entspricht der Interrupt-Eingangs-Task. Die Bits 0 bis 7 von S entsprechen den Interrupt-Eingangsnummern der entsprechenden Interrupt-Baugruppe. Der Befehl MSKS(690) maskiert (deaktiviert) den Interrupt-Eingang, wenn das entsprechende Bit auf EIN gesetzt ist, und demaskiert (aktiviert) den Interrupt-Eingang, wenn das entsprechende Bit auf AUS gesetzt ist.
C200HS-INT01	0 bis 3	
CJ1M CPU-Baugruppe, integrierte Interrupt-Eingänge	6 bis 9	

Baugruppe	N	Bedeutung
CS1W-INT01 oder CJ1W-IN01	2 oder 3	N entspricht der Interrupt-Eingangs-Task. S spezifiziert die steigende oder fallende Flanke als Trigger. (In der Werkseinstellung ist die steigende Flanke gewählt.)
CJ1M CPU-Baugruppe, integrierte Interrupt-Eingänge	10 bis 13	

Hinweis

1. Der Befehl MSKS(690) kann verwendet werden, um eine bestimmte E/A-Interrupt-Task nur in einem bestimmten Zyklus zu aktivieren und sie in anderen Zyklen zu deaktivieren.
2. Die Baugruppennummern werden den Interrupt-Eingangsbaugruppen entsprechend der Reihenfolge der Montage von links nach rechts zugewiesen.

N = 4 oder 5

Die Werte 4 und 5 entsprechen den zeitgesteuerten Interrupts 2 und 3.
Wenn N auf 4 oder 5 gesetzt ist, kann der Inhalt von S die Interrupt-Task entweder deaktivieren (S = 0000) oder sie auf den spezifizierte Zeitintervall einstellen. Als Zeiteinheit für die Einstellung des Intervalls für zeitgesteuerte Interrupts kann im SPS-Setup zwischen 10 ms, 1,0 ms und 0,1 ms gewählt werden.

N = 14 oder 15 (nur CJ1M CPU-Baugruppen)

Wenn N auf 14 oder 15 gesetzt ist, wird das in S spezifizierte Zeitintervall für zeitgesteuerte Interrupts bei der durch N spezifizierten zeitgesteuerten Interrupt-Task eingestellt und der Intervallzeitgeber für den zeitgesteuerten Interrupt wird zurückgesetzt. Die Zeit für den Rücksetzstart des ersten Interrupts wird beibehalten.

Hinweis

1. Die Zeiteinheit für den zeitgesteuerten Interrupt wird im SPS-Setup eingestellt.
2. Achten Sie darauf, dass das eingestellte Zeitintervall länger ist als die zur Ausführung der zeitgesteuerten Interrupt-Task erforderliche Zeit.
3. Bei zeitgesteuerten Interrupts wird der Befehl MSKS(690) nur zur Einstellung des Intervalls für den zeitgesteuerten Interrupt verwendet, nicht jedoch zur Einstellung der Zeit bis zum ersten zeitgesteuerten Interrupt. Zur präzisen Steuerung der Zeit bis zum ersten Interrupt und des Interrupt-Intervalls können Sie den Befehl CLI(691) unmittelbar vor der Programmierung des Befehls MSKS(690) programmieren, um die Zeit bis zum ersten zeitgesteuerten Interrupt einzustellen. Wenn der Befehl MSKS(690) bei einer CJ1M CPU-Baugruppe zum Neustart eines zeitgesteuerten Interrupts verwendet wird, ist die Zeit bis zum ersten zeitgesteuerten Interrupt selbst dann genau, wenn der Befehl CLI(691) nicht verwendet wird.

A440 enthält die maximale Verarbeitungszeit für Interrupt-Tasks. Das niederwertigste Byte von A441 enthält die Nummer der Interrupt-Task mit der längsten Verarbeitungszeit.

Merker

Name	Bezeich-nung	Funktion
Fehler-merker	ER	EIN, wenn N nicht innerhalb des spezifizierten Bereichs von 0 bis 5 (0 bis 15 bei den integrierten Interrupt-Eingängen einer CJ1M CPU-Baugruppe) liegt. EIN, wenn S nicht innerhalb des spezifizierten Bereichs von 0000 bis 00FF hex liegt, während N auf 0 bis 3 gesetzt ist (bei Verwendung einer C200HS-INT01 und Spezifikation der E/A-Interrupt-Verarbeitung). EIN, wenn S nicht innerhalb des spezifizierten Bereichs von 0000 bis 0003 hex liegt (bei Verwendung eines integrierten Interrupt-Eingangs einer CJ1M CPU-Baugruppe und Spezifikation der E/A-Interrupt-Verarbeitung). EIN, wenn S nicht innerhalb des spezifizierten Bereichs von 0000 bis 270F hex liegt, während N auf 4 oder 5 gesetzt ist (0005 bis 270F hex bei einem integrierten Interrupt-Eingangs einer CJ1M CPU-Baugruppe und einer Zeiteinheit von 0,1 ms). EIN, wenn der Befehl in einer Interrupt-Task ausgeführt wurde. In allen anderen Fällen AUS.
Gleich-Merker	=	AUS
Negativ-Merker	N	AUS

In der folgenden Tabelle sind die relevanten Merker im Zusatz-Systembereich aufgeführt.

Name	Adresse	Funktion
Interrupt-Task-Fehlermerker	A40213	Dieser Merker wird in folgenden Fällen auf EIN gesetzt: 1) Es wurde eine Interrupt-Task länger als 10 ms während der E/A-Aktualisierung bei einer C200H-Spezial-E/A-Baugruppe oder eines dezentralen E/A-Slave-Racks ausgeführt. (Nur CS-Serie.) 2) Der Befehl IORF(097) wurde in einer Interrupt-Task ausgeführt, ohne dass die zyklische Aktualisierung der Spezial-E/As deaktiviert wurde.
Interrupt-Task-Fehler, Ursachenmerker	A42615	Gibt an, ob Interrupt-Task-Fehler 1 oder 2 aufgetreten ist.
Interrupt-Task-Fehler, Task-Nummer	A42600 bis A42611	Bei Fehler 1: Angabe der Interrupt-Task-Nummer. Bei Fehler 2: Angabe der Baugruppennummer der Spezial-E/A-Baugruppe, bei der die mehrfache Aktualisierung aufgetreten ist.

Vorsichtsmaßnahmen

Für Interrupt-Tasks werden nur Interrupt-Eingänge von normalen Interrupt-Eingangsbaugruppen der CS/CJ-Serie und C200H-Interrupt-Eingangsbaugruppen unterstützt. Interrupt-Eingänge von internen Modulen und Spezial-E/A-Baugruppen werden nicht unterstützt.

Die Interrupt-Eingangsbaugruppe wird in den CPU-Baugruppenträger nahe an der CPU installiert. Bei Verwendung einer CJ1-H CPU-Baugruppe muss die Baugruppe in den Steckplätzen 0 bis 4, bei Verwendung einer CJ1M CPU-Baugruppe in den Steckplätzen 0 bis 2 installiert werden. Ein Start von E/A-Interrupt-Tasks ist nicht möglich, wenn die Interrupt-Eingangsbaugruppe nicht in einem dieser Steckplätze installiert ist.

Die Worte werden den Interrupt-Eingangsbaugruppen entsprechend der Reihenfolge der Installation von links nach rechts zugewiesen.

Interrupts haben verschiedene Prioritätsstufen. Ein Ausschalt-Interrupt hat die höchste Priorität, gefolgt von E/A-Interrupts und schließlich zeitgesteuerten Interrupts. E/A-Interrupts mit niedrigen Nummern haben Priorität gegenüber E/A-Interrupts mit hohen Nummern.

Achten Sie darauf, dass die Verarbeitung der E/A-Interrupt-Task nicht mehr als 10 ms in Anspruch nimmt, wenn eine C200H-Spezial-E/A-Baugruppe oder ein dezentrales SYSMAC BUS E/A-Slave-Rack mit dem System verbunden ist. Wenn eine Interrupt-Task länger als 10 ms während der E/A-Aktualisierung bei einer

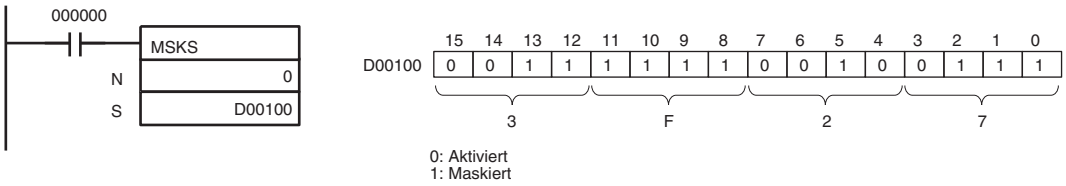
C200H-Spezial-E/A-Baugruppe oder eines dezentralen E/A-Slave-Racks ausgeführt wird, tritt ein nicht schwerwiegender Fehler auf und der Interrupt-Task-Fehlermerker (A40213) wird auf EIN gesetzt.

Wenn der Befehl IORF(097) innerhalb einer Interrupt-Task ausgeführt wird, um die E/A einer Spezial-E/A-Baugruppe zu aktualisieren, muss die zyklische Aktualisierung dieser Spezial-E/A-Baugruppe im SPS-Setup deaktiviert werden. Wenn die zyklische Aktualisierung der Spezial-E/A-Baugruppe nicht deaktiviert ist, kann es passieren, dass der Befehl IORF(097) während der zyklischen Aktualisierung ausgeführt wird, was dazu führt, dass ein nicht schwerwiegender Fehler aufgrund doppelter Aktualisierung auftritt und der Interrupt-Task-Fehlermerker (A40213) auf EIN gesetzt wird.

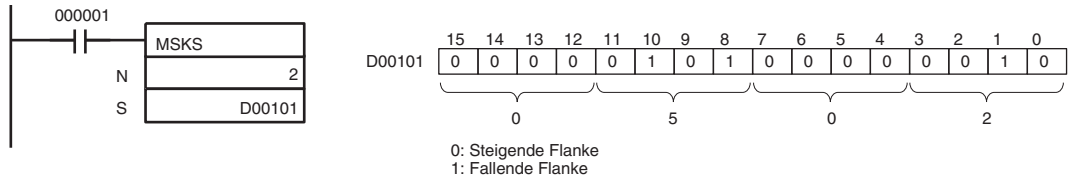
Beispiele

Beispiele für CS1W-INT01/CJ1W-INT01

Wenn CIO 000000 in dem folgenden Beispiel auf EIN wechselt, demaskiert (aktiviert) MSKS(690) die Interrupt-Eingänge von Interrupt-Eingangsbaugruppe 0.

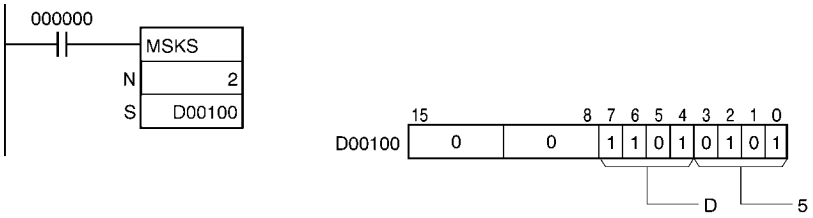


Wenn CIO 000001 in dem folgenden Beispiel auf EIN wechselt, setzt der Befehl MSKS(690) die Spezifikation von steigender/fallender Flanke bei Interrupt-Eingangsbaugruppe 0.



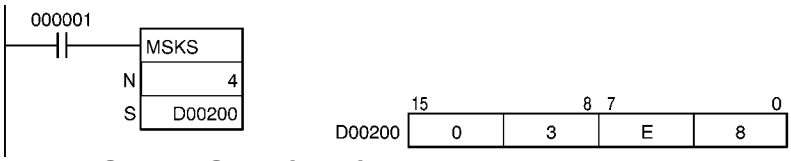
Beispiel für C200HS-INT01

Wenn CIO 000000 in dem folgenden Beispiel auf EIN wechselt, demaskiert (aktiviert) MSKS(690) die Interrupt-Eingänge 1, 3 und 5 von Interrupt-Eingangsbaugruppe 2.



Beispiel für zeitgesteuerte Interrupts

Wenn CIO 000001 im folgenden Beispiel auf EIN gesetzt ist, stellt der Befehl MSKS(690) ein Zeitintervall von 10 Sekunden für den zeitgesteuerten Interrupt 2 ein. (In diesem Fall ist die Zeiteinheit für das Intervall zeitgesteuerter Interrupts im SPS-Setup auf 10 ms eingestellt.)



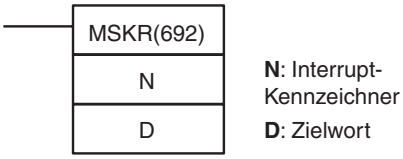
3-20-2 READ INTERRUPT MASK: MSKR(692)

Zweck

Liest die aktuellen Interrupt-Verarbeitungseinstellungen, die mit MSKS(690) gesetzt wurden.

Der Befehl MSKR(692) wird von CS1D CPU-Baugruppen nicht unterstützt.

Kontaktplansymbol



N: Interrupt-Kennzeichner

D: Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MSKR(692)
	Ausführung einmal bei steigender Flanke	@MSKR(692)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

CS1W-INT01/CJ1W-INT01

Lesen von Masken

Operand	Inhalt
N	Spezifikation der Baugruppennummer der Interrupt-Eingangsbaugruppe 0: Baugruppennummer 0 1: Baugruppennummer 1
D	Status der Interrupt-Maske 0000 bis FFFF hex (16 Bits pro Baugruppe) Die einzelnen Bits spezifizieren Folgendes: 0: Interrupt aktiviert 1: Interrupt maskiert

Der folgenden Tabelle können Sie die Beziehung zwischen Baugruppennummern der Interrupt-Eingangsbaugruppe und Interrupt-Task-Nummern entnehmen.

Baugruppen-nummer	Nummer der Interrupt-Task	
0	100 bis 115	Bits 00 bis 15 von S entsprechen den Eingangs-Interrupt-Tasks.
1	116 bis 131	

Lesen der Spezifikationen für steigende/fallende Flanke

Operand	Inhalt
N	Spezifikation der Baugruppennummer der Interrupt-Eingangsbaugruppe 2: Baugruppennummer 0 3: Baugruppennummer 1
D	Spezifikation der steigenden oder der fallenden Flanke des Interrupt-Eingangssignals. 0000 bis FFFF hex (16 Bits pro Baugruppe) Die einzelnen Bits spezifizieren Folgendes: 0: Steigende Flanke 1: Fallende Flanke

Hinweis

1. Die Baugruppen CS1W-INT01 und C200HS-INT01 können nicht gleichzeitig verwendet werden.
2. Die Interrupt-Eingangsbaugruppe CJ1W-INT01 kann nicht zusammen mit einer CJ1 CPU-Baugruppe verwendet werden. Zudem können E/A-Interrupt-Task nicht ausgeführt werden.

Der folgenden Tabelle können Sie die Beziehung zwischen Baugruppennummern der Interrupt-Eingangsbaugruppe und Interrupt-Task-Nummern entnehmen.

Baugruppen-nummer	Nummer der Interrupt-Task	
2	100 bis 115	Bits 00 bis 15 von S entsprechen den Eingangs-Interrupt-Tasks.
3	116 bis 131	

C200HS-INT01

Lesen von Masken

Operand	Inhalt
N	Spezifikation der Baugruppennummer der Interrupt-Eingangsbaugruppe 0: Baugruppennummer 0 1: Baugruppennummer 1 2: Baugruppennummer 2 3: Baugruppennummer 3
D	Status der Interrupt-Maske 0000 bis 00FF hex (8 Bits pro Baugruppe) Die einzelnen Bits spezifizieren Folgendes: 0: Interrupt aktiviert 1: Interrupt maskiert

Hinweis Die Baugruppen CS1W-INT01 und C200HS-INT01 können nicht gleichzeitig verwendet werden.

Der folgenden Tabelle können Sie die Beziehung zwischen Baugruppennummern der Interrupt-Eingangsbaugruppe und Interrupt-Task-Nummern entnehmen.

Baugruppennummer	Nummer der Interrupt-Task	
0	100 bis 107	Bits 00 bis 07 von S entsprechen den Eingangs-Interrupt-Tasks.
1	108 bis 115	
2	116 bis 123	
3	124 bis 131	

CJ1M CPU-Baugruppe mit integrierten Interrupt-Eingängen

Lesen von Masken

Operand	Inhalt
N	Spezifikation der Interrupt-Eingangsnummer 6: Interrupt-Eingang 0 7: Interrupt-Eingang 1 8: Interrupt-Eingang 2 9: Interrupt-Eingang 3
D	Interrupt-Maske 0000 hex: Interrupt aktiviert (Direktmodus) 0001 hex: Interrupt maskiert (Direktmodus) 0002 hex: Dekrementierungszähler gestartet und Interrupts aktiviert (Zählermodus) 0003 hex: Inkrementierungszähler gestartet und Interrupts aktiviert (Zählermodus)

Der folgenden Tabelle können Sie die Beziehung zwischen Interrupt-Eingangsnummern und Interrupt-Task-Nummern entnehmen.

Interrupt-Eingangsnummer	Nummer der Interrupt-Task	
Interrupt-Eingang 0	140	CIO 296000
Interrupt-Eingang 1	141	CIO 296001
Interrupt-Eingang 2	142	CIO 296002
Interrupt-Eingang 3	143	CIO 296003

Lesen der E/A-Interrupt-Verarbeitung, Spezifikation der steigenden/ fallenden Flanke

Operand	Inhalt
N	Spezifikation der Interrupt-Eingangsnummer 10: Interrupt-Eingang 0 11: Interrupt-Eingang 1 12: Interrupt-Eingang 2 13: Interrupt-Eingang 3
D	Spezifikation entweder der steigenden oder der fallenden Flanke des Interrupt-Eingangssignals. 0000 hex: Steigende Flanke 0001 hex: Fallende Flanke

Der folgenden Tabelle können Sie die Beziehung zwischen Interrupt-Eingangsnummern und Interrupt-Task-Nummern entnehmen.

Interrupt-Eingangsnummer	Nummer der Interrupt-Task	
Interrupt-Eingang 0	140	CIO 296000
Interrupt-Eingang 1	141	CIO 296001
Interrupt-Eingang 2	142	CIO 296002
Interrupt-Eingang 3	143	CIO 296003

Lesen des Intervalls zeitgesteuerter Interrupts

Operand	Inhalt
N	Spezifikation der Nummer des zeitgesteuerten Interrupts 4: Zeitgesteuerter Interrupt 0 (Interrupt-Task 2) 5: Zeitgesteuerter Interrupt 1 (Interrupt-Task 3)
D	0000: Zeitgesteuerter Interrupt deaktiviert. 0001 bis 270F hex: Intervall des zeitgesteuerten Interrupts (1 bis 9999) **Hinweis** Als Zeiteinheit für die Einstellung des Intervalls für zeitgesteuerte Interrupts kann in den Interrupt-Ein-stellungen im SPS-Setup zwischen 10 ms und 1,0 ms gewählt werden. Bei CJ1M-SS-Systemen kann auch eine Zeiteinheit von 0,1 ms eingestellt werden. In diesem Fall hat die Zeiteinstellung einen Bereich von 0005 bis 270F hex (5 bis 9999).

Lesen des Istwerts zeitgesteuerter Interrupts

Operand	Inhalt	
N	Spezifikation der Nummer des zeitgesteuerten Interrupts 14: Zeitgesteuerter Interrupt 0 (Interrupt-Task 2) 15: Zeitgesteuerter Interrupt 1 (Interrupt-Task 3)	
D	Zeit ab dem Start der Verarbeitung des zeitgesteuerten Interrupts bzw. Zeit ab dem vorherigen zeitgesteuerten Interrupt. 10-ms-Einheit: 0 bis 99.990 1-ms-Einheit: 0 bis 9.999 0,1-ms-Einheit: 0,0 bis 999,9 (nur CJ1M)	0000 bis 270F hex **Hinweis:** Selbst wenn der zeitgesteuerte Interrupt derzeit angehalten ist, kann die Zeit, die bis zum Anhalten verstrichen ist, gelesen werden. Wenn der zeitgesteuerte Interrupt noch nicht gestartet wurde, wird 0000 hex zurückgegeben.

Operanden-Spezifikationen

Bereich	N	D
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich	---	W000 bis W511
Haftmerkerbereich	---	H000 bis H511
Zusatz-Systembereich	---	A448 bis A959
Zeitgeberbereich	---	T0000 bis T4095

Bereich	N	D
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adresse in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	Nur spezifizierte Werte	---
Datenregister	---	DR0 bis DR15
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Mit MSKR(692) werden die aktuellen Interrupt-Einstellungen gelesen, die mit MSKS(690) eingestellt wurden.

N = 0 oder 1 (0 bis 3 bei C200HS-INT01)

Die Werte 0 und 1 (0 bis 3) entsprechen den Interrupt-Eingangsbaugruppen 0 und 1 (0 bis 3).

Die Bits 0 bis 7 von D entsprechen den Interrupt-Eingangsnummern 0 bis 7 der spezifizierten Baugruppe. Wenn ein Bit auf EIN gesetzt ist, ist der entsprechende Interrupt-Eingang maskiert (deaktiviert); wenn ein Bit auf AUS gesetzt ist, ist der entsprechende Interrupt-Eingang demaskiert (aktiviert).

N = 2 oder 3 (nur CS1W-INT01/CJ1W-INT01/CJ1M mit integrierten Interrupt-Eingängen)

Die Werte 2 und 3 entsprechen den Interrupt-Eingangsbaugruppen 0 und 1. Die Spezifikation von steigender/fallender Flanke für die Interrupt-Eingänge der durch N spezifizierten Interrupt-Eingangsbaugruppe wird nach D ausgegeben.

N = 4 oder 5

Die Werte 4 und 5 entsprechen den zeitgesteuerten Interrupts 2 und 3.

Wenn N auf 4 oder 5 gesetzt ist, gibt der Inhalt von D das Zeitintervall an, das für diesen Interrupt eingestellt wurde. Eine Einstellung auf 0000 bedeutet, dass der Interrupt deaktiviert wurde. Die Einheit für das Intervall zeitgesteuerter Interrupts kann im SPS-Setup eingestellt werden (00: 10 ms, 01: 1,0 ms), sodass der Bereich für das Zeitintervall 10 ms bis 99,99 s bzw. 1 ms bis 9,999 s beträgt.

N = 14 oder 15

Wenn N auf 14 oder 15 gesetzt ist, wird der Zeitgeber-Istwert des durch N spezifizierten zeitgesteuerten Interrupts in D gespeichert.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn N nicht innerhalb des spezifizierten Bereichs von 0 bis 5 (0 bis 15 bei CJ1M) liegt. In allen anderen Fällen AUS.

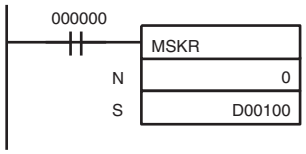
Vorsichtsmaßnahmen

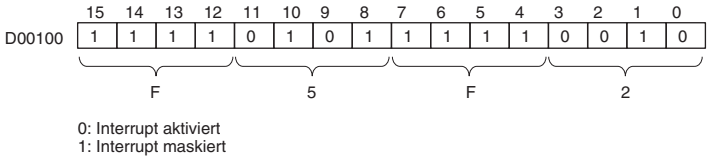
Der Befehl MSKR(692) kann im Hauptprogramm und in Interrupt-Tasks ausgeführt werden.

Beispiele

Beispiel für CS1W-INT01/CJ1W-INT01

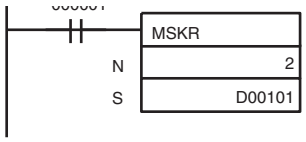
Wenn CIO 000000 im folgenden Beispiel auf EIN wechselt, liest der Befehl MSKR(692) den aktuellen Maskierungsstatus von Interrupt-Eingangsbaugruppe 2 und speichert ihn in D00100.

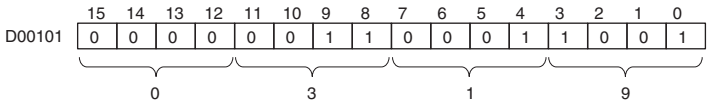




0: Interrupt aktiviert
1: Interrupt maskiert

Wenn CIO 000001 in dem folgenden Beispiel auf EIN wechselt, liest der Befehl MSKR(692) die Spezifikation von steigender/fallender Flanke bei Interrupt-Eingangsbaugruppe 0 und speichert sie in D00101.

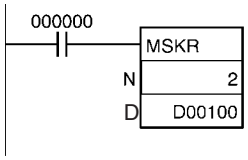


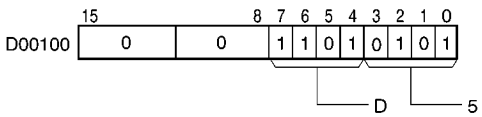


0: Steigende Flanke
1: Fallende Flanke

Beispiel für C200H-INT01

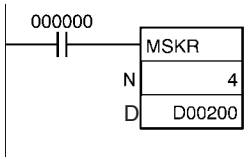
Wenn CIO 000000 im folgenden Beispiel auf EIN wechselt, liest der Befehl MSKR(692) den aktuellen Maskierungsstatus von Interrupt-Eingangsbaugruppe 2. In diesem Fall sind die Interrupt-Eingänge 1, 3 und 5 aktiviert.

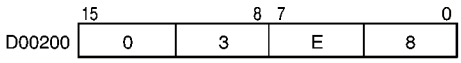




Beispiel für zeitgesteuerte Interrupts

Wenn CIO 000100 im folgenden Beispiel auf EIN wechselt, liest der Befehl MSKR(692) die Einstellung für den zeitgesteuerten Interrupt 2. In diesem Fall ist das Zeitintervall auf 1.000 (3E8 hexadezimal) eingestellt, was 10 s entspricht, wenn die Zeiteinheit für zeitgesteuerte Interrupts im SPS-Setup auf 10 ms eingestellt ist.





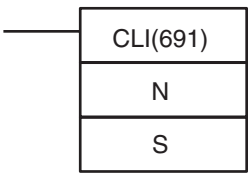
3-20-3 CLEAR INTERRUPT: CLI(691)

Zweck

Löscht oder erhält aufgezeichnete Interrupt-Eingänge für E/A-Interrupts oder legt die Zeit bis zum ersten Interrupt bei zeitgesteuerten Interrupts fest.

Der Befehl CLI(691) wird von CS1D CPU-Baugruppen nicht unterstützt.

Kontaktplansymbol



N: Interrupt-Kennzeichner
S: Interrupt-Daten

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CLI(691)
	Ausführung einmal bei steigender Flanke	@CLI(691)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

E/A-Interrupts bei CS1W-INT01/CJ1W-INT01

Operand	Inhalt
N	Spezifikation der Baugruppennummer der Interrupt-Eingangsbaugruppe 0: Baugruppennummer 0 1: Baugruppennummer 1
S	Spezifikation des Löschens der Interrupt-Maskierung (16 Bits/Baugruppe) 0000 bis FFFF hex Bedeutung der einzelnen Bits 0: Aufgezeichneter Interrupt-Eingang erhalten 1: Aufgezeichneter Interrupt-Eingang gelöscht

Hinweis

1. Die Baugruppen CS1W-INT01 und C200HS-INT01 können nicht gleichzeitig verwendet werden.
2. Die Interrupt-Eingangsbaugruppe CJ1W-INT01 kann nicht zusammen mit einer CJ1 CPU-Baugruppe verwendet werden. Zudem können E/A-Interrupt-Task nicht ausgeführt werden.

Der folgenden Tabelle können Sie die Beziehung zwischen Baugruppennummern der Interrupt-Eingangsbaugruppe und Interrupt-Task-Nummern entnehmen.

Baugruppen-nummer	Nummer der Interrupt-Task	
0	100 bis 115	Bits 00 bis 15 von S entsprechen den Eingangs-Interrupt-Tasks.
1	116 bis 131	

IE/A-Interrupts bei C200H-INT01

Operand	Inhalt
N	Spezifikation der Baugruppennummer der Interrupt-Eingangsbaugruppe 0: Baugruppennummer 0 1: Baugruppennummer 1 2: Baugruppennummer 2 3: Baugruppennummer 3
S	Spezifikation des Löschens der Interrupt-Maskierung (8 Bits/Baugruppe) 0000 bis 00FF hex Bedeutung der einzelnen Bits 0: Aufgezeichneter Interrupt-Eingang erhalten 1: Aufgezeichneter Interrupt-Eingang gelöscht

Hinweis Die Baugruppen CS1W-INT01 und C200HS-INT01 können nicht gleichzeitig verwendet werden.

Der folgenden Tabelle können Sie die Beziehung zwischen Baugruppennummern der Interrupt-Eingangsbaugruppe und Interrupt-Task-Nummern entnehmen.

Baugruppennummer	Nummer der Interrupt-Task	
0	100 bis 107	Bits 00 bis 07 von S entsprechen den Eingangs-Interrupt-Tasks.
1	108 bis 115	
2	116 bis 123	
3	124 bis 131	

CJ1M CPU-Baugruppe mit integrierten Interrupt-Eingängen

Operand	Inhalt
N	Spezifikation der Interrupt-Eingangsnummer 6: Interrupt-Eingang 0 7: Interrupt-Eingang 1 8: Interrupt-Eingang 2 9: Interrupt-Eingang 3
S	Spezifikation des Löschens der Interrupt-Maskierung 0000 hex: Aufgezeichneter Interrupt-Eingang erhalten 0001 hex: Aufgezeichneter Interrupt-Eingang gelöscht

Der folgenden Tabelle können Sie die Beziehung zwischen Interrupt-Eingangsnummern und Interrupt-Task-Nummern entnehmen.

Interrupt-Eingangsnummer	Nummer der Interrupt-Task	
Interrupt-Eingang 0	140	CIO 296000
Interrupt-Eingang 1	141	CIO 296001
Interrupt-Eingang 2	142	CIO 296002
Interrupt-Eingang 3	143	CIO 296003

Löschen von Schneller-Zähler-Interrupts (nur CJ1M)

Operand	Inhalt
N	Spezifikation des Schneller-Zähler-Eingangs 10: Schneller Zähler 0 11: Schneller Zähler 1
S	Spezifikation des Löschens der Interrupt-Maskierung 0000 hex: Aufgezeichneter Interrupt-Eingang erhalten 0001 hex: Aufgezeichneter Interrupt-Eingang gelöscht

Einstellung der Zeit bis zum ersten zeitgesteuerten Interrupt

Operand	Inhalt
N	Spezifikation der Nummer des zeitgesteuerten Interrupts 4: Zeitgesteuerter Interrupt 0 (Interrupt-Task 2) 5: Zeitgesteuerter Interrupt 1 (Interrupt-Task 3)
S	0000 bis 270F hex: Zeit bis zum ersten zeitgesteuerten Interrupt (0 bis 9999) **Hinweis** Als Zeiteinheit für die Einstellung des Intervalls für zeitgesteuerte Interrupts kann in den Interrupt-Einstellungen im SPS-Setup zwischen 10 ms und 1,0 ms gewählt werden. Bei CJ1M-SPS-Systemen kann die Zeiteinheit auch auf 0,1 ms eingestellt werden.

Operanden-Spezifikationen

Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich	---	W000 bis W511
Haftmerkerbereich	---	H000 bis H511

Bereich	N	S
Zusatz-Systembereich	---	A000 bis A959
Zeitgeberbereich	---	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adresse in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---	DR0 bis DR15
Datenregister	Nur spezifizierte Werte	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis – 2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

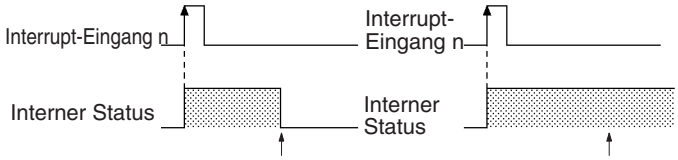
Beschreibung

Abhängig vom Wert für N löscht der Befehl CLI(691) die spezifizierten aufgezeichneten E/A-Interrupts oder er dient zur Einstellung der Zeit bis zur Ausführung des ersten zeitgesteuerten Interrupts. Bei CJ1M-SPS-Systemen kann der Befehl auch zum Löschen von Interrupts für schnelle Zähler verwendet werden.

N = 0 oder 1 (0 bis 3 bei C200HS-INT01 bzw. 6 bis 9 bei CJ1M CPU-Baugruppen mit integrierten Interrupt-Eingängen)

Die Werte 0 und 1 (0 bis 3) entsprechen den Interrupt-Eingangsbaugruppen 0 und 1 (0 bis 3).

Die Bits 0 bis 7 von S entsprechen den Interrupt-Eingangsnummern 0 bis 7 der spezifizierten Baugruppe. Der Befehl CLI(691) löscht einen aufgezeichneten Interrupt-Eingang, wenn das entsprechende Bit von S auf EIN gesetzt ist, und behält den aufgezeichneten Interrupt-Eingang bei, wenn das entsprechende Bit auf AUS gesetzt ist.



Wenn eine E/A-Interrupt-Task ausgeführt wird und ein Interrupt-Eingang mit einer anderen Interrupt-Nummer empfangen wird, wird diese Interrupt-Nummer intern aufgezeichnet. Die aufgezeichneten E/A-Interrupts werden dann später in der Reihenfolge ihrer Priorität (von der niedrigsten zur höchsten Nummer) ausgeführt. Der Befehl CLI(691) kann verwendet werden, um diese intern aufgezeichneten Interrupts zu löschen, bevor sie ausgeführt werden.

Hinweis

1. Der Befehl MSKS(690) kann verwendet werden, um eine bestimmte E/A-Interrupt-Task in einem bestimmten Zyklus zu aktivieren und sie in anderen Zyklen zu deaktivieren.

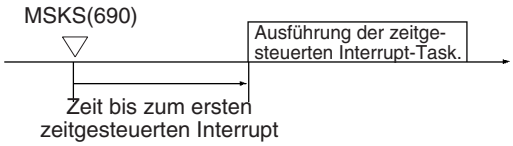
2. Die Baugruppennummern werden den Interrupt-Eingangsbaugruppen entsprechend der Reihenfolge der Montage von links nach rechts zugewiesen.

N = 4 oder 5

Die Werte 4 und 5 entsprechen den zeitgesteuerten Interrupts 2 und 3.
Wenn N auf 4 oder 5 gesetzt ist, spezifiziert der Inhalt von S das Zeitintervall bis zur ersten zeitgesteuerten Interrupt-Task nach Ausführung des Befehls MSKS(690).

Das Zeitintervall kann im Bereich von 0000 bis 270F (0 bis 9.999) eingestellt werden. Die Einheit für das Intervall zeitgesteuerter Interrupts kann im SPS-Setup eingestellt werden (00: 10 ms, 01: 1,0 ms), sodass sich der tatsächliche Bereich für das Zeitintervall von 10 ms bis 99,99 s bzw. 1 ms bis 9,999 s erstreckt.

Hinweis Stellen Sie das Zeitintervall bis zum ersten zeitgesteuerten Interrupt auf mindestens 10 ms ein.



N = 10 oder 11 (nur CJ1M)

Die Werte 10 und 11 entsprechen Interrupts für schnelle Zähler und können verwendet werden, um Interrupts für diese zu löschen oder zu halten (sowohl bei Zielwert- als auch bei Bereichsvergleich).

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn N nicht innerhalb des spezifizierten Bereichs von 0 bis 5 (0, 1 oder 4 bis 11 bei CJ1M) liegt. EIN, wenn S nicht innerhalb des spezifizierten Bereichs von 0000 bis 00FF hex liegt, während N auf 0 bis 3 gesetzt ist (nur bei Verwendung einer C200HS-INT01 und E/A-Interrupts). EIN, wenn S nicht den Wert 0000 oder 0001 hex hat (nur bei Schneller-Zähler-Interrupts und CJ1M mit integrierten Interrupt-Eingängen). EIN, wenn S nicht innerhalb des für zeitgesteuerte Interrupts spezifizierten Bereichs von 0000 bis 270F hex liegt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Es gibt keine Beschränkung für die Anzahl der Interrupt-Eingänge, die aufgezeichnet werden können. Zwar ist eine Aufzeichnung aller Interrupt-Eingänge möglich, jedoch wird ein neuer Interrupt-Eingang ignoriert, wenn dieser Interrupt bereits aufgezeichnet wurde. Darüber hinaus wird der aufgezeichnete Interrupt auch nicht gelöscht, bis die Ausführung seiner Interrupt-Task abgeschlossen ist. Daher wird ein neuer Interrupt ignoriert, wenn er während der Ausführung der zugehörigen Interrupt-Task empfangen wird.

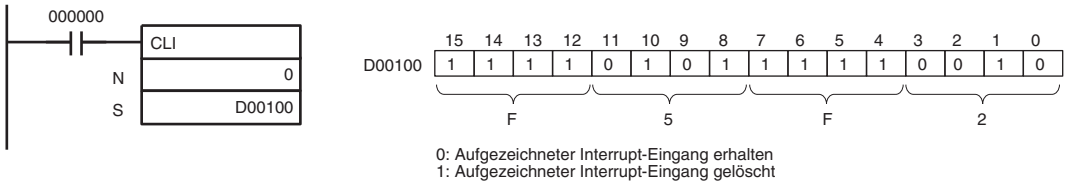
Für Interrupt-Tasks werden nur Interrupt-Eingänge von normalen Interrupt-Eingangsbaugruppen der CS/CJ-Serie und C200H-Interrupt-Eingangsbaugruppen unterstützt. Interrupt-Eingänge von internen Modulen und Spezial-E/A-Baugruppen werden nicht unterstützt.

Interrupts haben verschiedene Prioritätsstufen. Ein Ausschalt-Interrupt hat die höchste Priorität, gefolgt von E/A-Interrupts und schließlich zeitgesteuerten Interrupts. E/A-Interrupts mit niedrigen Nummern haben Priorität gegenüber E/A-Interrupts mit hohen Nummern.

Beispiele

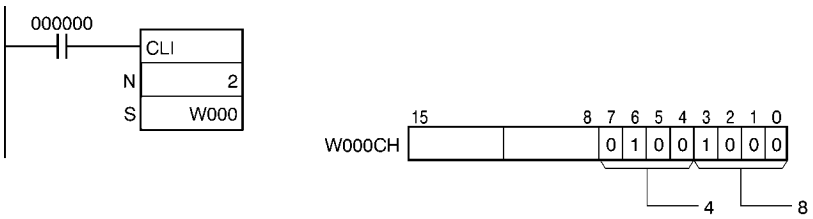
Beispiele für CS1W-INT01/CJ1W-INT01

Wenn CIO 000000 im folgenden Beispiel auf EIN wechselt, löscht der Befehl CLI(691) die aufgezeichneten Interrupts für Interrupt-Eingänge 1, 4 bis 8, 10 und 12 bis 15 in Interrupt-Eingangsbaugruppe 0.



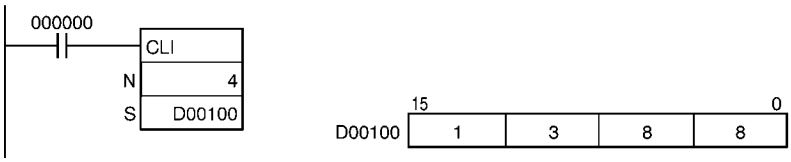
Beispiel für C200HS-INT01

Wenn CIO 000000 im folgenden Beispiel auf EIN wechselt, löscht der Befehl CLI(691) die aufgezeichneten Interrupts für die Interrupt-Eingänge 3 und 6 von Interrupt-Eingangsbaugruppe 2.



Beispiel für die Einstellung der Zeit bis zum ersten zeitgesteuerten Interrupt

Wenn CIO 000000 i folgenden Beispiel auf EIN wechselt, stellt der Befehl CLI(691) die Zeit bis zur ersten Ausführung des zeitgesteuerten Interrupts 2 auf 50 Sekunden ein.
(In diesem Fall ist die Zeiteinheit für das Intervall zeitgesteuerter Interrupts im SPS-Setup auf 10 ms eingestellt.)

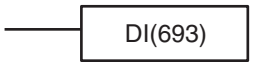


3-20-4 DISABLE INTERRUPTS: DI(693)

Zweck

Deaktiviert die Ausführung sämtlicher Interrupt-Tasks mit Ausnahme der Ausschalt-Interrupt-Task.
Bei Verwendung einer CS1D CPU-Baugruppe für Einzel-CPU-Systeme oder einer CS1-H, CJ1-H oder CJ1M CPU-Baugruppe, bei der die Ausschalt-Interrupt-Task deaktiviert ist, ist es möglich, die Ausschalt-Interrupt-Verarbeitung gleichzeitig zu deaktivieren.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	DI(693)
	Ausführung einmal bei steigender Flanke	@DI(693)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	Nicht zulässig

Beschreibung

Der Befehl DI(693) wird aus dem Hauptprogramm heraus ausgeführt, um alle Interrupt-Tasks mit Ausnahme der Ausschalt-Interrupt-Task (E/A-Interrupts, zeitgesteuerte Interrupts und externe interrupts) vorübergehend zu deaktivieren.

Alle Interrupt-Tasks bleiben so lange deaktiviert, bis sie durch die Ausführung des Befehls EI(694) wieder aktiviert werden.

CS1-H, CJ1-H und CJ1M CPU-Baugruppen und Ausschalt-Interrupts

Bei Verwendung einer CS1-H, CJ1-H oder CJ1M CPU-Baugruppe, kann die Verarbeitung des Ausschalt-Interrupts gleichzeitig deaktiviert werden, wenn A503 (die Deaktivierungseinstellung für Ausschalt-Interrupts) auf A5A5 hex gesetzt ist. Selbst wenn eine Unterbrechung der Spannungsversorgung nach Ausführung von DI(693) erkannt wird, wird die CPU-Baugruppe zurückgesetzt, nachdem die Befehle des Programms der Reihe nach bis zum Befehl EI(694) oder dem Befehl END(001) in der letzten Task ausgeführt wurden.

Wenn die Ausschalt-Interrupt-Task aktiviert ist, wird die CPU-Baugruppe nach Ausführung der Ausschalt-Interrupt-Task zurückgesetzt. Detaillierte Informationen hierzu finden Sie im Abschnitt über die Ausschalt-Interrupt-Task im *CS/CJ-Serie Programmierhandbuch*.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn DI(693) aus einer Interrupt-Task heraus ausgeführt wird. In allen anderen Fällen AUS.

Zugehörige Merker und Worte

Das folgende Wort befindet sich im Zusatz-Systembereich

Name	Adresse	Inhalt
Einstellung zur Deaktivierung von Ausschalt-Interrupts	A530	A5A5 hex: Aktivierung der Einstellung zur Deaktivierung von Ausschalt-Interrupts. Sie Ausschalt-Verarbeitung (ohne Ausführung der Ausschalt-Interrupt-Task) ist zwischen den Befehlen DI(694) und EI(694) maskiert, sodass die Befehle bis EI(694) ausgeführt werden.

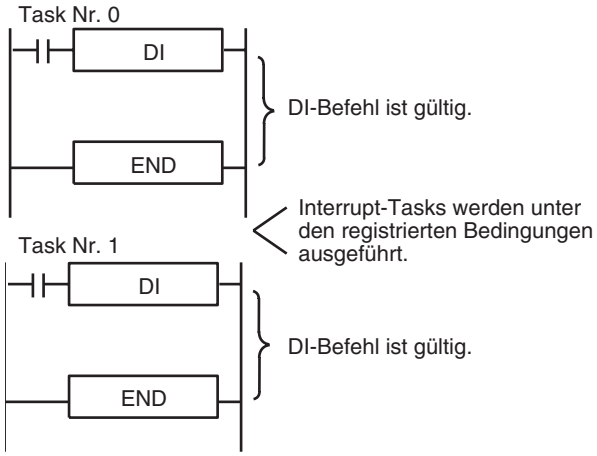
Vorsichtsmaßnahmen

Alle Interrupt-Tasks bleiben bis zur Ausführung des Befehls EI(694) deaktiviert.

DI(693) kann nicht aus einer Interrupt-Task heraus ausgeführt werden.

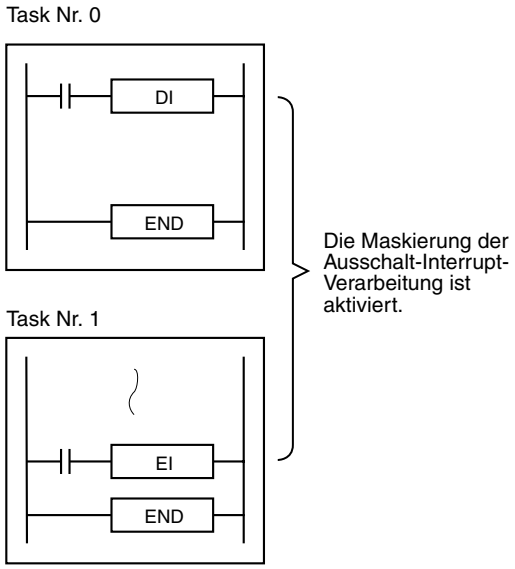
DI(693) kann nicht für mehr als eine zyklische Task ausgeführt werden. Zur Deaktivierung mehr als einer zyklisch ausgeführten Task muss der Befehl DI(693) in jeder dieser zyklischen Tasks eingefügt werden. Alle Interrupts, die während der Ausführung einer zyklischen Task auftreten, werden ausgeführt, wenn die Ausführung der zyklischen Task abgeschlossen ist, sofern die Interrupts nicht durch den Befehl CLI(691) deaktiviert sind (siehe folgendes Beispiel).

Bei Verwendung von DI(693) zur Deaktivierung der Ausschalt-Verarbeitung bei einer CS1-H, CJ1-H oder CJ1M CPU-Baugruppe, ist eine Deaktivierung der Verarbeitung durch die zyklischen Tasks möglich. (Der Deaktivierungszustand wird nach Abschluss der Ausführung aller gestarteten Tasks wieder aufgehoben.)



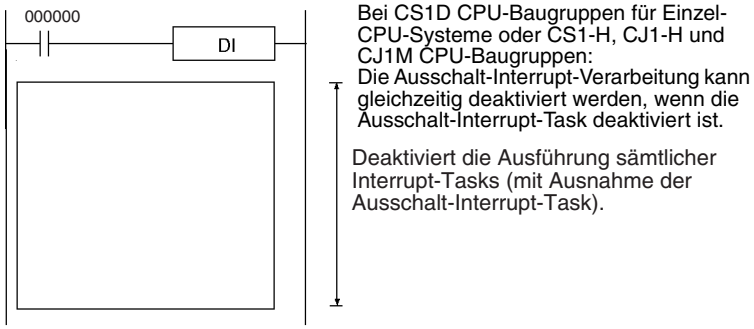
Wenn eine CS1D CPU-Baugruppe für Einzel-CPU-Systeme oder eine CS1-H, CJ1-H oder CJ1M CPU-Baugruppe verwendet wird, die Ausschalt-Interrupt-

Task deaktiviert ist und A530 auf A5A5 hex gesetzt ist, wird die CPU-Baugruppe im Falle der Erkennung einer Unterbrechung der Versorgungsspannung während der Ausführung der Befehle zwischen DI(693) und EI(694) nach der Ausführung von EI(694) zurückgesetzt.



Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt ist, deaktiviert der Befehl DI(693) alle Interrupt-Tasks mit Ausnahme der Ausschalt-Interrupt-Task.

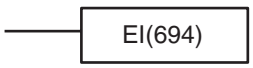


3-20-5 ENABLE INTERRUPTS: EI(694)

Zweck

Aktiviert die Ausführung aller durch DI(693) deaktivierten Interrupt-Tasks.
Bei Verwendung einer CS1D CPU-Baugruppe für Einzel-CPU-Systeme oder einer CS1-H, CJ1-H oder CJ1M CPU-Baugruppe, bei der die Ausschalt-Interrupt-Task deaktiviert ist, hebt der Befehl EI(694) gleichzeitig die Deaktivierung der Ausschalt-Interrupt-Verarbeitung auf.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	EI(694)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	Nicht zulässig

Beschreibung

Der Befehl EI(694) wird aus den Hauptprogramm heraus ausgeführt, um alle Interrupt-Tasks vorübergehend zu aktivieren, die durch den Befehl DI(693) deaktiviert wurden. Der Befehl DI(693) deaktiviert alle Interrupts (E/A-Interrupts, zeitgesteuerte Interrupts und externe Interrupts) mit Ausnahme des Ausschalt-Interrupts.

CS1-H, CJ1-H und CJ1M CPU-Baugruppen und Ausschalt-Interrupts

Wenn eine CS1-H, CJ1-H oder CJ1M CPU-Baugruppe verwendet wird und die Ausschalt-Interrupt-Verarbeitung durch DI(693) deaktiviert wurde, hebt der Befehl EI(694) auch die Ausführungssperre für die Ausschalt-Interrupt-Verarbeitung auf. Nach der Ausführung von DI(593) wird die CPU-Baugruppe nicht zurückgesetzt, selbst wenn eine Unterbrechung der Versorgungsspannung erkannt wird. Die CPU-Baugruppe wird zurückgesetzt, nachdem alle Befehle zwischen DI(693) und EI(694) ausgeführt wurden. Informationen zur Verwendung des Befehls DI(693) zur Deaktivierung der Ausschalt-Interrupt-Verarbeitung finden Sie unter *3-20-4 DISABLE INTERRUPTS: DI(693)*.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn EI(694) aus einer Interrupt-Task heraus ausgeführt wird. In allen anderen Fällen AUS.

Zugehörige Merker und Worte

Das folgende Wort befindet sich im Zusatz-Systembereich

Name	Adresse	Inhalt
Einstellung zur Deaktivierung von Ausschalt-Interrupts	A530	A5A5 hex: Aktivierung der Einstellung zur Deaktivierung von Ausschalt-Interrupts. Sie Ausschalt-Verarbeitung (ohne Ausführung der Ausschalt-Interrupt-Task) ist zwischen den Befehlen DI(694) und EI(694) maskiert, sodass die Befehle bis EI(694) ausgeführt werden. Jeder andere Wert: Deaktivierung der Maskierung für die Ausschalt-Interrupt-Verarbeitung.

Vorsichtsmaßnahmen

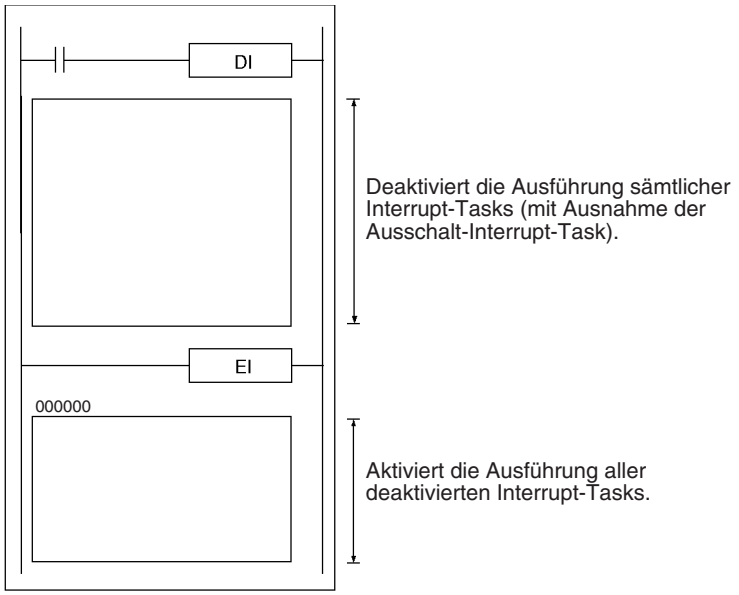
EI(694) erfordert keine Ausführungsbedingung. Der Befehl wird mit einer EIN-Ausführungsbedingung stets ausgeführt. EI(694) aktiviert die Interrupt-Tasks, die durch DI(693) deaktiviert wurden.

Der Befehl kann E/A-Interrupts nicht demaskieren, die nicht durch den Befehl MSKS(690) maskiert wurden, oder zeitgesteuerte Interrupts einstellen, die nicht durch MSKS(690) eingestellt wurden.

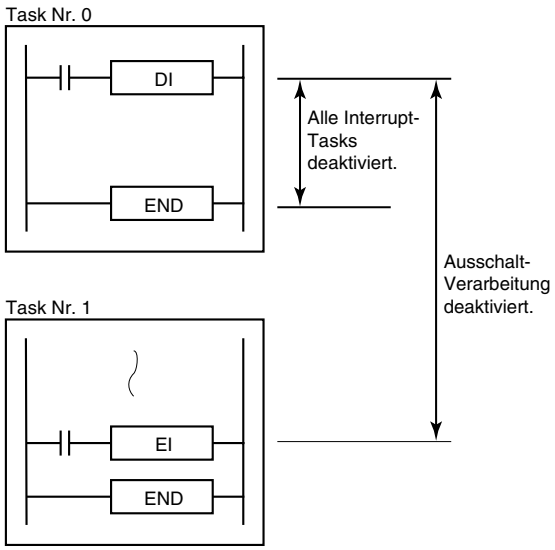
EI(694) kann nicht in einer Interrupt-Task ausgeführt werden.

Beispiele

In dem folgenden Beispiel aktiviert der Befehl EI(694) alle Interrupt-Tasks, die durch DI(693) deaktiviert wurden.



Hinweis Wenn die Ausschalt-Interrupt-Task bei einer CS1-H, CJ1-H oder CJ1M CPU-Baugruppe oder einer CS1D CPU-Baugruppe für Einzel-CPU-Systeme deaktiviert ist, wird gleichzeitig auch die Ausschalt-Interrupt-Verarbeitung aktiviert.



3-20-6 Übersicht über die Interrupt-Steuerung

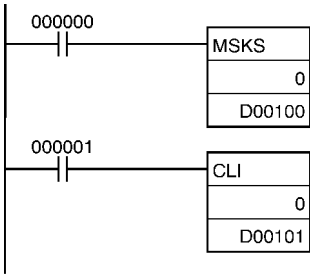
Die Befehle für die Steuerung von Interrupts steuern oder lesen die Einstellungen für E/A-Interrupts und zeitgesteuerte Interrupts. (Die Befehle DI(693) und EI(694) steuern die Funktion sowohl von externen Interrupts als auch von E/A-Interrupts und zeitgesteuerten Interrupts.)

Die Befehle, die sich auf einzelnen Interrupts auswirken, besitzen einen Operanden, N, der die Quelle des Interrupts identifiziert. Die Nummern 0 bis 3 spezifizieren die Interrupt-Eingangsbaugruppen 0 bis 3, die Nummern 4 und 5 spezifizieren die zeitgesteuerten Interrupts 2 und 3.

E/A-Interrupt-Verarbeitung (N = 0 bis 3)

Ein E/A-Interrupt wird durch ein Eingangssignal von einer Interrupt-Eingangsbaugruppe ausgelöst. An eine SPS können bis zu vier Interrupt-Eingangsbaugruppen angeschlossen werden. Den Baugruppen werden die Baugruppennummern 0 bis 3 gemäß ihrer Position im SPS-System von links nach rechts zugewiesen.

Das folgende Programmbeispiel zeigt die Funktion der Befehle MSKS(690) und CLI(691), wenn sie zur Steuerung von E/A-Interrupts verwendet werden.



D00100	0	0	F	0
D00101	0	0	F	2

Funktion von MSKS(690)

Beide Interrupt-Tasks (E/A und zeitgesteuert) sind maskiert (deaktiviert), wenn die SPS erstmalig eingeschaltet wird. MSKS(690) kann zum Maskieren oder Demaskieren von E/A-Interrupts verwendet werden sowie zum Einstellen der Intervallzeit für zeitgesteuerte Interrupts.

In diesem Beispiel verwendet der Befehl MSKS(690) den Inhalt von D00100, zur Demaskierung der Interrupt-Eingänge 0 bis 3 und zur Maskierung der Interrupt-Eingänge 4 bis 7 von Interrupt-Eingangsbaugruppe 0.

	F				0			
Interrupt-Eingänge von Baugruppe 0	7	6	5	4	3	2	1	0
Interrupt-Maskeneinstellungen	1	1	1	1	0	0	0	0

1 = Maskieren (Deaktivieren) 0 = Demaskieren (Aktivieren)

Wenn der Zustand von Interrupt-Eingang 3 von AUS zu EIN wechselt, wird die Ausführung des Hauptprogramms unterbrochen und E/A-Interrupt-Task 3 (Interrupt-Task 103) wird ausgeführt. Nach Abschluss der Ausführung von Interrupt-Task 3 wird die Ausführung des Hauptprogramms ab der Stelle der Unterbrechung wieder fortgesetzt.

E/A-Interrupt-Task Prioritätsstufen

Wenn zwei oder mehr Interrupt-Eingänge gleichzeitig empfangen werden, erfolgt die Ausführung der Interrupts in der Reihenfolge der Interrupt-Nummer von der niedrigsten zur höchsten Nummer (100 bis 131).

Baugruppe	Interrupt-Tasks
Interrupt-Eingangs-baugruppe 0	Eingänge 0 bis 7 entsprechen den E/A-Interrupt-Tasks 100 bis 107.
Interrupt-Eingangs-baugruppe 1	Eingänge 0 bis 7 entsprechen den E/A-Interrupt-Tasks 108 bis 115.
Interrupt-Eingangs-baugruppe 2	Eingänge 0 bis 7 entsprechen den E/A-Interrupt-Tasks 116 bis 123.
Interrupt-Eingangs-baugruppe 3	Eingänge 0 bis 7 entsprechen den E/A-Interrupt-Tasks 124 bis 131.

Wenn weitere Interrupt-Eingänge empfangen werden, während eine Interrupt-Task ausgeführt wird, werden die aufgezeichneten Interrupts in der Reihenfolge ihrer Priorität ausgeführt, nachdem die Ausführung der aktuellen Interrupt-Task angeschlossen ist.

Bei Auftreten eines zeitgesteuerten Interrupts hat die zeitgesteuerte Interrupt-Task Priorität gegenüber den E/A-Interrupt-Tasks.

Funktion von CLI(691)

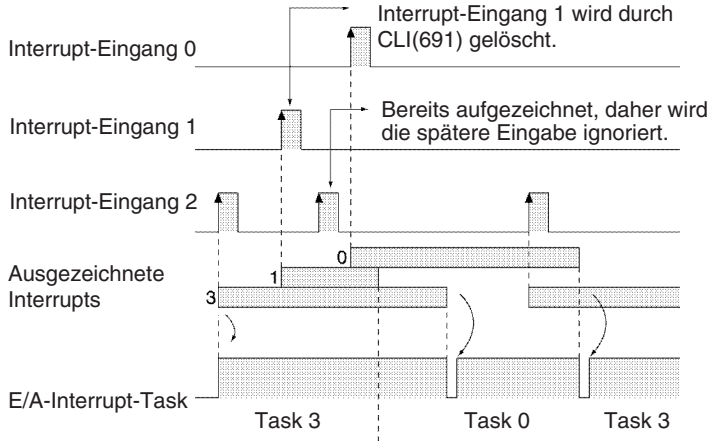
Wenn ein Interrupt-Eingang empfangen wird, während eine andere E/A-Interrupt-Task ausgeführt wird, erfolgt eine interne Aufzeichnung der Interrupt-Nummer des Eingangs, bis die Ausführung der aktuellen Interrupt-Task und aller anderen Tasks höherer Priorität abgeschlossen ist. Der Befehl CLI(691) kann dazu verwendet werden, die aufgezeichneten Interrupts zu löschen, bevor diese ausgeführt wurden. Ein Löschen von Interrupt-Tasks, die bereits ausgeführt werden, ist nicht möglich.

In diesem Beispiel verwendet der Befehl CLI(691) den Inhalt von D00101 zum Löschen aller aufgezeichneten Interrupt-Eingänge von Interrupt-Eingangsbaugruppe 0 mit Ausnahme der Eingänge 0, 2 und 3.

	F				2			
Interrupt-Eingänge von Baugruppe 0	7	6	5	4	3	2	1	0
Einstellungen für das Löschen/Erhalten von Interrupts	1	1	1	1	0	0	1	0

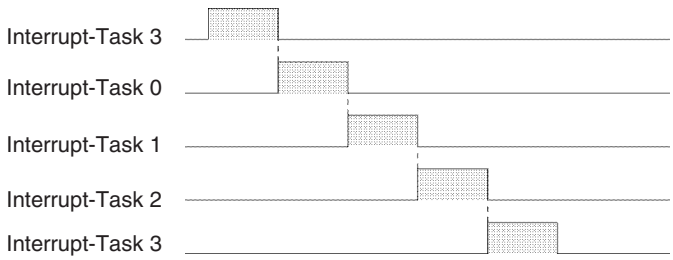
1 = Aufgezeichneten Eingang löschen 0 = Aufgezeichneten Eingang erhalten

Nach Abschluss der Ausführung von Interrupt-Task 3 werden die aufgezeichneten Interrupts in der Reihenfolge ihrer Priorität ausgeführt. Da ein Eingang von Interrupt-Eingang 0 aufgezeichnet wurde, wird E/A-Interrupt-Task 0 (Interrupt-Task 100) ausgeführt, wenn die Ausführung von Interrupt-Task 3 abgeschlossen ist. Interrupt-Eingang 1 wird nicht durch CLI(691) erhalten, sodass diese Eingabe gelöscht wird.



CLI(691) wird ausgeführt. Die Interrupt-Eingänge 0 und 3 bleiben erhalten, Eingang 1 wird gelöscht.

Wenn die Interrupt-Eingänge 0 bis 3 alle auf EIN wechseln und der Befehl CLI(691) nicht ausgeführt wird, werden diese Interrupts alle aufgezeichnet und die entsprechenden Interrupt-Tasks werden nach Abschluss der Ausführung von Interrupt-Task 3 in der Reihenfolge ihrer Priorität ausgeführt. (Die Interrupt-Tasks werden in der Reihenfolge ihrer Priorität ausgeführt, von der niedrigsten Interrupt-Nummer zur höchsten.)



Hinweis

1. Eine Verwendung des Befehls CLI(691) ist nicht immer erforderlich.
2. Wenn CLI(691) nicht ausgeführt wird, werden alle E/A-Interrupt-Eingänge aufgezeichnet, die während der Ausführung einer Interrupt-Task empfangen wurden. Wenn ein aufgezeichneter Interrupt-Eingang erneut empfangen wird, wird die zweite Eingabe ignoriert.
3. Wenn zwei oder mehr Interrupt-Eingänge aufgezeichnet sind, werden sie in der Reihenfolge ihrer Priorität ausgeführt. Die Reihenfolge, in der sie empfangen wurden, spielt keine Rolle,

Verarbeitung zeitgesteuerter Interrupts (N=4 oder 5)

Ein zeitgesteuerter Interrupt wird wiederholt in regelmäßigen Intervallen ausgeführt, die mit dem Befehl MSKS(690) eingestellt werden. Die Ausführungsintervalle sind unanhängig vom zeitlichen Ablauf des SPS-Zyklus. Die Nummern 4 und 5 für den Operanden N entsprechen den zeitgesteuerten Interrupts mit den Nummern 2 bzw. 3.

Verarbeitung zeitgesteuerter Interrupts

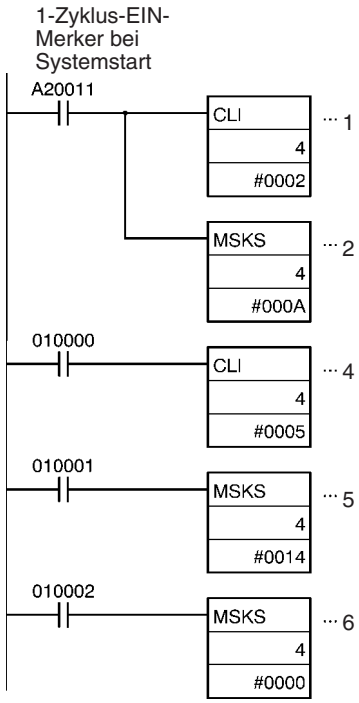
Nachfolgend finden Sie eine Aufstellung der wichtigsten Funktionen bei der Verarbeitung zeitgesteuerter Interrupts.

1,2,3...

1. Die zeitgesteuerten Interrupts sind beim ersten Einschalten der SPS maskiert (deaktiviert).
2. Stellen Sie die Zeit bis zum ersten zeitgesteuerten Interrupt (nach der Ausführung von MSKS(690)) mit dem Befehl CLI(691) ein. Die Zeit bis zum ersten zeitgesteuerten Interrupt ist nicht vorhersehbar, wenn sie nicht mit dem Befehl CLI(691) eingestellt wird.
3. Einstellung des Intervalls zeitgesteuerter Interrupts und Interrupt-Verarbeitung
   - Stellen Sie das Zeitintervall für zeitgesteuerte Interrupts mit dem Befehl MSKS(690) ein.
   - Nachdem der Befehl MSKS(690) ausgeführt wurde und die Zeit bis zum ersten zeitgesteuerten Interrupt (mit CLI(691) eingestellt) verstrichen ist, wird die derzeit ausgeführte Task unterbrochen und die zeitgesteuerte Interrupt-Task wird ausgeführt.
   - Wenn die Ausführung der zeitgesteuerten Interrupt-Task einen END(001)-Befehl erreicht, wird die Programmausführung an dem Punkt fortgesetzt, an dem sie durch den zeitgesteuerten Interrupt unterbrochen wurde.
   - Wenn das Intervall der zeitgesteuerten Interrupt-Task wieder abgelaufen ist, wird die Programmausführung wieder unterbrochen und die zeitgesteuerte Interrupt-Task wird erneut ausgeführt. Die zeitgesteuerte Interrupt-Task wird so lange wiederholt ausgeführt, bis sie deaktiviert wird.
4. Deaktivierung eines zeitgesteuerten Interrupts
   - Eine zeitgesteuerte Interrupt-Task kann deaktiviert werden, indem das Zeitintervall des zeitgesteuerten Interrupts durch den Befehl MSKS(690) auf 0000 gesetzt wird.
   - Wenn die zeitgesteuerte Interrupt-Task wieder aktiviert werden soll, muss darauf geachtet werden, zunächst die Zeit bis zum ersten zeitgesteuerten Interrupt mit dem Befehl CLI(691) einzustellen, bevor das Zeitintervall für den zeitgesteuerten Interrupt wieder mit dem Befehl MSKS(690) eingestellt wird.

Funktionsweise zeitgesteuerter Interrupts

Im folgenden Beispiel ist die Zeiteinheit für das Intervall zeitgesteuerter Interrupts im SPS-Setup auf 10 ms eingestellt.

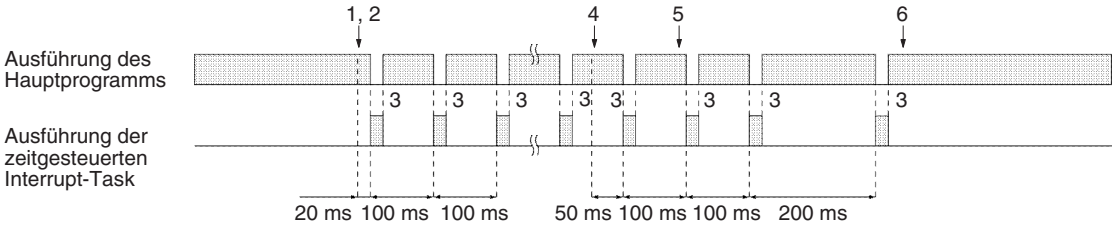


1,2,3...

1. Die Zeit bis zum ersten zeitgesteuerten Interrupt ist mit dem Befehl CLI(691) auf 20 ms eingestellt.

2. Durch den Befehl MSKS(690) ist das Zeitintervall für den zeitgesteuerten Interrupt auf 100 ms eingestellt und die Ausführung des zeitgesteuerten Interrupts 2 aktiviert.
3. Der zeitgesteuerte Interrupt 2 wird erstmalig 20 ms nach der Ausführung von MSKS(690) und in der Folge alle 100 ms ausgeführt.
4. Nachdem die Verarbeitung des zeitgesteuerten Interrupts begonnen hat, kann die Zeit bis zum nächsten zeitgesteuerten Interrupt mit dem Befehl CLI(690) geändert werden, jedoch ist diese Einstellung nur einmal wirksam.
5. Nachdem die Verarbeitung des zeitgesteuerten Interrupts begonnen hat, kann das Zeitintervall für den zeitgesteuerten Interrupt durch Ausführung des Befehls MSKS(690) geändert werden. In diesen Fall wird das Zeitintervall von 100 ms auf 200 ms geändert.
6. Die Verarbeitung des zeitgesteuerten Interrupts wird deaktiviert, indem der Befehl MSKS(690) ausgeführt wird, um das Zeitintervall auf 0000 zu setzen.

In dem folgenden Zeitablaufdiagramm ist die Funktionsweise des oben beschriebenen Beispiels dargestellt.

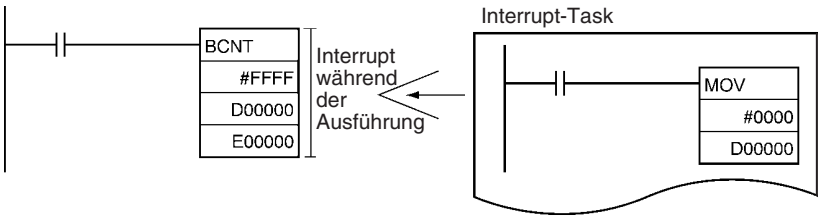


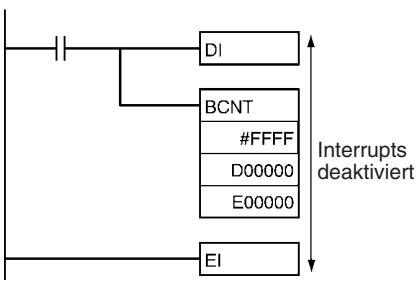
Vorsichtsmaßnahmen

Achten Sie darauf, dass das eingestellte Zeitintervall des zeitgesteuerten Interrupts länger ist als die zur Ausführung der zeitgesteuerten Interrupt-Task erforderliche Zeit. Wenn das Zeitintervall des zeitgesteuerten Interrupts zu kurz ist, wird die Interrupt-Task kontinuierlich ausgeführt und es tritt ein Zykluszeit-zu-lang-Fehler auf. (Eine zeitgesteuerte Interrupt-Task mit langer Ausführungszeit kann die Gesamtausführungszeit des Hauptprogramms stark verlängern.)

Die zeitgesteuerte Interrupt-Task wird nach Ablauf des spezifizierten Zeitintervalls zuzüglich der Ausführungszeit für einen Befehl ausgeführt. Normalerweise ist die Ausführungszeit eine Befehl zu vernachlässigen, jedoch können Fehler verursacht werden, wenn Befehle mit einer langen Ausführungszeit verwendet werden. Zudem können auch Fehler bei Zeitgebern (TIM und TIMH) sowie bei der Datenaufzeichnung verursacht werden. Lassen Sie besondere Sorgfalt walten, wenn die Zeiteinheit für das Intervall zeitgesteuerter Interrupts im SPS-Setup auf 0,5 ms oder 1 ms eingestellt ist.

Interrupts werden auch während der Ausführung eine Befehls akzeptiert. Dies hat zur Folge, dass möglicherweise keine korrekten Verarbeitungsergebnisse erhalten werden, da sowohl die Interrupt-Task als auch der ausgeführte Befehl auf die gleichen Daten zugreifen, wenn ein Interrupt während der Ausführung eines Befehls, der eine lange Verarbeitungszeit benötigt, akzeptiert wird. Verwenden Sie in einem derartigen Fall die Befehle DI(693) und EI(694) zur vorübergehenden Deaktivierung und Aktivierung des Interrupts.





3-21 Befehle für schnelle Zähler und Impulsausgaben

In diesem Abschnitt werden Befehle beschrieben, die zur Steuerung von schnellen Zählern und Impulsausgaben verwendet werden.

Befehl	AWL	Funktionscode	Seite
MODE CONTROL	INI	880	823
HIGH-SPEED COUNTER PV READ	PRV	881	827
COUNTER FREQUENCY CONVERT	PRV2	881	833
REGISTER COMPARISON TABLE	CTBL	882	837
SPEED OUTPUT	SPED	885	841
SET PULSES	PULS	886	846
PULSE OUTPUT	PLS2	887	849
ACCELERATION CONTROL	ACC	888	855
ORIGIN SEARCH	ORG	889	862
PULSE WITH VARIABLE DUTY FACTOR	PWM	891	865

3-21-1 MODE CONTROL: INI(880) (nur CJ1M-CPU21/22/23)

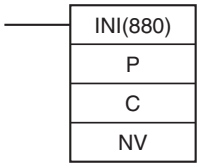
Zweck

INI(880) kann zur Ausführung der folgenden Operationen für integrierte E/A von CJ1M CPU-Baugruppen verwendet werden:

- Starten des Vergleichs mit der Vergleichstabelle des schnellen Zählers
- Anhalten des Vergleich mit der Vergleichstabelle des schnellen Zählers
- Ändern des Istwerts des schnellen Zählers
- Ändern des Istwerts der von Interrupt-Eingängen im Zählermodus
- Ändern des Istwerts der Impulsausgabe (Nullpunkt bei 0 festgelegt)
- Stoppen der Impulsausgabe

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 unterstützt.

Kontaktplansymbol



P: Schnittstellenbezeichner

C: Steuerdaten

NV: Erstes Wort mit neuem Istwert

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	INI(880)
	Ausführung einmal bei steigender Flanke	@INI(880)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

P: Schnittstellenbezeichner

P spezifiziert die Schnittstelle, auf die die Funktion angewendet wird.

P	Schnittstelle
0000 hex	Impulsausgang 0
0001 hex	Impulsausgang 1
0010 hex	Schneller Zähler 0

P	Schnittstelle
0011 hex	Schneller Zähler 1
0100 hex	Interrupt-Eingang 0 im Zählermodus
0101 hex	Interrupt-Eingang 1 im Zählermodus
0102 hex	Interrupt-Eingang 2 im Zählermodus
0103 hex	Interrupt-Eingang 3 im Zählermodus
1000 hex	PWM(891)-Ausgang 0
1001 hex	PWM(891)-Ausgang 1

C: Steuerdaten

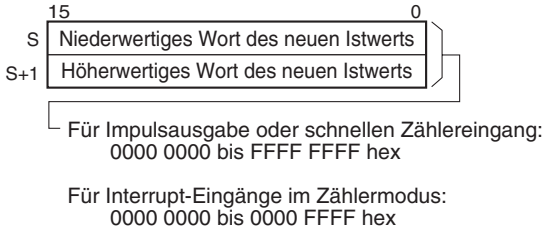
Die Funktion von INI(880) wird durch die Steuerdaten C bestimmt.

C	Funktion von INI(880)
0000 hex	Start des Vergleichs
0001 hex	Stoppen des Vergleichs
0002 hex	Ändern des Istwerts
0003 hex	Stoppen der Impulsausgabe

NV: Erstes Wort mit neuem Istwert

NV und NV+1 enthalten den neuen Istwert, wenn dieser geändert wird.

Wenn C auf 0002 hex gesetzt ist (d. h. beim Ändern eines Istwerts), enthalten NV und NV+1 den neuen Istwert. Alle Werte in NV und NV+1 werden ignoriert, wenn C nicht auf 0002 hex gesetzt ist.



Operanden-Spezifikationen

Bereich	P	C	NV
CIO-Bereich	---	---	CIO 0000 bis CIO 6142
Arbeitsbereich	---	---	W000 bis W510
Haftmerkerbereich	---	---	H000 bis H510
Zusatz-Systembereich	---	---	A448 bis A958
Zeitgeberbereich	---	---	T0000 bis T4094
Zählerbereich	---	---	C0000 bis C4094
DM-Bereich	---	---	D00000 bis D32766
EM-Bereich ohne Bank	---	---	---
EM-Bereich mit Bank	---	---	---
Indirekte DM/EM-Binäradresse	---	---	@ D00000 bis @ D32767
Indirekte DM/EM-Adresse in BCD	---	---	*D00000 bis *D32767
Konstanten	Siehe Beschreibung des Operanden.	Siehe Beschreibung des Operanden.	---
Datenregister	---	---	---

Bereich	P	C	NV
Indexregister	---	---	---
Indirektes Adressieren mittels Index-Registern	---	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

INI(880) führt die in C spezifizierte Funktion für die in P spezifizierte Schnittstelle aus. Mögliche Kombinationen von Funktionen und Schnittstellen sind in der folgenden Tabelle aufgeführt.

P: Schnittstellen-bezeichner	C: Steuerdaten			
	0000 hex: Vergleich starten	**0001 hex: Vergleich stoppen**	**0002 hex: Istwert ändern**	**0003 hex: Impuls-ausgabe stoppen**
0000 oder 0001 hex: Impulsausgang	Nicht zulässig	Nicht zulässig	OK	OK
0010 oder 0011 hex: Schneller Zählereingang	OK	OK	OK	Nicht zulässig
0100, 0101, 0102 oder 0103 hex: Interrupt-Eingang im Zählermodus	Nicht zulässig	Nicht zulässig	OK	Nicht zulässig
1000 oder 1001 hex: PWM(891)-Ausgang	Nicht zulässig	Nicht zulässig	Nicht zulässig	OK

■ **Vergleich starten (C = 0000 hex)**

Wenn C auf 0000 hex gesetzt ist, startet INI(880) den Vergleich des Istwerts eines schnellen Zählers mit der Vergleichstabelle, die mit CTBL(882) registriert wurde.

Hinweis Eine Zielwert-Vergleichstabelle muss im Vorfeld mit CTBL(882) registriert werden. Wenn INI(880) ohne Registrierung einer Tabelle ausgeführt wird, wird der Fehlermerker auf EIN gesetzt.

■ **Vergleich stoppen (C = 0001 hex)**

Wenn C auf 0001 hex gesetzt ist, stoppt INI(880) den Vergleich des Istwerts eines schnellen Zählers mit der Vergleichstabelle, die mit CTBL(882) registriert wurde.

■ **Istwert ändern (C = 0002 hex)**

Wenn C auf 0002 hex gesetzt ist, ändert INI(880) einen Istwert wie in der folgenden Tabelle gezeigt.

Schnittstelle und Modus			Funktion	Einstellbereich
Impulsausgang (P = 0000 oder 0001 hex)			Der Istwert der Impulsausgabe wird geändert. Der neue Wert ist in NV und NV+1 spezifiziert. **Hinweis:** Dieser Befehl kann nur ausgeführt werden, wenn die Impulsausgabe gestoppt wurde. Wenn dieser Befehl während einer Impulsausgabe ausgeführt wird, tritt ein Fehler auf.	8000 0000 bis 7FFF FFFF hex (-2.147.483.648 bis 2.147.483.647)
Schneller Zähleingang (P = 0010 oder 0011 hex)	Linearzähler-Modus	Differenzialeingänge, Inkrementier-/Dekrementier-Impulse, oder Impuls- + Richtungseingänge	Der Istwert des schnellen Zählers wird geändert. Der neue Wert ist in NV und NV+1 spezifiziert. **Hinweis:** Bei Ausführung des Befehls tritt ein Fehler auf, wenn die spezifizierte Schnittstelle nicht als schneller Zähler eingestellt ist.	8000 0000 bis 7FFF FFFF hex (-2.147.483.648 bis 2.147.483.647)
		Inkrementierender Impulseingang		0000 0000 bis FFFF FFFF hex (0 bis 4.294.967.295)
	Ringzähler-Modus			0000 0000 bis FFFF FFFF hex (0 bis 4.294.967.295)
Interrupt-Eingänge im Zählermodus (P = 0100, 0101, 0102 oder 0103 hex)			Der Istwert des Interrupt-Eingangs wird geändert. Der neue Wert ist in NV und NV+1 spezifiziert.	0000 0000 bis 0000 FFFF hex (0 bis 65.535) **Hinweis:** Es tritt ein Fehler auf, wenn ein außerhalb dieses Bereichs liegender Wert spezifiziert wird.

■ **Stoppen der Impulsausgabe (P = 1000 oder 1001 hex und C = 0003 hex)**

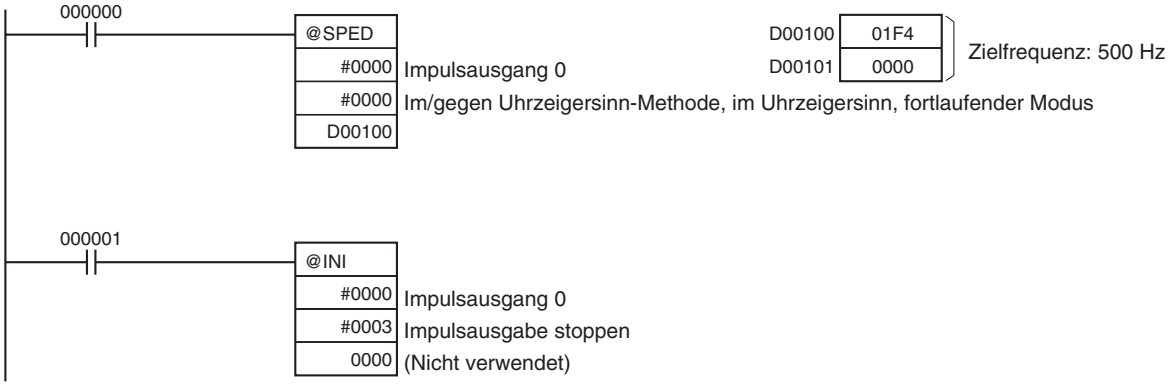
Wenn C auf 0003 hex gesetzt ist, stoppt INI(880) die Impulsausgabe über die spezifizierte Schnittstelle sofort. Wird dieser Befehl bei bereits gestoppter Impulsausgabe ausgeführt, wird die Einstellung der Impulsanzahl gelöscht.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der spezifizierte Bereich für P, C oder NV überschritten wird. EIN, wenn die Kombination von P und C nicht zulässig ist. EIN, wenn keine Vergleichstabelle registriert wurde, aber der Start des Vergleichs spezifiziert wird. EIN, wenn ein neuer Istwert für eine Schnittstelle spezifiziert wird, die derzeit Impulse ausgibt. EIN, wenn die Änderung des Istwerts eines schnellen Zählers für eine Schnittstelle spezifiziert wird, die nicht als schneller Zähler spezifiziert ist. EIN, wenn ein außerhalb des zulässigen Bereichs liegender Wert als Istwert für einen Interrupt-Eingang im Zählermodus spezifiziert wird. EIN, wenn INI(880) in einer Interrupt-Task für einen schnellen Zähler ausgeführt wird und bei Ausführung von CTBL(882) ein Interrupt auftritt. EIN, wenn der Befehl für eine nicht als Interrupt-Eingang im Zählermodus eingerichtete Schnittstelle ausgeführt wird.

Beispiel

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, beginnt SPED(885) mit der Ausgabe von Impulsen mit einer Frequenz von 500 Hz im fortlaufenden Modus über Impulsausgang 0. Wenn CIO 000001 auf EIN wechselt, wird die Impulsausgabe durch INI(880) gestoppt.



3-21-2 HIGH-SPEED COUNTER PV READ: PRV(881) (nur CJ1M-CPU21/22/23)

Zweck

PRV(881) liest die folgenden Daten der integrierten E/A von CJ1M-CPU-Baugruppen.

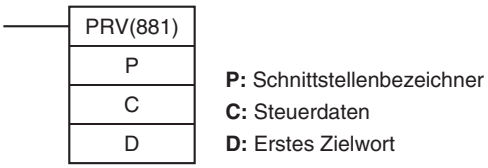
- Istwerte: Schneller-Zähler-Istwert, Impulsausgabe-Istwert, Interrupt-Eingangs-Istwert im Zählermodus.
- Die folgenden Statusinformationen.

Art des Status	Inhalt
Impulsausgabestatus	Impulsausgabestatus-Merker Istwert-Über-/Unterlauf-Merker Merker: Impulsausgabe-Sollzahl eingestellt Impulsausgabe-abgeschlossen-Merker Impulsausgabe-Merker Kein-Nullpunkt-Merker Nullpunkt-erreicht-Merker Fehlermerker: Impulsausgabe gestoppt
Schneller-Zähler-Eingangsstatus	Vergleichsvorgang-Merker Istwert-Über-/Unterlauf-Merker
PWM(891)-Ausgangsstatus	Merker: Impulsausgabe wird ausgeführt

- Bereichvergleichsergebnisse
- Impulsausgabefrequenz von Impulsausgang 0 oder Impulsausgang 1 (Nur von CJ1M CPU-Baugruppen ab Version 2.0 unterstützt.)
- Frequenz der Eingangsimpulse des schnellen Zählers 0.

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 unterstützt.

Kontaktplansymbol



Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	PRV(881)
	Ausführung einmal bei steigender Flanke	@PRV(881)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

P: Schnittstellenbezeichner

P spezifiziert die Schnittstelle, auf die die Funktion angewendet wird.

P	Schnittstelle
0000 hex	Impulsausgang 0
0001 hex	Impulsausgang 1
0010 hex	Schneller Zähler 0
0011 hex	Schneller Zähler 1
0100 hex	Interrupt-Eingang 0 im Zählermodus
0101 hex	Interrupt-Eingang 1 im Zählermodus
0102 hex	Interrupt-Eingang 2 im Zählermodus
0103 hex	Interrupt-Eingang 3 im Zählermodus
1000 hex	PWM(891)-Ausgang 0
1001 hex	PWM(891)-Ausgang 1

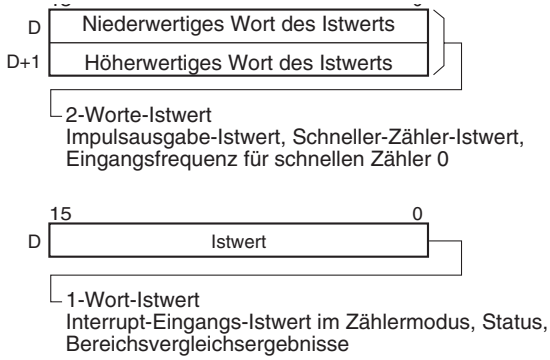
C: Steuerdaten

Die Funktion von INI(880) wird durch die Steuerdaten C bestimmt.

C	Funktion von PRV(881)	Variationen
0000 hex	Istwert wird gelesen.	---
0001 hex	Status wird gelesen	---
0002 hex	Bereichvergleichsergebnisse werden gelesen	---
00□3 hex	P = 0000 oder 0001 hex: Impulsausgabefrequenz von Impulsausgang 0 oder Impulsausgang 1 wird gelesen P = 0010 hex: Eingangsfrequenz des schnellen Zählers 0 wird gelesen.	C = 0003 hex: Standardbetrieb C = 0013 hex: 10-ms-Abtastmethode für Hochfrequenz (wird nur von CJ1M CPU-Baugruppen ab Version 3.0 unterstützt). C = 0023 hex: 100-ms-Abtastmethode für Hochfrequenz (wird nur von CJ1M CPU-Baugruppen ab Version 3.0 unterstützt). C = 0033 hex: 1-s-Abtastmethode für Hochfrequenz (wird nur von CJ1M CPU-Baugruppen ab Version 3.0 unterstützt).

D: Erstes Zielwort

Der Istwert wird in D bzw. in D und D+1 ausgegeben.



Operanden-Spezifikationen

Bereich	P	C	D
CIO-Bereich	---	---	CIO 0000 bis CIO 6142
Arbeitsbereich	---	---	W000 bis W510
Haftmerkerbereich	---	---	H000 bis H510
Zusatz-Systembereich	---	---	A448 bis A958
Zeitgeberbereich	---	---	T0000 bis T4094
Zählerbereich	---	---	C0000 bis C4094
DM-Bereich	---	---	D00000 bis D32766
EM-Bereich ohne Bank	---	---	---
EM-Bereich mit Bank	---	---	---
Indirekte DM/EM-Binäradresse	---	---	@ D00000 bis @ D32766
Indirekte DM/EM-Adresse in BCD	---	---	*D00000 bis *D32766
Konstanten	Siehe Beschreibung des Operanden.	Siehe Beschreibung des Operanden.	---
Datenregister	---	---	---
Indexregister	---	---	---
Indirektes Adressieren mittels Index-Registern	---	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

PRV(881) liest die in C spezifizierten Daten für die in P spezifizierte Schnittstelle. Mögliche Kombinationen von Daten und Schnittstellen sind in der folgenden Tabelle aufgeführt.

P: Schnittstellen-bezeichner	C: Steuerdaten			
	0000 hex: Istwert lesen	0001 hex: Status lesen	0002 hex: Bereich-vergleichs-ergebnisse lesen	0003 hex: Frequenz der Impuls-ausgabe/ des schnellen Zählers lesen
0000 oder 0001 hex: Impulsausgang	OK	OK	Nicht zulässig	OK (Nur CJ1M CPU-Baugruppen ab Ver. 3.0)
0010 oder 0011 hex: Schneller Zählereingang	OK	OK	OK	OK (nur schneller Zähler 0)
0100, 0101, 0102 oder 0103 hex: Interrupt-Eingang im Zählermodus	OK	Nicht zulässig	Nicht zulässig	Nicht zulässig
1000 oder 1001 hex: PWM(891)-Ausgang	Nicht zulässig	OK	Nicht zulässig	Nicht zulässig

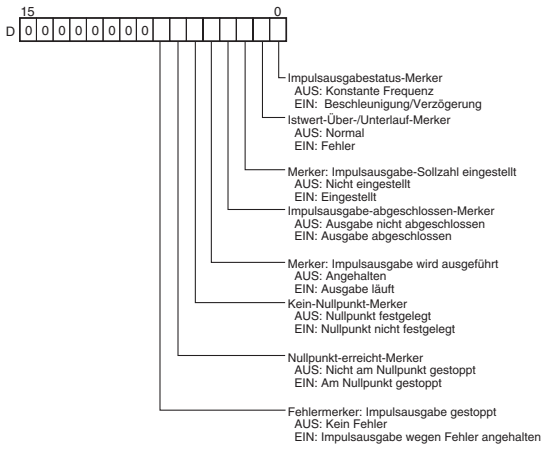
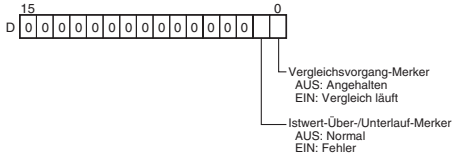
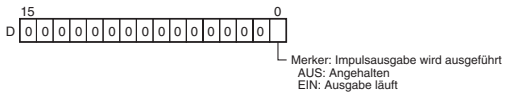
■ **Istwert lesen (C = 0000 hex)**

Wenn C auf 0000 hex gesetzt ist, liest PRV(881) einen Istwert wie in der folgenden Tabelle gezeigt.

Schnittstelle und Modus		Funktion	Einstellbereich
Impulsausgang (P = 0000 oder 0001 hex)		Der Istwert der1 Impulsausgabe wird in D und D+1 gespeichert.	8000 0000 bis 7FFF FFFF hex (-2.147.483.648 bis 2.147.483.647)
Schneller Zählereingang (P = 0010 oder 0011 hex)	Linear zähler-Modus	Der Istwert des schnellen Zählers wird in D und D+1 gespeichert.	8000 0000 bis 7FFF FFFF hex (-2.147.483.648 bis 2.147.483.647)
	Ring-zähler-Modus		0000 0000 bis FFFF FFFF hex (0 bis 4.294.967.295)
Interrupt-Eingänge im Zählermodus (P = 0100, 0101, 0102 oder 0103 hex)		Der Istwert des Interrupt-Eingangs wird in D gespeichert.	0000 bis FFFF hex (0 bis 65.535)

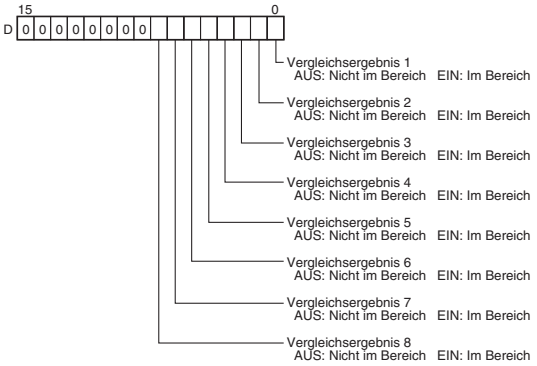
■ **Lesen des Status (C = 0001 hex)**

Wenn C auf 0001 hex gesetzt ist, liest PRV(881) den Status wie in der folgenden Tabelle gezeigt.

Schnittstelle und Modus	Funktion	Leseergebnisse
Impulsausgabe	Der Impulsausgabestatus wird in D gespeichert.	
Schneller Zähler-eingang	Der Status des schnellen Zählers wird in D gespeichert.	
PWM(891)-Ausgang	Der Status des PWM(891)-Ausgangs wird in D gespeichert.	

■ **Lesen der Ergebnisse des Bereichsvergleichs (C = 0002 hex)**

Wenn C auf 0002 hex gesetzt ist, liest PRV(881) die Ergebnisse des Bereichsvergleichs und speichert sie in D, wie im folgenden Schaubild dargestellt.



■ **Lesen der Frequenz des Impulsausgangs oder des schnellen Zählers (C = 00□3 hex)**

Wenn C auf 00□3 hex gesetzt ist, liest PRV(881) die am Impulsausgang 0 oder 1 ausgegebene oder die am Eingang des schnellen Zählers anliegende Frequenz (Hz) und speichert diese in D und D+1.

Frequenzbereiche

Wert von P	Konvertierungsresultat
0000 oder 0001 hex (Lesen der Frequenz von Impulsausgang 0 oder 1)	0000 0000 bis 0001 86A0 hex (0 bis 100.000)
0010 hex (Lesen der Frequenz des schnellen Zählers 0)	Zähler-Eingangseinstellung: Beliebige Eingangseinstellung außer der 4×-Differential-Phasen-Eingangseinstellung Resultat = 0000 0000 bis 0001 86A0 hex (0 bis 100.000) **Hinweis** Liegt die Eingangsfrequenz über 100 kHz, wird der Maximalwert 0001 86A0 hex ausgegeben.
	Zähler-Eingangseinstellung: 4×-Differential-Phasen-Eingangseinstellung Resultat = 0000 0000 bis 0003 0D40 hex (0 bis 200.000) **Hinweis** Liegt die Eingangsfrequenz über 200 kHz, wird der Maximalwert 0003 0D40 hex ausgegeben.

Impulsfrequenz-Berechnungsmethoden

CJ1M CPU-Baugruppen ab der Baugruppenversion 3.0 verfügen über zwei Möglichkeiten zur Berechnung der am Impulsausgang 0 oder 1 ausgegebenen bzw. am schnellen Zähler 0 anliegenden Frequenz.

1. Standard-Berechnungsmethode (bisherige Methode)

   Bestimmung der Frequenz durch von der Frequenz unabhängige Zählung jedes einzelnen Impulses. Bei höheren Frequenzen wird die steigende oder fallende Flanke mancher Impulse nicht korrekt wahrgenommen, wodurch es zu Messfehlern kommen kann (max. ca. 1 % Fehler bei 100 kHz).

2. Berechnungsmethode für hohe Frequenzen

   Bei dieser Methode erfolgt die Bestimmung bei hoher und niedriger Frequenz auf verschiedene Weise.

   - Frequenzbestimmung bei hohen Frequenzen

     Bei höheren Frequenzen (über 1 kHz) zählt die Funktion die Anzahl der innerhalb eines festen Intervalls (der so genannten Abtastzeit) auftretenden Impulse und berechnet aus dem Zählwert die vorliegende Frequenz. Die niederwertigsten zwei Stellen von C bestimmen, welche der drei zur Verfügung stehenden Abtastzeiten dabei angewandt wird.

Abtastzeit	Wert von C	Beschreibung
10 ms	0013 hex	Zählt über einen Zeitraum von 10 ms die Anzahl der Impulse. Bei 1 kHz liegt der Messfehler bei max. 10 %.
100 ms	0023 hex	Zählt über einen Zeitraum von 100 ms die Anzahl der Impulse. Bei 1 kHz liegt der Messfehler bei max. 1 %.
1 s	0033 hex	Zählt über einen Zeitraum von 1 s die Anzahl der Impulse. Bei 1 kHz liegt der Messfehler bei max. 0,1 %.

   - Frequenzbestimmung bei niedrigeren Frequenzen

     Bei Frequenzen unter 1 kHz wird ungeachtet der eingestellten Abtastzeit die Standard-Berechnungsmethode angewandt.

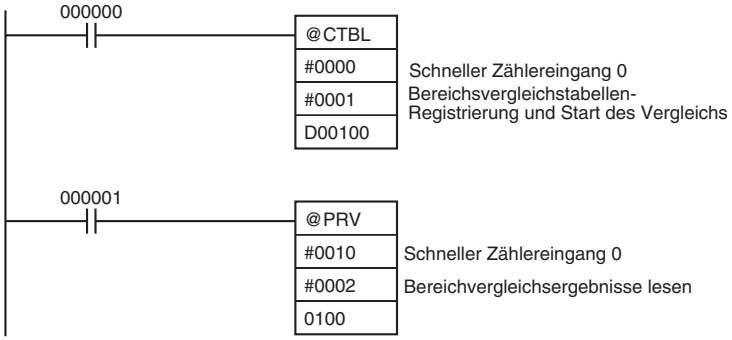
Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der spezifizierte Bereich für P oder C überschritten wird. EIN, wenn die Kombination von P und C nicht zulässig ist. EIN, wenn das Lesen von Bereichsvergleichsergebnissen spezifiziert ist, obwohl der Bereichsvergleich nicht ausgeführt wird. EIN, wenn das Lesen einer anderen Ausgabefrequenz außer der des schnellen Zählers 0 spezifiziert ist. EIN, wenn die spezifizierte Schnittstelle nicht als schneller Zähler eingestellt ist. EIN, wenn der Befehl für eine nicht als Interrupt-Eingang im Zählermodus eingerichtete Schnittstelle ausgeführt wird.

Beispiele

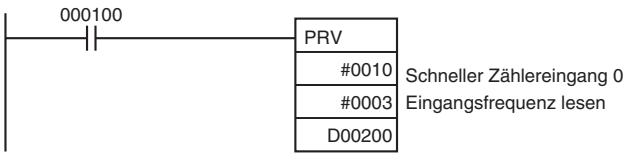
■ **Beispiel 1**

Wenn CIO 000000 im folgenden Programmbeispiel auf EIN wechselt, registriert CTBL(882) eine Bereichsvergleichstabelle für den schnellen Zähler 0 und startet den Vergleich. Wenn CIO 000001 auf EIN gesetzt wird, liest PRV(881) die Bereichsvergleichsergebnisse und speichert diese in CIO 0100.



■ **Beispiel 2**

Wenn CIO 000000 im folgenden Programmierbeispiel auf EIN gesetzt wird, liest PRV(881) die Frequenz der Impulse, die zu diesem Zeitpunkt an den schnellen Zähler 0 eingegeben werden, und speichert diese Frequenz als Hexadezimalwert in D00200 und D00201.



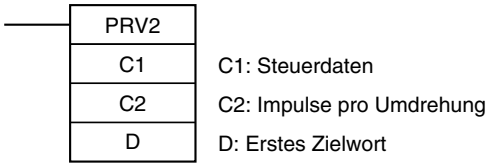
3-21-3 COUNTER FREQUENCY CONVERT: PRV2(883)

Zweck

PRV2(883) liest die Frequenz der Eingangsimpulse eines schnellen Zählers und konvertiert diese Frequenz entweder in eine Drehzahl oder den Zähler-Istwert in eine Gesamtzahl von Umdrehungen. Das Ergebnis wird als 8-stellige Hexadezimalzahl in die Zielworte ausgegeben. Impulsfrequenzen können nur vom schnellen Zähler 0 gelesen werden.

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 ab Baugruppenversion 2.0 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	PRV2(883)
	Ausführung einmal bei steigender Flanke	@PRV2(883)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

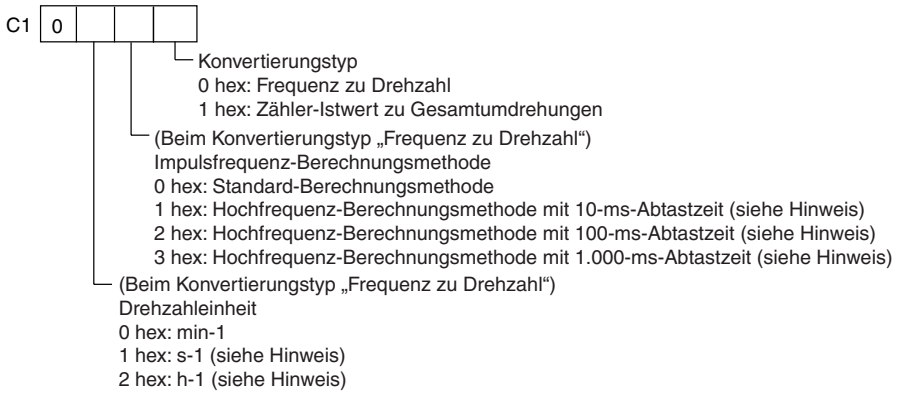
Operanden

C1: Steuerdaten

Die Funktion von PRV2(883) wird durch die Steuerdaten C1 bestimmt.

C1	Funktion von PRV2(883)
0□∗0 hex (Siehe Hinweis)	Konvertierung von Frequenz in Drehzahl
0001 hex	Konvertierung des Zähler-Istwerts in die Gesamtzahl von Umdrehungen

Hinweis Die zweite Stelle von C1 (□) spezifiziert die verwendete Einheit, die dritte Stelle (∗) die Impulsfrequenz-Berechnungsmethode.

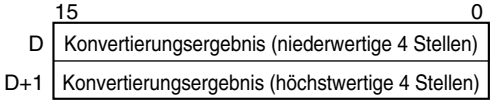


C2: Impulse pro Umdrehung

Spezifiziert die Anzahl von Impulsen pro Umdrehung (0001 bis FFFF hex).

D: Erstes Zielwort

Der Istwert wird in D bzw. in D und D+1 ausgegeben.



Operanden-Spezifikationen

Bereich	C1	C2	D
CIO-Bereich	---	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142
Arbeitsbereich	---	W000 bis W511	W000 bis W510

Bereich	C1	C2	D
Haftmerkerbereich	---	H000 bis H511	H000 bis H510
Zusatz-Systembereich	---	A448 bis A959	A448 bis A958
Zeitgeberbereich	---	T0000 bis T4095	T0000 bis T4094
Zählerbereich	---	C0000 bis C4095	C0000 bis C4094
DM-Bereich	---	D00000 bis D32767	D00000 bis D32766
EM-Bereich ohne Bank	---	---	---
EM-Bereich mit Bank	---	---	---
Indirekte DM/EM-Binäradresse	---	@ D00000 bis @ D32767	@ D00000 bis @ D32767
Indirekte DM/EM-Adresse in BCD	---	*D00000 bis *D32767	*D00000 bis *D32767
Konstanten	Siehe Beschreibung des Operanden.	---	---
Datenregister	---	---	---
Indexregister	---	---	---
Indirektes Adressieren mittels Index-Registern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

PRV2(883) konvertiert die Frequenz/Zahl der Eingangsimpulse des schnellen Zählers 0 gemäß der durch C1 spezifizierten Konvertierungsmethode und dem in C2 spezifizierten Koeffizienten für Impulse pro Umdrehung und gibt das Ergebnis nach D und D+1 aus.

Wählen Sie eine der folgenden Konvertierungsmethoden durch Setzen von C1 auf 0000 hex oder 0001 hex.

Konvertierung von Frequenz in Drehzahl (C1 = 0□∗0 hex)

Wenn C1 auf 0□∗0 hex gesetzt ist, berechnet PRV2(883) die Drehzahl (min-1) aus den Frequenzdaten und der Einstellung für Impulse/Umdrehung. Die zweite Stelle von C1 (□) spezifiziert die verwendete Einheit, die dritte Stelle (∗) die Impulsfrequenz-Berechnungsmethode.

1. Drehzahleinheiten
   - Drehzahleinheit = min-1

     Hat die zweite Stelle von C1 (□) den Wert 0, berechnet der Befehl PRV2(883) aus den Frequenzdaten und der Einstellung für Impulse/Umdrehung (C2) die Drehzahl in min-1.

     Drehzahl (min-1) = (Frequenz ÷ Impulse/Umdrehung) × 60
   - Drehzahleinheit = s-1 (nur CJ1M CPU-Baugruppen ab Ver. 3.0)

     Hat die zweite Stelle von C1 (□) den Wert 1, berechnet der Befehl PRV2(883) aus den Frequenzdaten und der Einstellung für Impulse/Umdrehung (C2) die Drehzahl in s-1.

     Drehzahl (s-1) = Frequenz ÷ Impulse/Umdrehung
   - Drehzahleinheit = h-1 (nur CJ1M CPU-Baugruppen ab Ver. 3.0)

     Hat die zweite Stelle von C1 (□) den Wert 2, berechnet der Befehl PRV2(883) aus den Frequenzdaten und der Einstellung für Impulse/Umdrehung (C2) die Drehzahl in h-1.

     Drehzahl (h-1) = (Frequenz ÷ Impulse/Umdrehung) × 60 × 60
   - Bereich der Konvertierungsresultate
     - Zähler-Eingangseinstellung: Beliebige Eingangseinstellung außer der 4×-Differential-Phasen-Eingangseinstellung
       Konvertierungsresultat = 0000 0000 bis 0001 86A0 hex
       (0 bis 100.000)

(Liegt die Eingangsfrequenz über 100 kHz, wird der Maximalwert 0001 86A0 hex ausgegeben.)

- Zähler-Eingangseinstellung: 4×-Differential-Phasen-Eingangseinstellung
  Konvertierungsresultat = 0000 0000 bis 0003 0D40 hex (0 bis 200.000)
  (Liegt die Eingangsfrequenz über 200 kHz, wird der Maximalwert 0003 0D40 hex ausgegeben.)

2. Frequenzberechnungsmethode

   CJ1M CPU-Baugruppen ab der Baugruppenversion 3.0 verfügen über zwei Möglichkeiten zur Berechnung der am schnellen Zähler 0 anliegenden Frequenz.

   a) Standard-Berechnungsnethode (C1 = 0□00)

   Bestimmung der Frequenz durch von der Frequenz unabhängige Zählung jedes einzelnen Impulses. Bei höheren Frequenzen wird die steigende oder fallende Flanke mancher Impulse nicht korrekt wahrgenommen, wodurch es zu Messfehlern kommen kann (max. ca. 1 % Fehler bei 100 kHz).

   b) Berechnungsmethode für hohe Frequenzen

   Bei dieser Methode erfolgt die Bestimmung bei hoher und niedriger Frequenz auf verschiedene Weise. (Nur von CJ1M CPU-Baugruppen ab Ver. 3.0 unterstützt)

   - Frequenzbestimmung bei hohen Frequenzen (C1 = 0□10, 0□20 oder 0□30)

     Bei höheren Frequenzen (über 1 kHz) zählt die Funktion die Anzahl der innerhalb eines festen Intervalls (der so genannten Abtastzeit) auftretenden Impulse und berechnet aus dem Zählwert die vorliegende Frequenz. Die dritte Stelle von C1 bestimmt, welche der drei zur Verfügung stehenden Abtastzeiten dabei angewandt wird.

Abtastzeit	Wert von C1	Beschreibung
10 ms	0□10 hex	Zählt über einen Zeitraum von 10 ms die Anzahl der Impulse. Bei 1 kHz liegt der Messfehler bei max. 10 %.
100 ms	0□20 hex	Zählt über einen Zeitraum von 100 ms die Anzahl der Impulse. Bei 1 kHz liegt der Messfehler bei max. 1 %.
1 s	0□30 hex	Zählt über einen Zeitraum von 1 s die Anzahl der Impulse. Bei 1 kHz liegt der Messfehler bei max. 0,1 %.

   - Frequenzbestimmung bei niedrigeren Frequenzen

     Bei Frequenzen unter 1 kHz wird ungeachtet der eingestellten Abtastzeit die Standard-Berechnungsmethode angewandt.

Konvertieren des Zähler-Istwerts in die Gesamtzahl von Umdrehungen (C1 = 0001 hex)

Wenn C1 auf 0001 hex gesetzt ist, berechnet PRV2(883) die Summe der Umdrehungen aus dem Zähler-Istwert und der Einstellung für Impulse/Umdrehung (C2).

Konvertierungsresultat = Zähler-Istwert ÷ Impulse/Umdrehung

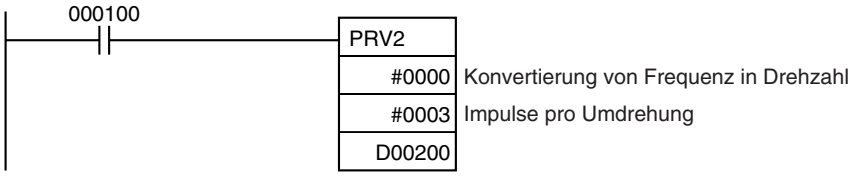
Merker

Name	Bezeichnung	Funktion
Fehler-merker	ER	EIN, wenn der schnelle Zähler 0 in den SPS-Einstellungen deaktiviert ist. EIN, wenn C1 kein zulässiger Wert ist (0000 oder 0001). EIN, wenn die Einstellung für Impulse/Umdrehung in C2 gleich 0000 ist.

Beispiele

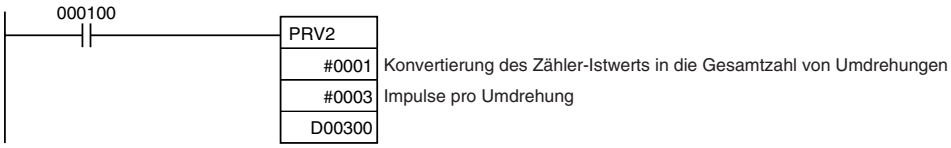
■ **Beispiel 1**

Wenn CIO 000100 im folgenden Programmierbeispiel auf EIN gesetzt wird, liest PRV(883) die aktuelle Frequenz der Impulse am schnellen Zähler 0, konvertiert diesen Wert in eine Drehzahl (min-1) und gibt das Ergebnis als Hexadezimalzahl nach D00201 und D00200 aus.



■ **Beispiel 2**

Wenn CIO 000100 im folgenden Programmierbeispiel auf EIN gesetzt wird, liest PRV(883) den Zähler-Istwert, konvertiert diesen Wert in eine Anzahl von Umdrehungen und gibt das Ergebnis als Hexadezimalzahl nach D00301 und D00300 aus.



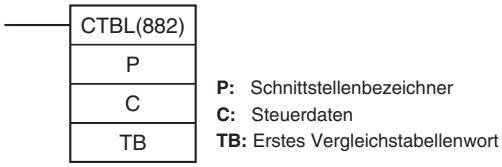
3-21-4 REGISTER COMPARISON TABLE: CTBL(882) (nur CJ1M-CPU21/22/23)

Zweck

CTBL(882) wird zur Registrierung einer Vergleichstabelle und zur Ausführung von Vergleichen mit dem Istwert eines schnellen Zählers verwendet. Es ist entweder ein Zielwert- oder ein Bereichsvergleich möglich. Bei Erfüllung einer spezifizierten Bedingung wird eine Interrupt-Task ausgeführt.

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CTBL(882)
	Ausführung einmal bei steigender Flanke	@CTBL(882)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

P: Schnittstellenbezeichner

P spezifiziert die Schnittstelle, deren Eingangsimpulse gezählt werden, wie in der folgenden Tabelle aufgeführt.

P	Schnittstelle
0000 hex	Schneller Zähler 0
0001 hex	Schneller Zähler 1

C: Steuerdaten

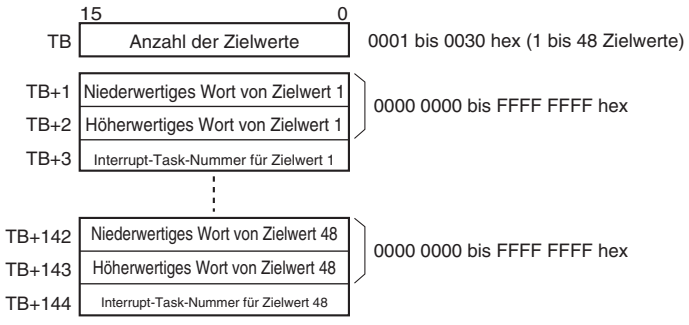
Die Funktion von CTBL(882) wird von den Steuerdaten C bestimmt, wie in der folgenden Tabelle aufgeführt.

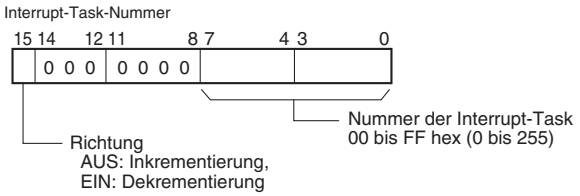
C	Funktion von CTBL(882)
0000 hex	Registrieren einer Zielwert-Vergleichstabelle und Starten des Vergleichs.
0001 hex	Registrieren einer Bereichs-Vergleichstabelle und Ausführen eines Vergleichs.
0002 hex	Registrieren einer Zielwert-Vergleichstabelle. Der Vergleich wird durch INI(880) gestartet.
0003 hex	Registrieren einer Bereichs-Vergleichstabelle. Der Vergleich wird durch INI(880) gestartet.

TB: Erstes Vergleichstabellenwort

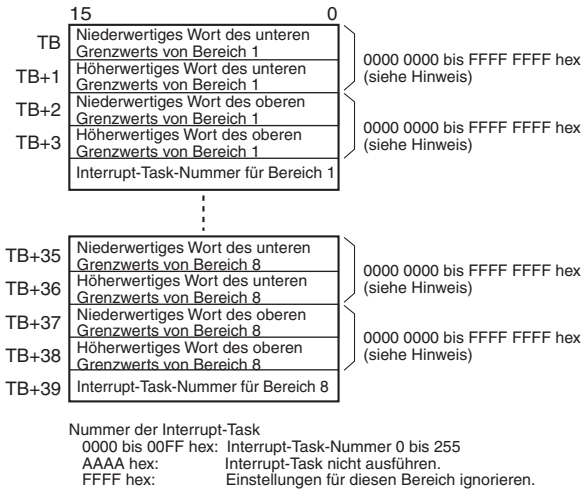
TB ist das erste Wort der Vergleichstabelle. Die Struktur der Vergleichstabelle hängt vom Typ des ausgeführten Vergleichs ab.

Für einen Zielwertvergleich wird die Länge der Vergleichstabelle durch die Anzahl der in TB spezifizierten Zielwerte bestimmt. Die Länge der Tabelle kann, wie unten gezeigt, zwischen 4 und 145 Worten betragen.





Für einen Bereichsvergleich enthält die Vergleichstabelle immer acht Bereiche. Die Länge der Tabelle beträgt, wie unten gezeigt, 40 Worte. Falls es nicht erforderlich ist, alle acht Bereiche zu verwenden, stellen Sie die Interrupt-Task-Nummer für alle nicht verwendeten Bereiche auf FFFF hex ein.



Nummer der Interrupt-Task
0000 bis 00FF hex: Interrupt-Task-Nummer 0 bis 255
AAAA hex: Interrupt-Task nicht ausführen.
FFFF hex: Einstellungen für diesen Bereich ignorieren.

Hinweis Stellen Sie die Obergrenze jedes Bereichs stets größer oder gleich der Untergrenze ein.

Operanden-Spezifikationen

Bereich	P	C	TB
CIO-Bereich	---	---	CIO 0000 bis CIO 6143
Arbeitsbereich	---	---	W000 bis W511
Haftmerkerbereich	---	---	H000 bis H511
Zusatz-Systembereich	---	---	A448 bis A959
Zeitgeberbereich	---	---	T0000 bis T4095
Zählerbereich	---	---	C0000 bis C4095
DM-Bereich	---	---	D00000 bis D32767
EM-Bereich ohne Bank	---	---	---
EM-Bereich mit Bank	---	---	---
Indirekte DM/EM-Binäradresse	---	---	@ D00000 bis @ D32767
Indirekte DM/EM-Adresse in BCD	---	---	*D00000 bis *D32767
Konstanten	Siehe Beschreibung des Operanden.	Siehe Beschreibung des Operanden.	---
Datenregister	---	---	---
Indexregister	---	---	---
Indirektes Adressieren mittels Index-Registern	---	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

CTBL(882) registriert eine Vergleichstabelle oder registriert eine Vergleichstabelle und startet den Vergleich für die in P spezifizierte Schnittstelle mit der in C spezifizierten Methode. Ist eine Vergleichstabelle registriert, bleibt sie so lange gültig, bis eine andere Tabelle registriert wird oder bis die CPU-Baugruppe in die PROGRAM-Betriebsart geschaltet wird.

Bei jeder Ausführung von CTBL(882) wird der Vergleich unter den spezifizierten Bedingungen gestartet. Wenn CTBL(882) zum Starten des Vergleichs verwendet wird, ist es normalerweise ausreichend, die Befehlsversion mit Flankenauswertung (@CTBL(882)) oder eine Ausführungsbedingung zu verwenden, die nur für einen Zyklus auf EIN gesetzt wird.

Hinweis Wenn eine nicht registrierte Interrupt-Task spezifiziert wird, erfolgt bei der ersten Erzeugung eines Interrupts ein schwerwiegender Programmfehler.

■ Registrieren einer Vergleichstabelle (C = 0002 oder 0003 hex)

Ist C auf 0002 oder 0003 hex gesetzt, wird eine Vergleichstabelle registriert, der Vergleich wird jedoch nicht gestartet. Der Vergleich wird durch INI(880) gestartet.

■ Registrieren einer Vergleichstabelle und Starten eines Vergleichs (C = 0000 oder 0001 hex)

Ist C auf 0000 oder 0001 hex gesetzt, wird eine Vergleichstabelle registriert und der Vergleich wird gestartet.

■ Stoppen des Vergleichs

Der Vergleich wird durch INI(880) gestoppt. Es macht dabei keinen Unterschied, welcher Befehl zum Start des Vergleichs verwendet wurde.

■ **Zielwertvergleich**

Wenn der Istwert mit einem Zielwert übereinstimmt, wird die entsprechende Interrupt-Task aufgerufen und ausgeführt.

- Die selbe Interrupt-Task-Nummer kann für mehr als einen Bereich spezifiziert werden.
- Es besteht die Möglichkeit die Richtung anzugeben, um zu spezifizieren, ob der Zielwert bei Inkrementierung oder Dekrementierung des Istwerts gültig ist. Wenn Bit 15 des zur Spezifizierung der Interrupt-Task-Nummer für den Bereich verwendeten Worts auf AUS gesetzt ist, wird der Istwert nur mit dem Zielwert verglichen, wenn der Istwert inkrementiert wird. Wenn Bit 00 auf EIN gesetzt ist, findet der Vergleich nur statt, wenn der Istwert dekrementiert wird.
- Die Vergleichstabelle kann bis zu 48 Zielwerte enthalten. Die Anzahl der Zielwerte wird im TB spezifiziert (d. h. die Länge der Tabelle ist von der Anzahl der spezifizierten Zielwerte abhängig).
- Vergleiche werden mit allen in der Tabelle registrierten Zielwerten durchgeführt.

Hinweis

1. Wenn derselbe Zielwert mit der gleichen Vergleichsrichtung in derselben Tabelle mehr als einmal registriert ist, tritt ein Fehler auf.
2. Bei Einstellung des schnellen Zählers auf Inkrement-Modus tritt ein Fehler auf, wenn die Dekrementierung in der Tabelle als Richtung für den Vergleich eingestellt ist.
3. Wenn sich die Zählrichtung ändert, während der Istwert einem Zielwert entspricht, der in Gegenrichtung zu der Richtung erreicht wurde, die als Vergleichsrichtung festgelegt wurde, wird die Vergleichsbedingung für diesen Zielwert nicht erfüllt. Stellen Sie Zielwerte nicht auf die Höchst- oder Mindestwerte des Zählers ein.

Bereichsvergleich

Wenn der Istwert innerhalb eines bestimmten Bereichs liegt, wird die entsprechende Interrupt-Task aufgerufen und ausgeführt.

- Dieselbe Interrupt-Task-Nummer kann für mehr als einen Bereich spezifiziert werden.
- Die Bereichsvergleichstabelle enthält 8 Bereiche, von denen jeder durch eine Ober- und Untergrenze definiert ist. Falls ein Bereich nicht verwendet werden soll, stellen Sie die Interrupt-Task-Nummer auf FFFF hex ein, um diesen Bereich zu deaktivieren.
- Die Interrupt-Task wird nur einmal ausgeführt, wenn der Istwert in den Bereich eintritt.
- Befindet sich der Istwert zum Zeitpunkt des Vergleichs in mehr als einem Bereich, wird der Interrupt-Task des Bereichs, der sich am nächsten zum Anfang der Tabelle befindet, Priorität gegeben und andere Interrupt-Tasks werden in den folgenden Zyklen ausgeführt.
- Gibt es keinen Grund für die Ausführung einer Interrupt-Task, kann AAAA hex als Interrupt-Task-Nummer spezifiziert werden. Die Ergebnisse des Bereichsvergleichs können mit PRV(881) gelesen werden oder durch Verwendung der Bereichsvergleich-in-Ausführung-Systemmerker.

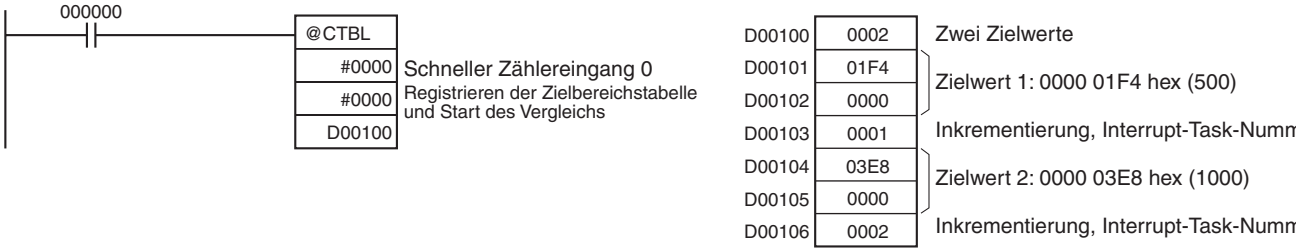
Hinweis Wenn die Obergrenze eines beliebigen Bereichs kleiner als seine Untergrenze ist, tritt ein Fehler auf.

Merker

Name	Bezeich-nung	Funktion
Fehler-merker	ER	EIN, wenn der spezifizierte Bereich für P oder C überschritten wird. EIN, wenn die Anzahl der für den Zielwertvergleich eingestellten Zielwerte auf 0 gesetzt ist. EIN, wenn die Anzahl der für den Zielwertvergleich eingestellten Zielwerte den Wert 48 überschreitet. EIN, wenn derselbe Zielwert mehr als einmal in derselben Vergleichsrichtung für den Zielvergleich spezifiziert wird. EIN, wenn der obere Grenzwert eines beliebigen Bereichs geringer als der untere Grenzwert ist. EIN, wenn die eingestellten Werte für alle Bereiche während eines Bereichsvergleichs deaktiviert sind. EIN, wenn der schnelle Zähler auf Inkrement-Modus und in der Tabelle Dekrementierung als Richtung für den Vergleich eingestellt ist. EIN, wenn ein Befehl bei Einstellung des schnellen Zählers auf Ringzählmodus ausgeführt wird und der spezifizierte Wert den maximalen Ringzählwert überschreitet. EIN, wenn die spezifizierte Schnittstelle nicht als schneller Zähler eingestellt ist. EIN, wenn der Befehl für eine andere Vergleichsmethode ausgeführt wird, während der Vergleich bereits im Gange ist.

Beispiel

Wenn CIO 000000 im folgenden Programmbeispiel auf EIN wechselt, registriert CTBL(882) eine Zielwertvergleichstabelle und beginnt mit dem Vergleich für den schnellen Zähler 0. Der Istwert des schnellen Zählers wird inkrementierend gezählt. Wenn er den Wert 500 erreicht, ist er gleich Zielwert 1 und dementsprechend wird Interrupt-Task 1 ausgeführt. Wenn der Istwert auf 1000 erhöht wird, entspricht er dem Zielwert 2 und Interrupt-Task 2 wird ausgeführt.



3-21-5 SPEED OUTPUT: SPED(885) (nur CJ1M-CPU21/22/23)

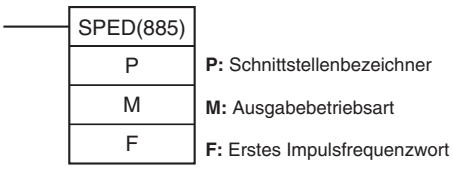
Zweck

SPED(885) wird zur Einstellung der Ausgangsimpulsfrequenz für eine spezifizierte Schnittstelle und zum Starten der Impulsausgabe ohne Beschleunigung oder Verzögerung verwendet. So ist eine Positionierung im einmaligen Modus oder eine Drehzahlregelung im fortlaufenden Modus möglich. Für die Positionierung im einmaligen Modus wird die Anzahl der Impulse unter Verwendung von PULS(886) eingestellt.

SPED(885) kann auch während der Impulsausgabe ausgeführt werden, um die Ausgabefrequenz zu ändern und auf diese Weise schrittweise Drehzahländerungen zu erzeugen.

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SPED(885)
	Ausführung einmal bei steigender Flanke	@SPED(885)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

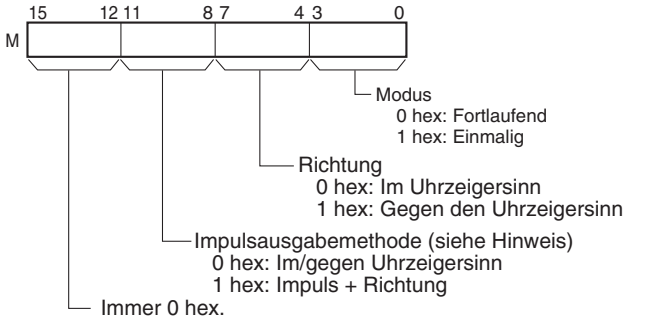
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

P: Schnittstellenbezeichner
Der Schnittstellenbezeichner spezifiziert die Schnittstelle, über die die Impulse ausgegeben werden.

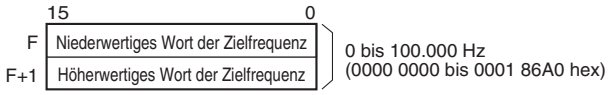
P	Schnittstelle
0000 hex	Impulsausgang 0
0001 hex	Impulsausgang 1

M: Ausgabebetriebsart
Der Wert von M bestimmt die Ausgabebetriebsart.



Hinweis: Verwenden Sie dieselbe Impulsausgabemethode, wenn Sie beide Impulsausgänge 0 und 1 verwenden.

F: Erstes Impulsfrequenzwort
Der Wert von F und F+1 bestimmt die Impulsfrequenz in Hz.



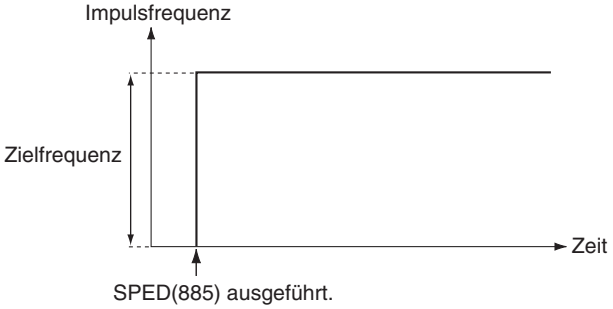
Operanden-Spezifikationen

Bereich	P	M	F
CIO-Bereich	---	---	CIO 0000 bis CIO 6142
Arbeitsbereich	---	---	W000 bis W510
Haftmerkerbereich	---	---	H000 bis H510
Zusatz-Systembereich	---	---	A448 bis A958
Zeitgeberbereich	---	---	T0000 bis T4094
Zählerbereich	---	---	C0000 bis C4094
DM-Bereich	---	---	D00000 bis D32766
EM-Bereich ohne Bank	---	---	---
EM-Bereich mit Bank	---	---	---
Indirekte DM/EM-Binäradresse	---	---	@ D00000 bis @ D32767
Indirekte DM/EM-Adresse in BCD	---	---	*D00000 bis *D32767

Bereich	P	M	F
Konstanten	Siehe Beschreibung des Operanden.	Siehe Beschreibung des Operanden.	Siehe Beschreibung des Operanden.
Datenregister	---	---	---
Indexregister	---	---	---
Indirektes Adressieren mittels Index-Registern	---	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

SPED(885) startet die Impulsausgabe über die in P spezifizierte Schnittstelle und verwendet dazu die in M spezifizierte Methode und die in F spezifizierte Frequenz. Die Impulsausgabe wird bei jeder Ausführung von SPED(885) gestartet. Es ist daher normalerweise ausreichend, die Befehlsvariante mit Flankenauswertung (@SPED(885)) oder eine Ausführungsbedingung zu verwenden, die nur für einen Zyklus auf EIN gesetzt wird.



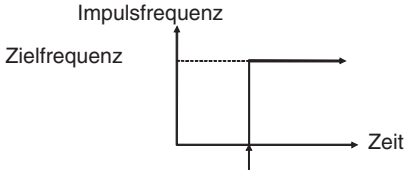
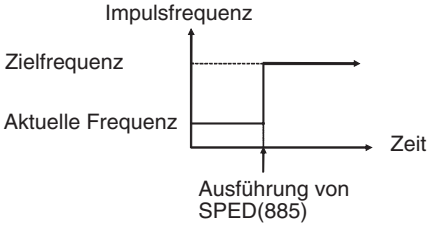
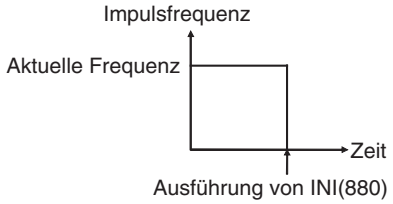
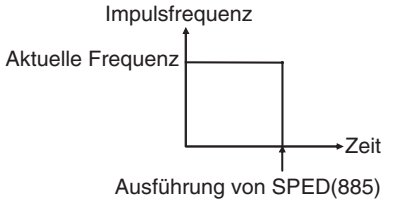
Im einmaligen Modus wird die Impulsausgabe automatisch gestoppt, wenn die im Voraus mit PULS(886) eingestellte Anzahl von Impulsen ausgegeben wurde. Im fortlaufenden Modus wird die Impulsausgabe so lange fortgesetzt, bis sie durch das Programm gestoppt wird.

Wenn während der Impulsausgabe zwischen dem einmaligen und fortlaufendem Modus gewechselt wird, tritt ein Fehler auf.

■ Fortlaufender Modus zur Drehzahlsteuerung

Wenn die Funktion im fortlaufenden Modus gestartet wird, geht die Impulsausgabe weiter, bis sie durch das Programm gestoppt wird.

Hinweis Die Impulsausgabe wird sofort gestoppt, wenn die CPU-Baugruppe in die PROGRAM-Betriebsart umgeschaltet wird.

Funktion	Zweck	Anwendung	Frequenzänderungen	Beschreibung	Verfahren/ Befehl
Starten der Impulsausgabe	Ausgabe mit spezifizierter Frequenz	Änderung der Drehzahl (Frequenz) in einem Schritt		Impulsausgabe mit festgelegter Frequenz.	SPED(885) (fortlaufend)
Ändern von Einstellungen	Ändern der Drehzahl in einem Schritt	Änderung der Drehzahl bei laufendem Betrieb		Ändert die Frequenz (auf- oder abwärts) der Impulsausgabe in einem Schritt.	SPED(885) (fortlaufend) ↓ SPED(885) (fortlaufend)
Stoppen der Impulsausgabe	Impulsausgabe stoppen	Unmittelbarer Stopp		Stoppt die Impulsausgabe unmittelbar.	SPED(885) (fortlaufend) ↓ INI(880)
	Impulsausgabe stoppen	Unmittelbarer Stopp		Stoppt die Impulsausgabe unmittelbar.	SPED(885) (fortlaufend) ↓ SPED(885) (fortlaufend, Zielfrequenz 0 Hz)

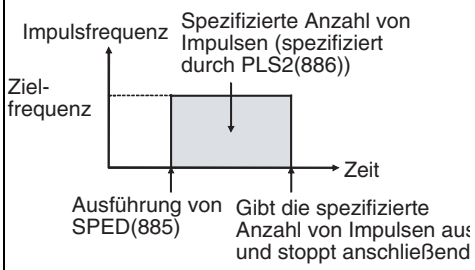
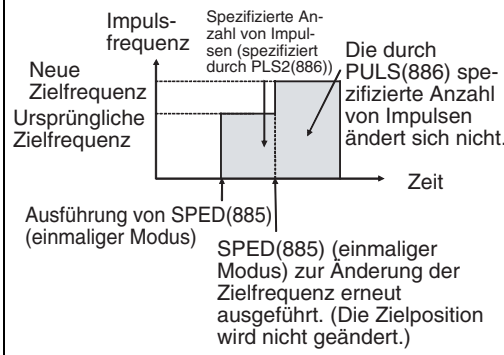
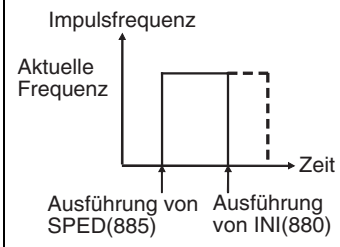
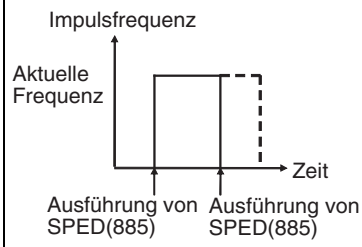
■ Einmaliger Modus zur Positionierung

Wenn die Funktion im einmaligen Modus gestartet wird, wird die Impulsausgabe so lange fortgesetzt, bis die spezifizierte Anzahl von Impulsen ausgegeben wurde.

Hinweis

1. Die Impulsausgabe wird sofort gestoppt, wenn die CPU-Baugruppe in die PROGRAM-Betriebsart umgeschaltet wird.
2. Die Anzahl der Ausgabeimpulse muss bei jedem Neustart der Ausgabe eingestellt werden.
3. Die Anzahl der Impulse muss im Vorfeld durch den Befehl PULS(886) eingestellt werden. Wenn nicht zuerst PULS(886) ausgeführt wird, werden durch SPED(885) keine Impulse ausgegeben.

4. Die im Operanden von SPED(885) festgelegte Richtung wird ignoriert, wenn die Anzahl der Impulse mit PULS(886) als Absolutwert eingestellt wird.

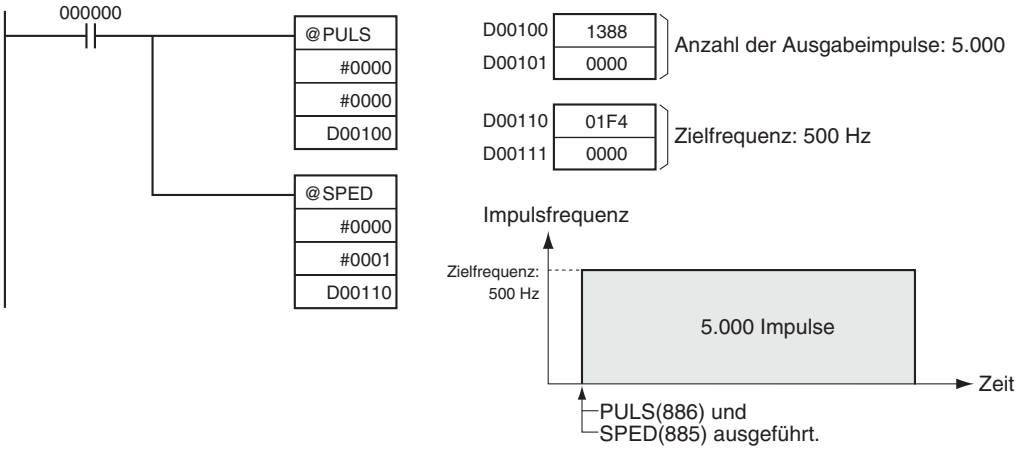
Funktion	Zweck	Anwendung	Frequenzänderungen	Beschreibung	Verfahren/ Befehl
Starten der Impulsausgabe	Ausgabe mit spezifizierter Frequenz	Positionierung ohne Beschleunigung oder Verzögerung		Startet die Impulsausgabe mit der spezifizierten Frequenz, und stoppt unmittelbar nach Ausgabe der spezifizierten Anzahl von Impulsen. **Hinweis** Die Zielposition (spezifizierte Impulsanzahl) kann während der Positionierung nicht verändert werden.	PULS(886) ↓ SPED(885) (einmalig)
Ändern von Einstellungen	Ändern der Drehzahl in einem Schritt	Änderung der Drehzahl in einem Schritt bei laufendem Betrieb		Der Befehl SPED(885) kann während der Positionierung ausgeführt werden, um die Frequenz der Impulsausgabe in einem Schritt zu ändern (anheben oder absenken). Die Zielposition (spezifizierte Impulsanzahl) kann nicht geändert werden.	PULS(886) ↓ SPED(885) (einmalig) ↓ SPED(885) (einmalig)
Stoppen der Impulsausgabe	Anhalten der Impulsausgabe (Einstellung der Impulsanzahl bleibt nicht erhalten.)	Unmittelbarer Stopp		Stoppt die Impulsausgabe unmittelbar und löscht die Einstellung für die Anzahl der Ausgabeimpulse.	PULS(886) ↓ SPED(885) (einmalig) ↓ INI(880) PLS2(887) ↓ INI(880)
	Anhalten der Impulsausgabe (Einstellung der Impulsanzahl bleibt nicht erhalten.)	Unmittelbarer Stopp		Stoppt die Impulsausgabe unmittelbar und löscht die Einstellung für die Anzahl der Ausgabeimpulse.	PULS(886) ↓ SPED(885) (einmalig) ↓ SPED(885) (einmalig, Zielfrequenz 0 Hz)

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der spezifizierte Bereich für P, M oder F überschritten wird. EIN, wenn PLS2(887) oder ORG(889) bereits zur Steuerung der Impulsausgabe über die spezifizierte Schnittstelle ausgeführt wird. EIN, wenn SPED(885) oder INI(880) während der Impulsausgabe zum Wechsel zwischen fortlaufendem und einmaligem Modus verwendet wird. EIN, wenn bei Ausführung von SPED(885) in einer Interrupt-Task bereits eine Impulsausgabe in einer zyklischen Task ausgeführt wird. EIN, wenn SPED(885) im einmaligen Modus mit einem Absolutwert für die Impulszahl ausgeführt wird und der Nullpunkt noch nicht bestimmt wurde.

Beispiel

Wenn CIO 000000 im folgenden Programmierbeispiel auf EIN gesetzt wird, stellt PULS(886) die Anzahl von Ausgabeimpulsen für Impulsausgang 0 ein. Es wird ein Absolutwert von 5000 Impulsen eingestellt. Anschließend wird SPED(885) mit der im/gegen den Uhrzeigersinn-Methode im Uhrzeigersinn und im einmaligen Modus mit einer Zielfrequenz von 500 Hz ausgeführt.



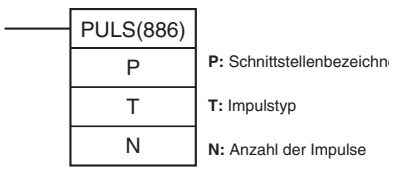
3-21-6 SET PULSES: PULS(886) (nur CJ1M-CPU21/22/23)

Zweck

PULS(886) wird zur Einstellung der Anzahl von Ausgabeimpulsen für Impulsausgaben verwendet, die später im Programm durch SPED(885) oder ACC(888) im einmaligen Modus gestartet werden.

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	PULS(886)
	Ausführung einmal bei steigender Flanke	@PULS(886)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbare Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

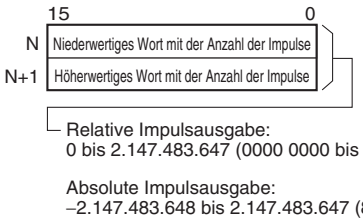
P: Schnittstellenbezeichner
Der Schnittstellenbezeichner spezifiziert die zu verwendende Schnittstelle. Die in D und N gesetzten Parameter gelten für den nächsten SPED(885)- oder ACC(888)-Befehl, in dem die gleiche Schnittstelle für die Ausgabe spezifiziert ist.

P	Schnittstelle
0000 hex	Impulsausgang 0
0001 hex	Impulsausgang 1

T: Impulstyp
T spezifiziert den Typ der ausgegebenen Impulse wie folgt:

T	Impulstyp
0000 hex	Relativ
0001 hex	Absolut

N und N+1: Anzahl der Impulse
N und N+1 spezifizieren die Anzahl von Impulsen bei relativer Impulsausgabe oder die absolute Zielposition bei absoluter Impulsausgabe als 8-stellige Hexadezimalzahl.



Die tatsächliche Anzahl der Bewegungsimpulse, die ausgegeben werden, ergibt sich wie folgt:

Bei der relativen Impulsausgabe ist die Anzahl der Bewegungsimpulse = die eingestellte Anzahl Impulse. Bei der absoluten Impulsausgabe ist die Anzahl der Bewegungsimpulse = die eingestellte Anzahl Impulse − dem Istwert.

Operanden-Spezifikationen

Bereich	P	T	N
CIO-Bereich	---	---	CIO 0000 bis CIO 6142
Arbeitsbereich	---	---	W000 bis W510
Haftmerkerbereich	---	---	H000 bis H510
Zusatz-Systembereich	---	---	A448 bis A958
Zeitgeberbereich	---	---	T0000 bis T4094
Zählerbereich	---	---	C0000 bis C4094
DM-Bereich	---	---	D00000 bis D32766
EM-Bereich ohne Bankangabe	---	---	---
EM-Bereich mit Bankangabe	---	---	---
Indirekte DM/EM-Binäradresse	---	---	@ D00000 bis @ D32767
Indirekte DM/EM-Adresse in BCD	---	---	*D00000 bis *D32767
Konstanten	Siehe Be-schreibung des Operanden.	Siehe Be-schreibung des Operanden.	Siehe Beschreibung des Operanden.
Datenregister	---	---	---

Bereich	P	T	N
Indexregister	---	---	---
Indirektes Adressieren mittels Index-Registern	---	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Mit PULS(886) wird der Impulstyp sowie die Anzahl von Impulsen (in T bzw. N spezifiziert) für die in P spezifizierte Schnittstelle eingestellt. Die tatsächliche Impulsausgabe wird später im Programm durch SPED(885) oder ACC(888) gestartet.

Merker

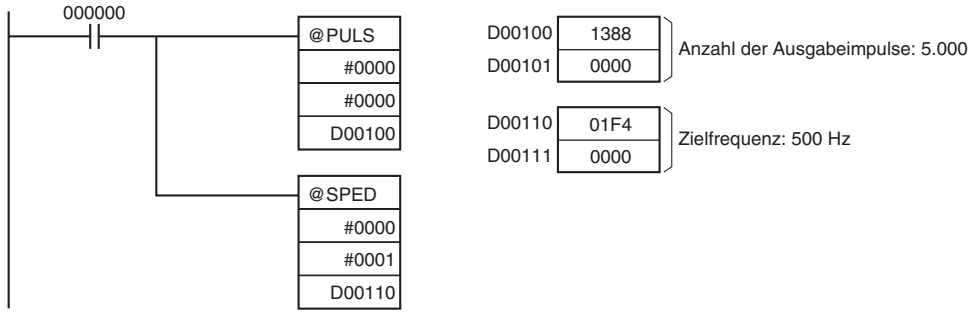
Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der spezifizierte Bereich für P, T oder N überschritten wird. EIN, wenn PULS(886) für eine Schnittstelle ausgeführt wird, die bereits Impulse ausgibt. EIN, wenn bei Ausführung von PULS(886) in einem Interrupt-Task bereits eine Impulsausgabe in einem zyklischen Task ausgeführt wird.

Vorsichtsmaßnahmen

- Wenn PULS(886) ausgeführt wird, obwohl bereits Impulse ausgegeben werden, tritt ein Fehler auf. Verwenden Sie die Befehlsvariante mit Flankenauswertung (@PULS(886)) oder eine Ausführungsbedingung, die nur für einen Zyklus auf EIN gesetzt wird, um dies zu vermeiden.
- Die berechnete Anzahl der durch PULS(886) ausgegebenen Impulse ändert sich nicht, auch wenn INI(880) zur Änderung des Istwerts für die Impulsausgabe verwendet wird.
- Die für SPED(885) oder ACC(888) festgelegte Richtung wird ignoriert, wenn die Anzahl der Impulse mit PULS(886) als Absolutwert eingestellt wird.
- Eine Überschreitung des Istwert-Bereichs für die Impulsausgabe ist möglich (–2.147.483.648 bis 2.147.483.647).

Beispiel

Wenn CIO 000000 im folgenden Programmierbeispiel auf EIN gesetzt wird, stellt PULS(886) die Anzahl von Ausgabeimpulsen für Impulsausgang 0 ein. Es wird ein Wert von 5000 Impulsen eingestellt. Anschließend werden mit SPED(885) im CW/CCW-Modus einmalig Pulse im Uhrzeigersinn mit einer Zielfrequenz von 500 Hz ausgegeben.



3-21-7 PULSE OUTPUT: PLS2(887) (nur CJ1M-CPU21/22/23)

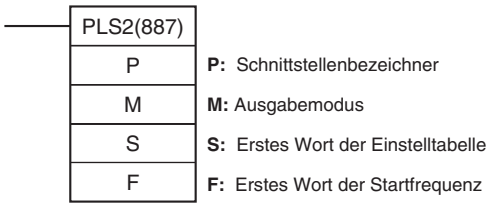
Zweck

PLS2(887) gibt eine spezifizierte Anzahl von Impulsen über die spezifizierte Schnittstelle aus. Die Impulsausgabe startet mit einer spezifizierten Startfrequenz, beschleunigt mit einer spezifizierten Beschleunigungsrate auf die Zielfrequenz, verzögert mit der spezifizierten Verzögerungsrate und stoppt bei einer Frequenz, die der Startfrequenz entspricht. Es wird nur die Positionierung im einmaligen Modus unterstützt.

Der Befehl PLS2(887) kann auch während einer Impulsausgabe ausgeführt werden, um die Anzahl der Ausgabeimpulse, Zielfrequenz, Beschleunigungsrate oder Verzögerungsrate zu ändern. Der Befehl PLS2 (887) kann dementsprechend auch für Drehzahländerungen mit unterschiedlichen Beschleunigungs- und Verzögerungsraten, Ziel- und Drehzahländerungen oder Richtungsänderungen verwendet werden.

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	PLS2(887)
	Ausführung einmal bei steigender Flanke	@PLS2 (887)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbare Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

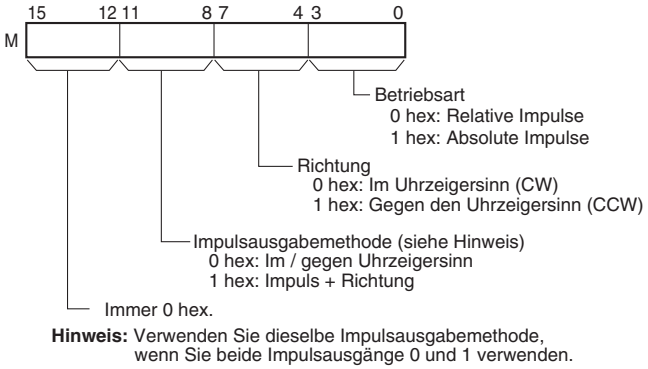
P: Schnittstellenbezeichner

Der Schnittstellenbezeichner spezifiziert die zu verwendende Schnittstelle.

P	Schnittstelle
0000 hex	Impulsausgang 0
0001 hex	Impulsausgang 1

M: Ausgabemodus

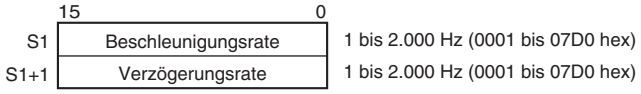
Der Inhalt von M spezifiziert die Parameter für die Impulsausgabe wie folgt:



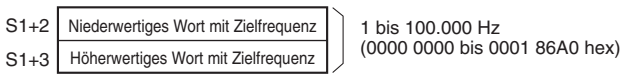
Hinweis: Verwenden Sie dieselbe Impulsausgabemethode, wenn Sie beide Impulsausgänge 0 und 1 verwenden.

S: Erstes Wort der Einstelltabelle

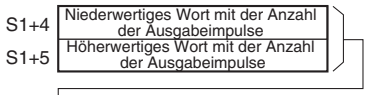
Der Inhalt von S bis S+5 steuert die Impulsausgabe wie in den folgenden Abbildungen dargestellt.



Spezifikation der Erhöhung bzw. Verringerung der Frequenz pro Impulssteuerungsperiode (4 ms).



Spezifikation der Frequenz in Hz nach Abschluss der Beschleunigung.



Relative Impulsausgabe: 0 bis 2.147.483.647
(0000 0000 bis 7FFF FFFF hex)

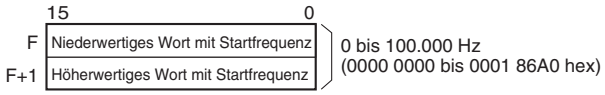
Absolute Impulsausgabe: -2.147.483.648 bis 2.147.483.647
(8000 0000 bis 7FFF FFFF hex)

Die tatsächliche Anzahl der Bewegungsimpulse, die ausgegeben werden, ergibt sich wie folgt:

Bei der relativen Impulsausgabe ist die Anzahl der Bewegungsimpulse = die eingestellte Anzahl Impulse. Bei der absoluten Impulsausgabe ist die Anzahl der Bewegungsimpulse = die eingestellte Anzahl Impulse − dem Istwert.

F: Erstes Wort der Startfrequenz

Die Startfrequenz ist in F und F+1 spezifiziert.



Spezifikation der Startfrequenz in Hz.

Operanden-Spezifikationen

Bereich	P	M	S	F
CIO-Bereich	---	---	CIO 0000 bis CIO 6138	CIO 0000 bis CIO 6142
Arbeitsbereich	---	---	W000 bis W506	W000 bis W510
Haftmerkerbereich	---	---	H000 bis H506	H000 bis H510
Zusatz-Systembereich	---	---	A448 bis A954	A448 bis A958
Zeitgeberbereich	---	---	T0000 bis T4090	T0000 bis T4094
Zählerbereich	---	---	C0000 bis C4090	C0000 bis C4094
DM-Bereich	---	---	D00000 bis D32762	D00000 bis D32766
EM-Bereich ohne Bankangabe	---	---	---	---
EM-Bereich mit Bankangabe	---	---	---	---
Indirekte DM/EM-Binäradresse	---	---	@ D00000 bis @ D32767	@ D00000 bis @ D32767
Indirekte DM/EM-Adresse in BCD	---	---	*D00000 bis *D32767	*D00000 bis *D32767
Konstanten	Siehe Beschreibung des Operanden.	Siehe Beschreibung des Operanden.	---	Siehe Beschreibung des Operanden.
Datenregister	---	---	---	---

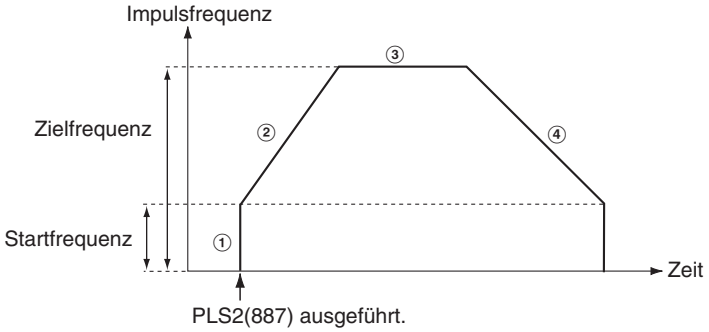
Bereich	P	M	S	F
Indexregister	---	---	---	---
Indirektes Adressieren mittels Index-Registern	---	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

PLS2(887) startet die Impulsausgabe über die in P spezifizierte Schnittstelle und verwendet den in M spezifizierten Modus mit der in F spezifizierten Startfrequenz. (1 in der Abbildung). Die Frequenz wird in jeder Impulssteuerungsperiode (4 ms) um die in S spezifizierte Beschleunigungsrate erhöht, bis die in S spezifizierte Zielfrequenz erreicht ist (2 in der Abbildung). Wenn die Zielfrequenz erreicht ist, wird die Beschleunigung gestoppt und die Impulsausgabe mit konstanter Frequenz (3 in der Abbildung) fortgesetzt.

Der Verzögerungspunkt wird aus der Anzahl der Ausgabeimpulse und der in S eingestellten Verzögerungsrate berechnet. Wenn dieser Punkt erreicht ist, wird die Frequenz in jeder Impulssteuerperiode (4 ms) um die in S spezifizierte Verzögerungsrate verringert, bis die in S spezifizierte Startfrequenz erreicht ist. An diesem Punkt wird die Impulsausgabe gestoppt (4 in der Abbildung).

Die Impulsausgabe wird bei jeder Ausführung von PLS2(887) gestartet. Es ist daher normalerweise ausreichend, die Befehlsvariante mit Flankenauswertung (@PLS(887)) oder eine Ausführungsbedingung zu verwenden, die nur für einen Zyklus auf EIN gesetzt wird.



PLS(887) kann nur zur Positionierung verwendet werden.

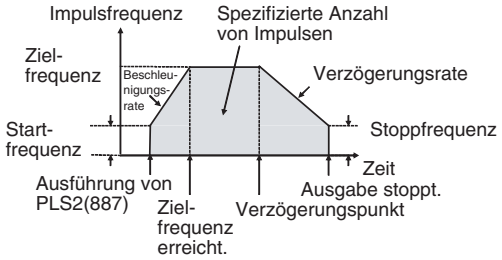
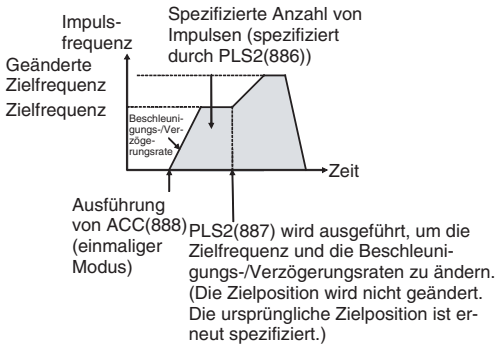
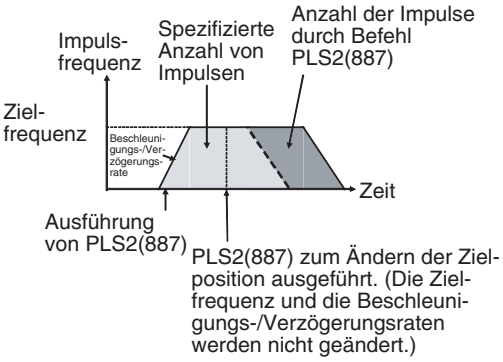
Bei den CJ1M CPU-Baugruppen kann PLS2(887) während der Impulsausgabe durch ACC(888) im einmaligen oder im fortlaufenden Modus sowie während der Beschleunigung, bei konstanter Frequenz oder Verzögerung ausgeführt werden. (Siehe Hinweis.) ACC(888) kann auch während der Impulsausgabe durch PLS2(887) während der Beschleunigung, bei konstanter Frequenz oder Verzögerung ausgeführt werden.

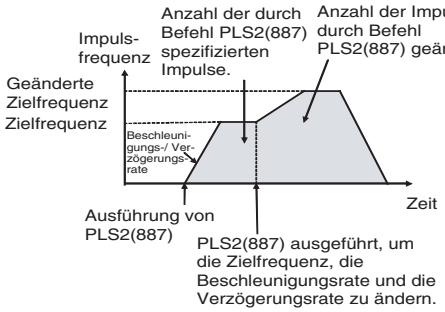
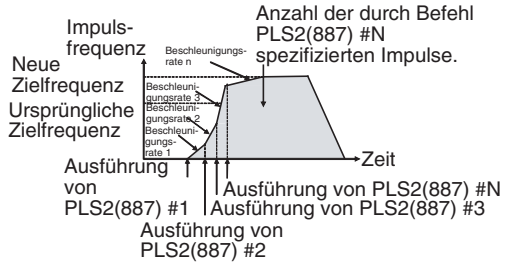
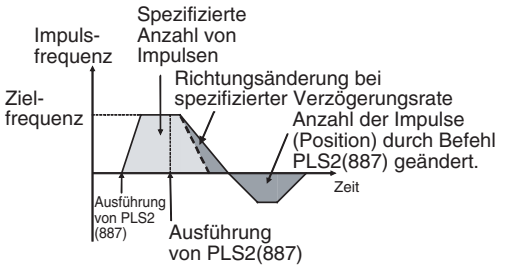
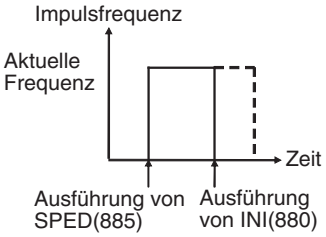
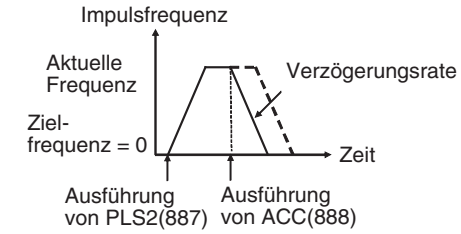
Hinweis

Die Ausführung von PLS2(887) kann während der Drehzahlsteuerung mit ACC(888) (fortlaufender Modus) mit der selben Zielfrequenz wie ACC(888) verwendet werden, um einen Interrupt-Vorschub um eine feste Strecke zu erzielen. Bei dieser Anwendung wird durch PLS2(887) keine Beschleunigung ausgeführt. Ist die Beschleunigungsrate jedoch auf 0 gesetzt, wird der Fehler-Merker auf EIN gesetzt und PLS2(887) wird nicht ausgeführt. Stellen Sie die Beschleunigungsrate immer auf einen anderen Wert als 0 ein.

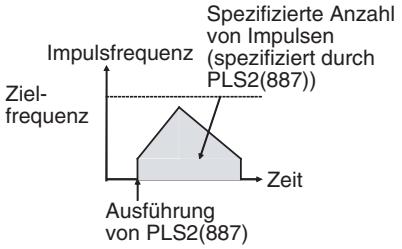
■ Einmaliger Modus zur Positionierung

Hinweis Die Impulsausgabe wird sofort gestoppt, wenn die CPU-Baugruppe in die PROGRAM-Betriebsart umgeschaltet wird.

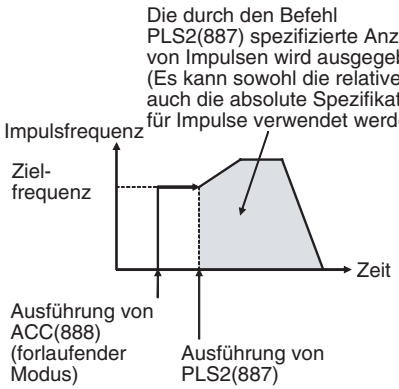
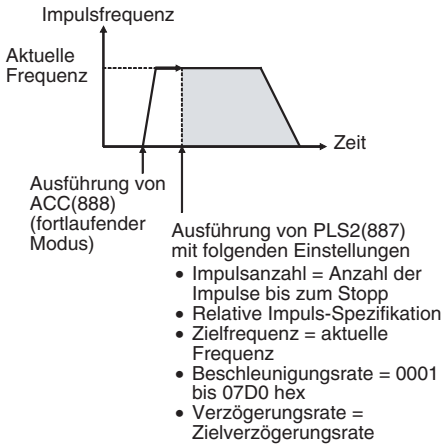
Funktion	Zweck	Anwendung	Frequenzänderungen	Beschreibung	Verfahren/ Befehl
Starten der Impulsausgabe	Komplexe trapezförmige Impulsausgabe	Positionierung mit trapezför-miger Beschleunigung und Verzögerung (unterschiedliche Raten für Beschleunigung und Verzögerung, Anfangsdrehzahl) Die Anzahl von Impulsen kann während der Positionierung geändert werden.		Beschleunigt und verzögert mit festen Raten. Die Impulsausgabe wird gestoppt, wenn die festgelegte Anzahl von Impulsen ausgegeben wurde. (Siehe Hinweis.) **Hinweis** Die Zielposition (festgelegte Impulsanzahl) kann während der Positionierung geändert werden.	PLS2(887)
Ändern von Einstellungen	Allmähliche Drehzahländerung (mit unterschiedlichen Beschleunigungs- und Verzögerungsraten)	Änderung der Zieldrehzahl (Frequenz) während der Positionierung (unterschiedliche Beschleunigungs- und Verzögerungsraten)		Der Befehl PLS2(887) kann während der Positionierung ausgeführt werden, um die Beschleunigungsrate, die Verzögerungsrate und die Zielfrequenz zu ändern. **Hinweis** Damit die Zielposition nicht versehentlich geändert wird, muss die ursprüngliche Zielposition in absoluten Koordinaten angegeben werden.	PLS2(887) ↓ PLS2(887) PULS(886) ↓ ACC(888) (einmalig) ↓ PLS2(887)
	Änderung der Zielposition	Änderung der Zielposition während der Positionierung (Mehrfachstartfunktion)		Der Befehl PLS2(887) kann während der Positionierung ausgeführt werden, um die Zielposition (Anzahl von Impulsen), die Beschleunigungsrate, die Verzögerungsrate und die Zielfrequenz zu ändern. **Hinweis** Wenn nach Änderung der Einstellungen keine konstante Drehzahl beibehalten werden kann, tritt ein Fehler auf und der ursprüngliche Betrieb wird zur ursprünglichen Zielposition fortgesetzt.	PLS2(887) ↓ PLS2(887) PULS(886) ↓ ACC(888) (einmalig) ↓ PLS2(887)

Funktion	Zweck	Anwendung	Frequenzänderungen	Beschreibung	Verfahren/ Befehl
Ändern von Einstellungen, Fortsetzung	Allmähliche Änderung von Zielposition und Drehzahl	Veränderung der Zielposition und Zieldrehzahl (Frequenz) während der Positionierung (Mehrfachstartfunktion)		Der Befehl PLS2(887) kann während der Positionierung ausgeführt werden, um die Zielposition (Anzahl von Impulsen), die Beschleunigungsrate, die Verzögerungsrate und die Zielfrequenz zu ändern. **Hinweis** Wenn nach Änderung der Einstellungen keine konstante Drehzahl beibehalten werden kann, tritt ein Fehler auf und der ursprüngliche Betrieb wird zur ursprünglichen Zielposition fortgesetzt.	PULS(886) ↓ ACC(888) (einmalig) ↓ PLS2(887)
		Änderung der Beschleunigungs- und Verzögerungsraten während der Positionierung (Mehrfachstartfunktion)		Der Befehl PLS2(887) kann während der Positionierung ausgeführt werden (Beschleunigung oder Verzögerung), um die Beschleunigungsrate oder die Verzögerungsrate zu ändern.	PLS2(887) ↓ PLS2(887) PULS(886) ↓ ACC(888) (einmalig) ↓ PLS2(887)
	Richtungsänderung	Änderung der Richtung während der Positionierung		Der Befehl PLS2(887) kann während der Positionierung mittels absoluter Spezifikation der Impulse ausgeführt werden, um die absolute Spezifikation der Impulse zu ändern und die Richtung umzukehren.	PLS2(887) ↓ PLS2(887) PULS(886) ↓ ACC(888) (einmalig) ↓ PLS2(887)
Stoppen der Impulsausgabe	Anhalten der Impulsausgabe (Einstellung der Impulsanzahl bleibt nicht erhalten.)	Unmittelbarer Stopp		Stoppt die Impulsausgabe unmittelbar und löscht die Anzahl der Ausgabeimpulse.	PLS2(887) ↓ INI(880)
	Allmählicher Stopp der Impulsausgabe. (Einstellung der Impulsanzahl bleibt nicht erhalten.)	Verzögerung bis zum Stopp		Verzögert die Impulsausgabe bis zum Stopp.	PLS2(887) ↓ ACC(888) (einmalig, Zielfrequenz 0 Hz)

Hinweis Dreieckförmige Impulsausgabe
Wenn die spezifizierte Anzahl von Impulsen geringer ist als die erforderliche Anzahl zum Erreichen der Zielfrequenz und Rückkehr auf Null, reduziert die Funktion automatisch die Beschleunigungs-/Verzögerungszeit und führt eine dreieckförmige Impulsausgabe aus (nur Beschleunigung und Verzögerung). Es tritt kein Fehler auf.



■ Wechsel vom fortlaufenden Modus (Drehzahlsteuerung) zum einmaligen Modus (Positionierung)

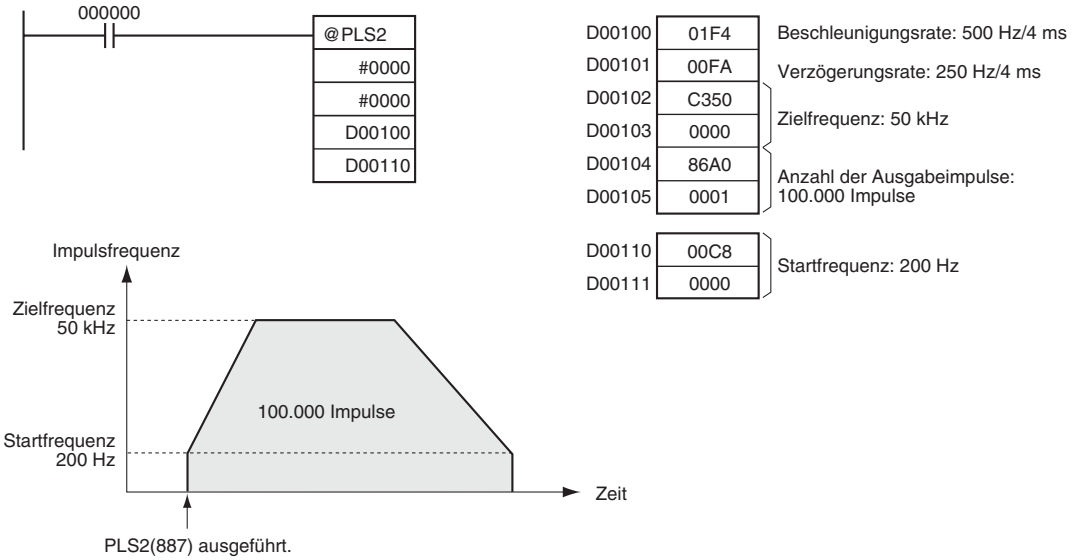
Beispielanwendung	Frequenzänderungen	Beschreibung	Verfahren/ Befehl
Wechsel von der Drehzahlsteuerung zur Positionierung bei laufendem Betrieb		Der Befehl PLS2(887) kann während einer durch einen ACC(888)-Befehl gestarteten Drehzahlsteuerung ausgeführt werden, um zur Positionierungs-funktion zu wechseln.	ACC(888) (fortlaufend) ↓ PLS2(887)
Interrupt zum Positionieren auf eine fixe Distanz			

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der spezifizierte Bereich für P, M, S oder F überschritten wird. EIN, wenn PLS2(887) für eine Schnittstelle ausgeführt wird, über die bereits Impulse durch SPED(885) oder ORG(889) ausgegeben werden. EIN, wenn bei Ausführung von PLS2(887) in einem Interrupt-Task bereits eine Impulsausgabe in einem zyklischen Task ausgeführt wird. EIN, wenn PLS2(887) für eine absolute Impulsausgabe ausgeführt wird, der Nullpunkt aber noch nicht bestimmt wurde.

Beispiel

Wenn CIO 000000 im folgenden Programmierbeispiel auf EIN gesetzt wird, startet PLS2(887) die Impulsausgabe über Impulsausgang 0 mit einer absoluten Spezifikation der Impulse von 100.000 Impulsen. Die Impulsausgabe wird beginnend bei 200 Hz alle 4 ms mit einer Rate von 500 Hz beschleunigt, bis die Zieldrehzahl von 50 kHz erreicht ist. Ab dem Verzögerungspunkt wird die Impulsausgabe alle 4 ms mit einer Rate von 250 Hz verzögert, bis die Drehzahl gleich der Anfangsfrequenz von 200 Hz erreicht ist. An diesem Punkt wird die Ausgabe gestoppt.



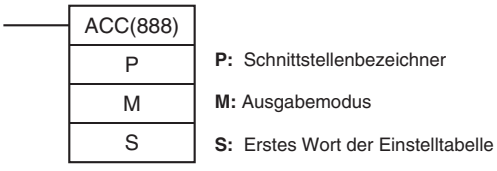
3-21-8 ACCELERATION CONTROL: ACC(888) (nur CJ1M-CPU21/22/23)

Zweck

ACC(888) gibt Impulse mit der spezifizierten Frequenz über die spezifizierte Schnittstelle aus, wobei die spezifizierte Beschleunigungs- und Verzögerungsrate verwendet wird. (Beschleunigungsrate und Verzögerungsrate sind identisch.) Es ist eine Positionierung im einmaligen Modus oder eine Drehzahlsteuerung im fortlaufenden Modus möglich. Zur Positionierung wird ACC(888) in Verbindung mit PULS(886) verwendet. ACC(888) kann auch während einer Impulsausgabe ausgeführt werden, um die Zielfrequenz oder Beschleunigungs-/Verzögerungsrate zu ändern und dadurch sanfte Drehzahländerungen zu bewirken.

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ACC(888)
	Ausführung einmal bei steigender Flanke	@ACC(888)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbare Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

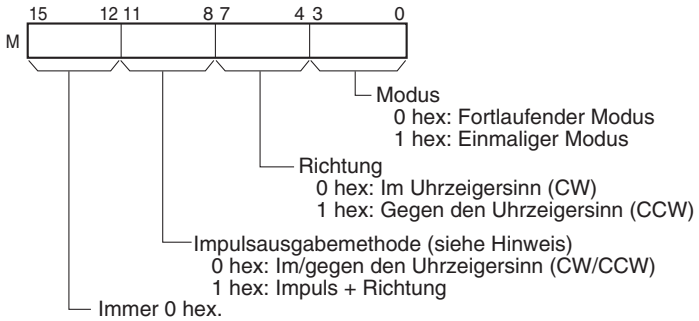
P: Schnittstellenbezeichner

Der Schnittstellenbezeichner spezifiziert die Schnittstelle, über die die Impulse ausgegeben werden.

P	Schnittstelle
0000 hex	Impulsausgang 0
0001 hex	Impulsausgang 1

M: Ausgabemodus

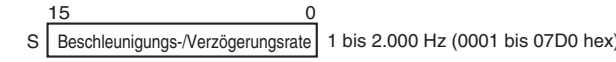
Der Inhalt von M spezifiziert die Parameter für die Impulsausgabe wie folgt:



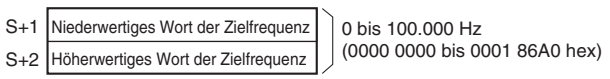
Hinweis: Verwenden Sie dieselbe Impulsausgabemethode, wenn Sie beide Impulsausgänge 0 und 1 verwenden.

S: Erstes Wort der Einstelltabelle

Der Inhalt von S bis S+2 steuert die Impulsausgabe wie in den folgenden Abbildungen dargestellt.



Spezifikation der Erhöhung bzw. Verringerung der Frequenz pro Impulssteuerungsperiode (4 ms).



Spezifikation der Frequenz in Hz nach Abschluss der Beschleunigung.

Operanden-Spezifikationen

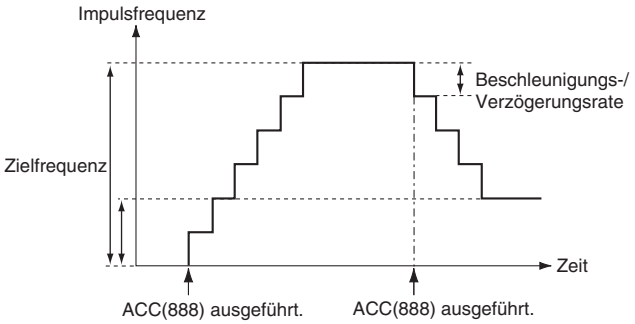
Bereich	P	M	S
CIO-Bereich	---	---	CIO 0000 bis CIO 6141
Arbeitsbereich	---	---	W000 bis W509
Haftmerkerbereich	---	---	H000 bis H509
Zusatz-Systembereich	---	---	A448 bis A957
Zeitgeberbereich	---	---	T0000 bis T4093
Zählerbereich	---	---	C0000 bis C4093
DM-Bereich	---	---	D00000 bis D32765
EM-Bereich ohne Bankangabe	---	---	---
EM-Bereich mit Bankangabe	---	---	---

Bereich	P	M	S
Indirekte DM/EM-Binäradresse	---	---	@ D00000 bis @ D32767
Indirekte DM/EM-Adresse in BCD	---	---	*D00000 bis *D32767
Konstanten	Siehe Beschreibung des Operanden.	Siehe Beschreibung des Operanden.	---
Datenregister	---	---	---
Indexregister	---	---	---
Indirektes Adressieren mittels Index-Registern	---	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

ACC(888) startet die Impulsausgabe über die in P spezifizierte Schnittstelle und verwendet den in M spezifizierten Modus und die in S spezifizierte Zielfrequenz sowie Beschleunigungs- und Verzögerungsrate. Die Frequenz wird in jeder Impulssteuerungsperiode (4 ms) um die in S spezifizierte Beschleunigungsrate erhöht, bis die in S spezifizierte Zielfrequenz erreicht ist.

Die Impulsausgabe wird bei jeder Ausführung von ACC(888) gestartet. Es ist daher normalerweise ausreichend, die Befehlsvariante mit Flankenauswertung (@ACC(888)) oder eine Ausführungsbedingung zu verwenden, die nur für einen Zyklus auf EIN gesetzt wird.



Im einmaligen Modus wird die Impulsausgabe automatisch gestoppt, wenn die spezifizierte Anzahl von Impulsen ausgegeben wurde. Im fortlaufenden Modus wird die Impulsausgabe so lange fortgesetzt, bis sie durch das Programm gestoppt wird.

Wenn der Versuch unternommen wird, während der Impulsausgabe zwischen dem einmaligem und fortlaufendem Modus zu wechseln, tritt ein Fehler auf.

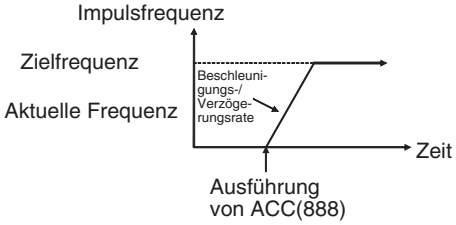
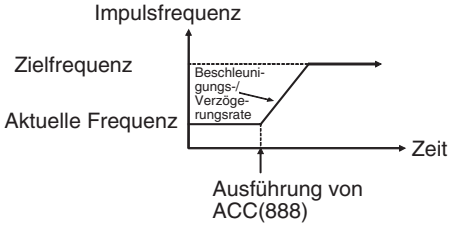
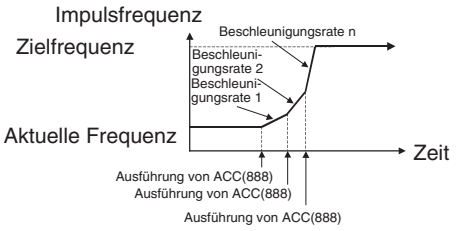
Bei den CJ1M CPU-Baugruppen kann PLS2(887) während der Impulsausgabe durch ACC(888) im einmaligen oder im fortlaufenden Modus sowie während der Beschleunigung, bei konstanter Frequenz oder Verzögerung ausgeführt werden. (Siehe Hinweis.) ACC(888) kann auch während der Impulsausgabe durch PLS2(887) während der Beschleunigung, bei konstanter Frequenz oder Verzögerung ausgeführt werden.

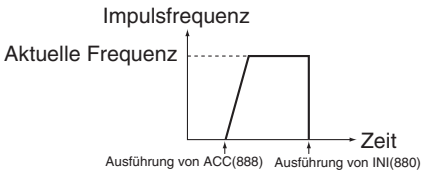
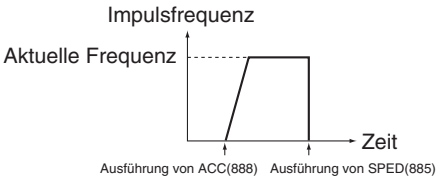
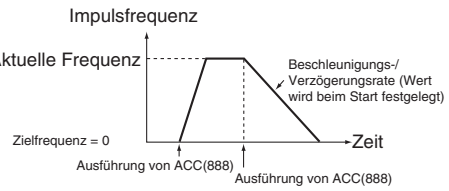
Hinweis Die Ausführung von PLS2(887) kann während der Drehzahlsteuerung mit ACC(888) (fortlaufender Modus) mit der selben Zielfrequenz wie ACC(888) verwendet werden, um einen Interrupt-Vorschub um eine feste Strecke zu erzielen. Bei dieser Anwendung wird durch PLS2(887) keine Beschleunigung ausgeführt. Ist die Beschleunigungsrate jedoch auf 0 gesetzt, wird der Fehler-Merker auf EIN gesetzt und PLS2(887) wird nicht ausgeführt. Stellen Sie die Beschleunigungsrate stets auf einen anderen Wert als 0 ein.

■ Fortlaufender Modus zur Drehzahlsteuerung

Die Impulsausgabe wird so lange fortgesetzt, bis sie durch das Programm gestoppt wird.

Hinweis Die Impulsausgabe wird sofort gestoppt, wenn die CPU-Baugruppe in die PROGRAM-Betriebsart umgeschaltet wird.

Funktion	Zweck	Anwendung	Frequenzänderungen	Beschreibung	Verfahren/ Befehl
Starten der Impulsausgabe	Ausgabe mit spezifizierter Beschleunigung und Frequenz	Erhöhung der Drehzahl (Frequenz) mit einer festen Rate		Ausgabe von Impulsen und Änderung der Frequenz um eine feste Rate.	ACC(888) (fortlaufend)
Ändern von Einstellungen	Allmähliche Drehzahländerung	Allmähliche Änderung der Drehzahl bei laufendem Betrieb		Ändert die Frequenz von der aktuellen Frequenz ausgehend um eine feste Rate. Die Frequenz kann erhöht oder verringert werden.	ACC(888) oder SPED(885) (fortlaufend) ↓ ACC(888) (fortlaufend)
		Drehzahländerung in Form einer Polygonlinie bei laufendem Betrieb.		Ändert die Beschleunigungs- oder Verzögerungsrate bei laufender Beschleunigung oder Verzögerung.	ACC(888) (fortlaufend) ↓ ACC(888) (fortlaufend)

Funktion	Zweck	Anwendung	Frequenzänderungen	Beschreibung	Verfahren/ Befehl
Stoppen der Impuls-ausgabe	Stoppen der Impuls-ausgabe	Unmittelbarer Stopp		Stopp die Impulsausgabe sofort.	ACC(888) (fortlaufend) ↓ INI(880) (Fort-laufend)
	Stoppen der Impuls-ausgabe	Unmittelbarer Stopp		Stopp die Impulsausgabe sofort.	ACC(888) (fortlaufend) ↓ SPED(885) (fortlaufend, Zielfrequenz 0 Hz)
	Impuls-ausgabe allmählich stoppen	Verzögerung bis zum Stopp		Verzögerte Impulsausgabe bis zum Stopp. **Hinweis** Wenn der Vorgang über ACC(888) gestartet wurde, bleibt die ursprüngliche Beschleunigungs-/Verzögerungsrate wirksam. Wenn der Vorgang über SPED(885) gestartet wurde, wird die Beschleunigungs-/Verzögerungsrate ungültig, und die Impulsausgabe stoppt unmittelbar.	ACC(888) (fortlaufend) ↓ ACC(888) (fortlaufend, Zielfrequenz 0 Hz)

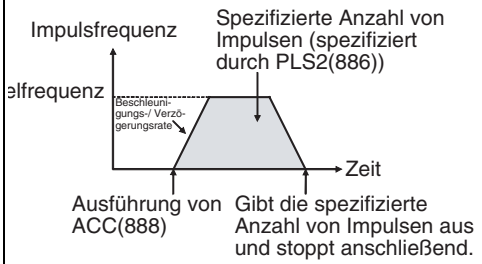
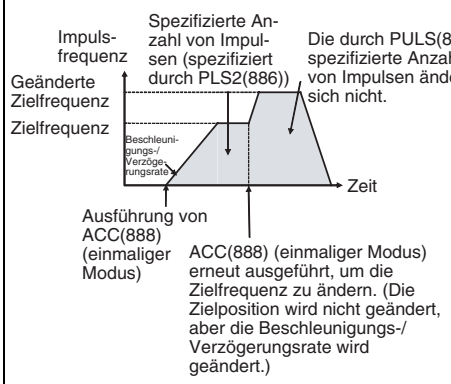
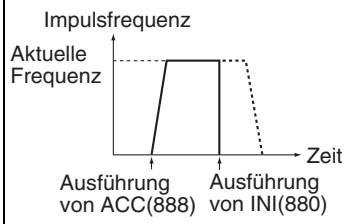
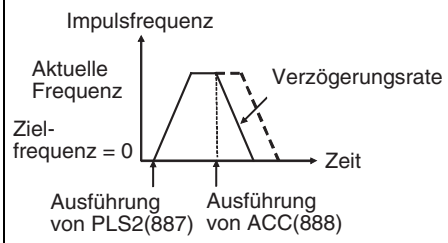
■ Einmaliger Modus zur Positionierung

Wenn die Funktion im einmaligen Modus gestartet wird, wird die Impulsausgabe so lange fortgesetzt, bis die spezifizierte Anzahl von Impulsen ausgegeben wurde.

Der Verzögerungspunkt wird aus der Anzahl der Ausgabeimpulse und aus der in S eingestellten Verzögerungsrate berechnet. Wenn dieser Punkt erreicht ist, wird die Frequenz in jeder Impulssteuerperiode (4 ms) um die in S spezifizierte Verzögerungsrate verringert, bis die spezifizierte Anzahl von Impulsen ausgegeben ist.

Hinweis

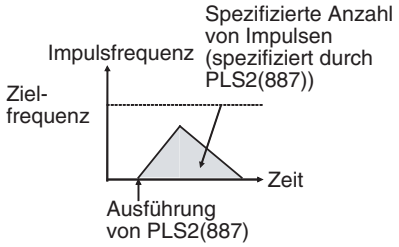
1. Die Impulsausgabe wird sofort gestoppt, wenn die CPU-Baugruppe in die PROGRAM-Betriebsart umgeschaltet wird.
2. Die Anzahl der Ausgabeimpulse muss bei jedem Neustart der Ausgabe eingestellt werden.
3. Die Anzahl der Impulse muss im Vorfeld durch den Befehl PULS(886) eingestellt werden. Wenn nicht zuerst PULS(886) ausgeführt wird, werden durch ACC(888) keine Impulse ausgegeben.
4. Die im Operanden von ACC(888) festgelegte Richtung wird ignoriert, wenn die Anzahl der Impulse mit PULS(886) als Absolutwert eingestellt wird.

Funktion	Zweck	Anwendung	Frequenzänderungen	Beschreibung	Verfahren/ Befehl
Starten der Impuls-ausgabe	Einfache trapez-förmige Impuls-ausgabe	Positionierung mit trapezförmiger Beschleunigung und Verzögerung (identische Rate für Beschleuni-gung und Verzö-gerung, keine Anfangsdrehzahl) Die Anzahl von Impulsen kann während der Positionierung nicht verändert werden.		Beschleunigt und verzögert mit derselben festen Rate, und stoppt unmittelbar nach der Ausgabe der festgelegten Anzahl von Impulsen. (Siehe Hinweis.) **Hinweis** Die Zielposi-tion (spezifi-zierte Impuls-anzahl) kann während der Positionierung nicht verän-dert werden.	PULS(886) ↓ ACC(888) (einmalig)
Ändern von Ein-stellungen	Allmähliche Drehzahl-änderung (mit identischer Beschleu-nigungs- und Verzöge-rungsrate)	Änderung der Zieldrehzahl (Frequenz) während der Positionierung (Beschleunigung srate = Verzö-gerungsrate)		Der Befehl ACC(888) kann während der Positionierung ausgeführt werden, um die Beschleunigungs-/ Verzögerungsrate und die Zielfrequenz zu ändern. Die Zielposition (spezifizierte Impulsanzahl) kann nicht geändert werden.	PULS(886) ↓ ACC(888) oder SPED(885) (einmalig) ↓ ACC(888) (einmalig)
Stoppen der Impuls-ausgabe	Stoppen der Impuls-ausgabe (Einstellung der Impuls-anzahl bleibt nicht erhalten.)	Unmittelbarer Stopp		Die Impulsausgabe wird sofort gestoppt und die verbleibende Anzahl der Ausgabeimpulse wird gelöscht.	PULS(886) ↓ ACC(888) (einmalig) ↓ INI(880)
	Impuls-ausgabe allmählich stoppen (Einstellung der Impuls-anzahl bleibt nicht erhalten.)	Verzögerung bis zum Stopp		Verzögert die Impuls-ausgabe bis zum Stopp. **Hinweis** Wenn der Vorgang über ACC(888) ge-startet wurde, bleibt die ur-sprüngliche Beschleuni-gungs-/Verzöge-rungsrate wirksam. Wenn der Vor-gang über SPED(885) gestartet wur-de, wird die Beschleuni-gungs-/Verzö-gerungsrate ungültig, und die Impulsaus-gabe stoppt unmittelbar.	PULS(886) ↓ ACC(888) oder SPED(885) (einmalig) ↓ ACC(888) (einmalig, Zielfrequenz 0 Hz) PLS2(887) ↓ ACC(888) (einmalig, Zielfrequenz 0 Hz)

Hinweis Dreieckförmige Impulsausgabe

Wenn die spezifizierte Anzahl von Impulsen geringer ist als die erforderliche Anzahl zum Erreichen der Zielfrequenz und Rückkehr auf Null, reduziert die Funktion automatisch die Beschleunigungs-/Verzögerungszeit und führt eine

dreieckförmige Impulsausgabe aus (nur Beschleunigung und Verzögerung). Es tritt kein Fehler auf.

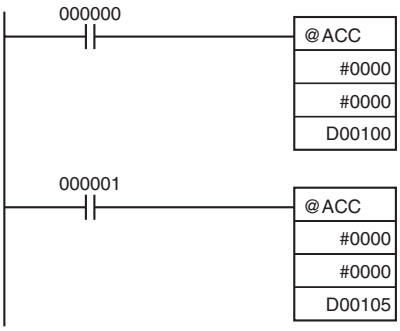


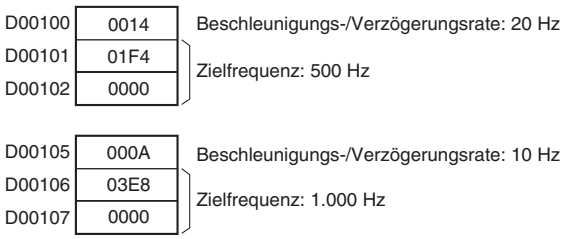
Merker

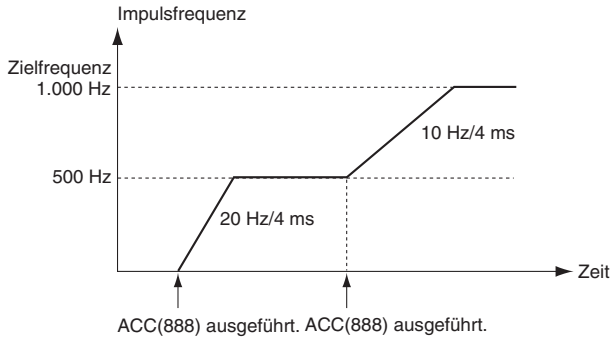
Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der spezifizierte Bereich für P, M oder S überschritten wird. EIN, wenn Impulse durch ORG(889) über die spezifizierte Schnittstelle ausgegeben werden. EIN, wenn ACC(888) zum Wechseln zwischen einmaligem und fortlaufendem Modus für eine Schnittstelle ausgeführt wird, über die bereits Impulse durch SPED(885), ACC(888) oder PLS2(887) ausgegeben werden. EIN, wenn bei der Ausführung von ACC(888) in einer Interrupt-Task bereits eine Impulsausgabe in einer zyklischen Task ausgeführt wird. EIN, wenn ACC(888) im einmaligen Modus für eine absolute Impulsausgabe ausgeführt wird, der Nullpunkt aber noch nicht bestimmt wurde.

Beispiel

Wenn CIO 000000 im folgenden Programmierbeispiel auf EIN gesetzt wird, startet ACC(888) die Impulsausgabe über Impulsausgang 0 im fortlaufenden Modus bei Verwendung der CW/CCW-Methode in Richtung des Uhrzeigersinns (CW). Die Impulsausgabe wird alle 4 ms mit einer Rate von 20 Hz beschleunigt, bis die Zielfrequenz von 500 Hz erreicht ist. Wenn CIO 000001 auf EIN gesetzt wird, wechselt ACC(888) auf eine Beschleunigungsrate von 10 Hz alle 4 ms, bis die Zielfrequenz von 1.000 Hz erreicht ist.







3-21-9 ORIGIN SEARCH: ORG(889) (nur CJ1M-CPU21/22/23)

Zweck

ORG(889) führt eine Nullpunktsuche oder Nullpunktrückkehr aus.

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 unterstützt.

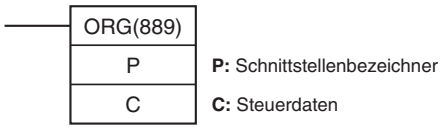
■ Nullpunktsuche

Impulse werden mit Hilfe der spezifizierten Methode ausgegeben, um letztendlich den Motor zu betreiben und den Nullpunkt auf Grundlage des Nullpunktnäherungs- und Nullpunkt-Eingangssignals zu bestimmen.

■ Rückkehr zum Nullpunkt

Das Positionierungssystem wird auf den zuvor bestimmten Nullpunkt zurückgeführt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ORG(889)
	Ausführung einmal bei steigender Flanke	@ORG(889)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

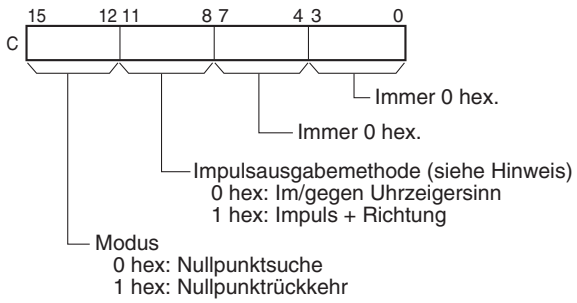
P: Schnittstellenbezeichner

Der Schnittstellenbezeichner spezifiziert die Schnittstelle, über die die Impulse ausgegeben werden.

P	Schnittstelle
0000 hex	Impulsausgang 0
0001 hex	Impulsausgang 1

C: Steuerdaten

Der Wert von C bestimmt die Methode der Nullpunktsuche.



Hinweis: Verwenden Sie dieselbe Impulsausgabemethode, wenn Sie beide Impulsausgänge 0 und 1 verwenden.

Operanden-Spezifikationen

Bereich	P	C
CIO-Bereich	---	---
Arbeitsbereich	---	---
Haftmerkerbereich	---	---
Zusatz-Systembereich	---	---
Zeitgeberbereich	---	---

Bereich	P	C
Zählerbereich	---	---
DM-Bereich	---	---
EM-Bereich ohne Bank	---	---
EM-Bereich mit Bank	---	---
Indirekte DM/EM-Binäradresse	---	---
Indirekte DM/EM-Adresse in BCD	---	---
Konstanten	Siehe Beschreibung des Operanden.	Siehe Beschreibung des Operanden.
Datenregister	---	---
Indexregister	---	---
Indirektes Adressieren mittels Index-Registern	---	---

Beschreibung

Der Befehl ORG(889) führt eine Nullpunktsuche oder Nullpunktrückkehr über die in P spezifizierte Schnittstelle unter Verwendung der in C spezifizierten Methode durch.

Die folgenden Parameter müssen im SPS-Setup eingestellt sein, bevor ORG(889) ausgeführt werden kann. Detaillierte Informationen hierzu finden Sie im *CJ-Serie Integrierte E/A Bedienerhandbuch*.

Nullpunktsuche	Nullpunktrückkehr
Nullpunktsuchfunktion aktivieren/ deaktivieren	Anfangsdrehzahl der Nullpunktsuche/ Nullpunktrückkehr
Nullpunktsuche-Betriebsmodus	Zieldrehzahl der Nullpunkt-Rückkehr
Suchvorgang der Nullpunktsuche	Beschleunigungsrate der Nullpunktrückkehr
Nullpunkterkennungsmethode	Verzögerungsrate der Nullpunktrückkehr
Suchrichtung der Nullpunktsuche	
Anfangsdrehzahl der Nullpunktsuche/ Nullpunktrückkehr	
Hohe Drehzahl der Nullpunktsuche	
Drehzahl ab Näherungsschalter bei der Nullpunktsuche	
Nullpunktkompensation	
Beschleunigungsrate der Nullpunktsuche	
Verzögerungsrate der Nullpunktsuche	
Art des Wegbegrenzungsschalter-Eingangssignals	
Art des Nullpunktnäherungs-Eingangssignals	
Art des Nullpunkt-Eingangssignals	

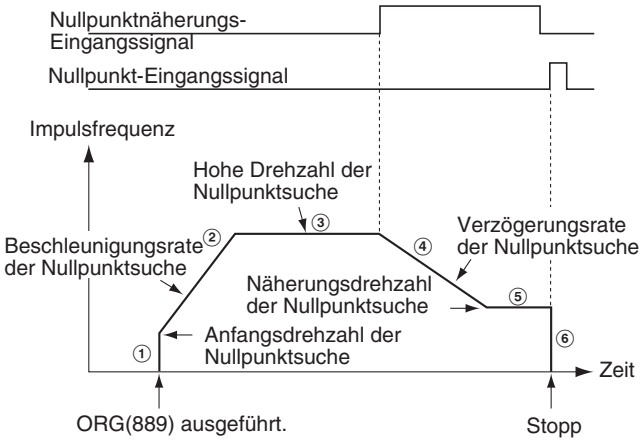
Bei jeder Ausführung von ORG(889) wird eine Nullpunktsuche bzw. Nullpunktrückkehr ausgeführt. Es ist daher normalerweise ausreichend, die Befehlsvariante mit Flankenauswertung (@ORG(889)) oder eine Ausführungsbedingung zu verwenden, die nur für einen Zyklus auf EIN gesetzt wird.

■ Nullpunktsuche (Bits 12 bis 15 von C = 0 hex)

ORG(889) startet mit der Ausgabe von Impulsen unter Verwendung der spezifizierten Methode mit der Anfangsdrehzahl der Nullpunktsuche (1 in der Abbildung). Die Impulsausgabe wird unter Verwendung der Beschleunigungsrate für die Nullpunktsuche auf die hohe Drehzahl für die Nullpunktsuche beschleunigt (2 in der Abbildung). Die Impulsausgabe wird dann mit konstanter Frequenz fortgeführt, bis das Nullpunktnäherungs-Eingangssignal auf EIN gesetzt wird (3 in der Abbildung). Ab diesem Punkt wird die Impulsausgabe unter Verwendung der Verzögerungsrate für die Nullpunktsuche auf die Näherungsdrehzahl der Nullpunktsuche verzögert (4 in der Abbildung).

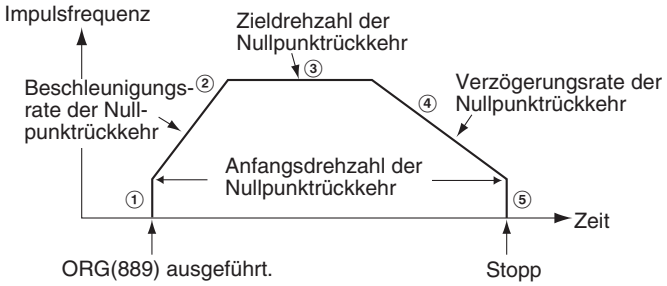
Impulse werden dann mit konstanter Frequenz ausgegeben, bis das Nullpunkt-Eingangssignal auf EIN gesetzt wird (5 in der Abbildung). Die Impulsausgabe wird gestoppt, wenn das Nullpunkt-Eingangssignal auf EIN gesetzt ist (6 in der Abbildung).

Wenn die Nullpunktsuche beendet ist, wird der Fehlerzähler-Rücksetzausgang auf EIN gesetzt. Die oben beschriebene Funktion hängt jedoch von der Betriebsart, Nullpunkterkennungsmethode und anderen Parametern ab. Detaillierte Informationen hierzu finden Sie im *CJ-Serie Integrierte E/A Bedienerhandbuch*.



■ **Nullpunktrückkehr (Bits 12 bis 15 von C = 1 hex)**

ORG(889) startet mit der Ausgabe von Impulsen unter Verwendung der spezifizierten Methode mit der Anfangsdrehzahl der Nullpunktrückkehr (1 in der Abbildung). Die Impulsausgabe wird unter Verwendung der Beschleunigungsrate für die Nullpunktrückkehr auf die Zieldrehzahl der Nullpunktrückkehr beschleunigt (2 in Abbildung) und die Impulsausgabe wird mit konstanter Frequenz fortgesetzt (3 in Abbildung). Der Verzögerungspunkt wird aus der Anzahl der bis zum Nullpunkt verbleibenden Impulse und der Verzögerungsrate berechnet. Wenn dieser Punkt erreicht ist, wird die Impulsausgabe mit der Verzögerungsrate für die Nullpunktrückkehr verzögert (4 in der Abbildung), bis die Nullpunktrückkehr-Anfangsdrehzahl erreicht ist, bei der die Impulsausgabe am Nullpunkt gestoppt wird (5 in der Abbildung).

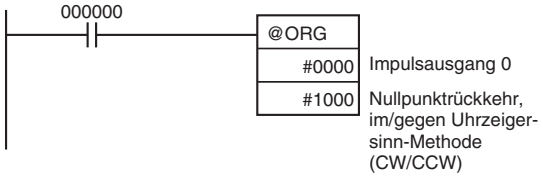


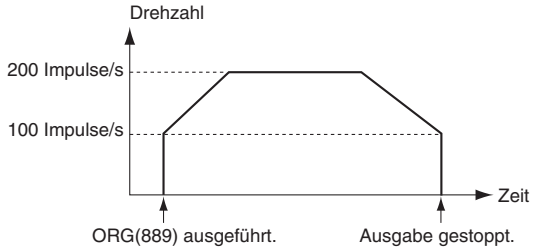
Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der spezifizierte Bereich für P oder C überschritten wird. EIN, wenn ORG(889) für eine Schnittstelle aufgeführt wird, über die bereits durch die Befehle SPED(885), ACC(888) oder PLS2(887) Impulse ausgegeben werden. EIN, wenn bei Ausführung von ORG(889) in einer Interrupt-Task bereits eine Impulsausgabe in einer zyklischen Task ausgeführt wird. EIN, wenn die im SPS-Setup eingestellten Parameter für Nullpunktsuche oder Nullpunktrückkehr nicht innerhalb des zulässigen Bereichs liegen. EIN, wenn die hohe Drehzahl der Nullpunktsuche niedriger als oder gleich der Näherungsdrehzahl der Nullpunktsuche ist oder die Näherungsdrehzahl der Nullpunktsuche niedriger als oder gleich der Anfangsdrehzahl der Nullpunktsuche ist. EIN, wenn die Zieldrehzahl der Nullpunktrückkehr niedriger als oder gleich der Anfangsdrehzahl der Nullpunktrückkehr ist. EIN, wenn eine Nullpunktrückkehr ohne vorherige Bestimmung des Nullpunkts versucht wird.

Beispiel

Wenn CIO 000000 im folgenden Programmierbeispiel auf EIN gesetzt wird, startet ORG(889) eine Nullpunktrückkehr durch Ausgabe von Impulsen über Impulsausgang 0 unter Verwendung der Methode im/gegen den Uhrzeigersinn. Entsprechend des SPS-Setup beträgt die Anfangsfrequenz 100 Impulse/s, die Zielfrequenz beträgt 200 Impulse/s und die Beschleunigungs- und Verzögerungsrate beträgt 50 Hz/4 ms.





Die Parameter im SPS-Setup haben folgende Einstellungen:

Parameter	Einstellung
Impulsausgang 0 - Anfangsdrehzahl für Nullpunktsuche und Nullpunktrückkehr	0000 0064 hex: 100 Impulse/s
Impulsausgang 0 - Zieldrehzahl der Nullpunktsuche	0000 00C8 hex: 200 Impulse/s
Impulsausgang 0 - Beschleunigungsrate für die Nullpunktrückkehr	0032 hex: 50 Hz/4 ms
Impulsausgang 0 - Verzögerungsrate für die Nullpunktrückkehr	0032 hex: 50 Hz/4 ms

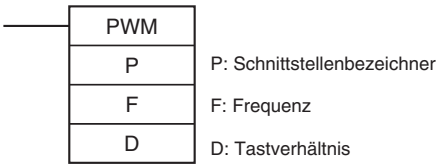
3-21-10 PULSE WITH VARIABLE DUTY FACTOR: PWM(891) (nur CJ1M-CPU21/22/23)

Zweck

PWM(891) wird zur Ausgabe von Impulsen mit spezifiziertem Tastverhältnis über die spezifizierte Schnittstelle verwendet.

Dieser Befehl wird nur von den CPU-Baugruppen CJ1M-CPU21/22/23 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	PWM(891)
	Ausführung einmal bei steigender Flanke	@PWM(891)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

P: Schnittstellenbezeichner

Der Schnittstellenbezeichner spezifiziert die Schnittstelle, über die die Impulse ausgegeben werden.

P	Schnittstelle
0000 hex	Impulsausgang 0 (Tastverhältnis: in Schritten von 1 %)
0001 hex	Impulsausgang 1 (Tastverhältnis: in Schritten von 1 %)
1000 hex (Nur CJ1M CPU-Baugruppen Version 2.0)	Impulsausgang 0 (Tastverhältnis: in Schritten von 0,1 %)
1001 hex (Nur CJ1M CPU-Baugruppen Version 2.0)	Impulsausgang 1 (Tastverhältnis: in Schritten von 0,1 %)

F: Frequenz

F spezifiziert die Frequenz der Impulsausgabe zwischen 0,1 und 6.553,5 Hz (in Einheiten von 0,1 Hz, 0001 bis FFFF hex). Die Genauigkeit der durch den Befehl PMW(891) ausgegebenen tatsächlichen Impulsform (EIN-Tastverhältnis +5 %/–0 %) gilt aufgrund von Beschränkungen in den Ausgangsschaltungen nur im Bereich zwischen 0,1 bis 1.000,0 Hz.

D: Tastverhältnis

D spezifiziert das Tastverhältnis der Impulsausgabe, d. h. den Prozentsatz der Zeit, in der der Ausgang eingeschaltet ist. Der Wert für D muss in folgendem Bereich liegen:

- CJ1M CPU-Baugruppen vor Version 2.0
  0 % bis 100 % (in Schritten von 1 %, 0000 bis 0064 hex)
- CJ1M CPU-Baugruppen ab Version 2.0
  0,0 % bis 100,0 % (in Schritten von 0,1 %, 0000 bis 03E8 hex)

Operanden-Spezifikationen

Bereich	P	F	D
CIO-Bereich	---	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6143
Arbeitsbereich	---	W000 bis W511	W000 bis W511
Haftmerkerbereich	---	H000 bis H511	H000 bis H511
Zusatz-Systembereich	---	A448 bis A959	A448 bis A959
Zeitgeberbereich	---	T0000 bis T4095	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767	D00000 bis D32767
EM-Bereich ohne Bank	---	---	---
EM-Bereich mit Bank	---	---	---
Indirekte DM/EM-Binäradresse	---	@ D00000 bis @ D32767	@ D00000 bis @ D32767
Indirekte DM/EM-Adresse in BCD	---	*D00000 bis *D32767	*D00000 bis *D32767
Konstanten	Siehe Beschreibung des Operanden.	0000 bis FFFF hex	0000 bis 0064 hex
Datenregister	---	DR0 bis DR15	DR0 bis DR15
Indexregister	---	---	---
Indirektes Adressieren mittels Index-Registern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

PWM(891) gibt die in F spezifizierte Frequenz mit dem in D spezifizierten Tastverhältnis über die in P spezifizierte Schnittstelle aus. PWM(891) kann auch während der Ausgabe von Impulsen mit Tastverhältnis ausgeführt werden, um das Tastverhältnis ohne Stoppen der Impulsausgabe zu ändern. Jegliche Versuche, die Frequenz zu ändern, werden ignoriert.

Die Impulsausgabe wird bei jeder Ausführung von PWM(891) gestartet. Es ist daher normalerweise ausreichend, die Befehlsvariante mit Flankenauswertung (@PWM(891)) oder eine Ausführungsbedingung zu verwenden, die nur für einen Zyklus auf EIN gesetzt wird.

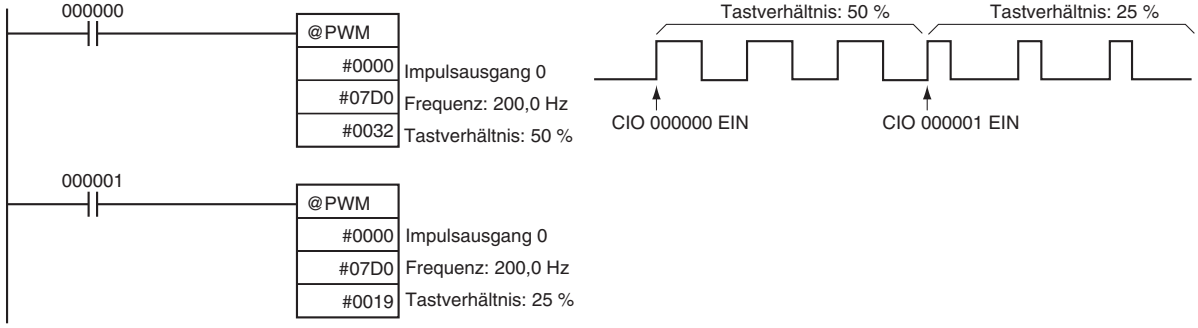
Die Impulsausgabe wird fortgesetzt, bis sie entweder durch Ausführung des Befehls INI(880) gestoppt wird (C = 0003 hex: Stopp der Impulsausgabe) oder bis die CPU-Baugruppe in die PROGRAM-Betriebsart umgeschaltet wird.

Merker

Name	Bezeich-nung	Funktion
Fehlermerker	ER	EIN, wenn der spezifizierte Bereich für P, F oder D überschritten wird. EIN, wenn Impulse durch ORG(889) über die spezifizierte Schnittstelle ausgegeben werden. EIN, wenn bei Ausführung von PWM(891) in einer Interrupt-Task bereits eine Impulsausgabe in einer zyklischen Task ausgeführt wird.

Beispiel

Wenn CIO 000000 im folgenden Programmierbeispiel auf EIN gesetzt wird, startet PWM(891) die Impulsausgabe über Impulsausgang 0 mit einer Frequenz von 200 Hz mit einem Tastverhältnis von 50 %. Wenn CIO 000001 auf EIN gesetzt wird, wird das Tastverhältnis auf 25 % geändert.



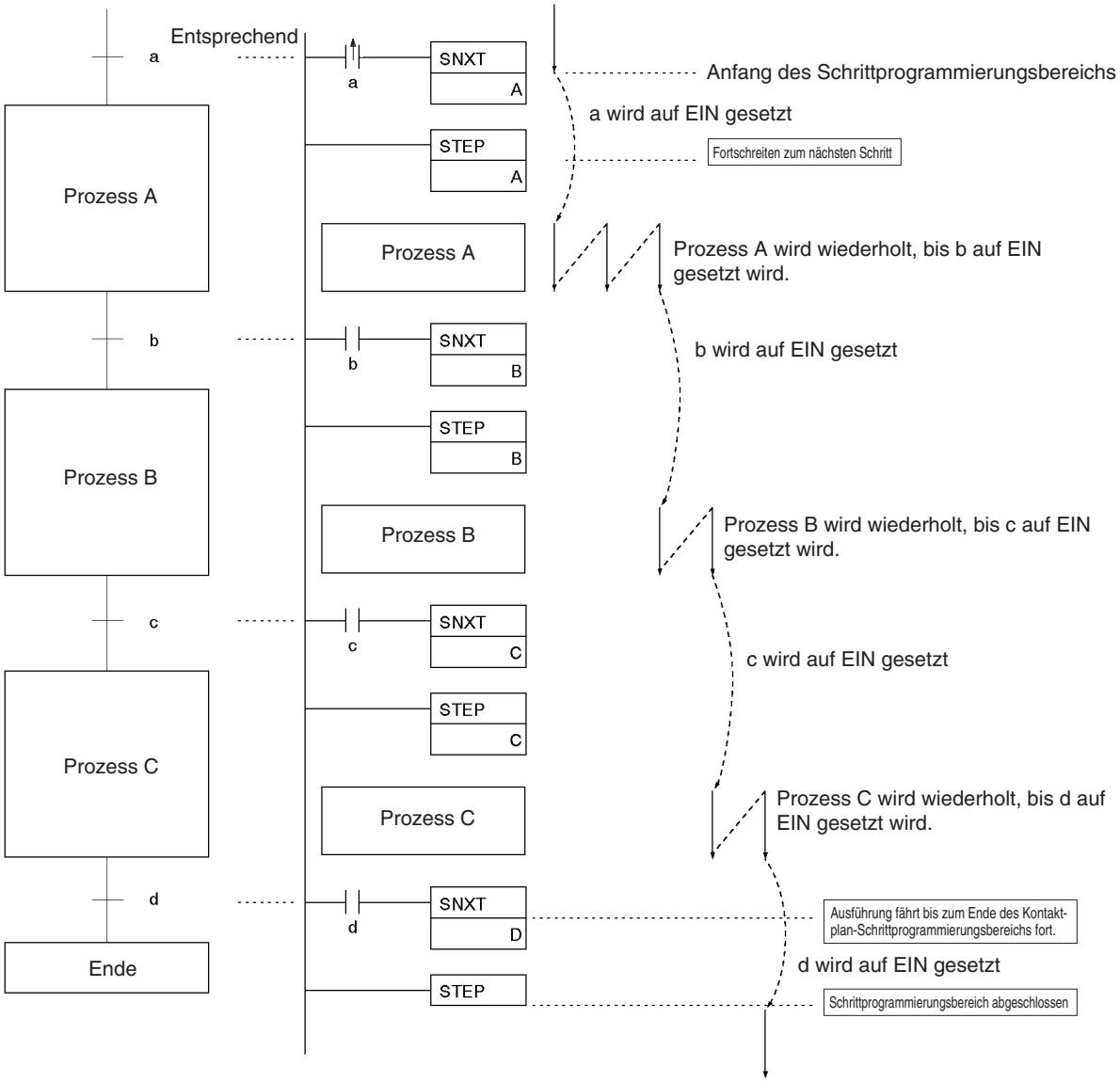
3-22 Schritt-Befehle

In diesem Abschnitt werden Schritt-Befehle beschrieben, die dazu verwendet werden können, einen Unterteilungspunkt zwischen einzelnen Abschnitten eines großen Programms einzurichten, damit diese Abschnitte als Einheiten ausgeführt und nach Abschluss der Ausführung zurückgesetzt werden können.

Befehl	AWL	Funktionscode	Seite
STEP DEFINE	STEP	008	868
STEP START	SNXT	009	868

Bei SPS der CS/CJ-Serie können die Befehle STEP(008)/SNXT(009) gemeinsam verwendet werden, um Schrittprogramme zu erstellen.

Befehl	Funktion	Diagramm
SNXT(009): STEP START	Steuert den Fortschritt zum nächsten Schritt des Programms.	Entsprechend
STEP(008): STEP DEFINE	Kennzeichnet den Anfang eines Schritts. Wiederholt das gleiche Schrittprogramm, bis die Bedingungen zum Fortschreiten zum nächsten Schritt erfüllt sind.	Entsprechend



Hinweis Als Steuerbits für A, B, C und D werden Arbeitsbits verwendet.

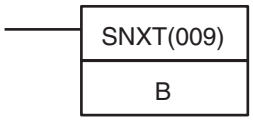
3-22-1 STEP DEFINE und STEP START: STEP(008)/SNXT(009)

Zweck

Der Befehl SNXT(009) wird unmittelbar vor dem Befehl STEP(008) platziert und steuert die Schrittausführung, indem er das spezifizierte Steuerbit auf EIN setzt. Wenn ein anderer Schritt unmittelbar vor SNXT(009) vorhanden ist, setzt der Befehl zudem das Steuerbit für diesen Prozess auf AUS.

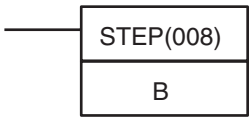
Der Befehl STEP(008) wird unmittelbar hinter dem Befehl SNXT(009) und vor dem jeweiligen Prozess platziert. Er definiert den Anfang des jeweiligen Prozesses und spezifiziert das Steuerbit für ihn. Zudem wird der Befehl auch am Ende des Schrittprogrammbereichs hinter dem letzten SNXT(009) platziert, um das Ende des Schrittprogrammbereichs zu kennzeichnen. Bei Verwendung zur Kennzeichnung des Endes eines Schrittprogrammbereichs benötigt der Befehl STEP(008) kein Steuerbit.

Kontaktplansymbole



B: Bit

Bei der Definition des Anfangs eines Schritts wird ein Steuerbit wie folgt spezifiziert:



B: Bit

Bei der Definition des Endes eines Schritts wird kein Steuerbit spezifiziert:



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	STEP(008)/ SNXT(009)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
Nicht zulässig	OK	Nicht zulässig	Nicht zulässig

Operanden-Spezifikationen

Bereich	B
CIO-Bereich	---
Arbeitsbereich	W00000 bis W51115
Haftmerkerbereich	---
Zusatz-Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradresse	---
Indirekte DM/EM-Adresse in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

SNXT(009)

SNXT(009) wird auf die folgenden drei Arten verwendet:

1,2,3...

1. Zum Start der Ausführung der Schrittprogrammierung
2. Zum Fortfahren mit dem nächsten Schritt-Steuerbit
3. Zum Beenden der Ausführung der Schrittprogrammierung,

Der Schrittprogrammbereich erstreckt sich vom ersten STEP(008)-Befehl (der stets ein Steuerbit benötigt) bis zum letzten STEP(008)-Befehl (der in keinem Fall ein Steuerbit benötigt).

Starten der Schrittausführung

Der Befehl SNXT(009) wird am Anfang des Schrittprogrammierungsbereichs platziert, um die Schrittausführung zu starten. Er setzt das für B des nächsten STEP(008)-Befehls spezifizierte Steuerbit auf EIN und schreitet mit Schritt B fort (alle Befehle nach STEP(008) B). Für den Befehl SNXT(009), der die Ausführung des Schrittprogrammbereichs startet, muss eine Ausführungsbedingung mit Flankenüberwachung verwendet werden, da die Schrittausführung andernfalls nur einen Zyklus lang erfolgt.

Fortschreiten zum nächsten Schritt

Wenn der Befehl SNXT(009) mitten im Schrittprogrammierungsbereich auftritt, wird er zum Fortschreiten zum nächsten Schritt eingesetzt. Er setzt das vorherige Steuerbit auf AUS und das nächste Steuerbit B für den nächsten Schritt auf EIN, wodurch die Ausführung von Schritt B (alle Befehle nach STEP(008) B) gestartet wird.

Beenden des Schrittprogrammierungsbereichs

Wenn der Befehl SNXT(009) ganz am Ende des Schrittprogrammbereichs platziert wird, beendet er die Schrittausführung und setzt das vorhergehende Steuerbit auf AUS. Das für B spezifizierte Steuerbit ist in diesem Fall ein Dummy-Bit. Dieses Bit wird dennoch auf EIN gesetzt, weshalb Sie darauf achten müssen, ein Bit zu wählen, durch das in diesem Fall keine Probleme verursacht werden.

STEP(008)

STEP(008) funktioniert je nach seiner Position und ob ein Steuerbit spezifiziert wurde oder nicht, auf die folgenden 2 Arten:

1,2,3... 1. Startet einen bestimmten Schritt.
2. Beendet den Schrittprogrammierungsbereich (d. h. die Schrittausführung).

Starten eines Schritts

Der Befehl STEP(008) wird am Anfang jedes Schritts platziert. Dabei wird der Befehl mit einem Operanden, B, der als Steuerbit für diesen Schritt dient, programmiert.

Das Steuerbit B wird durch den Befehl SNXT(009) auf EIN gesetzt, wodurch die Befehle innerhalb des Schritts ab dem Befehl, der unmittelbar auf STEP(008) folgt, ausgeführt werden. A20012 (Schritt-Merker) wird auch auf EIN gesetzt, wenn die Ausführung des Schritts beginnt.

Nach dem ersten Zyklus wird die Schrittausführung fortgesetzt, bis die Bedingungen für einen Schrittwechsel erfüllt sind, also bis der Befehl SNXT(009) das Steuerbit für den nächsten STEP(008)-Befehl auf EIN setzt.

Wenn der Befehl SNXT(009) das Steuerbit für einen Schritt auf EIN setzt, wird das Steuerbit B des aktuellen Befehl zurückgesetzt (auf AUS gesetzt) und der durch das Bit B gesteuerte Schritt wird gesperrt.

Die Behandlung von Ausgängen und Befehlen in einem Schritt ist vom EIN/AUS-Staus des Steuerbits B abhängig. (Der Status des Steuerbits wird durch den Befehl SNXT(009) gesteuert.) Wenn das Steuerbit auf AUS gesetzt wird, werden die Befehle in dem entsprechenden Schritt zurückgesetzt und gesperrt. Siehe folgende Tabellen.

Status des Steuerbits	Behandlung
EIN	Die Befehle im Schritt werden normal ausgeführt.
EIN→AUS	Bits und Befehle im Schritt werden gesperrt (siehe folgende Tabelle).
AUS	Alle Befehle im Schritt werden als NOP verarbeitet.

Sperrstatus

Ausgang des Befehls		Status
Für OUT, OUT NOT spezifizierte Bits		Alle AUS
Folgende Zeitgeber-Befehle: TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIML(542) und TIMLX(553)	Istwert	0000 hex (zurückgesetzt)
	Ablaufmerker	AUS (zurückgesetzt)
Bei anderen Befehlen spezifizierte Bits oder Worte (siehe Hinweis)		Vorheriger Status wird beibehalten (die Befehle werden jedoch nicht ausgeführt)

Hinweis Gilt für alle anderen Befehle, wie z. B. TTIM(087), TTIMX(555), MTIM(543), MTIMX(554), SET, RSET, CNT, CNTX(546), CNTR(012), CNTRX(548), SFT und KEEP(011).

Der Befehl STEP(008) muss am Anfang jedes Schritts platziert werden. STEP(008) wird am Anfang des Schrittprogrammbereichs platziert, um den Anfang des Schritts zu definieren.

Beenden des Schrittprogrammierungsbereichs

Der Befehl STEP(008) wird ohne Operand am Ende des Schrittprogrammbereichs platziert, um das Ende des Schrittprogramms zu definieren. Wenn das Steuerbit, das einem SNXT(009)-Befehl vorangeht, auf AUS gesetzt wird, wird die Schrittausführung durch den Befehl SNXT(009) gestoppt.

Merker: STEP(008)

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn das spezifizierte Bit B nicht im Arbeitsbereich (WR-Bereich) liegt. EIN, wenn STEP(008) in einem Interrupt-Programm verwendet wird. In allen anderen Fällen AUS.

Merker: SNXT(009)

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn das spezifizierte Bit B nicht im Arbeitsbereich (WR-Bereich) liegt. EIN, wenn SNXT(009) in einem Interrupt-Programm verwendet wird. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen Das Steuerbit B für die Befehle STEP(008)/SNXT(009) muss im Arbeitsbereich (W-Bereich) liegen.

Ein Bit, das als Steuerbit für die Befehle STEP(008)/SNXT(009) verwendet wird, kann nicht an anderen Stellen im Kontaktplan verwendet werden. Wenn das gleiche Bit zweimal verwendet wird, tritt ein Duplikationsfehler auf.

Wenn der Befehl SBS(091) dazu verwendet wird, ein Unterprogramm aus einem Schrittprogramm heraus aufzurufen, werden die Ausgänge und Befehle des Unterprogramms gesperrt, wenn das Steuerbit auf AUS gesetzt wird.

Innerhalb eines Abschnitts eines Schrittprogramms verwendete Steuerbits müssen aufeinander folgen und aus dem gleichen Wort stammen.

Der Befehl SNXT(009) wird nur einmal ausgeführt, und zwar an der steigenden Flanke der Ausführungsbedingung.

Geben Sie den Befehl SNXT(009) am Ende des Schrittprogrammbereichs ein. Achten Sie dabei darauf, dass das Steuerbit ein Dummy-Bit im Arbeitsbereich ist. wenn ein Steuerbit für einen Schritt im letzten SNXT(009)-Befehl im

Schrittprogrammbereich verwendet wird, wird der entsprechende Schritt bei Ausführung von SNXT(009) gestartet.

Wenn der für den Befehl SNXT(009) oder STEP(008) spezifizierte Operand B nicht im Arbeitsbereich (W-Bereich) liegt oder wenn das Schrittprogramm in einem anderen Programmbereich als in einer zyklischen Task platziert wurde, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

A20012 (Schritt-Merker) wird bei der Ausführung von STEP(008) einen Zyklus lang auf EIN gesetzt. Dieser Merker kann zur Steuerung einer Initialisierung bei Beginn der Ausführung des Schritts genutzt werden.

Bedingungen für die Platzierung von Schrittprogrammierungsbereichen (STEP B bis STEP)

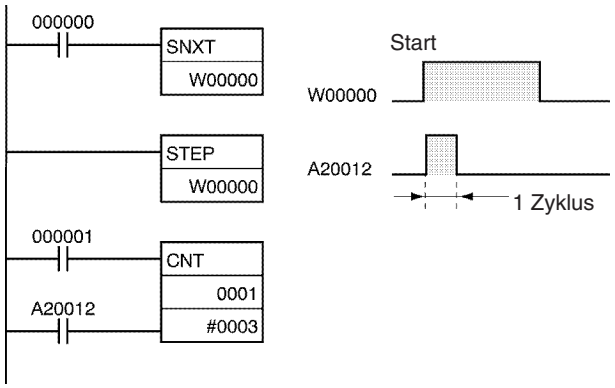
STEP(008) und SNXT(009) können nicht in Unterprogrammen, Interrupt-Programmen oder Blockprogrammen verwendet werden.

Achten Sie darauf, dass nicht zwei Schritte in gleichen Zyklus ausgeführt werden.

Befehle, die nicht innerhalb von Schrittprogrammen verwendet werden können

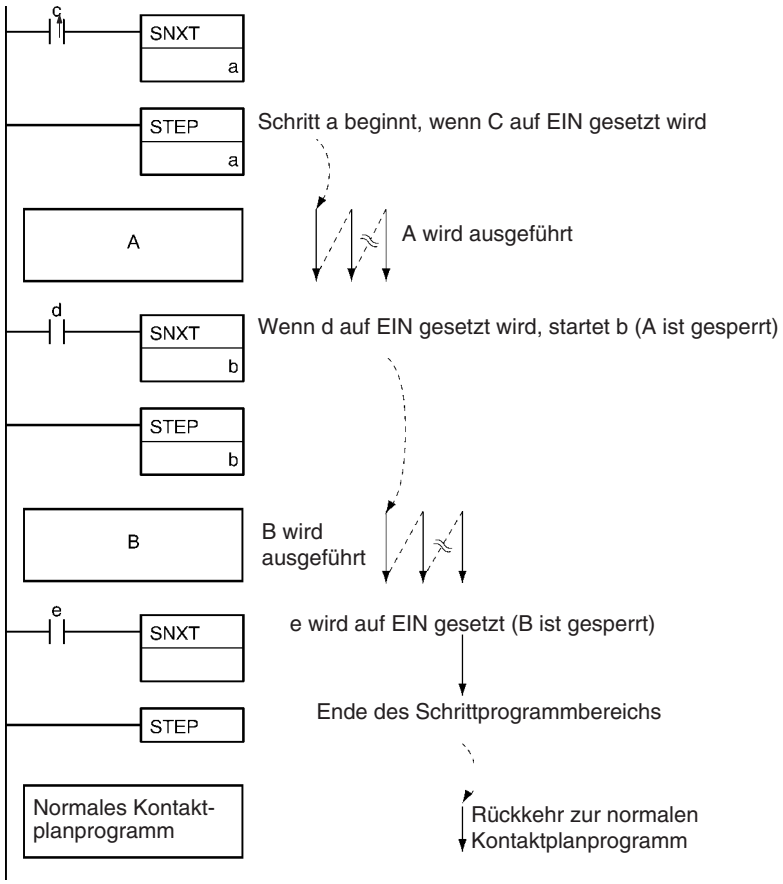
In der folgenden Tabelle sind die Befehle aufgeführt, die nicht innerhalb von Schrittprogrammen verwendet werden können.

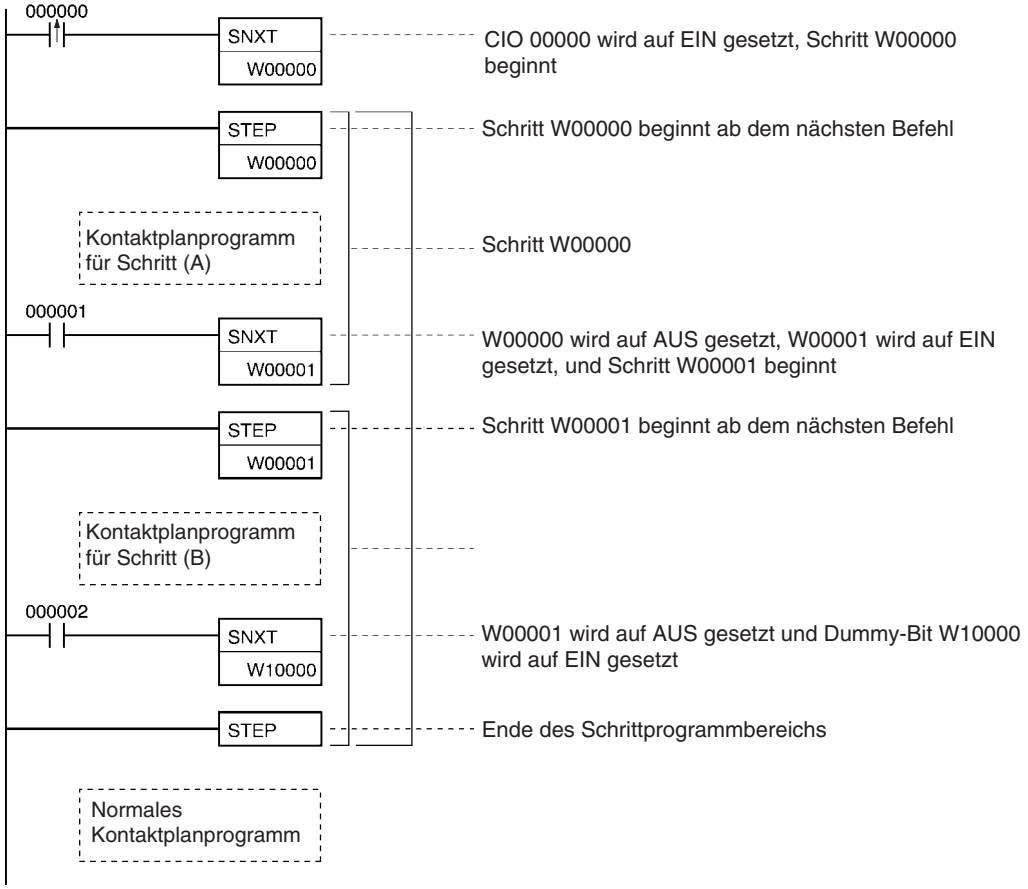
Funktion	Mnemonic	Name
Sequenzsteuerungsbefehle	END(001)	END
	IL(002)	INTERLOCK
	ILC(003)	INTERLOCK CLEAR
	JMP(004)	JUMP
	JME(005)	JUMP END
	CJP(510)	CONDITIONAL JUMP
	CJPN(511)	CONDITIONAL JUMP NOT
	JMP0(515)	MULTIPLE JUMP
	JME0(516)	MULTIPLE JUMP END
Unterprogrammbefehle	SBN(092)	SUBROUTINE ENTRY
	RET(093)	SUBROUTINE RETURN



Zugehörige Bits

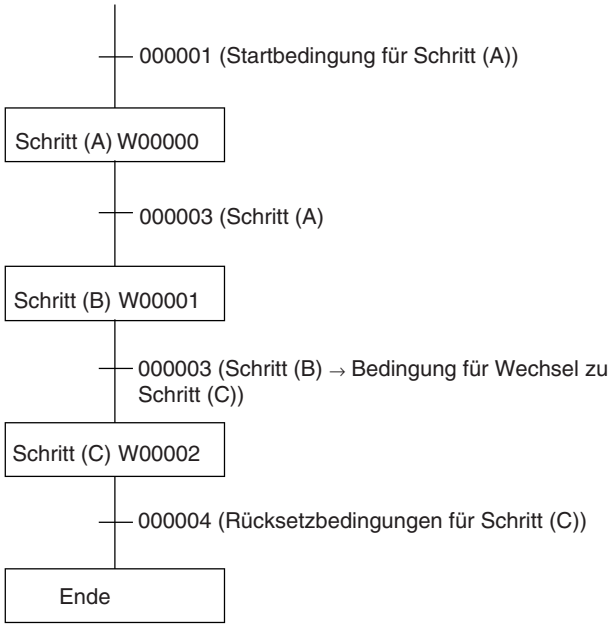
Name	Adresse	Details
Schritt-Merker	A20012	Dieser Merker ist nach Start der Schrittausführung mittels STEP(008) für einen Zyklus auf EIN gesetzt. Dieser Merker kann zur Rücksetzung von Zeitgebern oder zu anderen Verarbeitungen bei Beginn eines neuen Schritts genutzt werden.

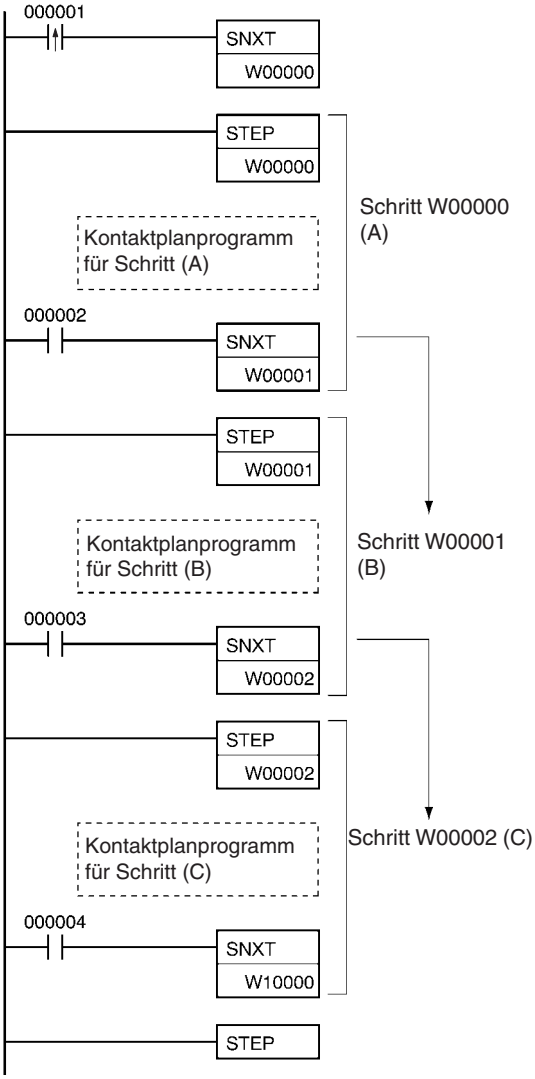

c
SNXT
a
STEP
a
Schritt a beginnt, wenn C auf EIN gesetzt wird
A
A wird ausgeführt
d
SNXT
b
Wenn d auf EIN gesetzt wird, startet b (A ist gesperrt)
STEP
b
B
B wird
ausgeführt
e
SNXT
e wird auf EIN gesetzt (B ist gesperrt)
STEP
Ende des Schrittprogrammbereichs
Normales Kontakt-
planprogramm
Rückkehr zur normalen
Kontaktplanprogramm



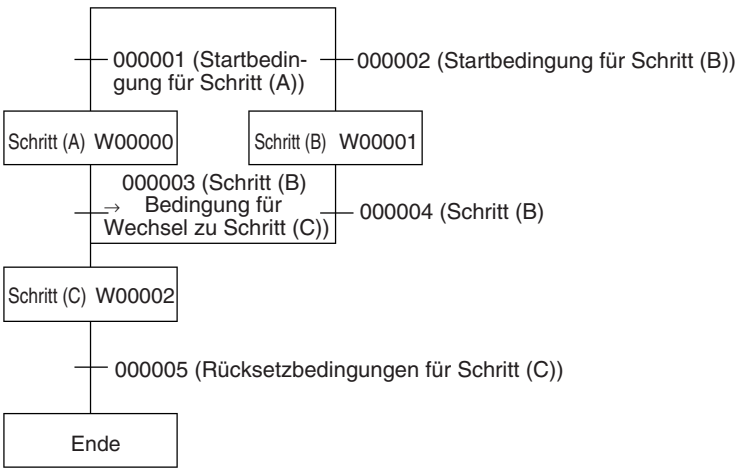
Beispiele

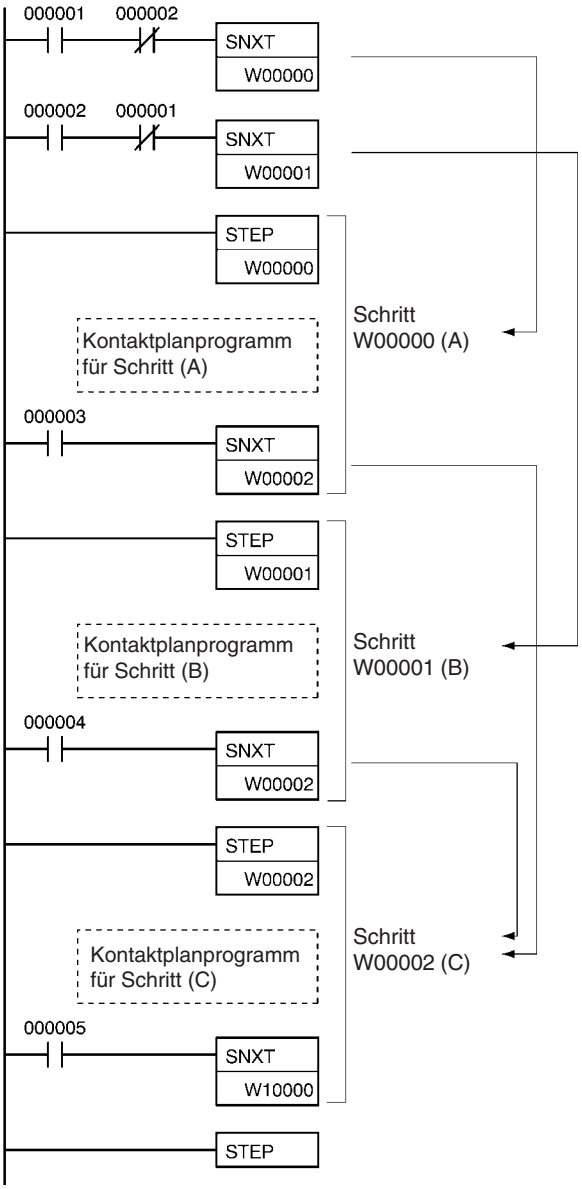
Sequenzielle Steuerung



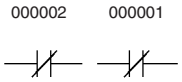


Verzweigungssteuerung



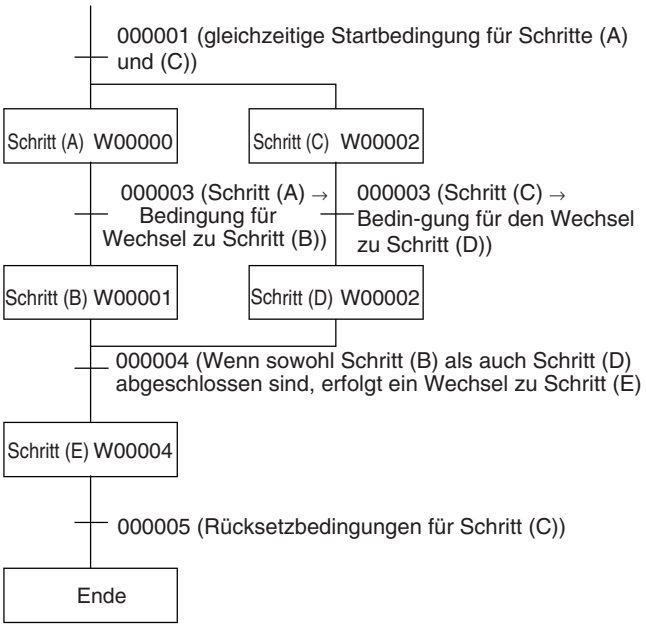


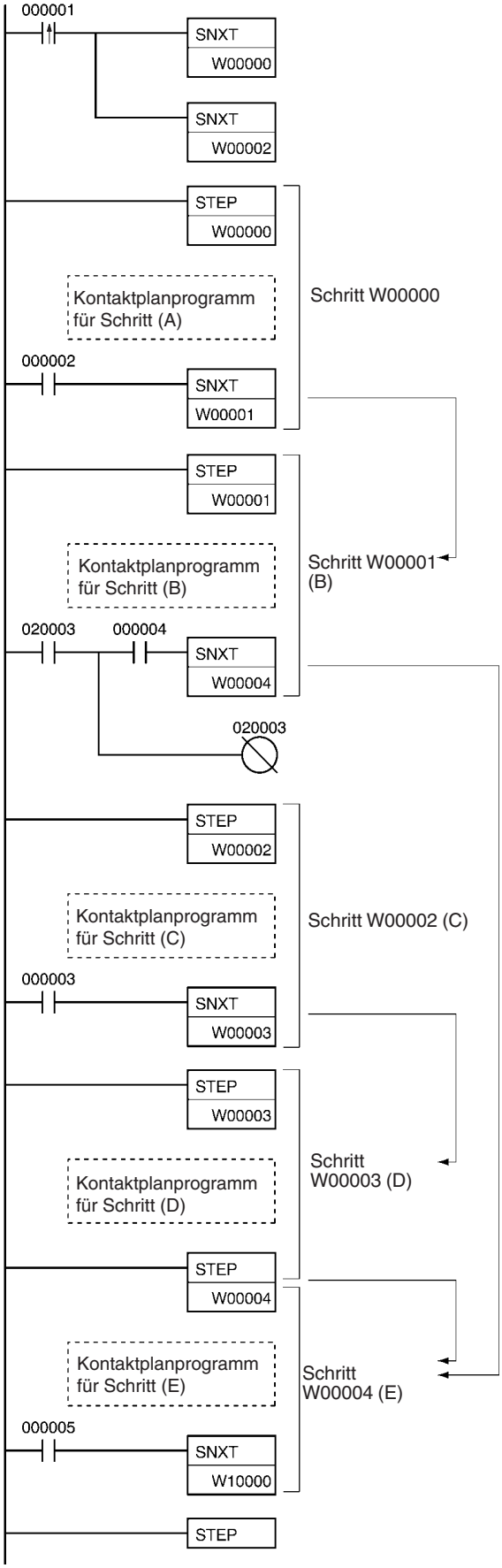
Die oben beschriebenen Programmierung wird verwendet, wenn die Schritte A und B nicht gleichzeitig ausgeführt werden können. Zur gleichzeitigen Ausführung der Schritte A und B muss die Ausführungsbedingung wie nachfolgend beschrieben gelöscht werden.



Hinweis In dem oben gezeigten Beispiel, in dem SNXT(009) für W00002 ausgeführt wird, schreitet die Verzweigung mit den nächsten Schritten fort, obwohl das gleiche Steuerbit doppelt verwendet wird. Dies wird bei der Programmprüfung mit CX-Programmer nicht als Fehler erkannt. Ein Duplikationsfehler tritt im Kontaktplanprogramm nur auf, wenn ein Steuerbit einen Schrittbefehls auch im normalen Kontaktplanprogramm verwendet wird.

Parallelsteuerung



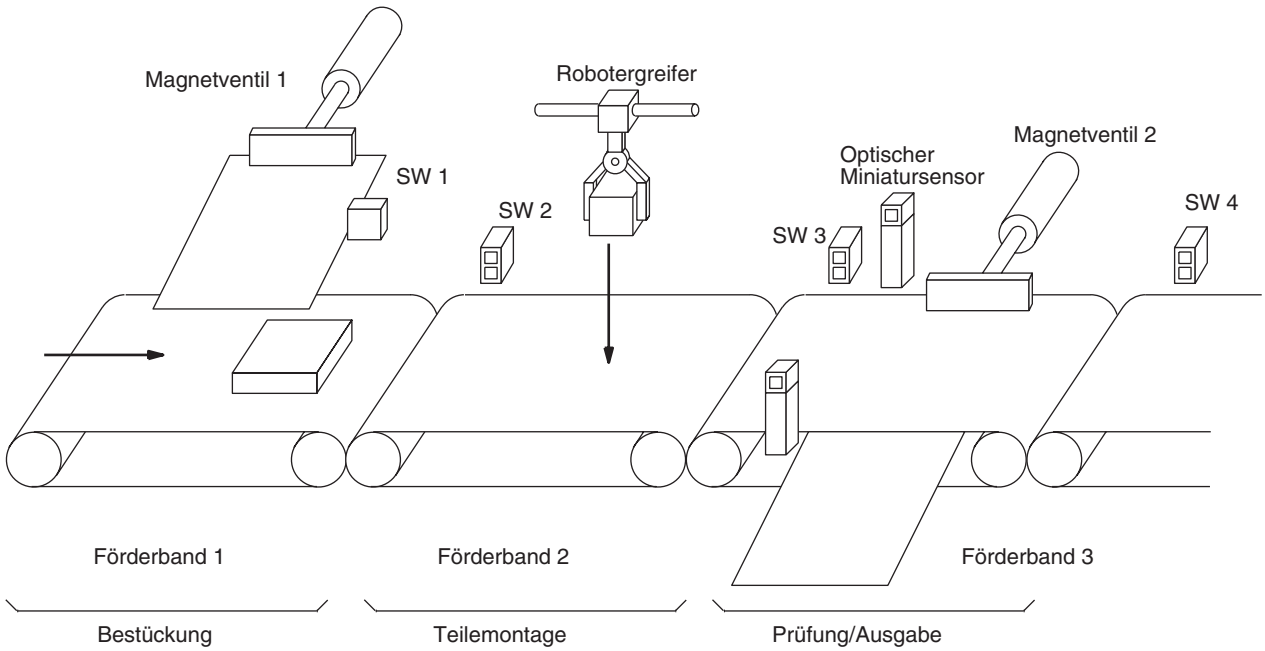

000001
SNXT
W00000
SNXT
W00002
STEP
W00000
Kontaktplanprogramm
für Schritt (A)
Schritt W00000
000002
SNXT
W00001
STEP
W00001
Kontaktplanprogramm
für Schritt (B)
Schritt W00001
(B)
020003
000004
SNXT
W00004
020003
STEP
W00002
Kontaktplanprogramm
für Schritt (C)
Schritt W00002 (C)
000003
SNXT
W00003
STEP
W00003
Kontaktplanprogramm
für Schritt (D)
Schritt
W00003 (D)
STEP
W00004
Kontaktplanprogramm
für Schritt (E)
Schritt
W00004 (E)
000005
SNXT
W10000
STEP

Anwendungsbeispiele

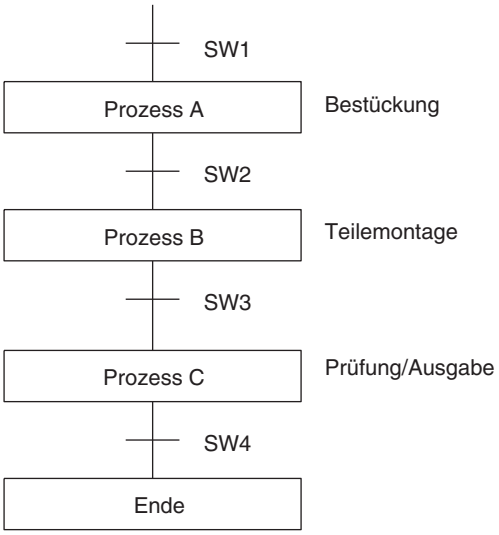
Die folgenden drei Beispiele zeigen die drei Arten der Ausführungssteuerung, die bei Schrittprogrammierung möglich sind. *Beispiel 1* zeigt die sequenzielle Ausführung, *Beispiel 2* die Verzweigungsausführung und *Beispiel 3* die Parallelausführung.

Beispiel 1: Sequenzielle Ausführung

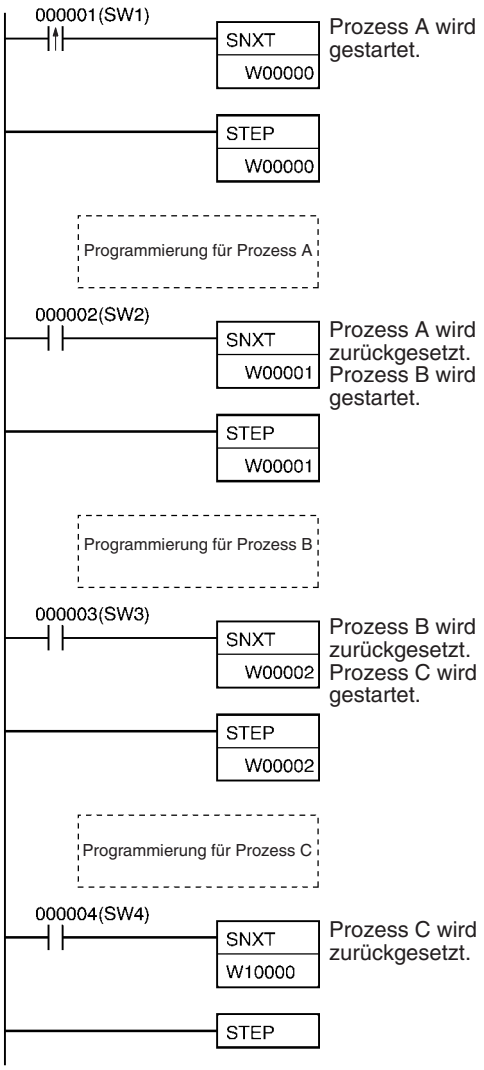
Bei dem folgenden Verfahren müssen die drei Prozesse zur Bestückung, Montage und Prüfung/Ausgabe nacheinander ausgeführt werden, wobei jeder Prozess zurückgesetzt wird, bevor der nächste beginnt. Verschiedene Sensoren (SW1, SW2, SW3 und SW4) sind so positioniert, dass sie Signale ausgeben, wenn die Prozesse gestartet bzw. beendet werden müssen.



Das folgende Diagramm zeigt den Ablauf der Verarbeitung und die Schalter, die zur Ausführungssteuerung eingesetzt werden.



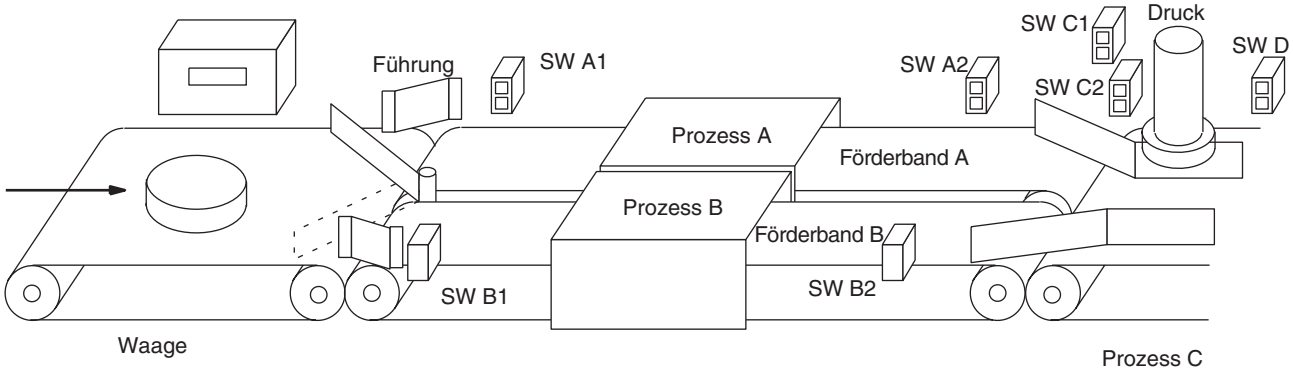
In dem nachfolgend gezeigten Programm zu Steuerung des Prozesses wird die Grundform der Schrittprogrammierung verwendet: jeder Schritt wird durch einen jeweiligen SNXT(009)-Befehl abgeschlossen, der den nächsten Schritt startet. Jeder der Schritte startet, wenn der Schalter, der den Abschluss des vorhergehenden Schritts signalisiert, auf EIN geschaltet wird.



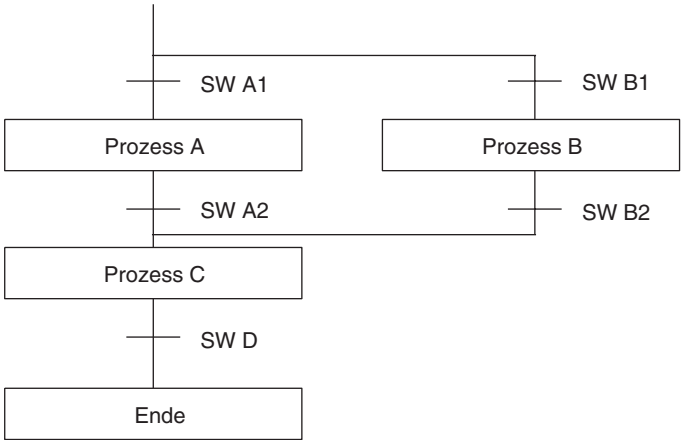
Adresse	Befehl	Operanden
000000	@LD	000001
000001	SNXT(009)	W00000
000002	STEP(008)	W00000
	Prozess A	
000100	LD	000002
000101	SNXT(009)	W00001
000102	STEP(008)	W00001
	Prozess B	
000100	LD	000003
000101	SNXT(009)	W00002
000102	STEP(008)	W00002
	Prozess C	
000200	LD	000004
000201	SNXT(009)	W00003
000202	STEP(008)	W00003

Beispiel 2: Verzweigungsausführung

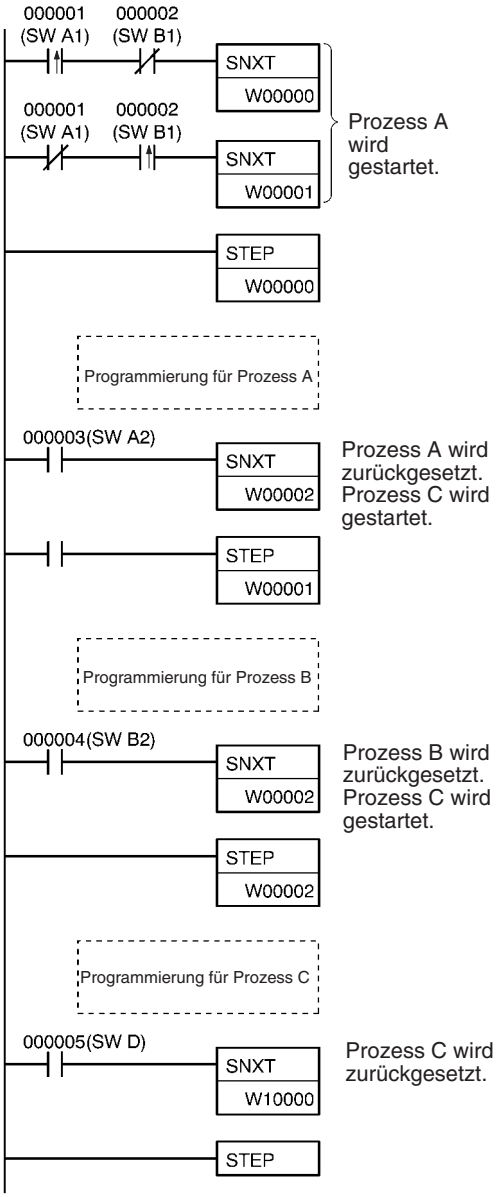
In dem folgenden Verfahren ist es erforderlich, dass ein Produkt je nach seinem Gewicht auf eine von zwei Arten verarbeitet wird, bevor es bedruckt wird. Der Druckprozess ist unabhängig davon, welcher der vorhergehenden Prozesse zum Einsatz kommt, stets der gleiche. Es kommen mehrere Sensoren zum Einsatz, die signalisieren, wenn die einzelnen Prozesse begonnen und beendet werden müssen.



Das folgende Diagramm zeigt den Ablauf der Verarbeitung und die Schalter, die zur Ausführungssteuerung eingesetzt werden. In diesem Beispiel wird je nach Status der Schalter SW A1 und SW B1 entweder Prozess A oder Prozess B verwendet.



Das nachfolgend dargestellte Programm für diesen Prozess beginnt mit zwei SNXT(009)-Befehlen, die die Prozesse A und B starten. Aufgrund der Art der Programmierung von CIO 000001 (SW A1) und CIO 000002 (SW B1) wird nur einer der beiden Prozesse mit einer EIN-Ausführungsbedingung zum Start von entweder Prozess A oder Prozess B ausgeführt. Beide der Schritte für die Prozesse enden mit dem Befehl SNXT(009), der den nächsten Schritt (Prozess C) startet.

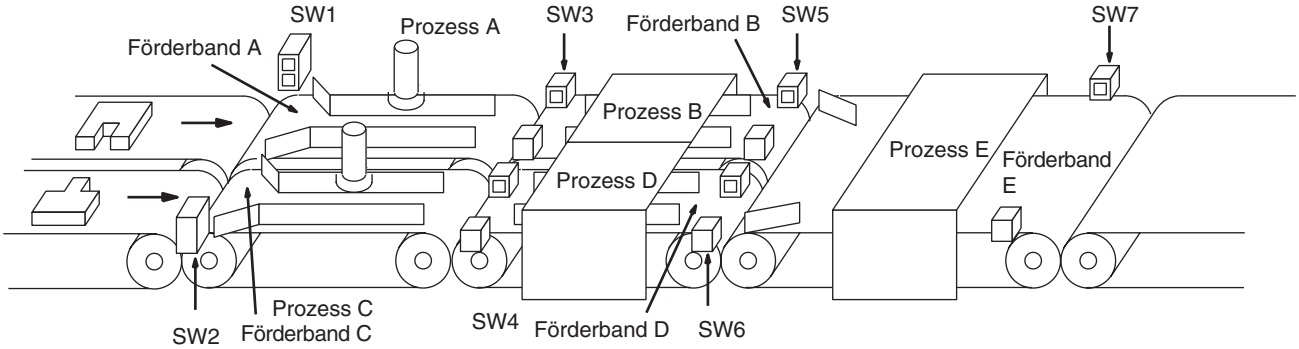


Adresse	Befehl	Operanden
000000	@LD	000001
000001	AND NOT	000002
000002	SNXT(009)	010000
000003	LD NOT	000001
000004	@AND	000002
000005	SNXT(009)	010001
000006	STEP(008)	010000
Prozess A		
000100	LD	000003
000101	SNXT(009)	010002
000102	STEP(008)	010001
Prozess B		
000100	LD	000004
000101	SNXT(009)	010002
000102	STEP(008)	010002
Prozess C		
000200	LD	000005
000201	SNXT(009)	024614
000202	STEP(008)	---

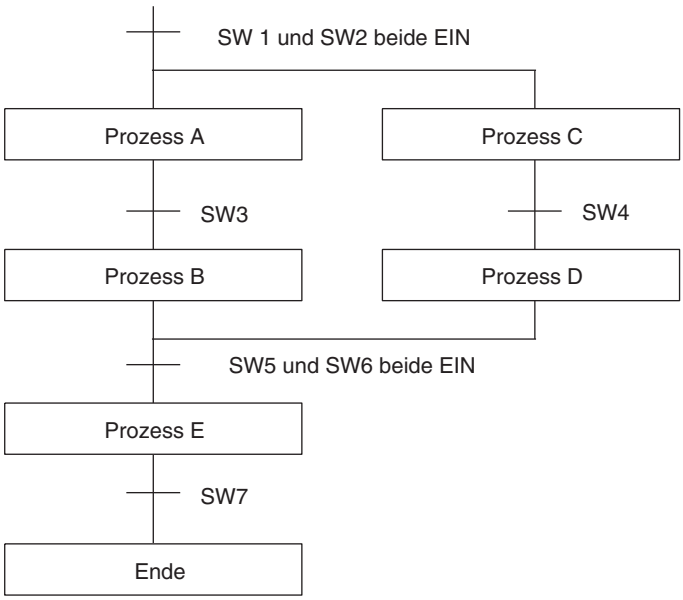
Hinweis In dem oben gezeigten Programmbeispiel wird CIO 010002 in zwei SNXT(009)-Befehlen verwendet. Dies führt bei der Programmprüfung nicht zu einem Duplizierungsfehler.

Beispiel 3: Parallelausführung

Bei dem folgenden Verfahren ist es erforderlich, dass zwei Teile eines Produkts gleichzeitig durch jeweils zwei Prozesse laufen, bevor sie in einem fünften Prozess zusammengesetzt werden. Es kommen mehrere Sensoren zum Einsatz, die signalisieren, wenn die einzelnen Prozesse begonnen und beendet werden müssen.

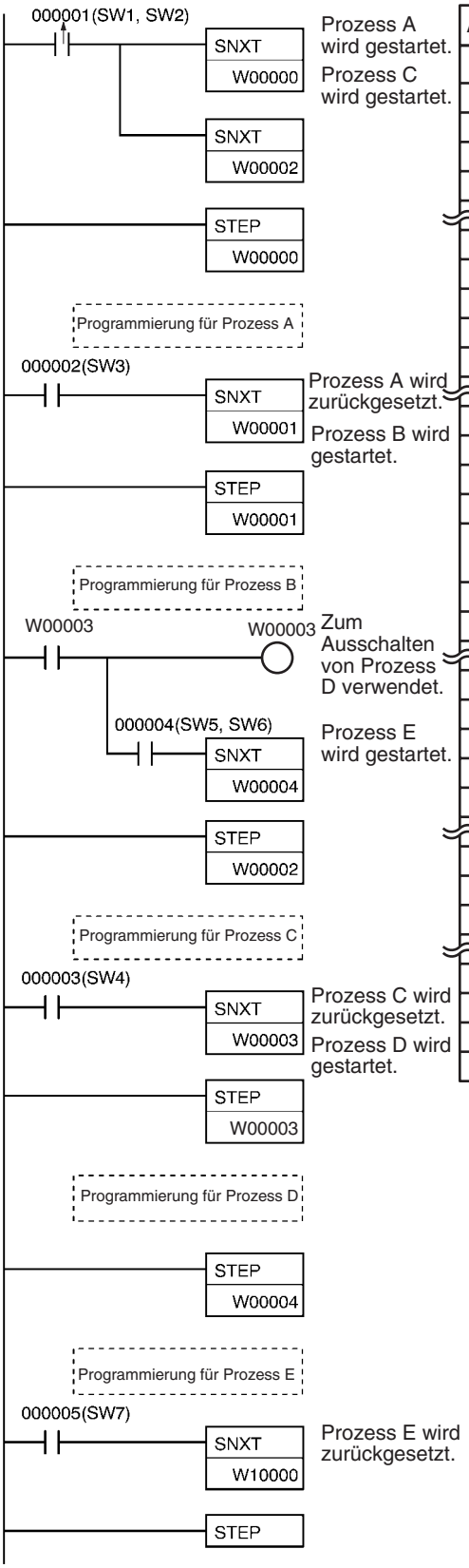


Das folgende Diagramm zeigt den Ablauf der Verarbeitung und die Schalter, die zur Ausführungssteuerung eingesetzt werden. In diesem Beispiel werden die Prozesse A und C gemeinsam gestartet. Wenn Prozess A abgeschlossen ist, startet Prozess B; wenn Prozess C abgeschlossen ist, startet Prozess D. Wenn die beiden Prozesse B und D abgeschlossen sind, wird Prozess E gestartet.



Das nachfolgend dargestellte Programm für dieses Verfahren beginnt mit zwei SNXT(009)-Befehlen, die die Prozesse A und C starten. Diese Befehle zweigen von der gleichen Befehlszeile ab, werden stets zusammen ausgeführt und starten die Programmschritte für die Prozesse A und C. Wenn die Programmschritte für A und C beide abgeschlossen sind, beginnen unmittelbar anschließend die Programmschritte für die Prozesse B und D.

Wenn sowohl Prozess B als auch Prozess D abgeschlossen sind (also wenn die Schalter SW5 und SW6 auf EIN schalten), werden die Prozesse B und D durch den Befehl SNXT(009) am Ende der Programmierung für Prozess B zusammen zurückgesetzt. Auch wenn keine SNXT(009)-Befehl am Ende von Prozess D vorhanden ist, wird das Steuerbit für diesen Prozess dennoch durch die Ausführung von SNXT(009) W00004 auf AUS gesetzt. Dies kommt zustande, da das OUT für Bit W00003 sich in dem durch SNXT(009) W00004 zurückgesetzten Schritt befindet, d. h., W00003 wird auf AUS gesetzt, wenn SNXT(009) W00004 ausgeführt wird. Prozess B wird also direkt zurückgesetzt und Prozess D wird unmittelbar vor der Ausführung des Schritts für Prozess E indirekt zurückgesetzt.



Adresse	Befehl	Operanden
000000	@LD	000001
000001	SNXT(009)	W00000
000002	SNXT(009)	W00002
000003	STEP(008)	W00000
	Prozess A	
000100	LD	000002
000101	SNXT(009)	W00001
000102	STEP(008)	W00001
	Prozess B	
000100	LD	000003
000101	OUT	W00003
000101	AND	000004
000101	SNXT(009)	W00004
000102	STEP(008)	W00002
	Prozess C	
000200	LD	000003
000201	SNXT(009)	W00003
000202	STEP(008)	W00003
	Prozess D	
000300	STEP(008)	W00004
	Prozess E	
000400	LD	000005
000401	SNXT(009)	024613
000402	STEP(008)	---

3-23 E/A-Baugruppenbefehle

In diesem Abschnitt werden die für E/A-Baugruppen verwendeten Befehle beschrieben.

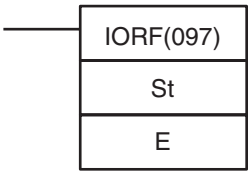
Befehl	Mnemonic	Funktionscode	Seite
I/O REFRESH	IORF	097	885
7-SEGMENT DECODER	SDEC	078	888
INTELLIGENT I/O READ	IORD	222	913
INTELLIGENT I/O WRITE	IOWR	223	917
DIGITAL SWITCH INPUT	DSW	210	890
TEN KEY INPUT	TKY	211	896
HEXADECIMAL KEY INPUT	HKY	212	899
MATRIX INPUT	MTR	213	904
7-SEGMENT DISPLAY OUTPUT	7SEG	214	908

3-23-1 I/O REFRESH: IORF(097)

Zweck

Aktualisiert die spezifizierten E/A-Worte.

Kontaktplansymbol



St: Anfangswort

E: Endwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	IORF(097)
	Ausführung einmal bei steigender Flanke	@IORF(097)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

St: Anfangswort
CIO 0000 bis CIO 0999 (E/A-Bit-Bereich) oder
CIO 2000 bis CIO 2959 (Spezial-E/A-Baugruppen-Bereich)

E: Endwort
CIO 0000 bis CIO 0999 (E/A-Bit-Bereich) oder
CIO 2000 bis CIO 2959 (Spezial-E/A-Baugruppen-Bereich)

Hinweis St und E müssen im gleichen Speicherbereich liegen.

Operanden-Spezifikationen

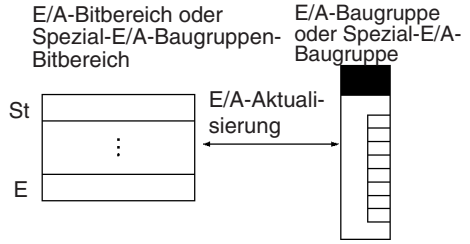
Bereich	St	E
CIO-Bereich	CIO 0000 bis CIO 0999 CIO 2000 bis CIO 2959	
Zusatz-Systembereich	---	
Haftmerkerbereich	---	
Spezial-E/A-Bit-Bereich	---	
Zeitgeberbereich	---	
Zählerbereich	---	
DM-Bereich	---	
EM-Bereich ohne Bank	---	

Bereich	St	E
EM-Bereich mit Bank	---	
Indirekte DM/EM-Binäradresse	---	
Indirekte DM/EM-Adresse in BCD	---	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047, IR0 bis IR15 DR0 bis DR15, IR0 bis IR15, IR0 bis IR15+(++) ,–(– –)IR0 bis IR15	

Beschreibung

Der Befehl IORF(097) aktualisiert die E/A-Worte zwischen St und E (einschließlich dieser Worte). IORF(097) wird zur Aktualisierung der Worte verwendet, die E/A-Baugruppen oder Spezial-E/A-Baugruppen zugewiesen sind, die im CPU-Block oder in Erweiterungsblöcken installiert sind. IORF(097) kann nicht zur gleichzeitigen Aktualisierung von Worten in beiden Bereichen (mit dem gleichen Befehl) verwendet werden). E/A-Baugruppen werden Worte im Bereich zwischen CIO 0000 und CIO 0999 zugewiesen, Spezial-E/A-Baugruppen werde Worte im Bereich zwischen CIO 2000 und CIO 2959 zugewiesen.

Wenn die Aktualisierung für Worte im Bit-Bereich einer Spezial-E/A-Baugruppe spezifiziert wird. werden alle 10 Worte, die der Baugruppe zugewiesen sind, aktualisiert, so lange das erste der 10 Worte, die der Baugruppe zugewiesen sind, im spezifizierten Wortbereich enthalten ist.



Falls innerhalb des Bereichs zwischen St und E Worte vorhanden sind, für die keine Baugruppe installiert ist, geschieht mit diesen Worten überhaupt nichts. Es werden nur Worte aktualisiert, die einer Baugruppe zugewiesen sind.

Mit dem gleichen Befehl können sowohl C200H Spezial-E/A-Baugruppen als auch CS Spezial-E/A-Baugruppen aktualisiert werden. (Nur CS-Serie.)

Alle Worte, die C200H Multipoint-E/A-Baugruppen der Gruppe 2 zugewiesen sind, müssen gleichzeitig aktualisiert werden. Die E/A-Worte der Baugruppe werden aktualisiert, wenn das erste der Baugruppe zugewiesene Wort innerhalb des spezifizierten Bereichs von E/A-Worten liegt. (Die Worte der Baugruppe werden nicht aktualisiert, wenn das Anfangswort hinter dem ersten der Baugruppe zugewiesenen Wort liegt. Hingegen werden sie selbst dann aktualisiert, wenn das Endwort vor dem letzten der Baugruppe zugewiesenen Wort liegt.) (Nur CS-Serie.)

Der Befehl IORF(097) kann in Interrupt-Tasks verwendet werden, wodurch eine schnelle Reaktion auf bestimmte E/A-Worte, die in der Interrupt-Task aktualisiert werden, möglich ist. (Siehe "Vorsichtsmaßnahmen".)

Eignung für Baugruppen

Die folgenden Baugruppen können mit dem Befehl IORF(097) aktualisiert werden. Die Aktualisierung dieser Baugruppen ist jedoch nur möglich, wenn sie im CPU-Rack oder in einem Erweiterungsrack installiert sind. Bei Installation in Slave-Rack ist keine Aktualisierung möglich.

CS-Serie E/A-Baugruppen, C200H E/A-Baugruppen (nur CS-Serie), C200H Multipoint-E/A-Baugruppen der Gruppe 2 (nur CS-Serie), CJ-Serie E/A-Bau-

gruppen und Spezial-E/A-Baugruppen (Einschließlich Multipoint-E/A-Baugruppen. Alle diesen Baugruppen zugewiesenen Worte können aktualisiert werden).

Hinweis Die Baugruppen, die mit dem Befehl IORF(097) aktualisiert werden können, sind nicht unbedingt die gleichen Baugruppen, die mit der Spezifikation der direkten Aktualisierung (!) aktualisiert werden können.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn St größer als E ist. EIN, wenn St und E in verschiedenen Speicherbereichen liegen. Bei CS1D CPU-Baugruppen: EIN, wenn die aktive und die Standby-CPU-Baugruppe nicht synchronisiert werden konnten. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Wenn Worte sowohl im E/A-Bitbereich (CIO 0000 bis CIO 0999) als auch im Spezial-E/A-Baugruppen-Bitbereich (CIO 2000 bis CIO 2959) im gleichen Befehl spezifiziert werden, tritt ein Fehler auf.

Für Baugruppen, bei denen ein E/A-Tabellenfehler aufgetreten ist, wird keine E/A-Aktualisierung vorgenommen. (Nur CS-Serie.)

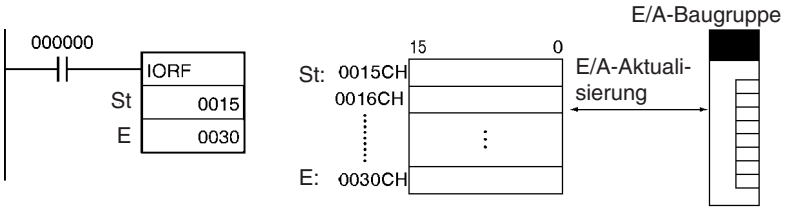
Die durch den Befehl IORF(097) initiierte E/A-Aktualisierung wird während der Verarbeitung abgebrochen, wenn während der E/A-Aktualisierung ein E/A-Busfehler auftritt.

Bei Verwendung des Befehl IORF(097) in einer Interrupt-Task muss darauf geachtet werden, dass die zyklische Aktualisierung von Spezial-E/A-Baugruppen im SPS-Setup deaktiviert wird. Wenn die zyklische Aktualisierung von Spezial-E/A-Baugruppen aktiviert ist und durch IORF(097) eine erneute Aktualisierung ausgeführt wird, tritt ein nicht schwerwiegender Fehler aufgrund doppelter Aktualisierung auf, und der Interrupt-Task-Fehlermerker (A40213) wird auf EIN gesetzt.

Beispiele

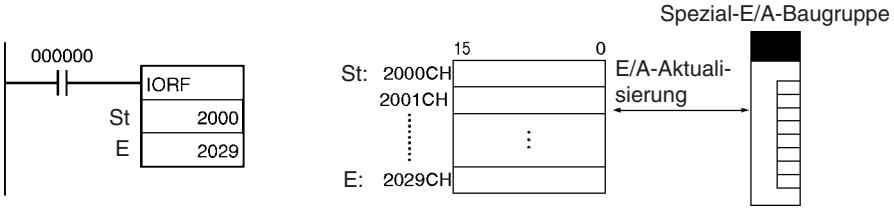
Aktualisierung von Worten im E/A-Bitbereich

Das folgende Beispiel veranschaulicht, wie die 16 Worte von CIO 0015 bis CIO 0030 aktualisiert werden, wenn CIO 000000 auf EIN gesetzt wird.



Aktualisierung von Worten im Spezial-E/A-Baugruppen-Bitbereich

Das folgende Beispiel veranschaulicht, wie die 30 Worte von CIO 2000 bis CIO 2029 aktualisiert werden, wenn CIO 000000 auf EIN gesetzt wird.

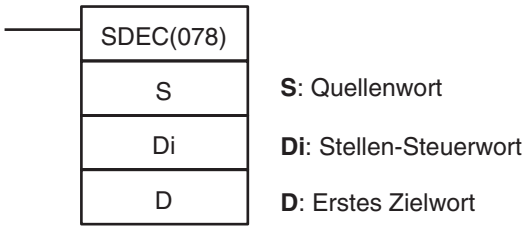


3-23-2 7-SEGMENT DECODER: SDEC(078)

Zweck

Konvertiert den hexadezimalen Inhalt der angegebenen Stelle(n) in 8-Bit-7-Segment- Anzeigecode und gibt ihn in die höher- bzw. niederwertigen 8 Bits der spezifizierten Zielworte aus.

Kontaktplansymbol



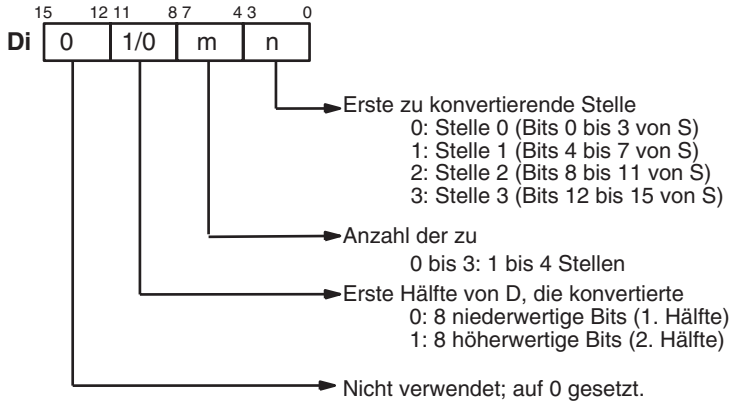
Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SDEC(078)
	Ausführung einmal bei steigender Flanke	@SDEC(078)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden: Stellen-Steuerwort



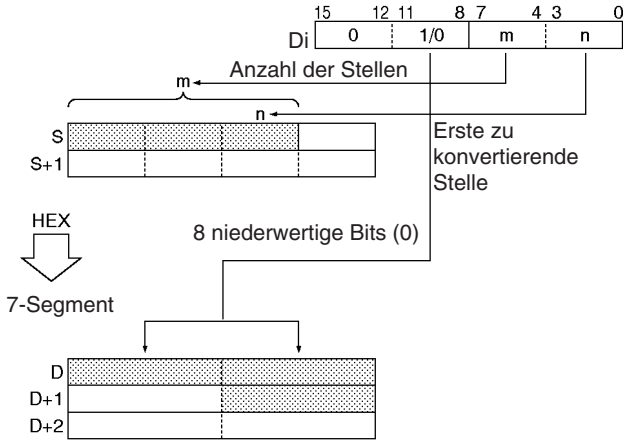
Operanden-Spezifikationen

Bereich	S	Di	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerkerbereich	H000 bis H511		
Zusatz-Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		

Bereich	S	Di	D
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	Nur spezifizierte Werte	---
Datenregister	DR0 bis DR15		---
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl SDEC(078) verarbeitet die durch S spezifizierten Daten als 4-stellige Hexadezimaldaten, konvertiert die durch Di spezifizierten Stellen von S (erste Stelle und Anzahl der Stellen) in 7-Segment-Daten und gibt das Ergebnis in die durch Di spezifizierten Bits von D aus.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Einstellungen in Di nicht innerhalb der spezifizierten Bereiche liegen. In allen anderen Fällen AUS.

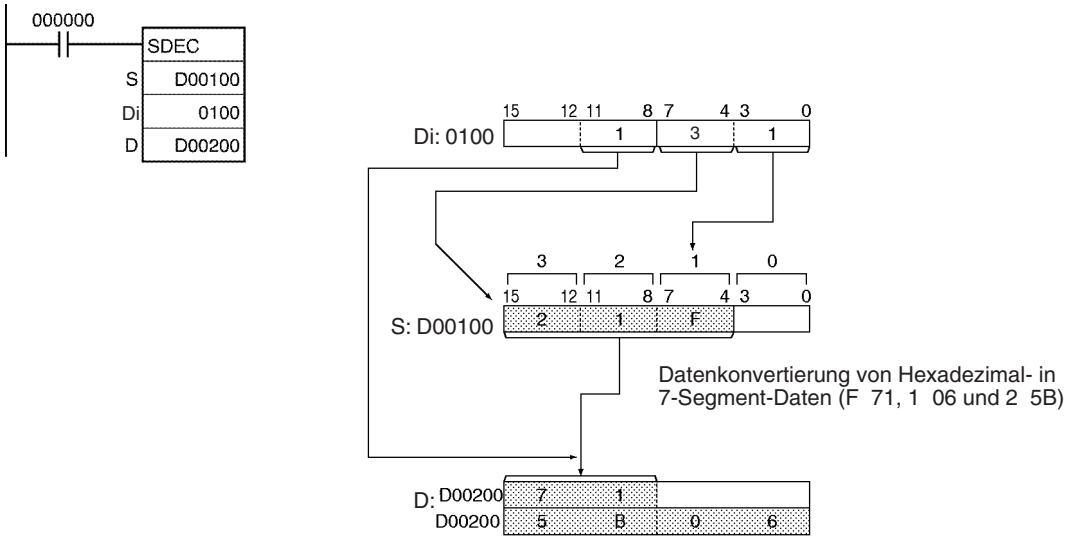
Vorsichtsmaßnahmen

Wenn in Di mehr als eine Stelle zu Konvertierung spezifiziert ist, werden die Stellen in der Reihenfolge von der nieder- zur höchstwertigen Stelle konvertiert. Die nächste Stelle nach Stelle 3 ist Stelle 0.

Die Ergebnisse werden in D in der Reihenfolge vom spezifizierten Teil zu den Worten mit höheren Adressen gespeichert. Wenn nur in einem der Bytes in einem Zielwort konvertierte Daten geschrieben werden, bleibt das andere Byte unverändert.

Beispiele

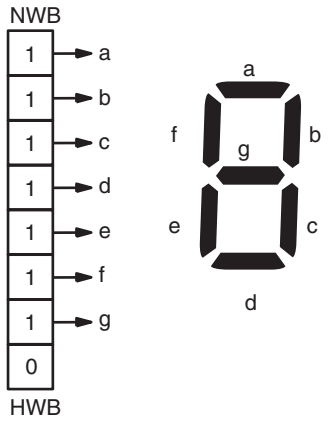
Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt wird, wird der Inhalt der 3 Stellen ab Stelle 1 in D00100 aus Hexadezimaldaten in 7-Segment-Daten konvertiert. Das Ergebnis wird in das höherwertige Byte von D00200 und beide Bytes von D00201 ausgegeben. Die Spezifikationen für die zu konvertierenden Bytes und die Adressen der Ausgabebytes erfolgen in CIO 0100.



7-Segment-Daten

In der folgenden Tabelle sind die Datenkonvertierungen aus (4-Bit-) Hexadezimalstellen in (8-Bit-) 7-Segment-Code aufgeführt.

Originaldaten					Konvertierter Code (Segmente)									Anzeige Originaldaten
Stelle	Bits				–	g	f	e	d	c	b	a	Hex	
0	0	0	0	0	0	0	1	1	1	1	1	1	3F	0
1	0	0	0	1	0	0	0	0	0	1	1	0	06	1
2	0	0	1	0	0	1	0	1	1	0	1	1	5B	2
3	0	0	1	1	0	1	0	0	1	1	1	1	4F	3
4	0	1	0	0	0	1	1	0	0	1	1	0	66	4
5	0	1	0	1	0	1	1	0	1	1	0	1	6D	5
6	0	1	1	0	0	1	1	1	1	1	0	1	7D	6
7	0	1	1	1	0	0	1	0	0	1	1	1	27	7
8	1	0	0	0	0	1	1	1	1	1	1	1	7F	8
9	1	0	0	1	0	1	1	0	1	1	1	1	6F	9
A	1	0	1	0	0	1	1	1	0	1	1	1	77	A
B	1	0	1	1	0	1	1	1	1	1	0	0	7C	b
C	1	1	0	0	0	0	1	1	1	0	0	1	39	C
D	1	1	0	1	0	1	0	1	1	1	1	0	5E	d
E	1	1	1	0	0	1	1	1	1	0	0	1	79	E
F	1	1	1	1	0	1	1	1	0	0	0	1	71	F



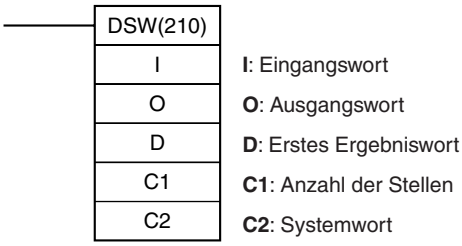
3-23-3 DIGITAL SWITCH INPUT: DSW(210)

Zweck

Liest den Wert, der auf einem an eine E/A-Baugruppe angeschlossenen externen Digital- oder Daumenradschalter eingestellt ist, und speichert den 4- bzw. 8-stelligen BCD-Wert in den spezifizierten Worten.

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	DSW(210)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

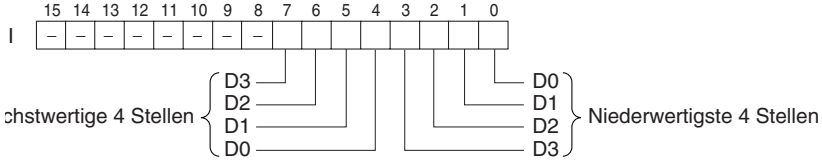
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

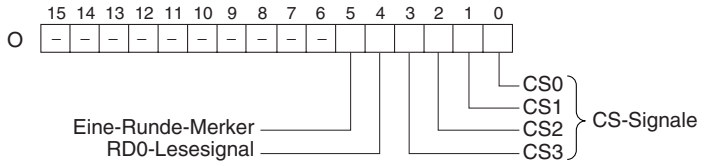
I: Eingangswort (Eingänge von Datenleitung D0 bis D3)

Spezifizieren Sie die der Eingangsbaugruppe zugewiesenen Worte, und schließen Sie die Datenleitungen D0 bis D3 des Digitalschalters wie in der folgenden Abbildung gezeigt an die Eingangsbaugruppe an.



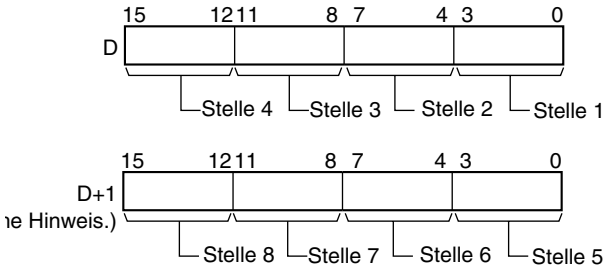
O: Ausgangswort (CS/RD-Steuersignalausgänge)

Spezifizieren Sie die der Ausgangsbaugruppe zugewiesenen Worte, und schließen Sie die Steuersignale des Digitalschalters (CS- und RD-Signale) wie in der folgenden Abbildung gezeigt an die Ausgangsbaugruppe an.



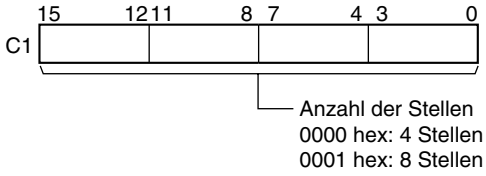
D: Erstes Ergebniswort

Spezifiziert die Adresse des ersten der Worte, in denen sie Sollwerte des externen Digitalschalters gespeichert werden.

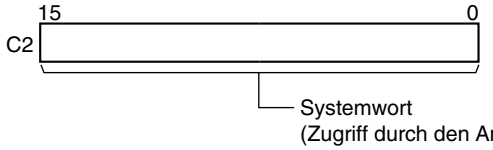


Hinweis: Nur bei C1 = 0001 hex zum Lesen von 8 Stellen.

C1: Anzahl der Stellen
Spezifiziert die Anzahl der Stellen, die vom externen Digitalschalter gelesen werden. Bei Einstellung von C1 auf 0000 hex werden 4 Stellen gelesen, bei Einstellung auf 0001 hex werden 8 Stellen gelesen.



C2: Systemwort
Spezifiziert ein Arbeitswort, das vom Befehl verwendet wird. Dieses Wort kann nicht in anderen Anwendungen genutzt werden.



Operanden-Spezifikationen

Bereich	I	O	D	C1	C2
CIO-Bereich	CIO 0000 bis CIO 6143			---	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511			---	W000 bis W511
Haftmerkerbereich	H000 bis H511			---	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A448 bis A953		---	A448 bis A959
Zeitgeberbereich	T0000 bis T4095			---	T0000 bis T4095
Zählerbereich	C0000 bis C4095			---	C0000 bis C4095
DM-Bereich	D00000 bis D32767			---	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767			---	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)			---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)			---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)			---	---
Konstanten	---			0000 oder 0001 hex	---
Datenregister	DR0 bis DR15				DR0 bis DR15

Bereich	I	O	D	C1	C2
Indexregister	---				
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15				,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl DSW(210) gibt Steuersignale an die Bits 00 bis 04 von O aus, liest die spezifizierte Anzahl von Stellen (entweder 4 oder 8 Stellen, in C1 spezifiziert) der Digitalschalter-Datenleitungsdaten mit I beginnend und speichert das Ergebnis in D und D+1. (Wenn 4 Stellen gelesen werden, wird das Ergebnis in D gespeichert. Wenn 8 Stellen gelesen werden, wird das Ergebnis in D und D+1 gespeichert.)

Der Befehl DSW(210) liest die 4- bzw. 8-stelligen Schalterdaten einmal alle 16 Zyklen, startet dann neu und fährt mit dem Lesen der Daten fort. Der Eine-Runde-Merker (Bit 05 von O) wird alle 16 Zyklen der CPU-Baugruppe auf EIN gesetzt.

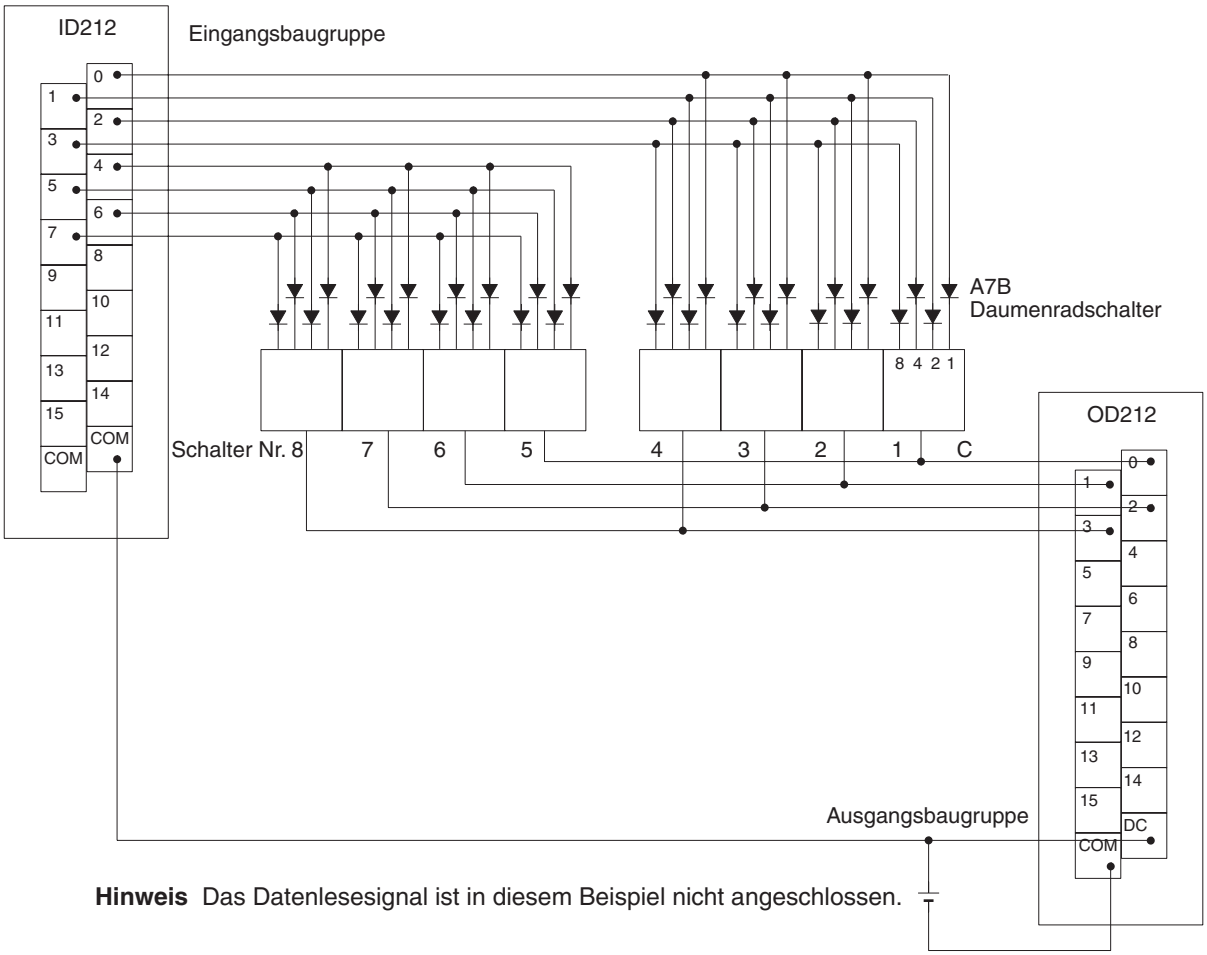
DSW(210) liest die 4- bzw. 8-stelligen Schalterdaten einmal alle 16 Zyklen, startet dann neu und liest die Daten in den nächsten 16 Zyklen erneut.

Bei der Ausführung beginnt der Befehl DSW(210) mit dem Lesen der Schalterdaten ab dem ersten der 16 Zyklen unabhängig davon, an welchem Punkt der letzte Befehl gestoppt wurde.

Es gibt keine Beschränkung wie oft der Befehl DSW(210) im Programm verwendet werden kann (im Gegensatz zur C200HX/HG/HE und zur CQM1H-Serie).

Externe Anschlüsse

Schließen Sie den Digital- bzw. Daumenradschalter an die Kontakte 0 bis 7 der Eingangsbaugruppe und an die Kontakte 0 bis 4 der Ausgangsbaugruppe an (siehe folgende Abbildung). In der folgenden Abbildung ist der Anschluss einen A7B Daumenradschalters dargestellt.

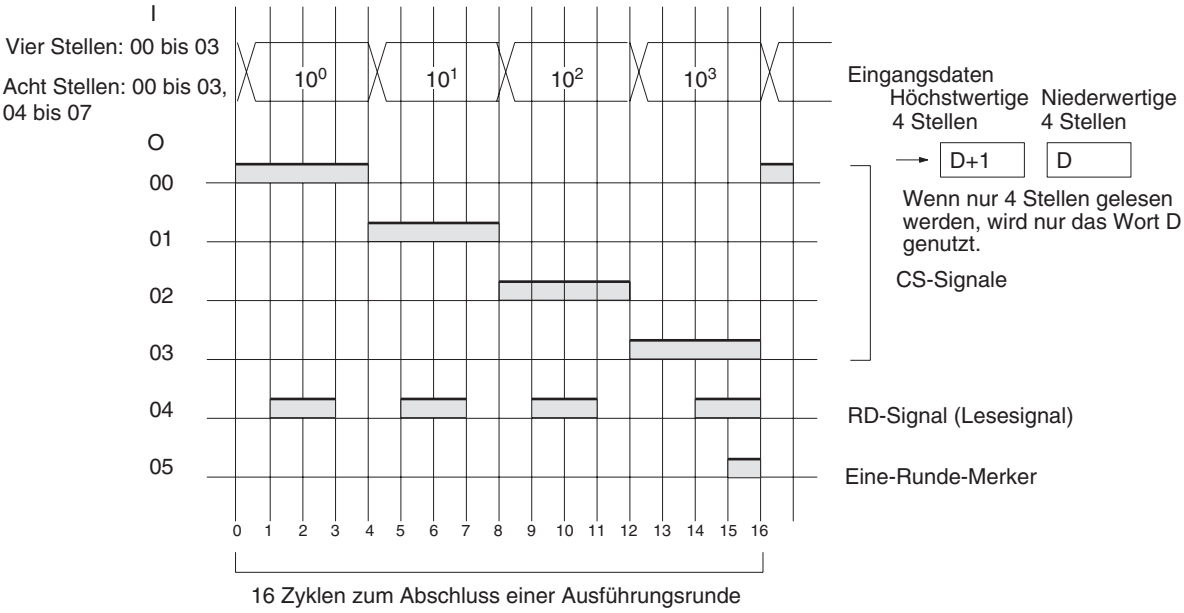


Hinweis Das Datenlesesignal ist in diesem Beispiel nicht angeschlossen.

Die Ein- und Ausgänge können an die folgenden Arten von E/A-Baugruppen und Multipoint-E/A-Baugruppen angeschlossen werde, sofern diese nicht in einem dezentralen SYSMAC BUS E/A-Block installiert sind.

- DC-Eingangsbaugruppen mit min. 8 Eingängen
- Transistor-Ausgangsbaugruppen mit min. 8 Ausgängen

Zeitablaufdiagramm



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS

Vorsichtsmaßnahmen

Das Systemwort (C2) darf nicht von anderen Befehlen gelesen oder beschrieben werden. Der Befehl DSW(210) funktioniert nicht einwandfrei, wenn durch einen anderen Befehl auf das Systemwort zugegriffen wird. Das Systemwort wird im ersten Zyklus bei Beginn der Programmausführung nicht durch DSW(210) initialisiert. Wenn der Befehl DSW(210) bereits ab dem ersten Programmzyklus ausgeführt werden soll, muss der Systemwort aus dem Programm heraus gelöscht werden.

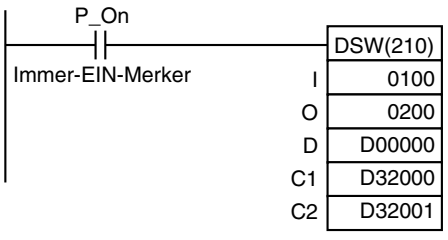
Der Befehl DSW(210) funktioniert nicht ordnungsgemäß, wenn nach der Ausführung von DSW(210) keine E/A-Aktualisierung bei an die Eingangs- und Ausgangsbaugruppe angeschlossenem Digital- bzw. Daumenradschalter erfolgt. Dementsprechend darf der Digital- bzw. Daumenradschalter nicht an die folgenden Baugruppen angeschlossen werden.

- E/A-Baugruppen oder Multipoint-E/A-Baugruppen, die in einem dezentralen SYSMAC BUS E/A-Slave-Block installiert sind
- Kommunikations-Slaves (DeviceNet- oder CompoBus/S-Slaves)

Beispiel

In diesem Beispiel wird der Befehl DSW(210) verwendet, um eine 8-stellige Zahl von einem Digitalschalter zu lesen und den resultierenden Wert ständig nach D00000 und D00001 auszugeben. Die Digitalschalter ist über CIO 0100 (einer CS1W-ID211 16-Punkt DC-Eingangsbaugruppe zugewiesen) und CIO 0200 (einer CS1W-OD211 16-Punkt Transistor-Ausgangsbaugruppe zugewiesen) verbunden.

Da 8 Datenstellen gelesen werden, ist C1 (in diesem Beispiel D32000) auf 0001 hex gesetzt. D32001 wird als Systemwort verwendet.



3-23-4 TEN KEY INPUT: TKY(211)

Zweck

Liest numerische Daten von einem an eine Eingangsbaugruppe angeschlossenen numerischen Tastenfeld und speichert bis zu 8 Stellen BCD-Daten in den spezifizierten Worten.

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TKY(211)
	Ausführung einmal bei steigender Flanke	@TKY(211)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

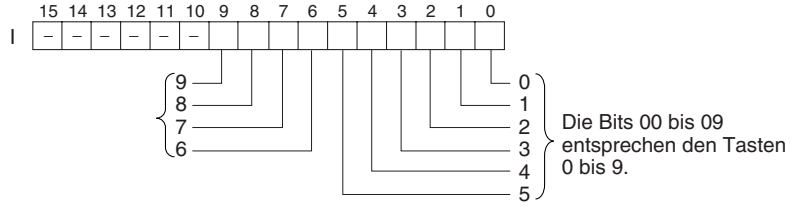
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

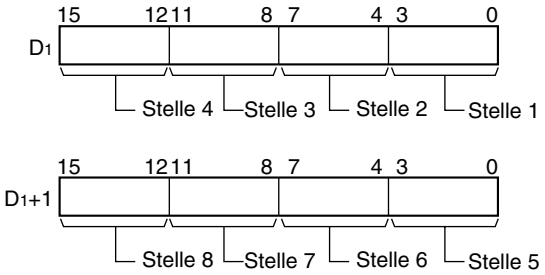
I: Eingangswort (Datenleitungseingänge)

Spezifizieren Sie die der Eingangsbaugruppe zugewiesenen Worte, und schließen Sie die Datenleitungen 0 bis 9 des numerischen Tastenfelds wie in der folgenden Abbildung gezeigt an die Eingangsbaugruppe an.



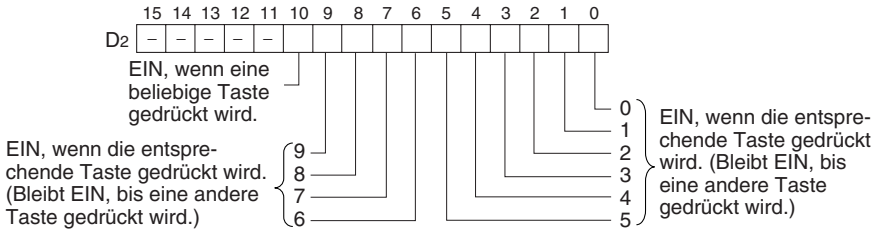
D_1: Erstes Registerwort

Spezifiziert die Adresse des ersten der Worte, in denen die numerischen Eingaben des numerischen Tastenfelds (bis zu 8 Stellen) gespeichert werden.



D_2: Tasteneingabewort

Bits 00 bis 10 von D_2 geben die Tasteneingaben wieder. Wenn eine der Tasten des Tastenfelds (0 bis 9) gedrückt wird, wird das entsprechende Bit von D_2 (0 bis 9) auf EIN gesetzt. Bit 10 von D_2 ist auf EIN gesetzt, während eine beliebige Taste gedrückt wird.



Hinweis Der Befehl TKY(211) benötigt, im Gegensatz zu anderen E/A-Befehlen, wie z. B. HKY(212), kein Systemwort.

Operanden-Spezifikationen

Bereich	I	D_1	D_2
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W510	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H510	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A448 bis A958	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl TKY(211) liest die numerischen Daten aus dem Eingangswort I, das einem an eine Eingangsbaugruppe angeschlossenen numerischen Tastenfeld zugewiesen ist, und speichert bis zu 8 Stellen DCD-Daten in den Registerworten D_1 und D_1+1. Zusätzlich wird bei jedem Drücken einer Taste das entsprechende Bit in D_2 (0 bis 9) auf EIN gesetzt und behält diesen Zustand, bis eine weitere Taste gedrückt wird. Bit 10 von D_2 wird auf EIN gesetzt, während eine Taste gedrückt wird. Wenn keine Taste gedrückt wird, ist es auf AUS gesetzt.

Das Zwei-Worte-Register in D_1 und D_1+1 arbeitet als 8-stelliges Schieberegister. Beim Drücken einer Taste des numerischen Tastenfeldes wird die entsprechende BCD-Stelle in die niederwertigste Stelle von D_1 geschoben. Die

anderen Stellen von D_1 und D_1+1 werden nach links verschoben und die höchstwertige Stelle von D_1+1 geht verloren.

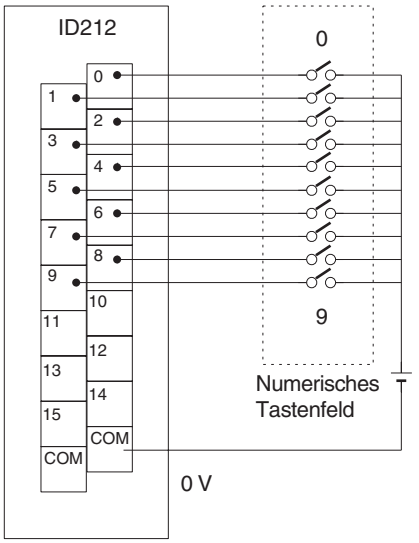
Bei der Ausführung beginnt der Befehl TKY(211) mit dem Lesen der Tasteneingabedaten ab dem ersten Zyklus unabhängig davon, an welchem Punkt der letzte Befehl gestoppt wurde.

Wenn eine der Tasten des Tastenfeldes gedrückt wird, wird die Eingabe von den anderen Tasten deaktiviert.

Es gibt keine Beschränkung für die Anzahl, mit der der Befehl TKY(211) im Programm verwendet werden kann (im Gegensatz zur C200HX/HG/HE und zur CQM1H-Serie).

Externe Anschlüsse

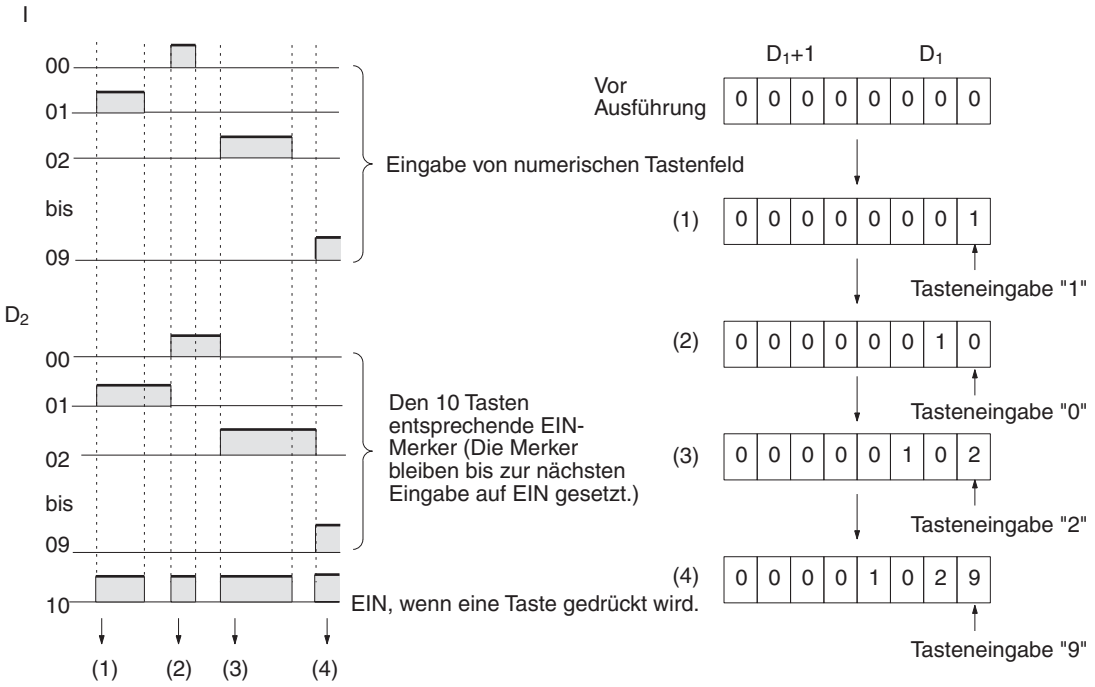
Schließen Sie das numerische Tastenfeld so an, dass die Schalter der Tasten 0 bis 9 über die Kontakte 0 bis 9 der Eingangsbaugruppe eingegeben werden (siehe folgende Abbildung).



DC-Eingangsbaugruppe

Bei der Eingangsbaugruppe muss es sich um eine DC-Eingangsbaugruppe oder Multipoint-E/A-Baugruppe mit mindestens 16 Eingängen handeln. Die Eingangsbaugruppe darf nicht in einem dezentralen SYSMAC BUS E/A-Block installiert sein.

Zeitablaufdiagramm



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS

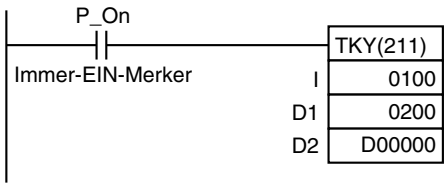
Vorsichtsmaßnahmen

Der Befehl TKY(211) funktioniert nicht ordnungsgemäß, wenn nach der Ausführung von TKY(211) keine E/A-Aktualisierung bei an die Eingangsbaugruppe angeschlossenem numerischen Tastenfeld erfolgt. Dementsprechend darf das numerische Tastenfeld nicht an die folgenden Baugruppen angeschlossen werden.

- E/A-Baugruppen oder Multipoint-E/A-Baugruppen, die in einem dezentralen SYSMAC BUS E/A-Slave-Block installiert sind
- Kommunikations-Slaves (DeviceNet- oder CompoBus/S-Slaves)

Beispiel

In diesem Beispiel liest der Befehl TKY(211) die Tasteneingaben über ein numerisches Tastenfeld und speichert die Eingaben in D00000 und D00001. Das numerische Tastenfeld ist mit CIO 0100 verbunden (einer CS1W-ID211 16-Punkt DC-Eingangsbaugruppe zugewiesen).



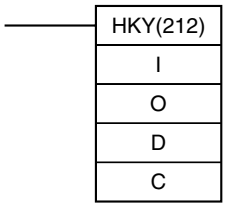
3-23-5 HEXADECIMAL KEY INPUT: HKY(212)

Zweck

Liest numerische Daten von einem an eine Eingangs- und Ausgangsbaugruppe angeschlossenen Hexadezimal-Tastenblock und speichert bis zu 8 Stellen Hexadezimaldaten in den spezifizierten Worten.

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



I: Eingangswort
O: Ausgangswort
D: Erstes Registerwort
C: Systemwort

Variationen

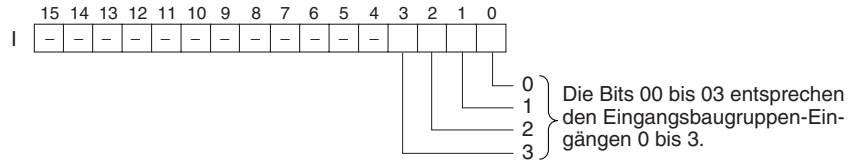
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	HKY(212)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

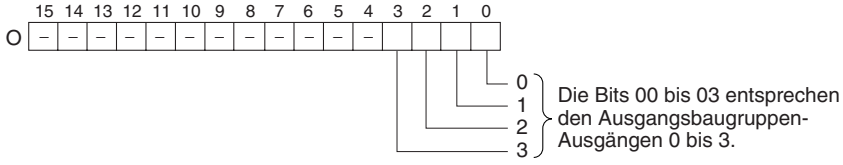
Operanden

I: **Eingangswort (Eingänge von Datenleitung D0 bis D3)**
Spezifizieren Sie die der Eingangsbaugruppe zugewiesenen Worte, und schließen Sie die Datenleitungen D0 bis D3 des Hexadezimal-Tastenblocks wie in der folgenden Abbildung gezeigt an die Eingangsbaugruppe an.



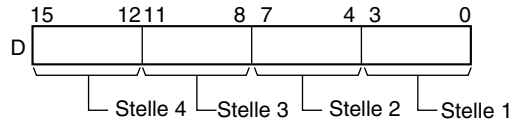
O: Ausgangswort (Auswahlsignal-Ausgänge)

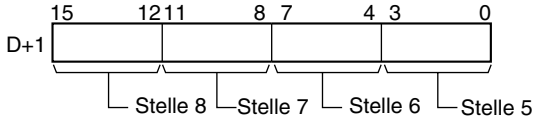
Spezifizieren Sie die der Ausgangsbaugruppe zugewiesenen Worte, und schließen Sie die Auswahlsignale des Hexadezimal-Tastenblocks wie in der folgenden Abbildung gezeigt an die Ausgangsbaugruppe an.

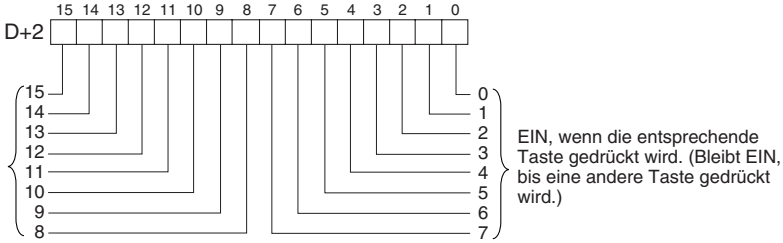


D: Erstes Registerwort

Spezifiziert die Adresse des ersten der Worte, in denen die numerischen Eingaben des Hexadezimal-Tastenblocks (bis zu 8 Stellen) gespeichert werden. (Zusätzlich wird bei jedem Drücken einer Taste das entsprechende Bit in D+2 (0 bis F) auf EIN gesetzt und behält diesen Zustand, bis eine weitere Taste gedrückt wird.)

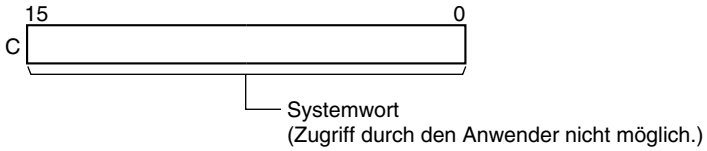






C: Systemwort

Spezifiziert ein Arbeitswort, das vom Befehl verwendet wird. Dieses Wort kann nicht in anderen Anwendungen genutzt werden.



Operanden-Spezifikationen

Bereich	I	O	D	C
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6141	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511		W000 bis W509	W000 bis W511
Haftmerkerbereich	H000 bis H511		H000 bis H509	H000 bis H511
Zusatz-Systembereich	A000 bis A957	A448 bis A959	A448 bis A957	A448 bis A959
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4093	T0000 bis T4095
Zählerbereich	C0000 bis C4095		C0000 bis C4093	C0000 bis C4095
DM-Bereich	D00000 bis D32767		D00000 bis D32765	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32765	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32765 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)			

Bereich	I	O	D	C
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)			
Konstanten	---			
Datenregister	DR0 bis DR15		---	DR0 bis DR15
Indexregister	---			
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15			

Beschreibung

Der Befehl HKY(212) gibt die Auswahlsignale in die Bits 00 bis 03 von O aus, liest die Daten von I in der Reihenfolge von Bit 00 bis 03 und speichert bis zu 8 Stellen Hexadezimaldaten in den Registerworten D und D+1.

Der Befehl HKY(212) liest jede Stelle in 3 bis 12 Zyklen ein, startet dann neu und setzt die Eingabe fort. Zusätzlich wird bei jedem Drücken einer Taste das entsprechende Bit in D+2 (0 bis F) auf EIN gesetzt und behält diesen Zustand, bis eine weitere Taste gedrückt wird.

HKY(212) ermittelt, welche Taste gedrückt wurde, durch die Bestimmung, welcher Eingang auf EIN gesetzt ist, wenn ein gegebenes Auswahlsignal EIN ist, sodass das Lesen einer Hexadezimalstelle zwischen 3 und 12 Zyklen dauern kann. Nachdem die Tasteneingabe gelesen wurde, startet der Befehl HKY(212) neu und liest in den nächsten 3 bis 12 Zyklen eine weitere Stelle.

Bei der Ausführung beginnt der Befehl HKY(212) mit dem Lesen der Tasteneingabedaten ab dem ersten Auswahlsignal unabhängig davon, an welchem Punkt der letzte Befehl gestoppt wurde.

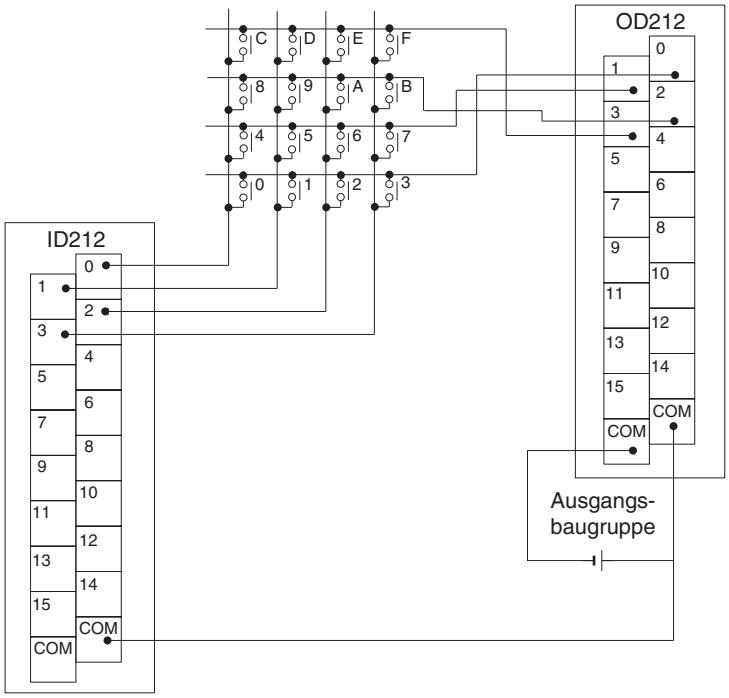
Das Zwei-Worte-Register in D_1 und D_1+1 arbeitet als 8-stelliges Schieberegister. Beim Drücken einer Taste des Hexadezimal-Tastenblocks wird die entsprechende Hexadezimalstelle in die niederwertigste Stelle von D_1 geschoben. Die anderen Stellen von D_1 und D_1+1 werden nach links verschoben und die höchstwertige Stelle von D_1+1 geht verloren.

Wenn eine der Tasten des Tastenblocks gedrückt wird, wird die Eingabe von den anderen Tasten deaktiviert.

Es gibt keine Beschränkung wie oft der Befehl HKY(212) im Programm verwendet werden kann (im Gegensatz zur CQM1H-Serie).

Externe Anschlüsse

Schließen Sie den Hexadezimal-Tastenblock an die Kontakte 0 bis 3 der Eingangsbaugruppe und an die Kontakte 0 bis 3 der Ausgangsbaugruppe an (siehe folgende Abbildung).

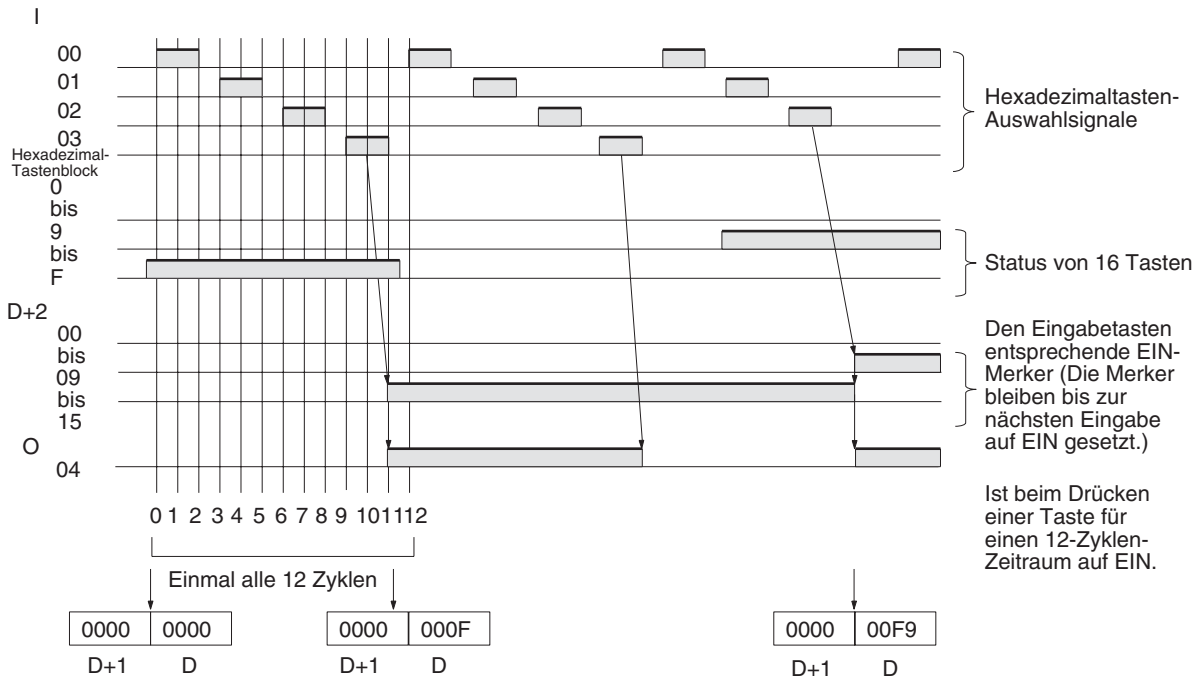


Eingangsbaugruppe

Die Ein- und Ausgänge können an die folgenden Arten von E/A-Baugruppen und Multipoint-E/A-Baugruppen angeschlossen werde, sofern diese nicht in einem dezentralen SYSMAC BUS E/A-Block installiert sind.

- DC-Eingangsbaugruppen mit min. 8 Eingängen
- Transistor-Ausgangsbaugruppen mit min. 8 Ausgängen

Zeitablaufdiagramm



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS

Vorsichtsmaßnahmen

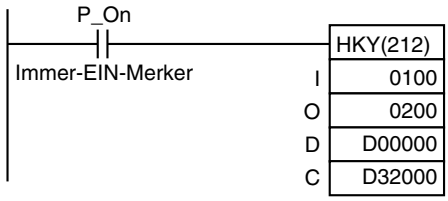
Das Systemwort (C) darf nicht von anderen Befehlen gelesen oder beschrieben werden. Der Befehl HKY(212) funktioniert nicht einwandfrei, wenn durch einen anderen Befehl auf das Systemwort zugegriffen wird. Das Systemwort wird im ersten Zyklus bei Beginn der Programmausführung nicht durch HKY(212) initialisiert. Wenn der Befehl HKY(212) bereits ab dem ersten Programmzyklus ausgeführt werden soll, muss der Systemwort aus dem Programm heraus gelöscht werden.

Der Befehl HKY(212) funktioniert nicht ordnungsgemäß, wenn nach der Ausführung von HKY(212) keine E/A-Aktualisierung bei an die Eingangs- und Ausgangsbaugruppe angeschlossenem Hexadezimal-Tastenblock erfolgt. Dementsprechend darf der Hexadezimal-Tastenblock nicht an die folgenden Baugruppen angeschlossen werden.

- E/A-Baugruppen oder Multipoint-E/A-Baugruppen, die in einem dezentralen SYSMAC BUS E/A-Slave-Block installiert sind
- Kommunikations-Slaves (DeviceNet- oder CompoBus/S-Slaves)

Beispiel

In diesem Beispiel liest der Befehl HKY(212) bis zu 8 Stellen Hexadezimaldaten von einem Hexadezimal-Tastenblock und speichert die Daten in D00000 und D00001. Der Hexadezimal-Tastenblock ist über CIO 0100 (einer CS1W-ID211 16-Punkt DC-Eingangsbaugruppe zugewiesen) und CIO 0200 (einer CS1W-OD211 16-Punkt Transistor-Ausgangsbaugruppe zugewiesen) verbunden. D32000 wird als Systemwort verwendet.



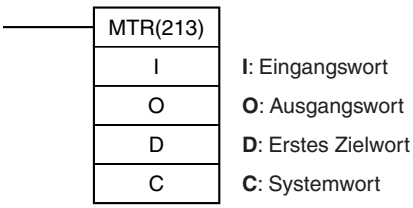
3-23-6 MATRIX INPUT: MTR(213)

Zweck

Gibt bis zu 64 Signale von einer an eine Eingangs- und Ausgangsbaugruppe (mit 8 Eingängen und 8 Ausgängen) angeschlossenen 8 × 8-Matrix ein und speichert diese 64-Bit-Daten in den 4 Zielworten.

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MTR(213)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

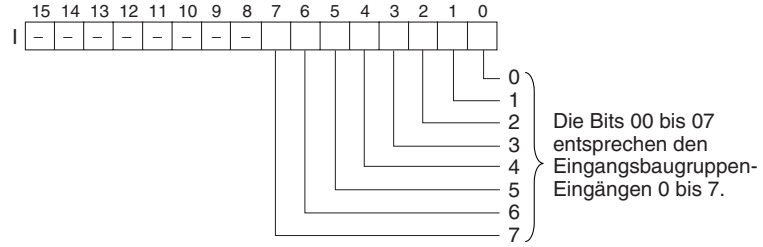
Anwendbarkeit auf Programmbereiche

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

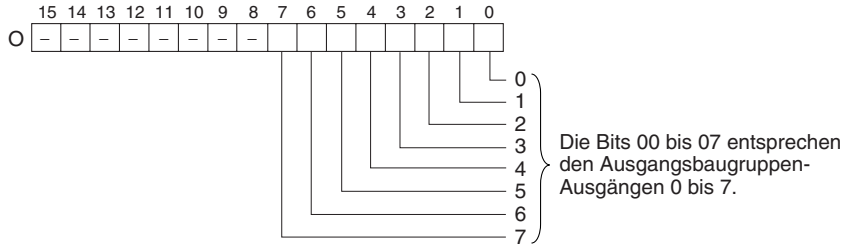
I: Eingangswort

Spezifizieren Sie die der Eingangsbaugruppe zugewiesenen Worte, und schließen Sie die 8 Eingangssignalleitungen wie in der folgenden Abbildung gezeigt an die Eingangsbaugruppe an.



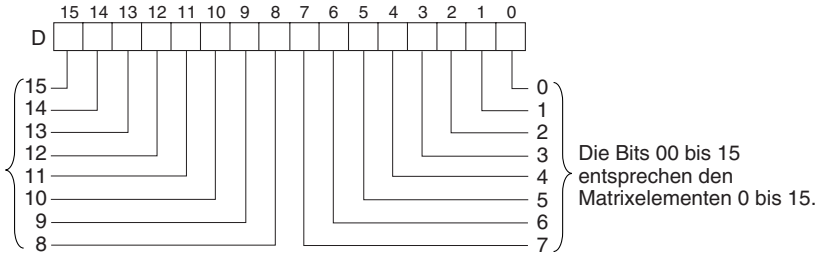
O: Ausgangswort (Auswahlsignal-Ausgänge)

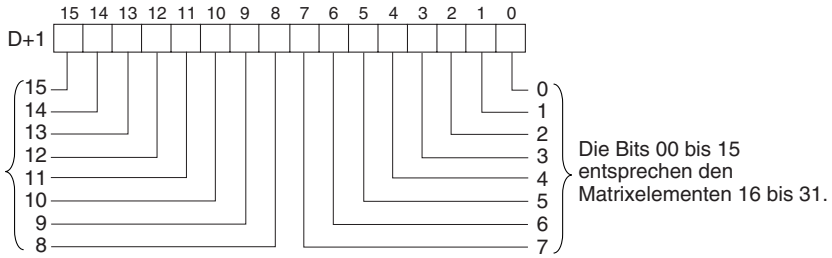
Spezifizieren Sie die der Ausgangsbaugruppe zugewiesenen Worte, und schließen Sie die 8 Auswahlsignale wie in der folgenden Abbildung gezeigt an die Ausgangsbaugruppe an.

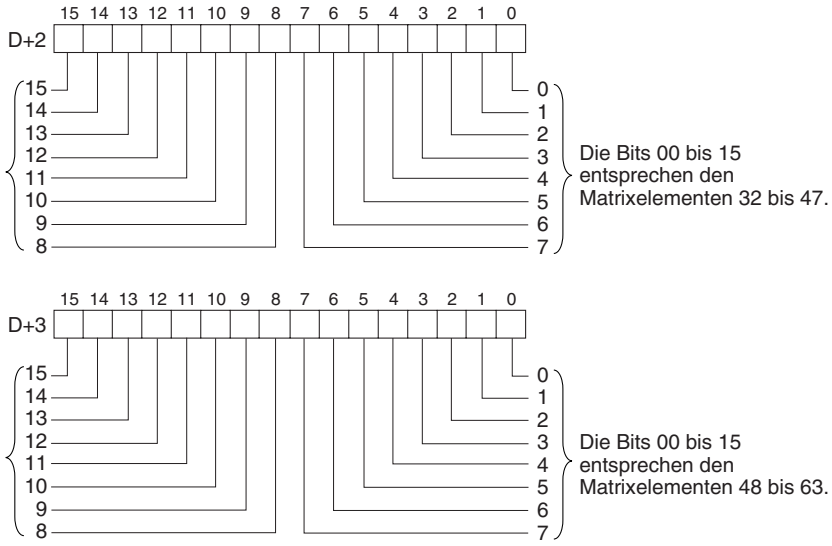


D: Erstes Registerwort

Spezifiziert die Adresse des ersten der 4 Worte, die die Daten aus der 8 × 8-Matrix enthalten.

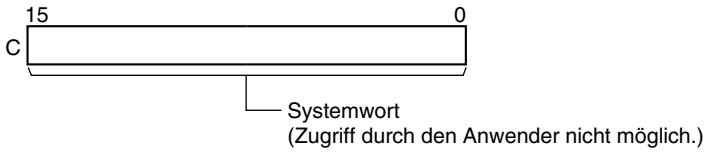






C: Systemwort

Spezifiziert ein Arbeitswort, das vom Befehl verwendet wird. Dieses Wort kann nicht in anderen Anwendungen genutzt werden.



Operanden-Spezifikationen

Bereich	I	O	D	C
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 614	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511		W000 bis W508	W000 bis W511
Haftmerkerbereich	H000 bis H511		H000 bis H508	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A448 bis A959	A448 bis A956	A448 bis A959
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4092	T0000 bis T4095
Zählerbereich	C0000 bis C4095		C0000 bis C4092	C0000 bis C4095
DM-Bereich	D00000 bis D32767		D00000 bis D32764	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32764	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32764 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)			
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)			
Konstanten	---			
Datenregister	DR0 bis DR15		---	DR0 bis DR15

Bereich	I	O	D	C
Indexregister	---			
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15			

Beschreibung

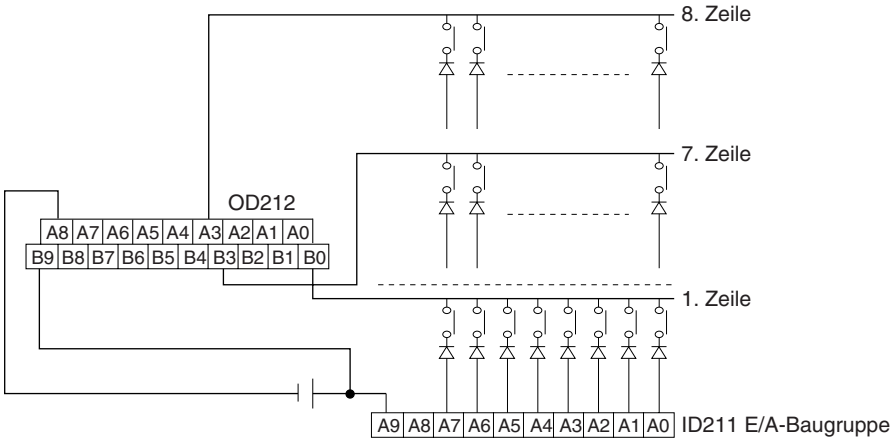
Der Befehl MTR(213) gibt die Auswahlsignale in die Bits 00 bis 07 von O aus, liest die Daten von I in der Reihenfolge von Bit 00 bis 07 und speichert die 64 Bits Daten in den 4 Registerworten D bis D+3. MTR(213) liest den Status der 64-Bit-Matrix alle 24 Zyklen der CPU-Baugruppe. Der Eine-Runde-Merker (Bit 08 von O) wird alle 24 Zyklen, nachdem jedes der Auswahlsignal auf EIN gesetzt wurde, für einen Zyklus auf EIN gesetzt.

Bei der Ausführung beginnt der Befehl MTR(213) mit dem Lesen des Matrixstatus ab dem Anfang der Matrix unabhängig davon, an welchem Punkt der letzte Befehl gestoppt wurde.

Es gibt keine Beschränkung wie oft der Befehl MTR(213) im Programm verwendet werden kann (im Gegensatz zur C200HX/HG/HE und zur CQM1H-Serie).

Externe Anschlüsse

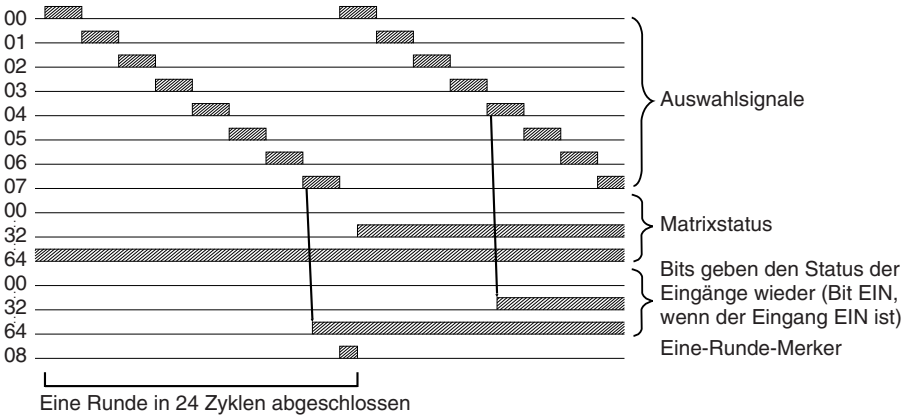
Schließen Sie die 8 x 8-Matrix an die Kontakte 0 bis 3 der Eingangsbaugruppe und an die Kontakte 0 bis 3 der Ausgangsbaugruppe an (siehe folgende Abbildung).



Die Ein- und Ausgänge können an die folgenden Arten von E/A-Baugruppen und Multipoint-E/A-Baugruppen angeschlossen werde, sofern diese nicht in einem dezentralen SYSMAC BUS E/A-Block installiert sind.

- DC-Eingangsbaugruppen mit min. 8 Eingängen
- Transistor-Ausgangsbaugruppen mit min. 8 Ausgängen

Zeitablaufdiagramm



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS

Vorsichtsmaßnahmen

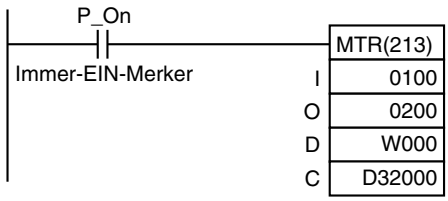
Das Systemwort (C) darf nicht von anderen Befehlen gelesen oder beschrieben werden. Der Befehl MTR(213) funktioniert nicht einwandfrei, wenn durch einen anderen Befehl auf das Systemwort zugegriffen wird. Das Systemwort wird im ersten Zyklus bei Beginn der Programmausführung nicht durch MTR(213) initialisiert. Wenn der Befehl MTR(213) bereits ab dem ersten Programmzyklus ausgeführt werden soll, muss der Systemwort aus dem Programm heraus gelöscht werden.

Der Befehl MTR(213) funktioniert nicht ordnungsgemäß, wenn nach der Ausführung von MTR(213) keine E/A-Aktualisierung bei an die Eingangs- und Ausgangsbaugruppe angeschlossener externer Matrix erfolgt. Dementsprechend darf die externe Matrix nicht an die folgenden Baugruppen angeschlossen werden.

- E/A-Baugruppen oder Multipoint-E/A-Baugruppen, die in einem dezentralen SYSMAC BUS E/A-Slave-Block installiert sind
- Kommunikations-Slaves (DeviceNet- oder CompoBus/S-Slaves)

Beispiel

In diesem Beispiel liest der Befehl MTR(213) die 64 Bits Daten von der 8 × 8-Matrix und speichert diese Daten in W000 bis W003. Die 8 × 8-Matrix ist über CIO 0100 (einer CS1W-ID211 16-Punkt DC-Eingangsbaugruppe zugewiesen) und CIO 0200 (einer CS1W-OD211 16-Punkt Transistor-Ausgangsbaugruppe zugewiesen) verbunden. D32000 wird als Systemwort verwendet.



3-23-7 7-SEGMENT DISPLAY OUTPUT: 7SEG(214)

Zweck

Konvertiert die Eingangsdaten (4-stellige oder 8-stellige BCD-Daten) in Daten für eine 7-Segment-Anzeige und gibt diese Daten in das spezifizierte Ausgangswort aus.

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol

7SEG(214)	
S	**S**: Quellenwort
O	**O**: Ausgangswort
C	**C**: Steuerdaten
D	**D**: Systemwort

Variationen

Variationen		7SEG(214)
	Ausführung in jedem Zyklus bei EIN-Bedingung	7SEG(214)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

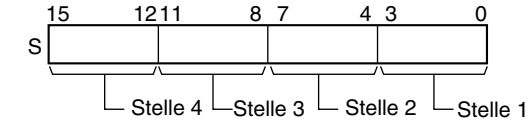
Anwendbarkeit auf Programmbereiche

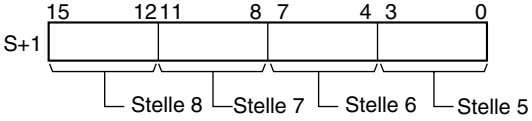
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

S: Quellenwort

Spezifizieren Sie das erste Quellenwort, das die Daten enthält, die in Daten für eine 7-Segment-Anzeige konvertiert werden sollen.

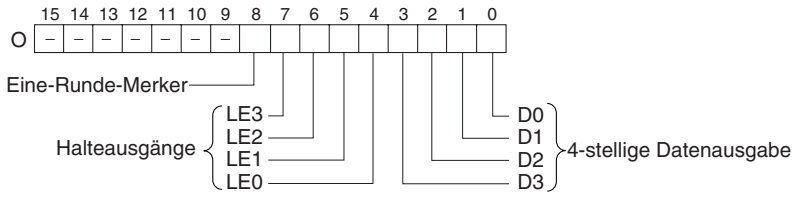




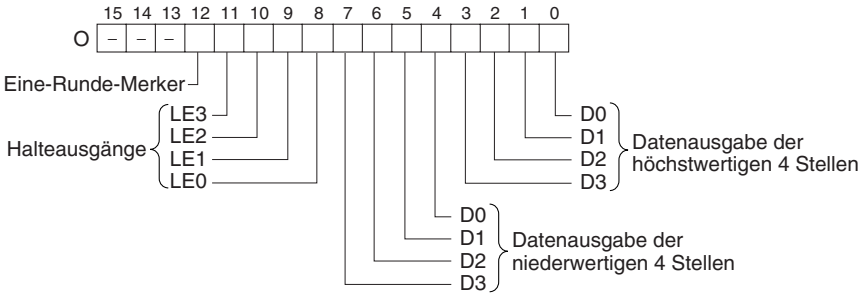
O: Ausgangswort (Daten- und Halteausgänge)

Spezifizieren Sie die der Ausgangsbaugruppe zugewiesenen Worte, und schließen Sie die 7-Segment-Anzeige wie in der folgenden Abbildung gezeigt an die Ausgangsbaugruppe an.

- Konvertierung von 4 Stellen



- Konvertierung von 8 Stellen



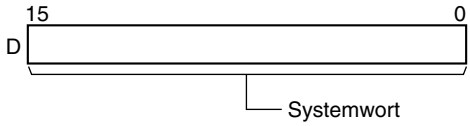
C: Steuerdaten

Der Wert für C spezifiziert die Stellenanzahl der Quelldaten und die Logik für die Eingangs- und Ausgangsbaugruppen (siehe nachfolgende Tabelle). (Mit Logik ist die NPN- bzw. PNP-Logik der Transistorausgänge gemeint.)

Quelldaten	Dateneingangslogik der Anzeige	Halteeingangslogik der Anzeige	C
4 Stellen (S)	Wie Ausgangsbaugruppe	Wie Ausgangsbaugruppe	0000
		Anders als Ausgangsbaugruppe	0001
	Anders als Ausgangsbaugruppe	Wie Ausgangsbaugruppe	0002
		Anders als Ausgangsbaugruppe	0003
8 Stellen (S, S+1)	Wie Ausgangsbaugruppe	Wie Ausgangsbaugruppe	0004
		Anders als Ausgangsbaugruppe	0005
	Anders als Ausgangsbaugruppe	Wie Ausgangsbaugruppe	0006
		Anders als Ausgangsbaugruppe	0007

D: Systemwort

Spezifiziert ein Arbeitswort, das vom Befehl verwendet wird. Dieses Wort kann nicht in anderen Anwendungen genutzt werden.



Operanden-Spezifikationen

Bereich	S	O	C	D
CIO-Bereich	CIO 0000 bis CIO 6143		---	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511		---	W000 bis W511
Haftmerkerbereich	H000 bis H511		---	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A448 bis A959	---	A448 bis A959
Zeitgeberbereich	T0000 bis T4095		---	T0000 bis T4095
Zählerbereich	C0000 bis C4095		---	C0000 bis C4095
DM-Bereich	D00000 bis D32767		---	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767		---	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		---	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)			
Konstanten	---	---	0000 bis 0007	---
Datenregister	---	DR0 bis DR15	---	DR0 bis DR15

Bereich	S	O	C	D
Indexregister	---			
Indirektes Adressieren mittels Index-Registern	IR0 bis IR15, –2048 bis +2047, IR0 bis IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl 7SEG(214) liest die Quelldaten, konvertiert sie in Daten für eine 7-Segment-Anzeige, und gibt diese Daten (höchstwertige 4 Stellen D0 bis D3, niederwertige 4 Stellen D0 bis D3 und Halteausgangssignale LE0 bis LE3) an die am durch O spezifizierten Ausgang angeschlossene 7-Segment-Anzeige aus. Der Wert von C spezifiziert die Stellenanzahl der Quellendaten (entweder 4 oder 8 Stellen) und die Logik für die Eingangs- und Ausgangsbaugruppen.

Der Befehl 7SEG(214) zeigt die 4- bzw. 8-stelligen Daten einmal alle 12 Zyklen an, startet dann neu und fährt mit der Anzeige der Daten fort.

Der Eine-Runde-Merker (Bit 08 von O bei Konvertierung von 4 Stellen, Bit 12 von O bei Konvertierung von 8 Stellen) wird alle 12 Zyklen, nachdem 7SEG(214) jedes der Halteausgangssignale auf EIN gesetzt hat, für einen Zyklus auf EIN gesetzt. Nachdem die 7-Segment-Daten in 12 Zyklen ausgegeben wurden, startet 7SEG(214) neu und konvertiert den aktuellen Inhalt der Quellenworte in den nächsten 12 Zyklen.

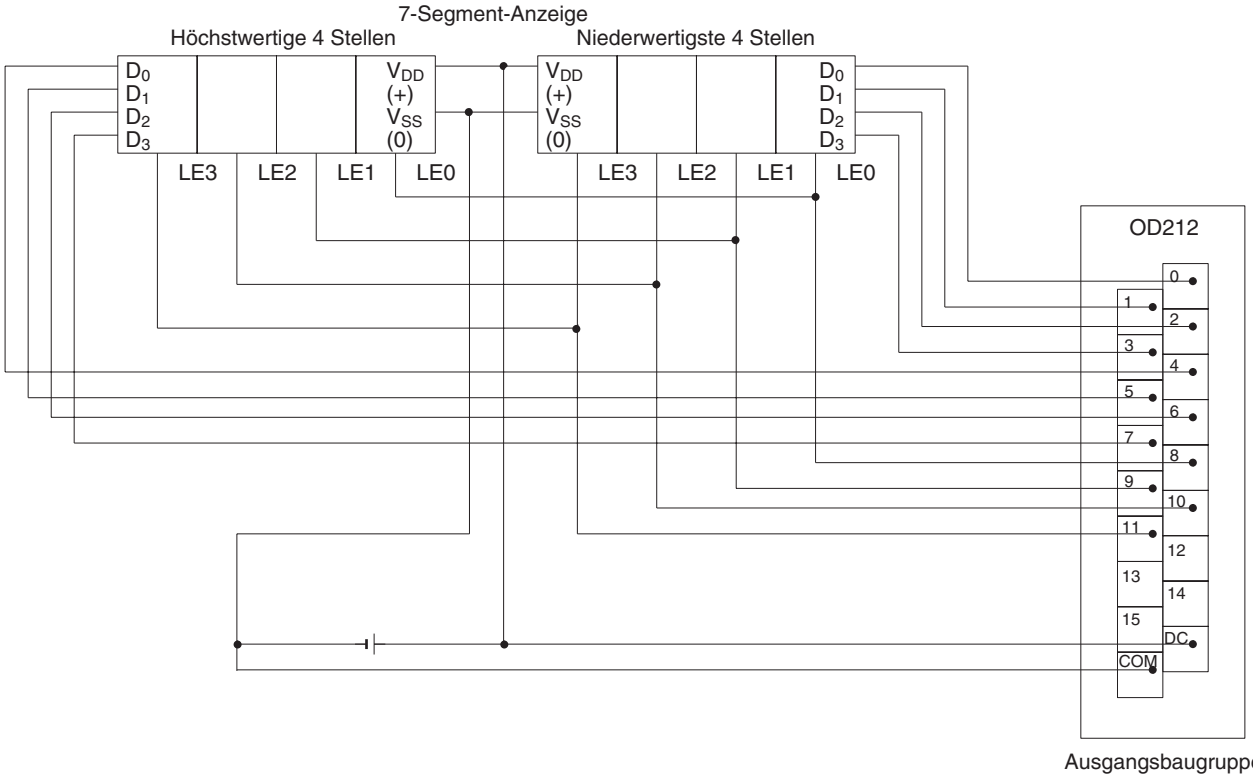
Bei der Ausführung beginnt der Befehl 7SEG(214) mit Halteausgang 0 am Anfang der Runde unabhängig davon, an welchem Punkt der letzte Befehl gestoppt wurde.

Selbst wenn die angeschlossene 7-Segment-Anzeige weniger als 4 bzw. 8 Stellen anzeigt, gibt der Befehl 7SEG(214) dennoch Daten für 4 bzw. 8 Stellen aus.

Externe Anschlüsse

Schließen Sie die 7-Segment-Anzeige wie in der folgenden Abbildung gezeigt an die Ausgangsbaugruppe an. In diesem Beispiel ist eine 8-stellige Anzeige dargestellt. Bei einer 4-stelligen Anzeige werden die Datenausgänge (D0 bis D3) an die Ausgänge 0 bis 3 angeschlossen und die Halteausgänge (LE0 bis LE3) an die Ausgänge 4 bis 7 angeschlossen. Ausgang 12 (bei 8-stelliger Anzeige) bzw. Ausgang 8 (bei 4-stelliger Anzeige) werden auf EIN geschaltet,

wenn Daten eine Runde ausgegeben wurden. Ein Anschluss an diesen Ausgang ist nicht notwendig, es sei denn, dies ist bei der Anwendung erforderlich.

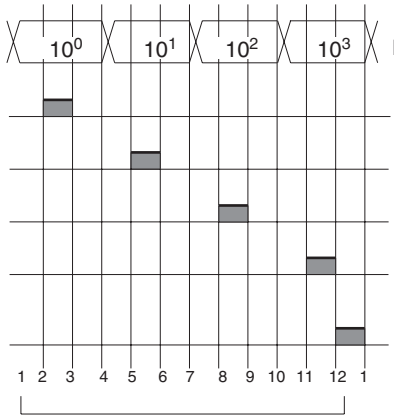


Ausgangsbaugruppe

Die Ein- und Ausgänge können an die folgenden Arten von E/A-Baugruppen und Multipoint-E/A-Baugruppen angeschlossen werde, sofern diese nicht in einem dezentralen SYSMAC BUS E/A-Block installiert sind.

- 4-stellige Anzeige: Transistor-Ausgangsbaugruppen mit min. 8 Ausgängen
- 8-stellige Anzeige: Transistor-Ausgangsbaugruppen mit min. 16 Ausgängen

Zeitablaufdiagramm

Funktion	Bit(s) in O		Ausgangsstatus (Daten- und Halteausgangslogik ist von C abhängig)
	(4 Stellen, 1 Block)	(4 Stellen, 2 Blöcke)	
Datenausgang	00 bis 03	00 bis 03 04 bis 07	
Halteausgang 0	04	08	
Halteausgang 1	05	09	
Halteausgang 2	06	10	
Halteausgang 3	07	11	
Eine-Runde-Merker	08	12	

Hinweis: 0 bis 3: Datenausgabe für Wort S
4 bis 7: Datenausgabe für Wort S+1

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	AUS

Vorsichtsmaßnahmen

Das Systemwort (D) darf nicht von anderen Befehlen gelesen oder beschrieben werden. Der Befehl 7SEG(214) funktioniert nicht einwandfrei, wenn durch einen anderen Befehl auf das Systemwort zugegriffen wird. Das Systemwort wird im ersten Zyklus bei Beginn der Programmausführung nicht durch 7SEG(214) initialisiert. Wenn der Befehl 7SEG(214) bereits ab dem ersten Programmzyklus ausgeführt werden soll, muss der Systemwort aus dem Programm heraus gelöscht werden.

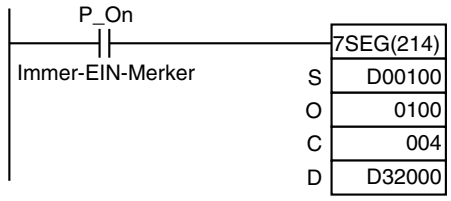
Der Befehl 7SEG(214) funktioniert nicht ordnungsgemäß, wenn nach der Ausführung von 7SEG(214) keine E/A-Aktualisierung bei an die Eingangsbaugruppe angeschlossener 7-Segment-Anzeige erfolgt. Dementsprechend darf die 7-Segment-Anzeige nicht an die folgenden Baugruppen angeschlossen werden.

- E/A-Baugruppen oder Multipoint-E/A-Baugruppen, die in einem dezentralen SYSMAC BUS E/A-Slave-Block installiert sind
- Kommunikations-Slaves (DeviceNet- oder CompoBus/S-Slaves)

Beispiel

In diesem Beispiel konvertiert der Befehl 7SEG(214) die 8 Stellen BCD-Daten in D00100 und D00101 und gibt die Daten über CIO 0100 an eine 7-Segment-Anzeige aus, die an eine CS1W-OD211 16-Punkt Transistor-Ausgangsbaugruppe angeschlossen ist.

Es werden 8 Stellen Daten ausgegeben und die Logik der 7-Segment-Anzeige ist mit der der Ausgangsbaugruppe identisch, sodass die Steuerdaten (C) auf 0004 gesetzt werden. D32000 wird als Systemwort (D) verwendet.



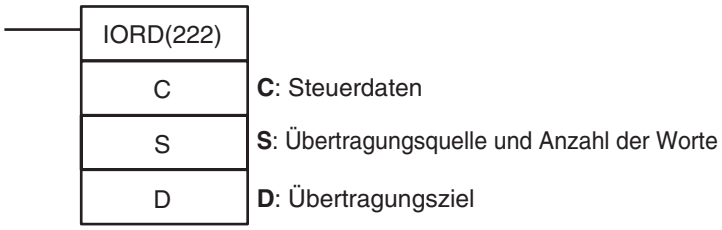
3-23-8 INTELLIGENT I/O READ: IORD(222)

Zweck

Liest den Inhalt des Speicherbereichs für eine Spezial-E/A-Baugruppe oder CPU-Bus-Baugruppe (siehe Hinweis).

Hinweis Bei CPU-Bus-Baugruppen gibt es Einschränkungen der Funktionalität. Weitere Informationen hierzu finden Sie unter *Einschränkungen* nachfolgend in diesem Abschnitt.

Kontaktplansymbol



Variationen

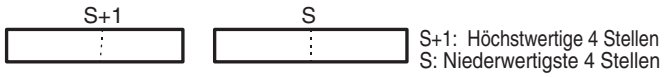
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	IORD(222)
	Ausführung einmal bei steigender Flanke	@IORD(222)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Von der Spezial-E/A- oder CPU-Bus-Baugruppe abhängig.
S: Spezial-E/A-Baugruppe: 0000 bis 005F hex (zur Spezifikation der Baugruppennummern 0 bis 95)
CPU-Bus-Baugruppe: 8000 bis 800F hex (zur Spezifikation der Baugruppennummern 0 bis F hex)
S+1: Anzahl der zu übertragenden Worte (0001 bis 0080 hex, von der Spezial-E/A- oder CPU-Bus-Baugruppe abhängig)

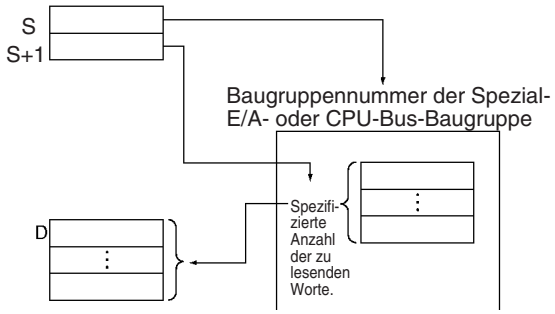


Operanden-Spezifikationen

Bereich	C	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W510	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H510	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A958	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	Nur spezifizierte Werte	---
Datenregister	DR0 bis DR15	---	
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl IORD(222) liest die in S+1 spezifizierte Anzahl von Worten aus dem Speicherbereich der Spezial-E/A- oder CPU-Bus-Baugruppe, deren Baugruppennummer durch S spezifiziert ist, und gibt die Daten nach D aus. Es können nur Spezial-E/A- oder CPU-Bus-Baugruppen spezifiziert werden, die in CPU-Racks oder E/A-Erweiterungsracks installiert sind. Informationen zu den spezifischen Details einzelner Baugruppen finden Sie im Bedienerhandbuch zu der Spezial-E/A- bzw. CPU-Bus-Baugruppe, von der die Daten gelesen werden sollen.



Einschränkungen Die folgenden Einschränkungen gelten beim Lesen von einer CPU-Bus-Baugruppe.

■ **Einschränkungen bei der CPU-Baugruppe**

CS1-H CPU-Baugruppen

Das Lesen von einer CPU-Bus-Baugruppe ist nur bei den folgenden Modellen von CPU-Baugruppen und nur bei CPU-Baugruppen möglich, die nach dem 18. April 2003 hergestellt wurden (Lot-Nummer ab 030418).

- CS1G-CPU□□H
- CS1H-CPU□□H

Das Herstellungsdatum kann anhand der Lot-Nummer geprüft werden, die an der Seite oder Unterseite der CPU-Baugruppe angegeben ist. Lot-Nummern geben das Herstellungsdatum im folgenden Format an:

JJMMTT nnnn

JJ = beide rechten Stellen der Jahreszahl, MM = Monat als Zahl, TT = Tag des Monats, nnnn = Seriennummer

CJ1-H, CJ1M und CS1D CPU-Baugruppen

Das Lesen von einer CPU-Bus-Baugruppe ist nur bei CPU-Baugruppen ab Baugruppenversion 2.0 möglich.

Hinweis Wenn der Befehl IORD(222) für eine CPU-Bus-Baugruppe ausgeführt wird, die von einer CPU-Baugruppe gesteuert wird, die die Verwendung des Befehls IORD(222) für CPU-Bus-Baugruppen nicht unterstützt, tritt ein Fehler auf und der Fehlermerker (ER) wird auf EIN gesetzt.

■ **Einschränkungen bei CX-Programmer**

Die Baugruppennummern von CPU-Bus-Baugruppen im Operanden S können mit CX-Programmer ab Version 3.0 spezifiziert werden.

Merker

Name	Bezeich-nung	Funktion
Fehler-merker	ER	EIN, wenn die Anzahl der zu übertragenden Worte (S) außerhalb des Bereichs von 0001 bis 0080 hex liegt. EIN, wenn die Baugruppennummer (S) außerhalb des Bereichs von 0000 bis 005F hex bzw. 8000 bis 800F hex liegt. EIN, wenn die spezifizierte Spezial-E/A-Baugruppe in einem dezentralen SYSMAC BUS-Rack installiert ist. EIN, wenn eine Spezial-E/A- oder CPU-Bus-Baugruppe spezifiziert ist, die den Befehl IORD(222) nicht unterstützt. EIN, wenn eine Spezial-E/A-Baugruppe spezifiziert ist, bei der ein Spezial-E/A-Baugruppen-Einstellungsfehler oder ein Spezial-E/A-Baugruppen-Fehler vorliegt. EIN, wenn eine CPU-Bus-Baugruppe spezifiziert ist, bei der ein CPU-Bus-Baugruppen-Einstellungsfehler oder ein CPU-Bus-Baugruppen-Fehler vorliegt. Bei CS1D CPU-Baugruppen: EIN, wenn die aktive und die Standby-CPU-Baugruppe nicht synchronisiert werden konnten. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn der Lesevorgang normal abgeschlossen wurde. AUS, wenn der Lesevorgang nicht normal abgeschlossen wurde.

Vorsichtsmaßnahmen

Der Gleich-Merker wird auf EIN gesetzt, wenn der Lesevorgang normal abgeschlossen wurde.

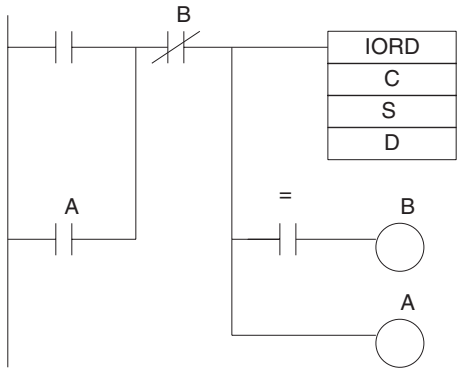
Der Gleich-Merker wird auf AUS gesetzt, wenn der Lesevorgang nicht normal abgeschlossen werden kann, da die Spezial-E/A-Baugruppe bzw. CPU-Bus-Baugruppe ausgelastet ist.

Wenn einer der folgenden Zustände eintritt, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

- Die Anzahl der zu übertragenden Worte (S) liegt außerhalb des Bereichs von 0001 bis 0080 hex.
- Die Baugruppennummer (S) liegt außerhalb des Bereichs von 0000 bis 005F hex bzw. 8000 bis 800F hex.
- Die spezifizierte Spezial-E/A-Baugruppe ist in einem dezentralen SYSMAC BUS-Rack installiert.
- Eine Spezial-E/A- oder CPU-Bus-Baugruppe wurde spezifiziert, die den Befehl IORD(222) nicht unterstützt.
- Eine Spezial-E/A-Baugruppe wurde spezifiziert, bei der ein Spezial-E/A-Baugruppen-Einstellungsfehler oder ein Spezial-E/A-Baugruppen-Fehler vorliegt.
- Eine CPU-Bus-Baugruppe wurde spezifiziert, bei der ein CPU-Bus-Baugruppen-Einstellungsfehler oder ein CPU-Bus-Baugruppen-Fehler vorliegt.

Die Ausführungsergebnisse bei Ausführung von IORD(222) werden in den Bedingungsmerkern wiedergegeben. Insbesondere wird der Gleich-Merker auf EIN gesetzt, wenn der Lesevorgang abgeschlossen ist. Geben Sie die Bedingungsmerker, wie den Gleich-Merker, mit Ausgangsverzweigung von den gleichen Eingangsbedingungen wie die des Befehls IORD(222) ein.

Wenn die Spezial-E/A- bzw. CPU-Bus-Baugruppe ausgelastet ist, wird der Lesevorgang nicht ausgeführt. Verwenden Sie den Gleich-Merker wie nachfolgend gezeigt zum Erstellen eines selbst erhaltenden Programms, damit der Befehl IORD(222) in jedem Zyklus ausgeführt wird, bis der Lesevorgang ausgeführt wurde.

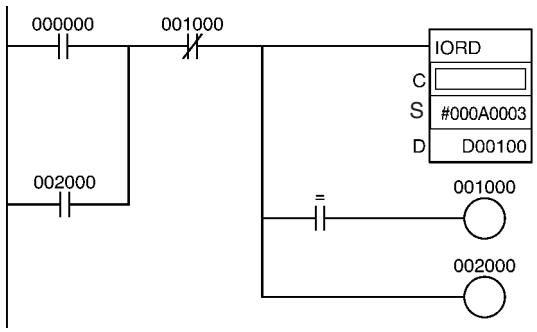


Wenn die Eingangsbedingung erfüllt ist, erfolgt das Selbsthalten durch Ausgang A und der Befehl IORD(222) wird in jedem Zyklus ausgeführt, bis der Gleich-Merker auf EIN gesetzt wird. Wenn der Lesevorgang abgeschlossen ist und der Gleich-Merker auf EIN gesetzt wird, wechselt Ausgang B auf EIN und die Selbsthaltung wird aufgehoben.

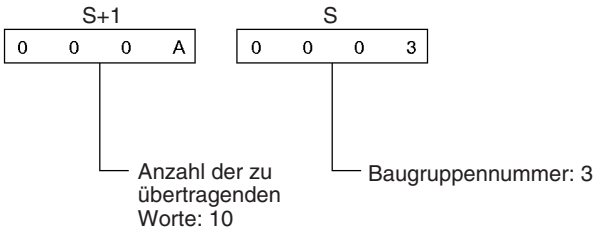
Achten Sie darauf, die Bedingungsmerker unmittelbar hinter IORD(222)-Befehlen zu platzieren, ohne dass andere Befehle dazwischen liegen. Wenn ein Bedingungsmerker hinter einem anderen Befehl platziert wird, wird er durch die Ergebnisse der Ausführung dieses Befehls beeinflusst.

Beispiel

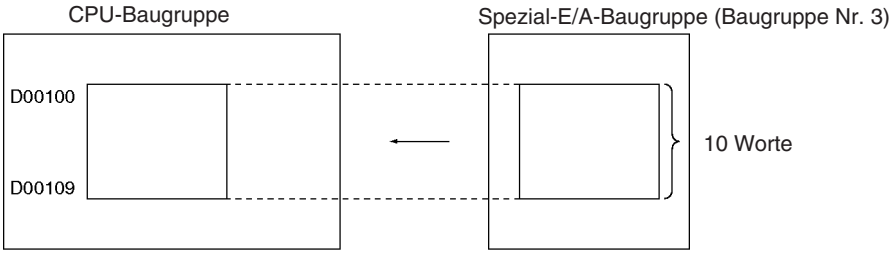
In diesem Beispiel wird der Befehl IORD(222) zum Lesen von Daten verwendet.



Wenn CIO 000000 auf EIN gesetzt ist, werden 10 Worte von der Spezial-E/A-Baugruppe mit Baugruppennummer 3 gelesen und in D00100 bis D00109 gespeichert.



Der Steuercode (C) ist von der Spezial-E/A-Baugruppe abhängig.



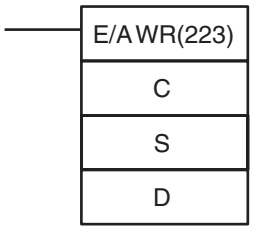
3-23-9 INTELLIGENT I/O WRITE: IOWR(223)

Zweck

Gibt den Inhalt des E/A-Speicherbereichs der CPU-Baugruppe an eine Spezial-E/A-Baugruppe oder CPU-Bus-Baugruppe aus (siehe Hinweis).

Hinweis Bei CPU-Bus-Baugruppen gibt es Einschränkungen der Funktionalität. Weitere Informationen hierzu finden Sie unter *Einschränkungen* nachfolgend in diesem Abschnitt.

Kontaktplansymbol



C: Steuerdaten

S: Übertragungsquelle und Anzahl der Worte

D: Übertragungsziel und Anzahl der Worte

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	IOWR(223)
	Ausführung einmal bei steigender Flanke	@IOWR(223)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

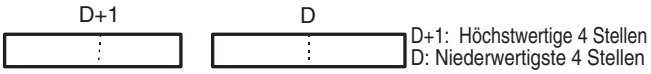
Operanden

C: Von der Spezial-E/A- oder CPU-Bus-Baugruppe abhängig.

D: Spezial-E/A-Baugruppe: 0000 bis 005F hex (zur Spezifikation der Baugruppennummern 0 bis 95)

CPU-Bus-Baugruppe: 8000 bis 800F hex (zur Spezifikation der Baugruppennummern 0 bis F hex)

D+1: Anzahl der zu übertragenden Worte (0000 bis 0080 hex, von der Spezial-E/A- oder CPU-Bus-Baugruppe abhängig)



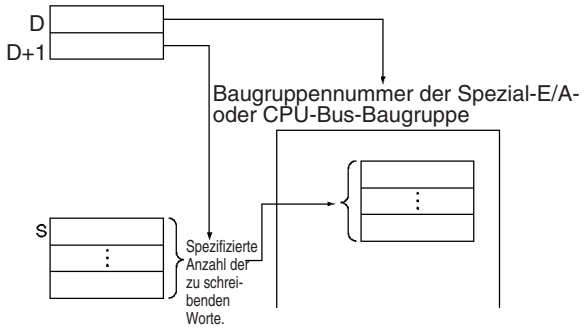
Operanden-Spezifikationen

Bereich	C	S	D
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6142
Arbeitsbereich	W000 bis W511		W000 bis W510
Haftmerkerbereich	H000 bis H511		H000 bis H510
Zusatz-Systembereich	A000 bis A959		A000 bis A958
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4094
Zählerbereich	C0000 bis C4095		C0000 bis C4094
DM-Bereich	D00000 bis D32767		D00000 bis D32766
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32766
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32766 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		

Bereich	C	S	D
Konstanten	#0000 bis #FFFF (binär)		Nur spezifizierte Werte
Datenregister	DR0 bis DR15	---	---
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl IOWR(223) liest die spezifizierte Anzahl von Worten (D) mit dem ersten Quellenwort (durch S spezifiziert) beginnend und gibt sie an die Spezial-E/A-Baugruppe oder CPU-Bus-Baugruppe mit der durch D spezifizierten Baugruppennummer aus. Es können nur Spezial-E/A- oder CPU-Bus-Baugruppen spezifiziert werden, die in CPU-Racks oder E/A-Erweiterungsracks installiert sind.



Einschränkungen

Die folgenden Einschränkungen gelten beim Lesen von einer CPU-Bus-Baugruppe.

■ **Einschränkungen bei der CPU-Baugruppe**

CS1-H CPU-Baugruppen

Das Schreiben in eine CPU-Bus-Baugruppe ist nur bei den folgenden Modellen von CPU-Baugruppen und nur bei CPU-Baugruppen möglich, die nach dem 18. April 2003 hergestellt wurden (Lot-Nummer ab 030418).

- CS1G-CPU□□H
- CS1H-CPU□□H

Das Herstellungsdatum kann anhand der Lot-Nummer geprüft werden, die an der Seite oder Unterseite der CPU-Baugruppe angegeben ist. Lot-Nummern geben das Herstellungsdatum im folgenden Format an:

JJMMTT nnnn

JJ = beide rechten Stellen der Jahreszahl, MM = Monat als Zahl, TT = Tag des Monats, nnnn = Seriennummer

CJ1-H, CJ1M und CS1D CPU-Baugruppen

Das Schreiben in eine CPU-Bus-Baugruppe ist nur bei CPU-Baugruppen ab Baugruppenversion 2.0 möglich.

Hinweis Wenn der Befehl IOWR(223) für eine CPU-Bus-Baugruppe ausgeführt wird, die von einer CPU-Baugruppe gesteuert wird, die die Verwendung des Befehls IOWR(223) für CPU-Bus-Baugruppen nicht unterstützt, tritt ein Fehler auf und der Fehlermerker (ER) wird auf EIN gesetzt.

■ **Einschränkungen bei CX-Programmer**

Die Baugruppennummern von CPU-Bus-Baugruppen im Operanden S können mit CX-Programmer ab Version 3.0 spezifiziert werden.

Merker

Name	Bezeich-nung	Funktion
Fehler-merker	ER	EIN, wenn die Anzahl der zu übertragenden Worte (D) außerhalb des Bereichs von 0001 bis 0080 hex liegt. EIN, wenn die Baugruppennummer (D) außerhalb des Bereichs von 0000 bis 005F hex bzw. 8000 bis 800F hex liegt. EIN, wenn S durch eine Konstante spezifiziert wird und Anzahl der zu übertragenden Worte (D+1) nicht 0001 hex ist. EIN, wenn die spezifizierte Spezial-E/A-Baugruppe in einem dezentralen SYSMAC BUS-Rack installiert ist. EIN, wenn eine Spezial-E/A- oder CPU-Bus-Baugruppe spezifiziert ist, die den Befehl IOWR(223) nicht unterstützt. EIN, wenn eine Spezial-E/A-Baugruppe spezifiziert ist, bei der ein Spezial-E/A-Baugruppen-Einstellungsfehler oder ein Spezial-E/A-Baugruppen-Fehler vorliegt. EIN, wenn eine CPU-Bus-Baugruppe spezifiziert ist, bei der ein CPU-Bus-Baugruppen-Einstellungsfehler oder ein CPU-Bus-Baugruppen-Fehler vorliegt. Bei CS1D CPU-Baugruppen: EIN, wenn die aktive und die Standby-CPU-Baugruppe nicht synchronisiert werden konnten. In allen anderen Fällen AUS.
Gleich-Merker	=	EIN, wenn der Schreibvorgang normal abgeschlossen wurde. AUS, wenn der Schreibvorgang nicht normal abgeschlossen wurde.

Vorsichtsmaßnahmen

Wenn für die Anzahl der zu übertragenden Worte (D+1) „0001“ spezifiziert ist, können die Daten für S durch eine Konstante spezifiziert werden. Wenn S durch eine Konstante spezifiziert wird und Anzahl der zu übertragenden Worte (D+1) nicht 0001 hex ist, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

Der Gleich-Merker wird auf EIN gesetzt, wenn der Schreibvorgang normal abgeschlossen wurde.

Der Gleich-Merker wird auf AUS gesetzt, wenn der Schreibvorgang nicht normal abgeschlossen werden kann, da die Spezial-E/A-Baugruppe bzw. CPU-Bus-Baugruppe ausgelastet ist.

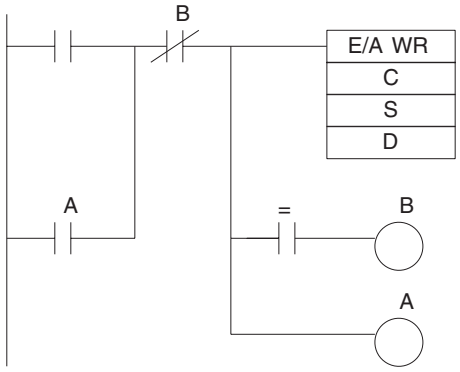
Wenn einer der folgenden Zustände eintritt, tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

- Bei der Spezial-E/A-Baugruppe liegt ein E/A-Baugruppen-Verifizierungsfehler, ein Spezial-E/A-Baugruppen-Einstellungenfehler oder ein Spezial-E/A-Baugruppen-Fehler vor.
- Bei der CPU-Bus-Baugruppe liegt ein E/A-Baugruppen-Verifizierungsfehler, ein CPU-Bus-Baugruppen-Einstellungsfehler oder ein CPU-Bus-Baugruppen-Fehler vor.
- Die Anzahl der zu übertragenden Worte (D) liegt außerhalb des Bereichs von 0001 bis 0080 hex.
- Die Baugruppennummer (D) liegt außerhalb des Bereichs von 0000 bis 005F hex bzw. 8000 bis 800F hex.
- Die spezifizierte Spezial-E/A-Baugruppe ist in einem dezentralen SYSMAC BUS-Block installiert.
- Eine Spezial-E/A- oder CPU-Bus-Baugruppe wurde spezifiziert, die den Befehl IOWR(223) nicht unterstützt.
- Eine Spezial-E/A-Baugruppe wurde spezifiziert, bei der ein Spezial-E/A-Baugruppen-Einstellungsfehler oder ein Spezial-E/A-Baugruppen-Fehler vorliegt.
- Eine CPU-Bus-Baugruppe wurde spezifiziert, bei der ein CPU-Bus-Baugruppen-Einstellungsfehler oder ein CPU-Bus-Baugruppen-Fehler vorliegt.

Die Ausführungsergebnisse bei Ausführung von IOWR(223) werden in den Bedingungsmerkern wiedergegeben. Insbesondere wird der Gleich-Merker auf EIN gesetzt, wenn der Schreibvorgang abgeschlossen ist. Geben Sie die Bedingungsmerker, wie den Gleich-Merker, mit Ausgangsverzweigung von den gleichen Eingangsbedingungen wie die des Befehls IOWR(223) ein.

Wenn die Spezial-E/A- bzw. CPU-Bus-Baugruppe ausgelastet ist, wird der Schreibvorgang nicht ausgeführt. Verwenden Sie den Gleich-Merker wie

nachfolgend gezeigt zum Erstellen eines selbst erhaltenden Programms, damit der Befehl IOWR(223) in jedem Zyklus ausgeführt wird, bis der Schreibvorgang ausgeführt wurde.

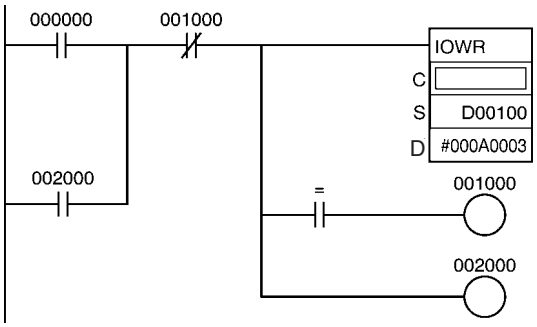


Wenn die Eingangsbedingung erfüllt ist, erfolgt das Selbsthalten durch Ausgang A und der Befehl IOWR(223) wird in jedem Zyklus ausgeführt, bis der Gleich-Merker auf EIN gesetzt wird. Wenn der Schreibvorgang abgeschlossen ist und der Gleich-Merker auf EIN gesetzt wird, wechselt Ausgang B auf EIN und die Selbsthaltung wird aufgehoben.

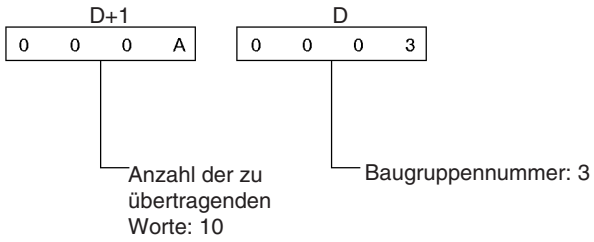
Achten Sie darauf, die Bedingungsmerker unmittelbar hinter IOWR(223)-Befehlen zu platzieren, ohne dass andere Befehle dazwischen liegen. Wenn ein Bedingungsmerker hinter einem anderen Befehl platziert wird, wird er durch die Ergebnisse der Ausführung dieses Befehls beeinflusst.

Beispiel

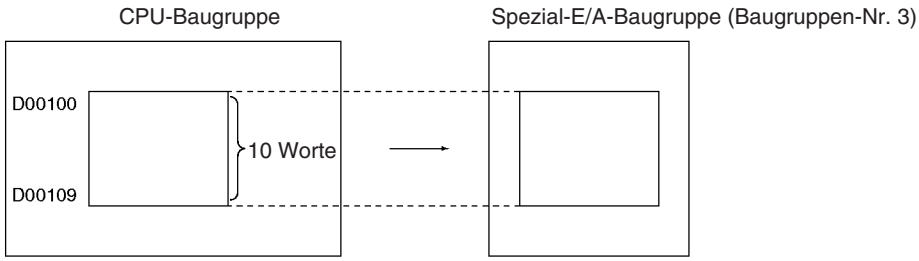
In diesem Beispiel wird der Befehl IOWR(223) zum Schreiben von Daten verwendet.



Wenn CIO 000000 auf EIN gesetzt ist, werden die 10 Worte in D00100 bis D00109 in die Spezial-E/A-Baugruppe geschrieben.



Der Steuercode (C) ist von der Spezial-E/A-Baugruppe abhängig.



3-23-10 CPU BUS UNIT I/O REFRESH: DLNK(226)

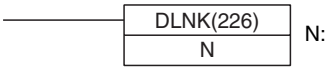
Zweck

Führt unmittelbar eine E/A-Aktualisierung bei der CPU-Bus-Baugruppe mit der spezifizierten Baugruppennummer durch. Die folgenden Daten werden aktualisiert:

- Die der CPU-Bus-Baugruppe zugewiesenen Worte in den CPU-Bus-Baugruppen-Bereichen der SPS (25 Worte im CIO-Bereich und 100 Worte im DM-Bereich)
- Aktualisierung spezifischer Daten von Baugruppen, wie bei Baugruppen, die Data-Links unterstützen

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	DLNK(226)
	Ausführung einmal bei steigender Flanke	@DLNK(226)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Baugruppennummer
Spezifiziert die Baugruppennummer der CPU-Bus-Baugruppe (0000 bis 000F hex bzw. 0 bis 15 dezimal).

Operanden-Spezifikationen

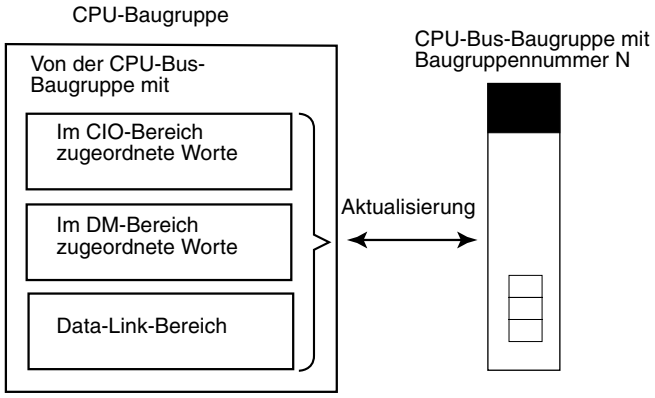
Bereich	N
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511
Haftmerkerbereich	H000 bis H511
Zusatz-Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	#0000 bis 000F (binär) oder 0 bis 15 (dezimal)
Datenregister	DR0 bis DR15
Indexregister	---
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl DLNK(226) führt eine unmittelbare E/A-Aktualisierung bei der CPU-Bus-Baugruppe mit der spezifizierten Baugruppennummer durch. Die nachfolgend aufgeführten Daten werden aktualisiert. Informationen über Details zu den Ausführungsbedingungen, die zur unmittelbaren Aktualisierung zu verwenden sind, finden Sie unten unter *Vorsichtsmaßnahmen*.

1. Die der CPU-Bus-Baugruppe zugewiesenen Worte in den CPU-Bus-Baugruppen-Bereichen der SPS (25 Worte im CIO-Bereich und 100 Worte im DM-Bereich)
2. Für die CPU-Bus-Baugruppe spezifische Daten, wie Data-Link-Daten oder Daten der dezentralen DeviceNet-E/A-Kommunikation (die Aktualisierung erfolgt zusammen mit den Daten in den CPU-Bus-Baugruppen-Bereichen)

CPU-Bus-Baugruppe	Aktualisierung für die spezifischen Daten der Baugruppe
Controller-Link-Baugruppe und SYSMAC LINK-Baugruppe	Data-Link-Aktualisierung
DeviceNet-Baugruppe (Ausgenommen C200H DeviceNet-Master-Baugruppen.)	Aktualisierung der dezentralen E/A-Kommunikation



In der folgenden Tabelle ist zusammengestellt, wie sich der Befehl DLNK(226) vom Befehl IORF(097) unterscheidet.

Befehl	Funktion
DLNK(226)	• E/A-Aktualisierung des CS1 CPU-Bus-Baugruppen-Bereichs in CIO-Bereich (25 Worte) • E/A-Aktualisierung des CS1 CPU-Bus-Baugruppen-Bereichs in DM-Bereich (100 Worte) • Aktualisierung von spezifischen Daten für die CPU-Bus-Baugruppe, wie Data-Link-Daten oder Daten der dezentralen DeviceNet-E/A-Kommunikation
IORF(097)	• E/A-Aktualisierung von Worten, die durch E/A-Baugruppen verwendet werden • E/A-Aktualisierung der 10 CIO-Worte, die einer Spezial-E/A-Baugruppe zugewiesen sind

DLNK(226) aktualisiert Daten zwischen der CPU-Baugruppe und der spezifizierten CPU-Bus-Baugruppe. Es gibt zwei Besonderheiten, die bei der Verwendung des Befehls DLNK(226) berücksichtigt werden müssen:

1,2,3...

1. Beim Austausch von Daten über einen Data-Link oder dezentrale DeviceNet-E/A-Kommunikation erfolgt während der Ausführung von DLNK(226) kein gleichzeitiger Datenaustausch mit anderen Baugruppen. Der Datenaustausch erfolgt, wenn der Netzwerk-Kommunikationszyklus die fragliche Baugruppe erreicht und Daten mit dieser Baugruppe ausgetauscht werden. Demzufolge kann der tatsächliche Datenaustausch um bis zu eine Zykluszeit der Netzwerkkommunikation verzögert erfolgen.
2. Der Befehl DLNK(226) kann keine E/A-Aktualisierung mit einer CPU-Bus-Baugruppe durchführen, wenn diese Baugruppe zu dem Zeitpunkt Daten austauscht. Wenn DLNK(226) zu häufig ausgeführt wird, erfolgt keine E/A-Aktualisierung. Es wird daher empfohlen, zwischen der Ausführung von zwei DLNK(226)-Befehlen eine Verzögerung einzuhalten, die länger als die Kommunikationszykluszeit ist.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die spezifizierte Baugruppennummer nicht zwischen 0000 und 000F hex (zwischen 0 und 15 dezimal) liegt. EIN, wenn in der SPS keine CPU-Bus-Baugruppe mit der spezifizierten Baugruppennummer vorhanden ist. Bei CS1D CPU-Baugruppen: EIN, wenn die aktive und die Standby-CPU-Baugruppe nicht synchronisiert werden konnten. In allen anderen Fällen AUS.
Gleich-Merker	=	AUS, wenn die E/A-Aktualisierung nicht erfolgen konnte, da die CPU-Bus-Baugruppe Daten aktualisiert hat. AUS, wenn in der spezifizierten CPU-Bus-Baugruppe ein CPU-Bus-Baugruppen-Fehler oder ein CPU-Bus-Baugruppen-Einstellungsfehler vorlag. AUS, wenn DLNK(226) in einer Interrupt-Task ausgeführt wurde, es zu einem Konflikt mit der zyklischen E/A-Aktualisierung gekommen ist und so eine überlagerte Aktualisierung aufgetreten ist. EIN, wenn die E/A-Aktualisierung normal abgeschlossen wurde.

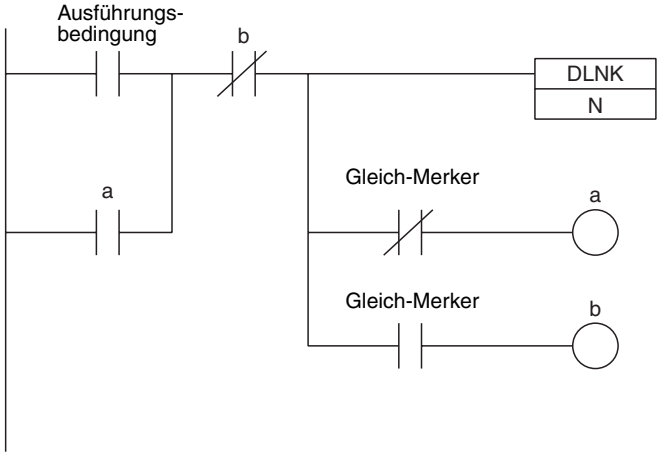
Vorsichtsmaßnahmen

Die E/A-Aktualisierung wird nicht durchgeführt, wenn in der spezifizierten CPU-Bus-Baugruppe ein CPU-Bus-Baugruppen-Fehler (A40207) oder ein CPU-Bus-Baugruppen-Einstellungsfehler (A40203) aufgetreten ist.

Die E/A-Aktualisierung wird gestoppt, wenn während der Durchführung der E/A-Aktualisierung durch DLNK(226) ein E/A-Bus-Fehler auftritt.

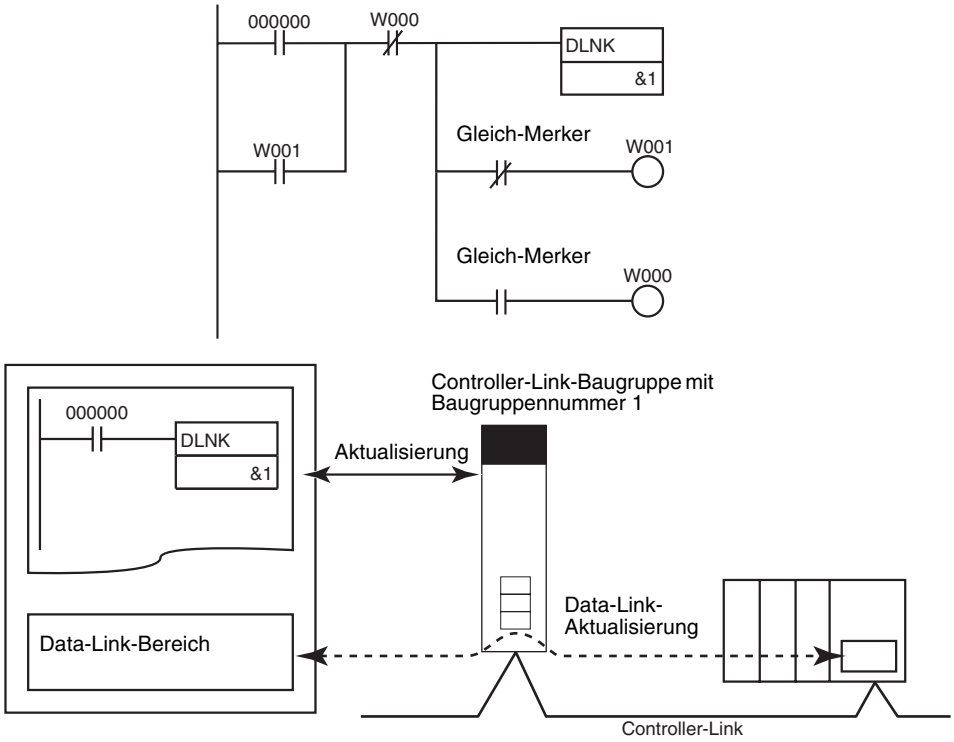
DLNK(226) aktualisiert Daten zwischen der CPU-Baugruppe und der spezifizierten CPU-Bus-Baugruppe. Für den Datenaustausch mit der CPU-Bus-Baugruppe ist eine gewisse Zeit erforderlich (beispielsweise über einen Data-Link mit einer Controller-Link-Baugruppe).

Wenn die spezifizierte CPU-Bus-Baugruppe einen Datenaustausch durchführt, wird der Befehl DLNK(226) nicht ausgeführt und der Gleich-Merker wird auf AUS gesetzt. Es ist empfehlenswert, die Ausführungsbedingungen wie nachfolgend gezeigt zu programmieren, damit automatisch Wiederholversuche für die Ausführung von DLNK(226) erfolgen.



Beispiel

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, führt der Befehl DLNK(226) ein unmittelbare E/A-Aktualisierung (in diesem Fall eine Data-Link-Aktualisierung innerhalb der SPS) bei der CPU-Bus-Baugruppe mit Baugruppennummer 1 (in diesem Fall eine Controller-Link-Baugruppe) durch. Wenn die E/A-Aktualisierung nicht durchgeführt werden kann, da die Controller-Link-Baugruppe zu diesem Zeitpunkt Daten aktualisiert, wird der Gleich-Merker auf AUS gesetzt, wodurch W001 auf EIN gesetzt wird, damit die Ausführung des Befehls im nächsten Zyklus erneut versucht wird. Wenn die E/A-Aktualisierung normal abgeschlossen ist, wird der Gleich-Merker auf EIN gesetzt und im nächsten Zyklus erfolgt kein erneuter Versuch der Ausführung des Befehls.

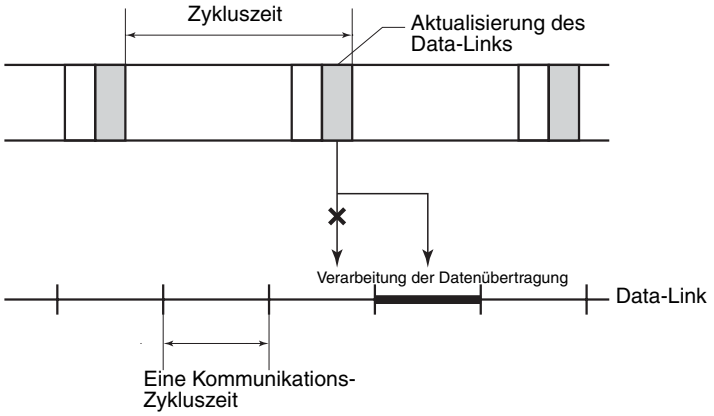


In diesem Beispiel ist der tatsächliche zeitliche Ablauf der Aktualisierung des Data-Link-Bereichs wie folgt:

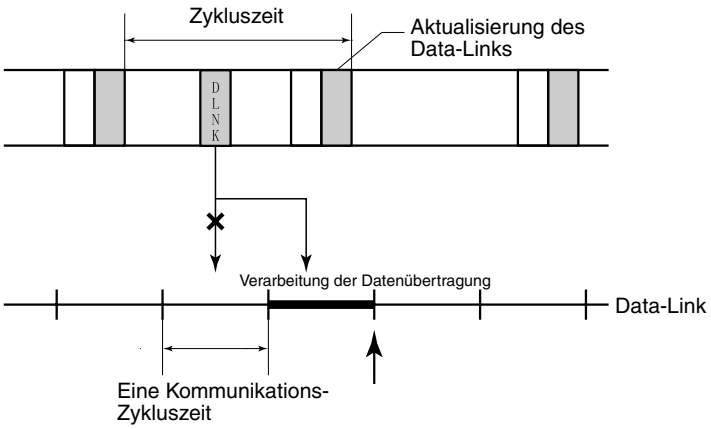
- Bei Übertragung: Daten werden beim nächsten Mal, wenn das Token-Recht erlangt wird, über das Netzwerk übertragen. (Die Datenübertragung erfolgt um bis zu eine Kommunikationszykluszeit verzögert.)
- Beim Empfangen: Die eingegebenen Daten wurden beim letzten Mal, als das Token-Recht erlangt wurde, über das Netzwerk empfangen (Der Datenempfang erfolgt um bis zu eine Kommunikationszykluszeit verzögert.)

Beispiele für die Verarbeitung der Datenübertragung:

- Übertragen der Daten von der vorherigen E/A-Aktualisierung



• Übertragen von Daten durch Ausführung von DLNK(226)



3-24 Befehle für serielle Kommunikation

In diesem Abschnitt werden die Befehle für die serielle Kommunikation beschrieben.

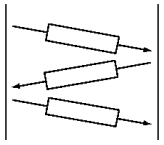
Befehl	AWL	Funktionscode	Seite
PROTOCOL MACRO	PMCR	260	928
TRANSMIT	TXD	236	937
RECEIVE	RXD	235	944
TRANSMIT VIA SERIAL COMMUNICATIONS UNIT	TXDU	256	952
RECEIVE VIA SERIAL COMMUNICATIONS UNIT	RXDU	255	960
CHANGE SERIAL PORT SETUP	STUP	237	968

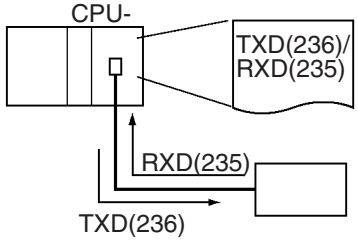
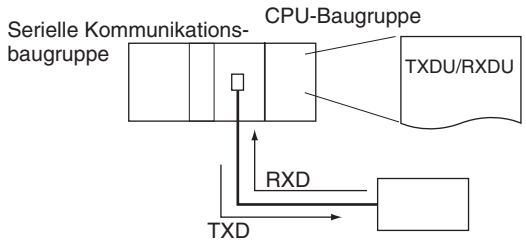
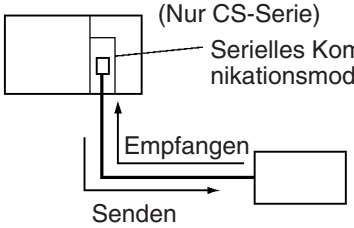
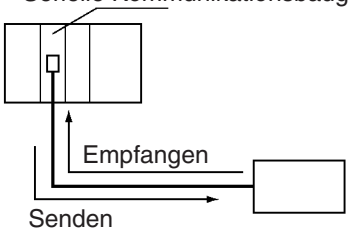
3-24-1 Serielle Kommunikation

Es gibt zwei Arten von Befehlen für die serielle Kommunikation. Die Befehle TXD(236), RXD(235), TXDU(256) und RXDU(255) senden bzw. empfangen Daten mittels protokollfreier Kommunikation mit einem externen Gerät. Der Befehl PMCR(260) sendet bzw. empfängt Daten durch Kommunikation mit einem externen Gerät, bei der anwenderdefinierte Protokolle eingesetzt werden. Die Unterschiede sind in den folgenden Tabellen zusammengestellt.

Hinweis

1. Die Befehle TXD(236) und RXD(235) übertragen Daten nur über die integrierte serielle Schnittstelle der CPU-Baugruppe oder die serielle Schnittstelle eines seriellen Kommunikationsmoduls (ab Version 1.2).
2. Die Befehle TXDU(256) und RXDU(255) übertragen Daten nur über eine serielle Kommunikationsbaugruppe (ab Version 1.2).

Befehle	Kommunikations-Frames	Funktion
TXD(236), RXD(235), TXDU(256) und RXDU(255)	Es kann eine beliebige der folgenden Arten verwendet werden. Kein Start- oder Endcode: Daten Start- und Endcode: ST | Daten | ED Nur Startcode: ST | Daten CR+LF-Endcode: Daten | CR | LF Nur Endcode: Daten | ED Start- und CR+LF-Endcode: ST | Daten | CR | LF	Senden oder Empfangen von Daten in nur einer Richtung. Eine Sendeverzögerung kann eingestellt werden.
PMCR(260)	Die folgenden Arten von Frames (Meldungen) können erstellt werden, um den Anforderungen des externen Geräts zu entsprechen. Kopfzeile | Adresse | Daten | Fehlerprüfung | Endezeichen Kommunikationsschritte können erstellt werden.  E/A-Speicher Lesen/Schreiben	Für das Senden und Empfangen können bis zu 16 Schritte definiert werden. Auf Grundlage der Antworten können Schritte geändert werden und eine Wiederholungsverarbeitung erfolgen. Kommunikations-Überwachungszeiten können eingestellt werden. Symbole können in der SPS gelesen/ geschrieben werden. Wiederholungssymbole können verwendet werden. Sonstiges

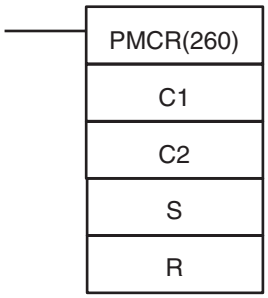
Befehle	Modus	Kommunikationsschnittstellen
TXD(236) und RXD(235)	Protokollfrei	Serielle Schnittstelle der CPU-Baugruppe oder serielles Kommunikationsmodul  Die Befehle TXD(236) und RXD(235) verwenden die serielle Schnittstellen der CPU-Baugruppe oder des seriellen Kommunikationsmoduls (ab Version 1.2).
TXDU(256) und RXDU(255)	Protokollfrei	Serielle Schnittstelle der seriellen Kommunikationsbaugruppe (ab Version 1.2) 
PMCR(260)	Protokoll-Makro	Serielles Kommunikationsmodul (Nur CS-Serie)  Serielle Kommunikationsbaugruppe 

3-24-2 PROTOCOL MACRO: PMCR(260)

Zweck

Ruft eine bei einem seriellen Kommunikationsmodul (nur CS-Serie) oder einer seriellen Kommunikationsbaugruppe registrierte Kommunikationssequenz auf und führt diese aus.

Kontaktplansymbol



C1: Steuerwort 1

C2: Steuerwort 2

S: Erstes Sendewort

R: Erstes Empfangswort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	PMCR(260)
	Ausführung einmal bei steigender Flanke	@PMCR(260)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

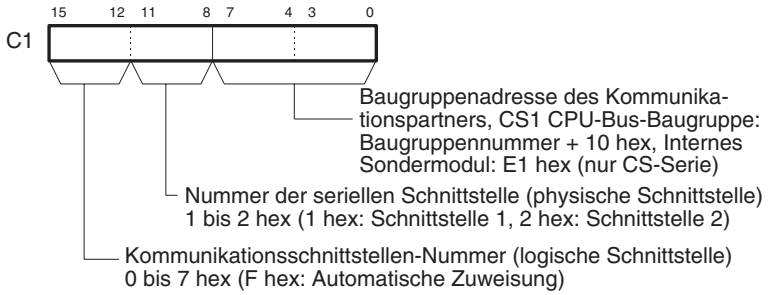
Anwendbarkeit auf Programmbereiche

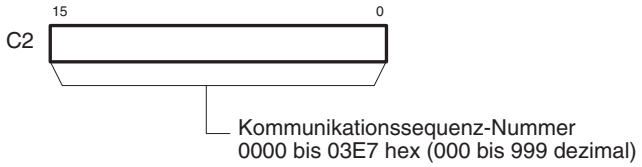
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C1: Steuerwort 1; C2: Steuerwort 2

Die Inhalte der beiden Steuerworte sind in der folgenden Abbildung angegeben.



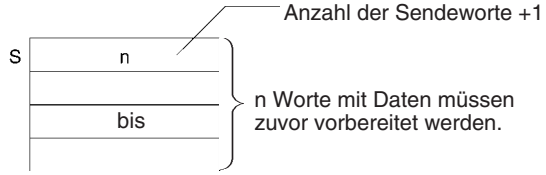


Hinweis Einzelheiten über die Verwendung der automatischen Zuweisung der Kommunikationsschnittstellen-Nummer (logische Schnittstelle) finden Sie unter *Automatische Zuweisung von Kommunikationsschnittstellen* auf Seite 979.

S: Erstes Sendewort und Sendebereich

Das erste der Worte, die zum Senden von Daten erforderlich sind, wird spezifiziert. S enthält die Anzahl der zu sendenden Worte +1 (d. h., das Wort S wird mitgezählt), die Sendedaten beginnen in S+1. Die Anzahl der Worte kann zwischen 0000 und 00FA hex (0 und 250 dezimal) eingestellt werden.

Wenn in der Ausführungssequenz kein Operand, wie ein direktes oder verknüpftes Wort, spezifiziert ist, spezifizieren Sie für S die Konstante #0000. Wenn eine Wortadresse oder ein Register spezifiziert ist, müssen die Daten in dem Wort bzw. Register stets 0000 sein. Wenn eine andere Konstante oder Wortadresse spezifiziert wird, tritt ein Fehler auf, der Fehlermerker wird auf EIN gesetzt und der Befehl PMCR(260) wird nicht ausgeführt.



R: Erstes Empfangswort und Empfangsbereich

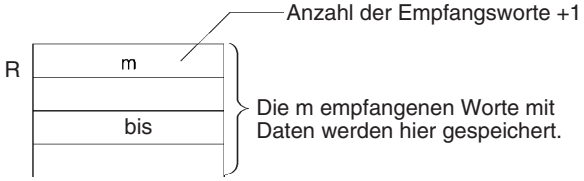
Empfangene Daten werden automatisch in den mit R+1 beginnenden Worten gespeichert. Die Anzahl der empfangenen Worte plus das Wort R (d. h., R wird mitgezählt) wird mit einem gültigen Wertebereich von 0000 bis 00FA hex (0 bis 250 dezimal) in R geschrieben.

Einstellung vor Ausführung von PMCR

Stellen Sie die durch m spezifizierten Daten (mit D beginnend) als die Anfangsdaten für dem Empfangspuffer (Sicherungsdaten bei Empfangsfehler) ein. Die Daten für m können zwischen 0002 und 00FA (hex) (2 und 255) eingestellt werden. Wenn 0000 (hex) oder 0001 (hex) für m spezifiziert wird, wird der Anfangswert des Empfangspuffers auf 0 gelöscht.

Spezifizieren Sie stets eine Wortadresse für R, selbst wenn keine Empfangsdaten vorhanden sind. Wenn eine Konstante spezifiziert wird, tritt ein Fehler auf, der Fehlermerker wird auf EIN gesetzt und der Befehl PMCR(260) wird nicht ausgeführt. Sind keine Empfangsdaten vorhanden, wird R nicht verwendet und kann für andere Zwecke genutzt werden.

Wenn in der Ausführungssequenz kein Operand, wie ein direktes oder verknüpftes Wort, spezifiziert ist, spezifizieren Sie für R die Konstante #0000. Wenn eine Wortadresse oder ein Register spezifiziert ist, müssen die Daten in dem Wort bzw. Register stets 0000 sein.



Operanden-Spezifikationen

Bereich	C1	C2	S	R
CIO-Bereich	CIO 0000 bis CIO 6143			
Arbeitsbereich	W000 bis W511			
Haftmerkerbereich	H000 bis H511			
Zusatz-Systembereich	A000 bis A447 A448 bis A959			A448 bis A959
Zeitgeberbereich	T0000 bis T4095			
Zählerbereich	C0000 bis C4095			
DM-Bereich	D00000 bis D32767			
EM-Bereich ohne Bank	E00000 bis E32767			
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)			
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)			

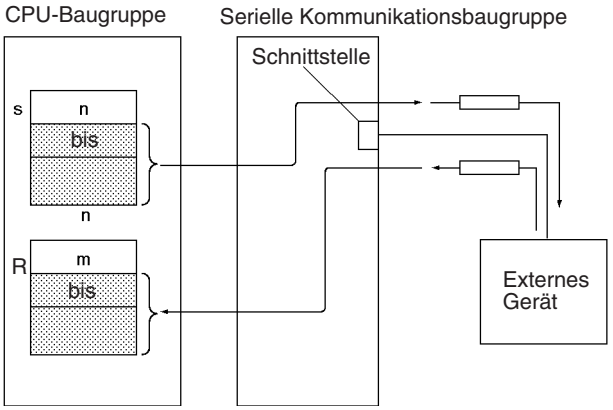
Bereich	C1	C2	S	R
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)			
Konstanten	Nur spezifizierte Werte	0000 bis 03E7 hex (0 bis 999)	#0000 (binär)	
Datenregister	DR0 bis DR15		---	
Indexregister	---			
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15			

Beschreibung

Der Befehl PMCR(260) führt die in C2 spezifizierte Kommunikationssequenz für die in den Bit 0 bis 7 von C1 spezifizierte Baugruppenadresse aus und verwendet dabei die in den Bits 12 bis 15 von C1 spezifizierte logische Schnittstelle und die in den Bits 8 bis 11 von C1 spezifizierte physische Schnittstelle.

Wenn ein Symbol als Operand für eine Sendemeldung spezifiziert ist, wird die in S spezifizierte Anzahl der Sendeworte und die mit S+1 beginnenden Adressen als Sendebereich verwendet. Wenn ein Symbol als Operand für eine Empfangsmeldung spezifiziert ist, werden die empfangenen Daten in die mit R+1 beginnenden Speicheradressen geschrieben und die Anzahl der empfangenen Worte wird automatisch in R geschrieben, wenn die Überragung erfolgreich war.

Ist die Überragung nicht erfolgreich, werden die vor Ausführung von PMCR(260) gesetzten Daten (ab R+1) aus dem Empfangspuffer gelesen und erneut in die mit R+1 beginnenden Worte geschrieben.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn der Kommunikationsschnittstellen-Frei-Merker für die spezifizierte logische Schnittstelle bei der Ausführung von PMCR(260) auf AUS gesetzt ist. EIN, wenn C1 nicht in den spezifizierten Bereichen liegt. (Der Fehlermerker wird nicht auf EIN gesetzt, wenn die Daten in C2 außerhalb der spezifizierten Bereiche liegen. Der Endcode wird im Kommunikationsschnittstellen-Abschlusscode (A203 bis A210) im Zusatz-Systembereich gespeichert.) EIN, wenn die in S bzw. R spezifizierte Anzahl von Worten 249 übersteigt (bei Spezifikation von Worten). In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

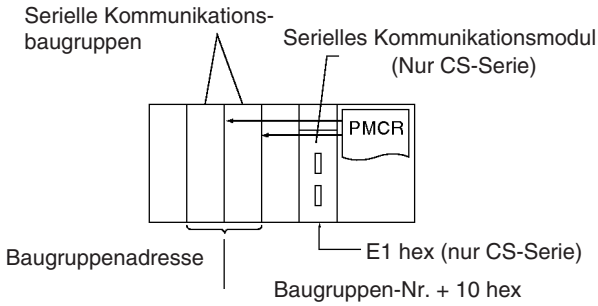
Die Daten in dem durch S spezifizierten Sendebereich werden tatsächlich unter Verwendung der Symbolleseoption, R( ), in einer Sendemeldung gesendet.

Die Daten werden tatsächlich unter Verwendung der Symbolschreiboption, W( ), in einer Empfangsmeldung in den durch R spezifizierten Empfangsbereich empfangen.

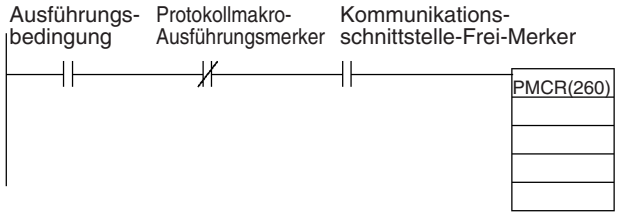
Informationen zu den Verfahren für das Spezifizieren der Symbole R( ) and W( ) finden Sie im *CX-Protocol Bedienerhandbuch (W344)*.

Der Befehl PMCR(260) kann für eine serielle Kommunikationsschnittstelle eines seriellen Kommunikationsmoduls (nur CS-Serie) oder einer seriellen Kommunikationsbaugruppe ausgeführt werden. Im CPU-Rack und in E/A-Erweiterungsracks können bis zu 16 serielle Kommunikationsbaugruppen installiert werden. Die Baugruppenadresse des Kommunikationspartners muss in den Bits 0 bis 7 von C1 eingestellt werden, um zu spezifizieren, welche Baugruppe/welches Modul verwendet werden soll. Die Nummer der seriellen Schnittstelle muss in den Bits 8 bis 11 von C1 eingestellt werden. Die Baugruppenadressen werden wie in der folgenden Tabelle angegeben spezifiziert.

Baugruppe/Modul	Baugruppenadresse
Serielles Kommunikationsmodul (Nur CS-Serie)	E1 hex
Serielle Kommunikationsbaugruppe	Baugruppennummer + 10 hex

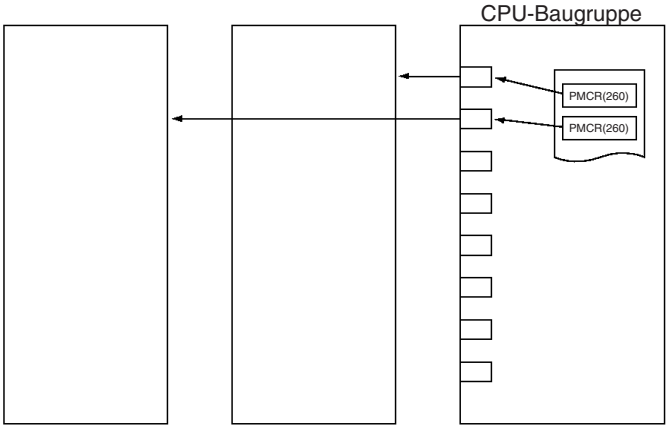


Der entsprechende Protokollmakro-Ausführungsmerker wird bei Beginn der Ausführung von PMCR(260) auf EIN gesetzt. Dieser Merker wird wieder auf AUS gesetzt, nachdem die Kommunikationssequenz abgeschlossen wurde und die Daten in den spezifizierten Empfangsbereich geschrieben wurden. Bei jeder Ausführung von PMCR(260) sollte ein Öffner-Eingang für den entsprechenden Protokollmakro-Ausführungsmerker als Teil der Ausführungsbedingung verwendet werden, um sicherzustellen, dass über die gleiche physische Schnittstelle keine zwei Kommunikationssequenzen gleichzeitig ausgeführt werden. Nachfolgend ist ein Beispiel dargestellt.



Auch die Befehl SEND(090), RECV(098) und CMND(490) verwenden die logischen Schnittstellen 0 bis 7 zur Ausführung von Kommunikationssequenzen über serielle Kommunikationsbaugruppen und -module (unter interner Verwendung von FINS-Befehlen). Der Befehl PMCR(260) kann nicht für eine logische Schnittstelle ausgeführt werden, die bereits durch einen der Befehle SEND(090), RECV(098), CMND(490) oder PMCR(260) verwendet wird. Um

zu vermeiden, dass mehrere Kommunikationssequenzen gleichzeitig für die gleiche logische Schnittstelle ausgeführt werden, sollte der entsprechende Kommunikationsschnittstelle-Frei-Merker (A20200 bis A20207) als Schließer-Eingang in der Ausführungsbedingung für PMCR(260) verwendet werden, wie in der Abbildung oben dargestellt.



In den folgenden Fällen wird der Fehlermerker auf EIN gesetzt.

- Der entsprechende Kommunikationsschnittstelle-Frei-Merker der spezifizierten logische Schnittstelle (0 bis 7) ist bei der Ausführung von PMCR(260) auf AUS gesetzt.
- C1 liegt nicht innerhalb der spezifizierten Bereiche.

Festlegung des Empfangsbereichs

Vor der Ausführung von PMCR(260) muss der Anwender für den Fall von Empfangsverarbeitungsfehlern Backup-Daten im Empfangsbereich einstellen. Sobald PMCR(260) ausgeführt wird, werden die Daten im Empfangspuffer im Empfangsbereich gespeichert. Ein Beispiel für die Anwendung von Backup-Daten: Im Voraus wird ein bestimmter Wert (Backup-Daten) eingestellt, damit beim Auftreten einer Übertragungsstörung während der Ausführung des Protokolls zum Lesen des Istwerts eines Controllers nicht Null als Istwert gelesen wird.

Zugehörige Merker und Worte

Die folgenden Merker und Worte können bei Bedarf bei der Ausführung von PMCR(260) verwendet werden.

Zusatz-Systembereich

Name	Adresse	Inhalt
Kommunikations-schnittstelle-Frei-Merker	A20200 bis A20207	EIN, wenn die Netzwerkkommunikation aktiviert ist (einschließlich PMCR(260). Die Bits 00 bis 07 entsprechen den logischen Schnittstellen 0 bis 7. Ein Kommunikationsschnittstelle-Frei-Merker wird beim Start der Netzwerkkommunikation auf AUS gesetzt und nach deren Abschluss wieder auf EIN gesetzt (unabhängig davon, ob die Kommunikation normal oder mit einem Fehler endete).

Name	Adresse	Inhalt
Kommunikations-schnittstellen-Fehlermerker	A21900 bis A21907	EIN, wenn bei der Netzwerkkommunikation ein Fehler auftritt. Die Bits 00 bis 07 entsprechen den logischen Schnittstellen 0 bis 7. Der Status des Merkers bleibt bis zum Start der nächsten Netzwerkkommunikation erhalten. Der Merker wird bei erneutem Start der Kommunikation selbst dann auf AUS gesetzt, wenn bei der vorherigen Kommunikation ein Fehler auftrat.
Kommunikations-schnittstellen-Abschluss-Codes	A203 bis A210	Enthält den bei der Durchführung der Netzwerkkommunikation gespeicherten Abschlusscode. Die Worte A203 bis A210 entsprechen den logischen Schnittstellen 0 bis 7. Während der Kommunikationsbefehl ausgeführt wird, ist der Abschlusscode auf 00 gesetzt. Der neue Antwortcode wird gespeichert, wenn die Ausführung abgeschlossen ist. Der Inhalt dieser Worte wird beim Start des Betriebs gelöscht.

Kommunikationsantworten

Code	Inhalt
1106 (hex)	Keine entsprechende Programmnummer Die spezifizierte Sende-/Empfangssequenz-Nummer ist nicht registriert. Ändern Sie die Sende-/Empfangssequenz-Nummer oder fügen Sie die Nummer mit CX-Programmer hinzu.
2201 (hex)	Aufgrund der Protokollausführung nicht betriebsbereit Da bereits ein Protokollmakro ausgeführt wurde, wird keine weitere Ausführung akzeptiert. Fügen Sie in das Programm ein Öffner-Bedingung für den Protokollmakro-Ausführungsmerker ein.
2202 (hex)	Durch Anhalten nicht betriebsbereit Da das Protokollmakro gewechselt wird, wird keine weitere Ausführung akzeptiert. Fügen Sie in das Programm ein Öffner-Bedingung für den Änderungsmerker - serielle Einstellung ein.
2401 (hex)	Keine Registrierungstabelle In den Protokollmakrodaten ist ein Fehler aufgetreten oder Daten werden übertragen. Übertragen Sie die Protokollmakrodaten mit Hilfe von CX-Programmer.
Sonstiges	Informationen zu weiteren Antwortcodes finden Sie im *CS/CJ-Serie Kommunikationsbefehle Referenzhandbuch (W342)*.

Bereich für interne Sondermodule (Nur CS-Serie)

Name	Adresse	Inhalt
Schnittstelle 1 - Protokollmakro-Ausführungsmerker	CIO 190915	EIN, während PMCR(260) ausgeführt wird. Der Merker bleibt auf AUS gesetzt, wenn die Ausführung fehlschlägt. Der Merker wird bei Abschluss der Kommunikationssequenz (bei Ende oder Abbruch) auf AUS gesetzt.
Schnittstelle 2 - Protokollmakro-Ausführungsmerker	CIO 191915	

CPU-Bus-Baugruppenbereich

n = 1500 + 25 x Baugruppennummer

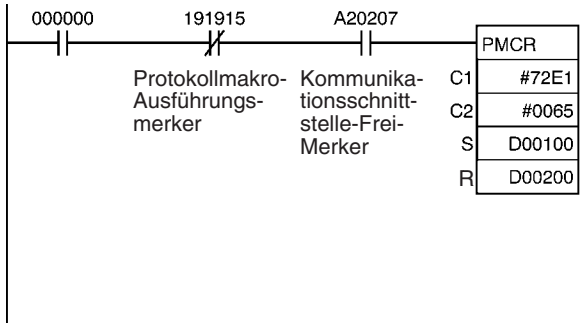
Name	Adresse	Inhalt
Schnittstelle 1 - Protokollmakro-Ausführungsmerker	Bit 15 von CIO n+9	EIN, während PMCR(260) ausgeführt wird. Der Merker bleibt auf AUS gesetzt, wenn die Ausführung fehlschlägt. Der Merker wird bei Abschluss der Kommunikationssequenz (bei Ende oder Abbruch) auf AUS gesetzt.
Schnittstelle 2 - Protokollmakro-Ausführungsmerker	Bit 15 von CIO n+19	

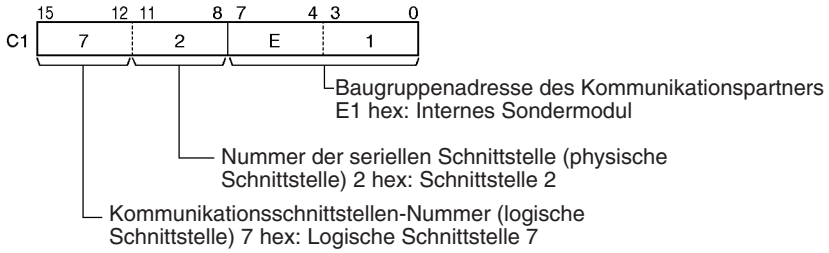
Beispiele

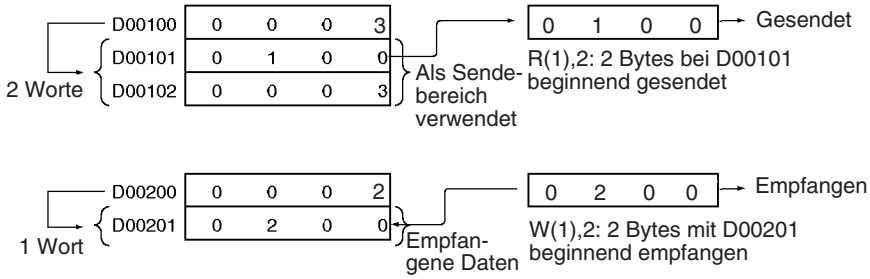
Wenn CIO 0000 in dem folgenden Beispiel auf EIN gesetzt ist, wird die Kommunikationssequenz Nr. 101 (0065 hex) ausgeführt, sofern der Kommunikationsschnittstelle-Frei-Merker für Schnittstelle 7 (A20207) auf EIN und der Protokollmakro-Ausführungsmerker für Schnittstelle 1 (CIO 190915) auf AUS gesetzt ist.

Wenn für das Symbol in einer Sendemeldung ein Operand spezifiziert ist, werden 2 Worte Daten mit D00101 beginnend als Sendebereich verwendet (da D00100 den Inhalt #0003 hat).

Wenn für das Symbol in einer Empfangsmeldung ein Operand spezifiziert ist, werden 2 Worte Daten mit D00201 beginnend gespeichert, und die Anzahl der empfangenen Wort +1 wird in D00200 geschrieben.







Hinweis Wie oben gezeigt, erfolgt das tatsächliche Senden/Empfangen mit der Symbolleseoption, R( ), in der Sendemeldung oder der Symbolschreiboption, W( ), in der Empfangsmeldung.

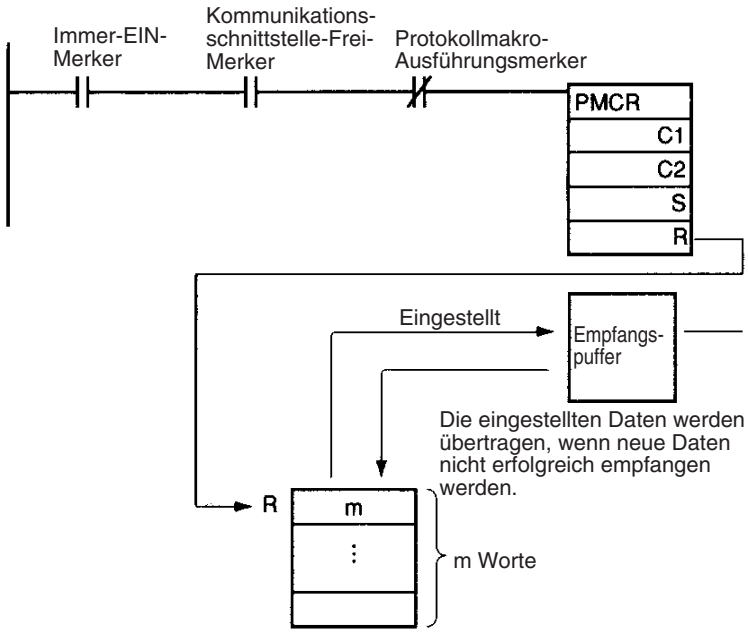
Halten des Empfangsbereichs

Der Empfangspuffer wird unmittelbar von der Ausführung einer Kommunikationssequenz durch PMCR(260) komplett auf Nullen gelöscht. Wenn eine Programmierung wie die unten dargestellte verwendet wird, um in regelmäßigen Abständen Istwerte oder andere Daten zu lesen, und die Daten können aufgrund eines Empfangsfehlers oder einer anderen Ursache nicht gelesen werden, werden die gelesenen Daten bis zum nächsten erfolgreichen Lesen gelöscht.

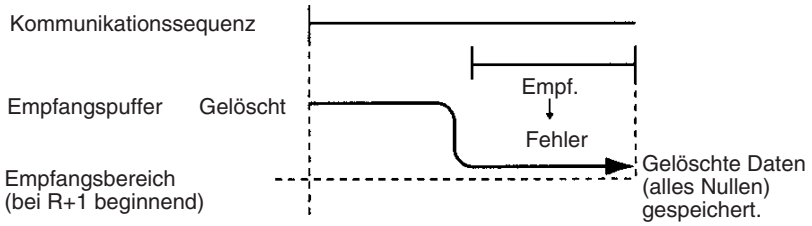
Es ist jedoch eine Funktion verfügbar, um die Daten im Empfangsbereich zu erhalten, selbst wenn ein Empfangsfehler auftritt. Bei Verwendung dieser Funktion werden die Daten aus den ersten m Worten des Empfangsbereichs in den Empfangspuffer übertragen, nachdem der Puffer auf Nullen gelöscht wurde, jedoch bevor die Kommunikationssequenz ausgeführt wird. Dadurch wird vermieden, dass der Empfangsbereich beim Schreiben der letzten Empfangsdaten vorübergehend auf Nullen gelöscht wird, was eintritt, wenn neue Empfangsdaten nicht erfolgreich erhalten werden.

Spezifizieren Sie als Wert m die Anzahl der Worte im Empfangsbereich, die erhalten bleiben sollen. Wenn 0 oder 1 spezifiziert wird, ist die Haltefunktion deaktiviert und der Empfangsbereich wird auf Nullen gelöscht.

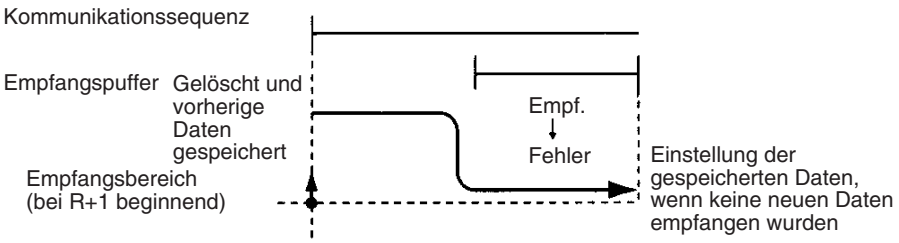
Das folgende Programmierungsbeispiel zeigt die Befehle, die zum ständigen oder periodischen Ausführen von PMCR(260) zum Lesen von Daten durch einen einzelnen Empfangsvorgang verwendet werden.



Empfangsdaten nicht gehalten



Empfangsdaten gehalten

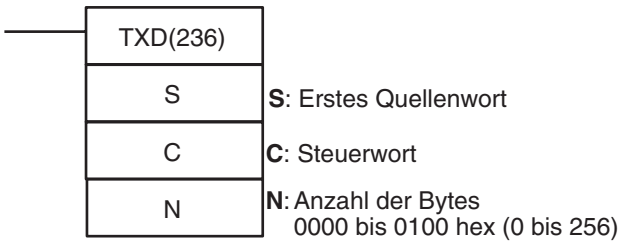


3-24-3 TRANSMIT: TXD(236)

Zweck

Gibt die spezifizierte Anzahl von Datenbytes über die integrierte RS-232C-Schnittstelle der CPU-Baugruppe oder über eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls aus. (Das serielle Kommunikationsmodul muss mindestens die Version 1.2 haben.)

Kontaktplansymbol



Variationen

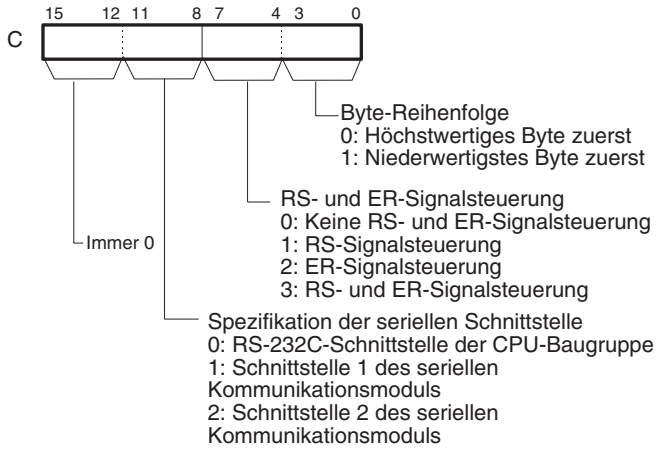
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TXD(236)
	Ausführung einmal bei steigender Flanke	@TXD(236)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Der Inhalt des Steuerworts C ist nachfolgend beschrieben.



Operanden-Spezifikationen

Bereich	S	C	N
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerkerbereich	H000 bis H511		
Zusatz-Systembereich	A000 bis A447 A448 bis A959		
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		

Bereich	S	C	N
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	Nur spezifizierte Werte	#0000 bis #0100 (binär) bzw. &0 bis &256 (dezimal)
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

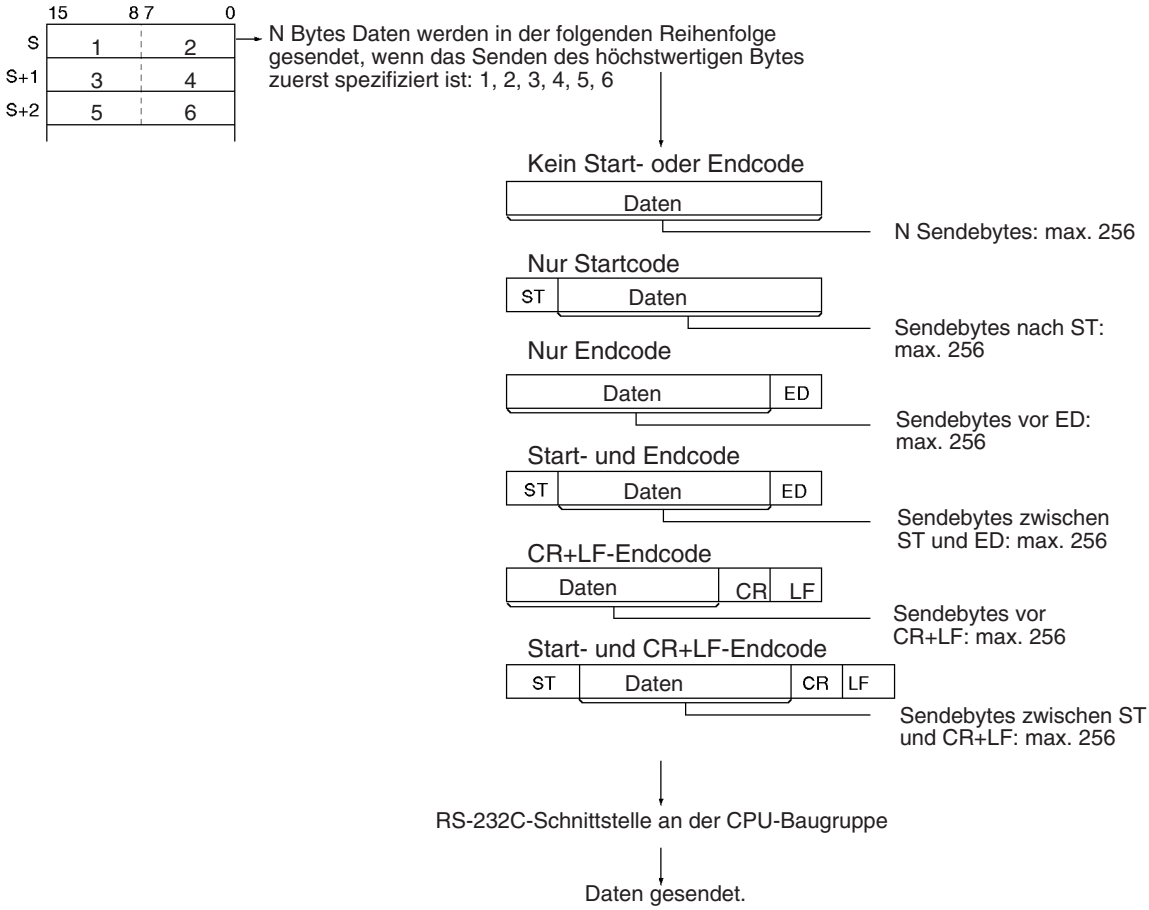
Mit TXD(236) werden n Byte Daten aus den Wörtern S to S+(N÷2)–1 ausgelesen und die Rohdaten im protokollfreien Modus über die integrierte RS-232C-Schnittstelle der CPU-Baugruppe oder eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls ausgegeben. (Die Ausgabeschnittstelle wird durch die Bits 8 bis 11 von C spezifiziert.)

Die für den protokollfreien Modus angegebenen Start- und End-Codes werden den Daten vor der Datenausgabe hinzugefügt. Die Start- und End-Codes (für die RS-232C-Schnittstelle der CPU-Baugruppe) sind im SPS-Setup oder im zugewiesenen DM-Konfigurationsbereich (für die Schnittstellen des seriellen Kommunikationsmoduls) spezifiziert.

Es können nur dann Daten gesendet werden, wenn der Sendebereitschafts-Merker der Schnittstelle auf EIN gesetzt ist. Der Sendebereitschafts-Merker für die RS-232C-Schnittstelle der CPU-Baugruppe hat die Adresse A39205, der für Schnittstelle 1 des seriellen Kommunikationsmoduls die Adresse A39605 und der für Schnittstelle 2 des seriellen Kommunikationsmoduls die Adresse A39613.

Es können bis zu 259 Byte gesendet werden, dazu gehören die Sendedaten (N = 256 Byte maximal), der Start-Code und der End-Code.

In der folgenden Abbildung ist die Reihenfolge dargestellt, in der Daten gesendet werden, sowie der Inhalt des Sende-Frames bei verschiedenen Start- und Endcode-Einstellungen.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die RS-232C-Schnittstelle der CPU-Baugruppe als Sendeschnittstelle spezifiziert ist, im SPS-Setup jedoch nicht der protokollfreie Modus eingestellt ist. EIN, wenn eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls als Sendeschnittstelle spezifiziert ist, in dem der Schnittstelle zugewiesenen DM-Konfigurationsbereich jedoch nicht der protokollfreie Modus eingestellt ist. EIN, wenn der Wert von C außerhalb des gültigen Bereichs liegt. EIN, wenn der Wert von N nicht zwischen 0000 und 0100 hex liegt. EIN, wenn das Senden versucht wird, während der Sendebereitschafts-Merker auf AUS gesetzt ist. (Der Sendebereitschafts-Merker für die RS-232C-Schnittstelle der CPU-Baugruppe hat die Adresse A39205, der für Schnittstelle 1 des seriellen Kommunikationsmoduls die Adresse A39605 und der für Schnittstelle 2 des seriellen Kommunikationsmoduls die Adresse A39613.) EIN (ER-Merker in Interrupt-Tasks), wenn ein TXD(236)- oder RXD(235)-Befehl für das serielle Kommunikationsmodul in der zyklischen Task ausgeführt wird, die zyklische Task unterbrochen wird und ein weiterer TXD(236)- oder RXD(235)-Befehl für das serielle Kommunikationsmodul in der Interrupt-Task ausgeführt wird. (Siehe Hinweis.) **Hinweis** Der Fehlermerker (ER) wird unmittelbar nach Ausführung des anderen TXD(236)- bzw. RXD(235)-Befehls in der Interrupt-Task auf EIN gesetzt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Befehl TXD(236) kann nur für die RS-232C-Schnittstelle der CPU-Baugruppe oder eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls verwendet werden. Darüber hinaus muss diese Schnittstelle auf den protokollfreien Modus eingestellt sein.

Das folgende Sendemeldungs-Frameformat kann im SPS-Setup (für die RS-232C-Schnittstelle der CPU-Baugruppe) bzw. im zugewiesenen DM-Konfigurationsbereich (für die Schnittstellen des seriellen Kommunikationsmoduls) eingestellt werden.

- Startcode: Keiner oder 00 bis FF hex.
- Endcode: Keiner, CR+LF oder 00 bis FF hex.

Die Daten werden mit einen beliebigen Start- und/oder Endcode gesendet, der im SPS-Setup bzw. im DM-Konfigurationsbereich spezifiziert ist. Wenn Start- und Endcodes spezifiziert sind, werden diese Codes an die Sendedaten (N) angefügt. In diesem Fall beträgt die maximale Anzahl von Bytes, die für N spezifiziert werden kann, 256.

Es können nur dann Daten gesendet werden, wenn der Sendebereitschafts-Merker der Schnittstelle auf EIN gesetzt ist. (Der Sendebereitschafts-Merker für die RS-232C-Schnittstelle der CPU-Baugruppe hat die Adresse A39205, der für Schnittstelle 1 des seriellen Kommunikationsmoduls die Adresse A39605 und der für Schnittstelle 2 des seriellen Kommunikationsmoduls die Adresse A39613.)

Die Daten werden in der in C spezifizierten Reihenfolge gesendet.

Wenn für N der Wert 0 spezifiziert ist, wird nichts gesendet.

Wenn die RS-Signalsteuerung in C spezifiziert ist, wird Bit 15 von S als RS-Signal verwendet.

Wenn die ER-Signalsteuerung in C spezifiziert ist, wird Bit 15 von S als ER-Signal verwendet.

Wenn die RS- und ER-Signalsteuerung in C spezifiziert ist, wird Bit 15 von S als ER-Signal und Bit 14 von S als ER-Signal verwendet.

Wenn in C für die RS- und ER-Signalsteuerung 1, 2 oder 3 hex spezifiziert ist, wird der Befehl TXD(236) unabhängig vom Status des Sendebereitschafts-Merkers (A39205, A39605 oder A39613, je nach verwendeter Schnittstelle) ausgeführt.

Wenn der Befehl TXD(236) für ein Modul ausgeführt wird, das den protokollfreien Modus nicht unterstützt (ein serielles Kommunikationsmodul ohne Versionsnummer), werden der Merker für Deaktivierung des Dienstes für das interne Sondermodul (A42404) sowie der Fehlermerker auf EIN gesetzt.

In den folgenden Fällen tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

- Die RS-232C-Schnittstelle der CPU-Baugruppe ist als Sendeschnittstelle spezifiziert, im SPS-Setup ist jedoch nicht der protokollfreie Modus eingestellt.
- Eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls ist als Sendeschnittstelle spezifiziert, in dem der Schnittstelle zugewiesenen DM-Konfigurationsbereich ist jedoch nicht der protokollfreie Modus eingestellt.
- Eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls ist als Sendeschnittstelle spezifiziert, das Modul unterstützt den protokollfreien Modus jedoch nicht (das Modul hat keine Versionsnummer).
- Der Wert von C liegt außerhalb des gültigen Bereichs.
- Der Wert von N liegt nicht zwischen 0000 und 0100 hex.
- Senden wurde versucht, während der Sendebereitschafts-Merker auf AUS gesetzt war. (Der Sendebereitschafts-Merker für die RS-232C-Schnittstelle der CPU-Baugruppe hat die Adresse A39205, der für Schnittstelle 1 des seriellen Kommunikationsmoduls die Adresse A39605 und der für Schnittstelle 2 des seriellen Kommunikationsmoduls die Adresse A39613.)
- Der Befehl TXD(236) oder RXD(235) wurde für das serielle Kommunikationsmodul in der zyklischen Task ausgeführt, die zyklische Task wurde unterbrochen und ein weiterer TXD(236)- oder RXD(235)-Befehl für das serielle Kommunikationsmodul in der Interrupt-Task wurde ausgeführt.

Hinweis Programmieren Sie die Befehle TXD(236)/RXD(235) für eine Schnittstelle (Schnittstelle 1 oder 2) eines seriellen Kommunikationsmoduls nicht sowohl in der zyklischen Task als auch in einer Interrupt-Task. Ein TXD(236)/RXD(235)-Befehl für das serielle Kommunikationsmodul kann nicht in der Interrupt-Task ausgeführt werden, während ein TXD(236)/RXD(235)-Befehl für das serielle Kommunikationsmodul in der zyklischen Task ausgeführt wird. Wenn ein TXD(236)/RXD(235)-Befehl für das serielle Kommunikationsmodul in der Interrupt-Task ausgeführt wird, während ein anderer TXD(236)/RXD(235)-Befehl für das serielle Kommunikationsmodul in der zyklischen Task ausgeführt wird, tritt ein Fehler auf und der ER-Merker wird auf EIN gesetzt. (Diese Befehle können nicht sowohl in der zyklischen Task als auch in Interrupt-Tasks programmiert werden, selbst wenn sie für verschiedene Schnittstellen des seriellen Kommunikationsmoduls ausgeführt werden.)

Zugehörige Merker und Worte

Die folgenden Einstellungen im SPS-Setup und Merker im Zusatz-Systembereich können wie erforderlich bei der Ausführung von TXD(236) verwendet werden.

Einstellungen für die RS-232C-Schnittstelle der CPU-Baugruppe im SPS-Setup

Programmierkonsolen-Adresse		Name	Einstellungen
Wort	Bit		
162	0 bis 15	Sendeverzögerung im protokollfreien Modus	0000 bis 210F hex, 0 bis 99.990 ms dezimal (in 10-ms-Schritten)
164	8 bis 15	Startcode für protokollfreien Modus	00 bis FF hex
	0 bis 7	Endcode für protokollfreien Modus	00 bis FF hex
165	12	Startcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Startcode verwenden.
	8 und 9	Endcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Endcode verwenden. 2: CR+LF verwenden.
	0 bis 7	Anzahl der Datenbytes für protokollfreien Modus	00: 256 Bytes 01 bis FF: 1 bis 255 Bytes

Einstellungen für die Schnittstellen des seriellen Kommunikationsmoduls im DM-Konfigurationsbereich

Konfigurationsbereichs-Wort		Bit	Name	Einstellungen
Schnittstelle 1	Schnittstelle 2			
D32002	D32012	15	Sendeverzögerungs-Spezifikation für protokollfreien Modus	0: Standardeinstellung (0 ms) 1: Die in den Bits 1 bis 14 eingestellte Verzögerung verwenden.
		0 bis 14	Sendeverzögerungszeit für protokollfreien Modus	0000 bis 7530 hex 0 bis 300.000 ms dezimal (in 10-ms-Schritten)
D32004	D32014	8 bis 15	Startcode für protokollfreien Modus	00 bis FF hex
		0 bis 7	Endcode für protokollfreien Modus	00 bis FF hex
D32005	D32015	12 bis 15	Startcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Startcode verwenden.
		8 bis 11	Endcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Endcode verwenden. 2: CR+LF verwenden.

Zusatz-Systembereich

Sendebereitschafts-Merker

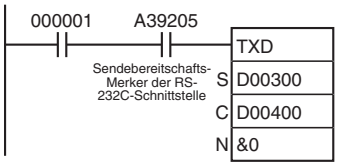
Schnittstelle	Adresse	Inhalt
Integrierte RS-232C-Schnittstelle der CPU-Baugruppe	A39205	EIN, wenn Daten im protokollfreien Modus gesendet werden können.
Schnittstelle 1 des seriellen Kommunikationsmoduls	A39605	
Schnittstelle 2 des seriellen Kommunikationsmoduls	A39613	

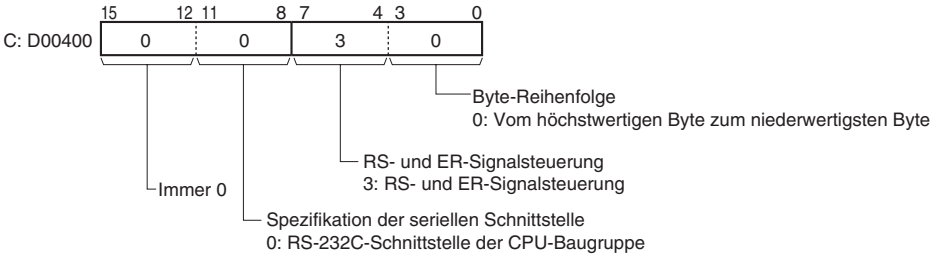
Internes-Sondermodul-Merker für serielle Kommunikationsmodule (Schnittstellen 1 und 2)

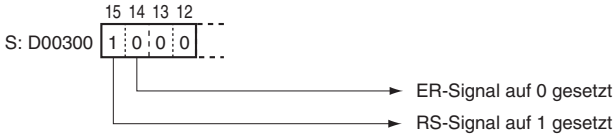
Name	Adresse	Inhalt
Merker: Dienst für das interne Sondermodul deaktiviert	A42404	EIN, wenn der Befehl TXD(236) für ein serielles Kommunikationsmodul ausgeführt wird, das den protokollfreien Modus nicht unterstützt (ein Modul ohne Versionsnummer).

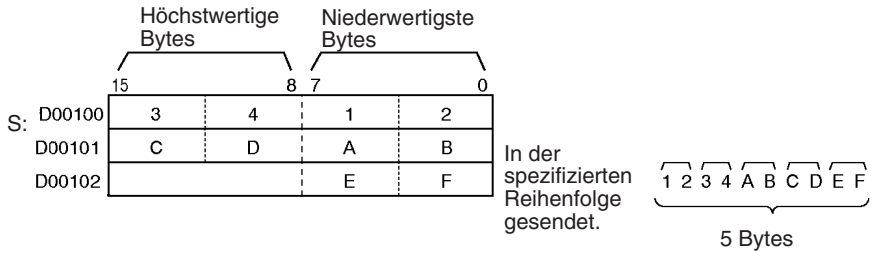
Beispiel: Senden von Daten

Wenn in dem folgenden Beispiel sowohl CIO 000001 als auch der Sendebereitschafts-Merker der RS-232C-Schnittstelle (A39205) auf EIN gesetzt sind, wird das RS-Signal entsprechend des Status von D00300, Bit 15, und das ER-Signal entsprechend des Status von D00300, Bit 14, gesetzt.









Start- und Endcode werden gemäß der Einstellung im SPS-Setup angefügt (in diesem Beispiel wird angenommen, dass die Verwendung von Start- und Endcode eingestellt ist).

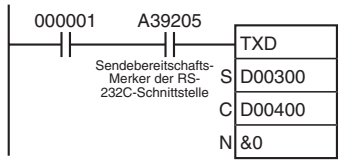
ST	12	34	AB	CD	EF	ED

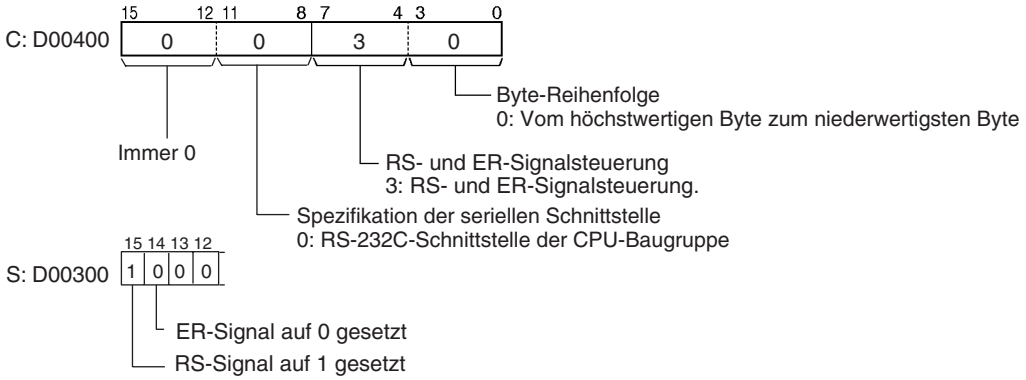
ST: Startcode (z. B. 02 hex)
ED: Endcode (z. B. 03 hex)

Gesendet

Beispiel: Durchführen der Signalsteuerung

Wenn in dem folgenden Beispiel sowohl CIO 000001 als auch der Sendebereitschafts-Merker der RS-232C-Schnittstelle (A39205) auf EIN gesetzt sind, wird das RS-Signal entsprechend des Status von D00300, Bit 15, und das ER-Signal entsprechend des Status von D00300, Bit 14, gesetzt.



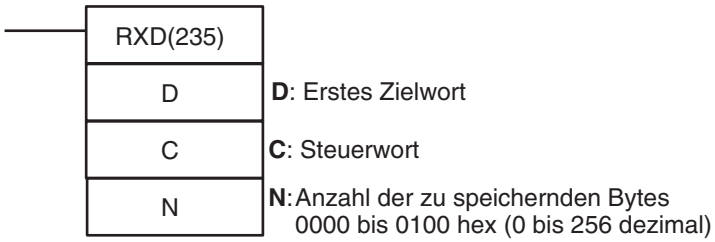


3-24-4 RECEIVE: RXD(235)

Zweck

Liest die spezifizierte Anzahl von Datenbytes von der integrierten RS-232C-Schnittstelle der CPU-Baugruppe oder von einer der seriellen Schnittstellen des seriellen Kommunikationsmoduls. (Das serielle Kommunikationsmodul muss mindestens die Version 1.2 haben.)

Kontaktplansymbol



Variationen

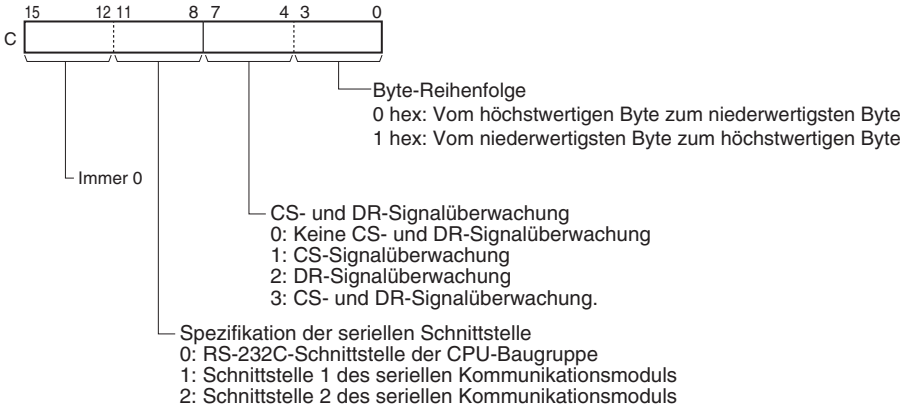
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	RXD(235)
	Ausführung einmal bei steigender Flanke	@RXD(235)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Der Inhalt des Steuerworts C ist nachfolgend beschrieben.



Operanden-Spezifikationen

Bereich	D	C	N
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich	W000 bis W511		
Haftmerkerbereich	H000 bis H511		
Zusatz-Systembereich	A448 bis A959	A000 bis A447 A448 bis A959	
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	Nur spezifizierte Werte	#0000 bis #0100 (binär) oder &0 bis &256 (dezimal)
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

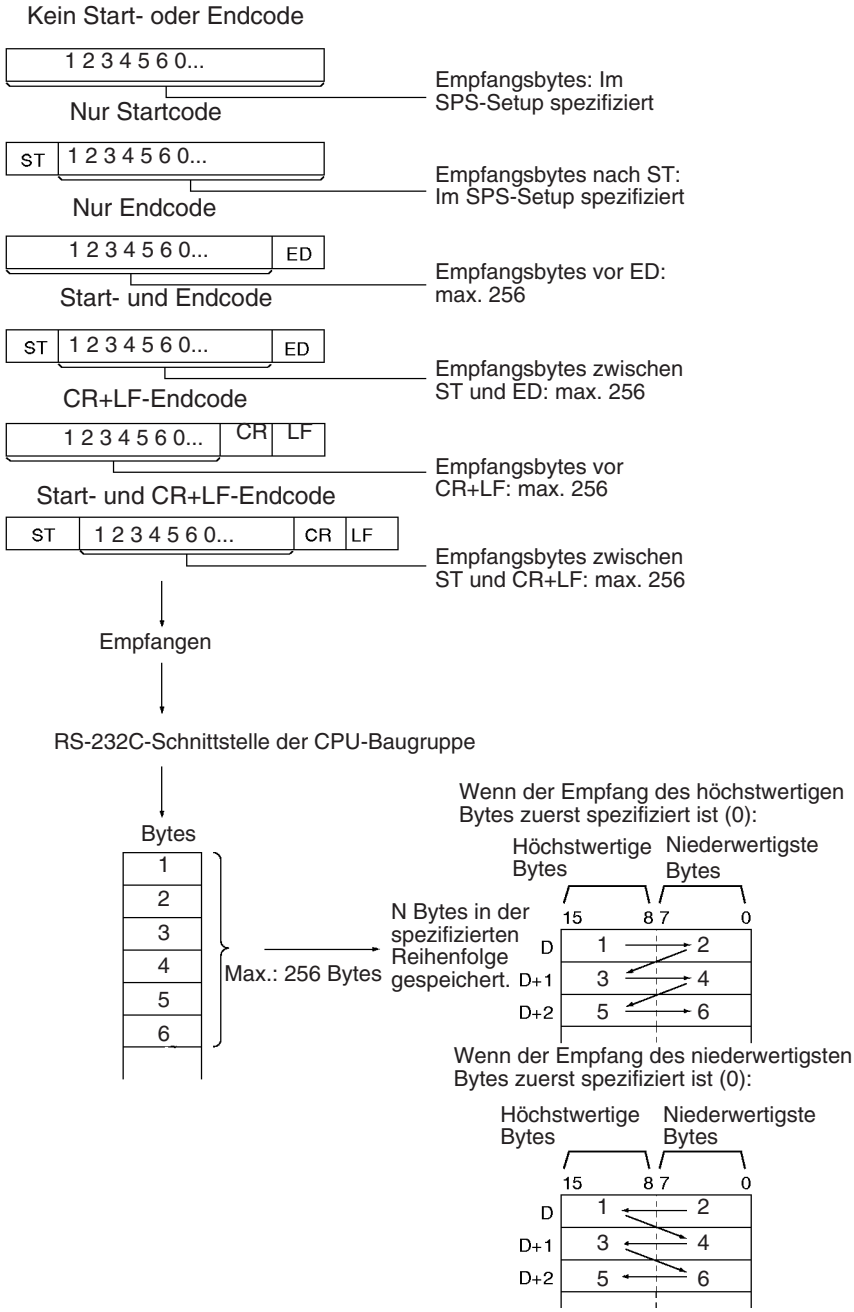
Beschreibung

Mit RXD(235) werden Daten, die im protokollfreien Modus an der internen RS-232C-Schnittstelle oder einer der seriellen Schnittstellen des seriellen Kommunikationsmoduls empfangen wurden (die Schnittstelle wird durch die Bits 8 bis 11 von C spezifiziert), eingelesen und N Byte Daten werden in den Worten D bis D+(N÷2)-1 gespeichert. Wenn nicht N Bytes Daten über die Schnittstelle empfangen wurde, werden nur die Daten, die empfangen wurden, gespeichert.

Es können nur dann Daten empfangen werden, wenn der Empfangsbereitschafts-Merker der Schnittstelle auf EIN gesetzt ist. Der Empfangsbereitschafts-Merker der RS-232C-Schnitstelle der CPU-Baugruppe hat die Adresse A39206, der von Schnittstelle 1 des seriellen Kommunikationsmoduls die Adresse A39606 und der von Schnittstelle 2 des seriellen Kommunikationsmoduls die Adresse A39614. Führen Sie den Befehl RXD(235) nur aus, wenn der entsprechende Empfangsbereitschafts-Merker auf EIN gesetzt ist.

Es können bis zu 259 Bytes empfangen werden, dazu gehören die Empfangsdaten (N = maximal 256 Bytes), der Startcode und der Endcode.

In der folgenden Abbildung ist die Reihenfolge dargestellt, in der Daten empfangen werden, sowie der Inhalt des Empfangs-Frames bei verschiedenen Einstellungen.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die RS-232C-Schnittstelle der CPU-Baugruppe als Empfangsschnittstelle spezifiziert ist, im SPS-Setup jedoch nicht der protokollfreie Modus eingestellt ist. EIN, wenn eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls als Empfangsschnittstelle spezifiziert ist, in dem der Schnittstelle zugewiesenen DM-Konfigurationsbereich jedoch nicht der protokollfreie Modus eingestellt ist. EIN, wenn der Wert von C außerhalb des gültigen Bereichs liegt. EIN, wenn der Wert von N nicht zwischen 0000 und 0100 hex liegt. EIN, wenn das Empfangen versucht wird, während der Empfangsbereitschafts-Merker auf AUS gesetzt ist. Der Empfangsbereitschafts-Merker der RS-232C-Schnittstelle der CPU-Baugruppe hat die Adresse A39206, der von Schnittstelle 1 des seriellen Kommunikationsmoduls die Adresse A39606 und der von Schnittstelle 2 des seriellen Kommunikationsmoduls die Adresse A39614.) EIN (ER-Merker in Interrupt-Tasks), wenn ein TXD(236)- oder RXD(235)-Befehl für das serielle Kommunikationsmodul in der zyklischen Task ausgeführt wird, die zyklische Task unterbrochen wird und ein weiterer TXD(236)- oder RXD(235)-Befehl für das serielle Kommunikationsmodul in der Interrupt-Task ausgeführt wird. (Siehe Hinweis.) **Hinweis** Der Fehlermerker (ER) wird unmittelbar nach Ausführung des anderen TXD(236)- bzw. RXD(235)-Befehls in der Interrupt-Task auf EIN gesetzt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Befehl RXD(235) kann nur für die RS-232C-Schnittstelle der CPU-Baugruppe oder eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls verwendet werden. Darüber hinaus muss diese Schnittstelle auf den protokollfreien Modus eingestellt sein.

Das folgende Empfangsmeldungs-Frameformat kann im SPS-Setup (für die RS-232C-Schnittstelle der CPU-Baugruppe) bzw. im zugewiesenen DM-Konfigurationsbereich (für die Schnittstellen des seriellen Kommunikationsmoduls) eingestellt werden.

- Startcode: Keiner oder 00 bis FF hex.
- Endcode: Keiner, CR+LF oder 00 bis FF hex. Wenn keine Endcode spezifiziert ist, kann die Anzahl für zu empfangene Bytes auf 00 bis FF hex (1 bis 256 dezimal; 00 spezifiziert 256 Bytes) eingestellt werden.

Der Empfang-Abgeschlossen-Merker (Hinweis 1) wird auf EIN gesetzt, wenn die im SPS-Setup (bei der RS-232C-Schnittstelle der CPU-Baugruppe) bzw. im zugewiesenen DM-Konfigurationsbereich (bei den Schnittstellen des seriellen Kommunikationsmoduls) spezifizierte Anzahl von Bytes empfangen wurden. Wenn der Empfang-Abgeschlossen-Merker auf EIN gesetzt wird, hat die Anzahl der Bytes im Empfangszähler (Hinweis 2) den gleichen Wert wie die Anzahl der im SPS-Setup bzw. im zugewiesenen DM-Konfigurationsbereich spezifizierten Empfangsbytes. Werden mehr Bytes als spezifiziert empfangen, so wird der Empfangsüberlauf-Merker (Hinweis 3) auf EIN gesetzt.

Wenn im SPS-Setup bzw. im zugewiesenen DM-Konfigurationsbereich ein Endcode spezifiziert ist, wird der Empfang-Abgeschlossen-Merker (Hinweis 1) auf EIN gesetzt, wenn der Endcode empfangen wird oder wenn 256 Bytes Daten empfangen wurden. Wenn weitere Daten empfangen werden, nachdem der Empfang-Abgeschlossen-Merker (Hinweis 1) auf EIN gesetzt ist, wird der Empfangsüberlauf-Merker (Hinweis 3) auf EIN gesetzt.

Der Empfang wird gestoppt, nachdem 259 Bytes Daten empfangen wurden. Wenn danach weitere Daten eingegeben werden, werden der Überlauf-Fehlermerker (Hinweis 5) und der Übertragungs-Fehlermerker (Hinweis 6) auf EIN gesetzt.

Wenn mehr Daten als in N spezifiziert an der seriellen Schnittstelle des seriellen Kommunikationsmoduls eingegeben werden, werden diese Daten bei der Ausführung von RXD(235) verworfen. Hingegen werden übermäßige Daten, die über die RS-232C-Schnittstelle der CPU-Baugruppe eingegeben werden, bei der Ausführung von RXD(235) nicht verworfen.

Wenn der Befehl RXD(235) ausgeführt wird, werden die Daten in den mit D beginnenden Worten gespeichert, der Empfang-Abgeschlossen-Merker (Hinweis 1) wird auf AUS gesetzt (selbst falls der Empfangsüberlauf-Merker (Hinweis 3) auf EIN gesetzt ist) und der Empfangszähler (Hinweis 2) wird auf 0 zurückgesetzt.

Wenn das Neustart-Bit für die RS-232C-Schnittstelle der CPU-Baugruppe (Hinweis 4) auf EIN gesetzt ist, wird der Empfang-Abgeschlossen-Merker (Hinweis 1) auf AUS gesetzt (selbst falls der Empfangsüberlauf-Merker (Hinweis 3) auf EIN gesetzt ist) und der Empfangszähler (Hinweis 2) wird auf 0 zurückgesetzt.

Die Daten werden in der in C spezifizierten Reihenfolge im Speicher gespeichert.

Ist für N der Wert 0 spezifiziert, wird der Empfang-Abgeschlossen-Merker (Hinweis 1) auf AUS gesetzt, der Empfangszähler (Hinweis 2) wird auf 0 zurückgesetzt und es werden keine Daten in den Speicher geschrieben.

Wenn die CS-Signalüberwachung in C spezifiziert ist, wird der Status des CS-Signals in Bit 15 von D gespeichert.

Wenn die DR-Signalüberwachung in C spezifiziert ist, wird der Status des DR-Signals in Bit 15 von D gespeichert.

Wenn die CS- und DR-Signalüberwachung in C spezifiziert ist, wird der Status des CS-Signals in Bit 15 von D und der Status des DR-Signals in Bit 14 von D gespeichert.

Falls CS- oder DR-Signalüberwachung spezifiziert ist, werden keine Daten gespeichert.

Wenn für die RS- und ER-Signalsteuerung in C der Wert 1, 2 oder 3 hex spezifiziert ist, wird der Befehl RXD(235) ungeachtet des Status des Empfang-Abgeschlossen-Merkers (Hinweis 1) ausgeführt.

Wenn der Befehl RXD(235) für ein Modul ausgeführt wird, das den protokollfreien Modus nicht unterstützt (ein serielles Kommunikationsmodul ohne Versionsnummer), werden der Merker für Deaktivierung des Dienstes für das interne Sondermodul (A42404, nicht schwerwiegender Fehler) sowie der Fehlermerker auf EIN gesetzt.

Hinweis

1. Empfang-Abgeschlossen-Merker

Integrierte RS232C-Schnittstelle	A39206
Schnittstelle 1 des seriellen Kommunikationsmoduls:	A35606
Schnittstelle 2 des seriellen Kommunikationsmoduls:	A35614

2. Empfangszähler

Integrierte RS232C-Schnittstelle	A393
Schnittstelle 1 des seriellen Kommunikationsmoduls:	A357
Schnittstelle 2 des seriellen Kommunikationsmoduls:	A358

3. Empfangsüberlaufmerker

Integrierte RS232C-Schnittstelle	A39207
Schnittstelle 1 des seriellen Kommunikationsmoduls:	A35607
Schnittstelle 2 des seriellen Kommunikationsmoduls:	A35615

4. RS-232C-Schnittstellen-Neustartbit

Integrierte RS232C-Schnittstelle	A52600

5. Überlauf-Fehlermerker

Schnittstelle 1 des seriellen Kommunikationsmoduls:	CIO 190804
Schnittstelle 2 des seriellen Kommunikationsmoduls:	CIO 191804

6. Übertragungs-Fehlermerker

Schnittstelle 1 des seriellen Kommunikationsmoduls:	CIO 190815
Schnittstelle 2 des seriellen Kommunikationsmoduls:	CIO 191815

7. Merker: Dienst für das interne Sondermodul deaktiviert

   Schnittstellen 1 und 2 des seriellen Kommunikationsmoduls: A42404

In den folgenden Fällen tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

- Die RS-232C-Schnittstelle der CPU-Baugruppe ist als Empfangsschnittstelle spezifiziert, im SPS-Setup ist jedoch nicht der protokollfreie Modus eingestellt.
- Eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls ist als Empfangsschnittstelle spezifiziert, in dem der Schnittstelle zugewiesenen DM-Konfigurationsbereich ist jedoch nicht der protokollfreie Modus eingestellt.
- Eine der seriellen Schnittstellen des seriellen Kommunikationsmoduls ist als Empfangsschnittstelle spezifiziert, das Modul unterstützt den protokollfreien Modus jedoch nicht (das Modul hat keine Versionsnummer).
- Der Wert von C liegt außerhalb des gültigen Bereichs.
- Der Wert von N liegt nicht zwischen 0000 und 0100 hex.
- Der Befehl TXD(236) oder RXD(235) wurde für das serielle Kommunikationsmodul in der zyklischen Task ausgeführt, die zyklische Task wurde unterbrochen und ein weiterer TXD(236)- oder RXD(235)-Befehl für das serielle Kommunikationsmodul in der Interrupt-Task wurde ausgeführt.
- Es können keine weiteren Daten empfangen werden, bis die Empfangsdaten durch den Befehl RXD(235) aus dem Empfangspuffer ausgelesen sind. Wenn der Empfang-Abgeschlossen-Merker auf EIN gesetzt wird, sollten die Daten unverzüglich durch RXD(235) ausgelesen werden, bevor weitere Daten über die Schnittstelle eingegeben werden.
- Wenn der Befehl RXD(235) zum Lesen von Daten verwendet wird, die über die RS-232C-Schnittstelle der CPU-Baugruppe empfangen wurden, werden die im Empfangspuffer der Schnittstelle verbliebenen Daten nicht gelöscht, sodass RXD(235) wiederholt ausgeführt werden kann, um einen Datenblock stückweise zu lesen.
  Wenn der Befehl RXD(235) hingegen zum Lesen von Daten verwendet wird, die über eine der seriellen Schnittstellen eines seriellen Kommunikationsmoduls (serielles Kommunikationsmodul ab Version 1.2) empfangen wurden, wird der Empfangspuffer der Schnittstelle nach der Ausführung von RXD(235) gelöscht. Demzufolge kann der Befehl RXD(235) **nicht** wiederholt ausgeführt werden, um einen Datenblock stückweise zu lesen.

Zugehörige Merker und Worte

Die folgenden Einstellungen im SPS-Setup und Merker im Zusatz-Systembereich können wie erforderlich bei der Ausführung von RXD(235) verwendet werden.

Einstellungen für die RS-232C-Schnittstelle der CPU-Baugruppe im SPS-Setup

Programmierkonsolen-Adresse		Name	Einstellungen
Wort	Bit		
162	0 bis 15	Sendeverzögerung im protokollfreien Modus	0000 bis 210F hex, 0 bis 99.990 ms dezimal (in 10-ms-Schritten)
164	8 bis 15	Startcode für protokollfreien Modus	00 bis FF hex
	0 bis 7	Endcode für protokollfreien Modus	00 bis FF hex
165	12	Startcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Startcode verwenden.
	8 und 9	Endcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Endcode verwenden. 2: CR+LF verwenden.
	0 bis 7	Anzahl der Datenbytes für protokollfreien Modus	00: 256 Bytes 01 bis FF: 1 bis 255 Bytes

Einstellungen für die Schnittstellen des seriellen Kommunikationsmoduls im DM-Konfigurationsbereich

Konfigurations-bereichs-Wort		Bit	Name	Einstellungen
Schnitt-stelle 1	Schnitt-stelle 2			
D32004	D32014	8 bis 15	Startcode für protokollfreien Modus	00 bis FF hex
		0 bis 7	Endcode für protokollfreien Modus	00 bis FF hex
D32005	D32015	12 bis 15	Startcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Startcode verwenden.
		8 bis 11	Endcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Endcode verwenden. 2: CR+LF verwenden.

Merker im Zusatz-Systembereich für die RS-232C-Schnittstelle der CPU-Baugruppe

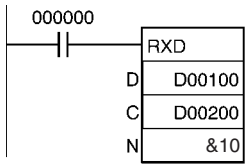
Name	Adresse	Inhalt
RS-232C-Schnittstellen-Empfang-abgeschlossen-Merker	A39206	EIN, wenn der protokollfreie Empfang abgeschlossen ist. Anzahl der spezifizierten Empfangsbytes: Dieser Merker wird auf EIN gesetzt, wenn die spezifizierte Anzahl von Bytes empfangen wurde. Spezifizierter Endcode: Dieser Merker wird auf EIN gesetzt, wenn der Endcode empfangen wird oder wenn 256 Bytes Daten empfangen wurden.
RS-232C-Schnittstellen-Empfangsüberlauf-Merker	A39207	EIN, wenn mehr als die spezifizierte Anzahl von Empfangsbytes empfangen wurden. Anzahl der spezifizierten Empfangsbytes: Dieser Merker wird auf EIN gesetzt, wenn nach Abschluss des Empfangs und Ausführung des nächsten RXD(235)-Befehls noch weitere Daten empfangen werden. Spezifizierter Endcode: Dieser Merker wird auf EIN gesetzt, wenn nach Empfang des Endcodes und Ausführung des nächsten RXD(235)-Befehls noch weitere Daten empfangen werden, oder wenn das 257. Datenbyte vor dem Endcode empfangen wird.
RS-232C-Schnittstellen-Empfangszähler	A393	Zählt die Anzahl der im protokollfreien Modus empfangenen Bytes als Hexadezimalzahl.

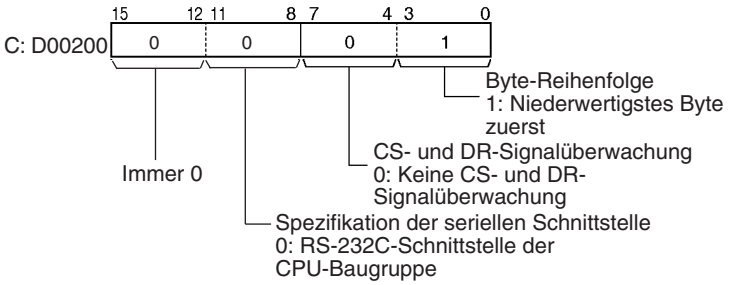
Merker für die Schnittstellen des seriellen Kommunikationsmoduls im Zusatz-Systembereich

Schnitt-stelle	Name	Adresse	Inhalt
Schnitt-stelle 1	Empfang-Abgeschlossen-Merker	A35606	EIN, wenn der protokollfreie Empfang abgeschlossen ist. Anzahl der spezifizierten Empfangsbytes: Dieser Merker wird auf EIN gesetzt, wenn die spezifizierte Anzahl von Bytes empfangen wurde. Spezifizierter Endcode: Dieser Merker wird auf EIN gesetzt, wenn der Endcode empfangen wird oder wenn 256 Bytes Daten empfangen wurden.
	Empfangsüber-lauf-Merker	A35607	EIN, wenn mehr als die spezifizierte Anzahl von Empfangsbytes im protokollfreien Modus empfangen wurden. Anzahl der spezifizierten Empfangsbytes: Dieser Merker wird auf EIN gesetzt, wenn nach Abschluss des Empfangs, jedoch vor dem Auslesen der empfangenen Daten aus dem Empfangspuffer mit dem Befehl RXD(235) weitere Daten empfangen werden. Spezifizierter Endcode: Dieser Merker wird auf EIN gesetzt, wenn 257 oder mehr Bytes Daten ohne einen Endcode empfangen werden.
	Empfangs-zähler	A357	Zählt die Anzahl der im protokollfreien Modus empfangenen Bytes als Hexadezimalzahl (0 bis 256 dezimal).
	Überlauf-Fehlermerker	CIO 1908, Bit 04	EIN, wenn vor der Ausführung des Befehls RXD(235) 260 Bytes Daten oder mehr im Empfangspuffer empfangen wurden.
Schnitt-stelle 2	Empfang-Abgeschlossen-Merker	A35614	EIN, wenn der protokollfreie Empfang abgeschlossen ist. Anzahl der spezifizierten Empfangsbytes: Dieser Merker wird auf EIN gesetzt, wenn die spezifizierte Anzahl von Bytes empfangen wurde. Spezifizierter Endcode: Dieser Merker wird auf EIN gesetzt, wenn der Endcode empfangen wird oder wenn 256 Bytes Daten empfangen wurden.
	Empfangsüber-lauf-Merker	A35615	EIN, wenn mehr als die spezifizierte Anzahl von Empfangsbytes im protokollfreien Modus empfangen wurden. Anzahl der spezifizierten Empfangsbytes: Dieser Merker wird auf EIN gesetzt, wenn nach Abschluss des Empfangs, jedoch vor dem Auslesen der empfangenen Daten aus dem Empfangspuffer mit dem Befehl RXD(235) weitere Daten empfangen werden. Spezifizierter Endcode: Dieser Merker wird auf EIN gesetzt, wenn 257 oder mehr Bytes Daten ohne einen Endcode empfangen werden.
	Empfangs-zähler	A358	Zählt die Anzahl der im protokollfreien Modus empfangenen Bytes als Hexadezimalzahl (0 bis 256 dezimal).
	Überlauf-Fehlermerker	CIO 1918, Bit 04	EIN, wenn vor der Ausführung des Befehls RXD(235) 260 Bytes Daten oder mehr im Empfangspuffer empfangen wurden.
Schnitt-stellen 1 und 2	Merker: Dienst für das interne Sondermodul deaktiviert	A42404	EIN, wenn der Befehl RXD(235) für ein serielles Kommunikationsmodul ausgeführt wird, das den protokollfreien Modus nicht unterstützt (ein Modul ohne Versionsnummer).

Beispiele

Wenn CIO 000000 in dem folgenden Beispiel auf EIN gesetzt ist, werden die über die RS-232C-Schnittstelle empfangenen Daten und 10 Bytes Daten in den mit D00100 beginnenden Adressen gespeichert.





In diesem Beispiel wird davon ausgegangen, dass im SPS-Setup ein Start- und ein Endcode spezifiziert wurden.

ST	31	32	33	34	41	42	43	44	45	46	ED

Speicherung →

ST: Startcode (z. B. 02 hex),
ED: Endcode (z. B. 03 hex)

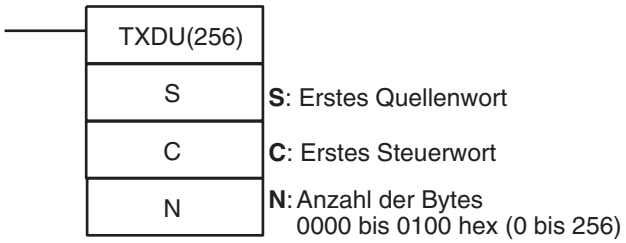
	Höchstwertige Bytes (15–8)		Niederwertigste Bytes (7–0)	
D: D00100	3	2	3	1
D00101	3	4	3	3
D00102	4	2	4	1
D00103	4	4	4	3
D00104	4	6	4	5

3-24-5 TRANSMIT VIA SERIAL COMMUNICATIONS UNIT: TXDU(256)

Zweck

Ausgabe der spezifizierten Anzahl von Datenbytes über eine der seriellen Schnittstellen der seriellen Kommunikationsbaugruppe. (Die serielle Kommunikationsbaugruppe muss mindestens die Version 1.2 haben.)

Kontaktplansymbol



Variationen

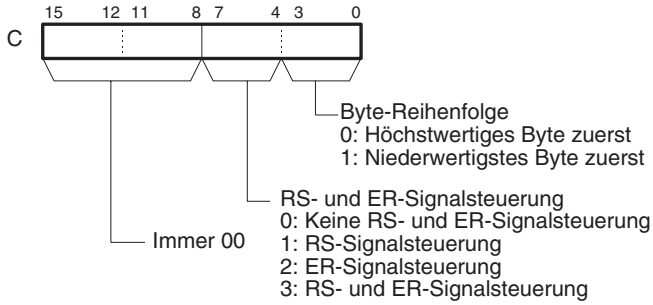
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TXDU(256)
	Ausführung einmal bei steigender Flanke	@TXDU(256)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

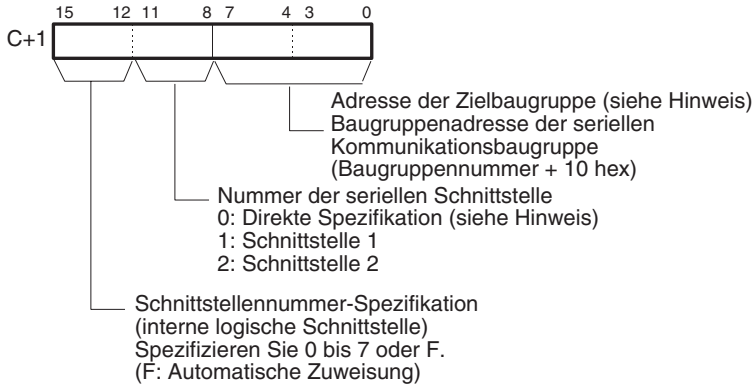
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Der Inhalt der Steuerworte C und C+1 ist nachfolgend beschrieben.





Hinweis Die Baugruppenadresse der seriellen Schnittstelle kann direkt spezifiziert werden, indem die Nummer der seriellen Schnittstelle auf 0 und die Adresse der Zielbaugruppe auf die Adresse der Baugruppe mit der seriellen Schnittstelle eingestellt wird. (Stellen Sie die Adresse der Zielbaugruppe auf 80 hex + 4 × Baugruppennummer für Schnittstelle 1 bzw. 81 hex + 4 × Baugruppennummer für Schnittstelle 2 ein.)

Operanden-Spezifikationen

Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W510	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H510	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A958	A000 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		

Bereich	S	C	D
Konstanten	---	Nur spezifizierte Werte	#0000 bis #0100 (binär) oder &0 bis &256 (dezimal)
Datenregister	---	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

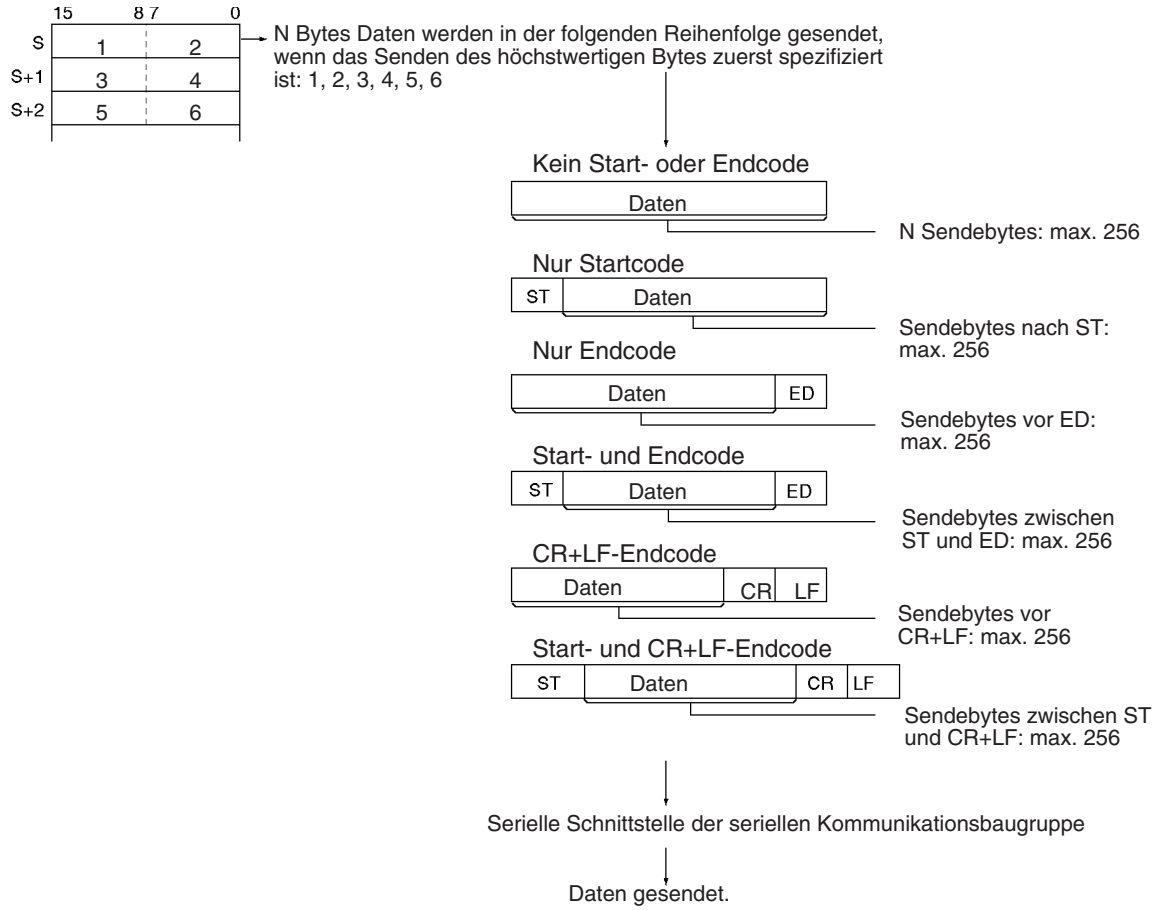
Der Befehl TXDU(256) liest N Bytes Daten aus den Worten S bis S+(N÷2)–1 und gibt diese Rohdaten im protokollfreien Modus von der seriellen Kommunikationsbaugruppe mit der in den Bits 0 bis 7 von C+1 spezifizierten Baugruppenadresse über die in den Bits 8 bis 11 von C+1 spezifizierte Schnittstelle aus. Die logische Schnittstellennummer kann auf einen beliebigen Wert zwischen 0 und 7 eingestellt werden. Sie wird mit den Bits 12 bis 15 von C+1 spezifiziert.

Die im zugewiesenen DM-Konfigurationsbereich für den protokollfreien Modus spezifizierten Start- und Endcodes werden den Daten vor der Datenausgabe hinzugefügt. Es können bis zu 259 Byte gesendet werden, dazu gehören die Sendedaten (N = 256 Byte maximal), der Startcode und der Endcode.

Die Daten können nur gesendet werden, wenn der Kommunikationsschnittstelle-Frei-Merker der spezifizierten logischen Schnittstelle (A20200 bis A20207 für die Schnittstellen 0 bis 7) auf EIN gesetzt ist und der TXDU-Befehl-Ausführungsmerker im zugewiesenen DM-Konfigurationsbereich auf AUS gesetzt ist.

Hinweis Die logische Schnittstellennummer kann auch automatisch zugewiesen werden, indem die Bits 12 bis 15 von C+1 auf F gesetzt werden. Weitere Informationen dazu finden Sie unter *Automatische Zuweisung von Kommunikationsschnittstellen* auf Seite 979.

In der folgenden Abbildung ist die Reihenfolge dargestellt, in der Daten gesendet werden, sowie der Inhalt des Sende-Frames bei verschiedenen Start- und Endcode-Einstellungen.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn bei Ausführung des Befehls alle logischen Schnittstellen belegt sind oder der Kommunikationsschnittstelle-Frei-Merker für die spezifizierte logische Schnittstelle auf AUS gesetzt ist. EIN, wenn der Wert von C außerhalb des gültigen Bereichs liegt. EIN, wenn der Wert von N nicht zwischen 0000 und 0100 hex liegt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Befehl TXDU(256) kann nur für eine serielle Schnittstelle einer seriellen Kommunikationsbaugruppe verwendet werden, die auf den protokollfreien Modus eingestellt ist.

Die folgenden Frame-Formate für Sendemeldungen können im zugewiesenen DM-Konfigurationsbereich eingestellt werden.

- Startcode: Keiner oder 00 bis FF hex.
- Endcode: Keiner, CR+LF oder 00 bis FF hex.

Die Daten werden mit einer beliebigen Kombination von Start- und/oder Endcodes gesendet, die im zugewiesenen DM-Konfigurationsbereich spezifiziert ist. Wenn Start- und Endcodes spezifiziert sind, werden diese Codes an die Sendedaten (N) angefügt. In diesem Fall beträgt die maximale Anzahl von Bytes, die für N spezifiziert werden kann, 256.

Es können nur dann Daten gesendet werden, wenn der Sendebereitschafts-Merker der Schnittstelle auf EIN gesetzt ist. (Der Sendebereitschafts-Merker für die RS-232C-Schnittstelle der CPU-Baugruppe hat die Adresse A39205, der für Schnittstelle 1 der seriellen Kommunikationsbaugruppe die Adresse A39605 und der für Schnittstelle 2 der seriellen Kommunikationsbaugruppe die Adresse A39613.)

Die Daten werden in der in C spezifizierten Reihenfolge gesendet.

Wenn für N der Wert 0 spezifiziert ist, wird nichts gesendet.

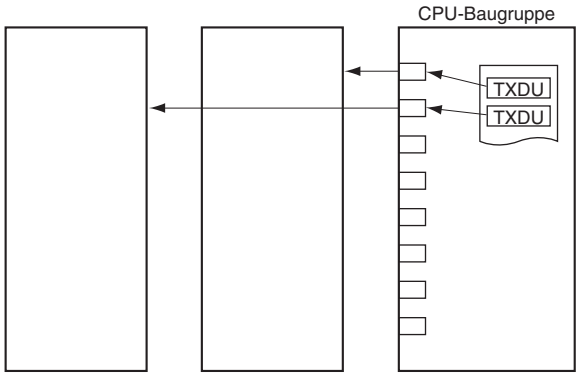
Wenn die RS-Signalsteuerung in C spezifiziert ist, wird Bit 15 von S als RS-Signal verwendet.

Wenn die ER-Signalsteuerung in C spezifiziert ist, wird Bit 14 von S als ER-Signal verwendet.

Wenn die RS- und ER-Signalsteuerung in C spezifiziert ist, wird Bit 15 von S als ER-Signal und Bit 14 von S als ER-Signal verwendet.

Der Befehl TXDU(256) verwendet eine logische Schnittstelle (da er einen internen FINS-Befehl sendet) zur Ausgabe eines Sendesequenz-Befehls an die serielle Kommunikationsbaugruppe (ab Versionsnummer 1.2). Da die Befehle SEND(090), RECV(098), CMND(490), PMCR(260) und RXDU(255) ebenfalls die logischen Schnittstellen 0 bis 7 verwenden, kann der Befehl TXDU(256) nicht für eine logische Schnittstelle ausgeführt werden, wenn diese logische Schnittstelle bereits von einem dieser Befehle oder einem anderen TXDU(256)-Befehl verwendet wird.

Um sicherzustellen, dass TXDU(256) nicht ausgeführt wird, wenn die spezifizierte logische Schnittstelle belegt ist, muss der Kommunikationsschnittstellen-Frei-Merker (A20200 bis A20207) als Schließerbedingung programmiert werden.



Der Befehl TXDU(256) kann nicht ausgeführt werden, während der TXDU-Befehl-Ausführungsmerker (Bit 5 von n+9 bzw. n+19, wobei n = CIO 1500 + 25 × Baugruppennummer) auf EIN gesetzt ist. Um sicherzustellen, dass kein weiterer TXDU(256)-Befehl für die Schnittstelle ausgeführt wird, bevor die Ausführung des ersten TXDU(256)-befehls abgeschlossen ist, muss der TXDU-Befehl-Ausführungsmerker der Schnittstelle als Öffner-Bedingung programmiert werden.

In den folgenden Fällen tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

- Der Kommunikationsschnittstellen-Frei-Merker für die spezifizierte logische Schnittstelle ist bei der Ausführung von TXDU(256) auf AUS gesetzt.
- Der Wert von C liegt außerhalb des gültigen Bereichs.
- Der Wert von N liegt nicht zwischen 0000 und 0100 hex.

Hinweis Abhängig vom externen Gerät ist es möglicherweise erforderlich, eine Sendevorzögerung beim Senden von Daten mittels TXDU(256) einzustellen. Falls eine Sendeverzögerung erforderlich ist, stellen Sie die Verzögerungszeit im zugewiesenen DM-Konfigurationsbereich ein, bzw. passen Sie die vorhandene Einstellung an.

Zugehörige Merker und Worte

Die folgenden Einstellungen im SPS-Setup und Merker im Zusatz-Systembereich können wie erforderlich bei der Ausführung von TXD(236) verwendet werden.

Einstellungen im DM-Konfigurationsbereich

(m = D30000 + 100 × Baugruppennummer)

Konfigurations-bereichs-Wort		Bit	Name	Einstellungen
Schnitt-stelle 1	Schnitt-stelle 2			
m+2	m+12	15	Sendeverzögerungs-Spezifikation für protokollfreien Modus	0: Standardeinstellung (0 ms) 1: Die in den Bits 1 bis 14 eingestellte Verzögerung verwenden.
		0 bis 14	Sendeverzögerungszeit für protokollfreien Modus	0000 bis 7530 hex 0 bis 300.000 ms dezimal (in 10-ms-Schritten)
m+4	m+14	8 bis 15	Startcode für protokollfreien Modus	00 bis FF hex
		0 bis 7	Endcode für protokollfreien Modus	00 bis FF hex
m+5	m+15	12 bis 15	Startcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Startcode verwenden.
		8 bis 11	Endcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Endcode verwenden. 2: CR+LF verwenden.

Zusatz-Systembereich

Name	Adresse	Beschreibung
Kommunikationsschnittstellen-Frei-Merker	A20200 bis A20207	EIN, wenn ein Kommunikationsbefehl (einschließlich TXDU(256)) mit der entsprechenden Schnittstellennummer ausgeführt werden kann. Die Bits 00 bis 07 entsprechen den Kommunikationsschnittstellen 0 bis 7. Der Merker ist auf AUS gesetzt, während ein Kommunikationsbefehl ausgeführt wird, und auf EIN gesetzt, wenn die Ausführung abgeschlossen ist (normaler Abschluss oder Abschluss mit Fehler).
Kommunikationsschnittstellen-Abschluss-Codes	A203 bis A210	Diese Wörter enthalten den Abschluss-Code für die entsprechenden Schnittstellennummern bei Ausführung von Kommunikationsbefehlen. Die Worte A203 bis A210 entsprechen den Kommunikationsschnittstellen 0 bis 7. Während der Befehl ausgeführt wird, ist der Code auf 00 gesetzt. Nach Abschluss der Ausführung enthält er den relevanten Code. Diese Worte werden bei Betriebsstart der SPS auf 0000 gelöscht.
Kommunikationsschnittstellen-Fehlermerker	A219	EIN, wenn bei der Ausführung eines Kommunikationsbefehls ein Fehler auftritt. Wenn ein Merker auf EIN gesetzt ist, muss der Abschlusscode in A203 bis A210 zwecks Fehlersuche geprüft werden. AUS, wenn die Ausführung normal abgeschlossen wurde. Die Bits 00 bis 07 entsprechen den Kommunikationsschnittstellen 0 bis 7. Der Merkerstatus wird bis zur Ausführung des nächsten Kommunikationsbefehls beibehalten. Selbst wenn ein Fehler aufgetreten ist, wird ein Merker bei der nächsten Ausführung eines Kommunikationsbefehls für betreffende Schnittstelle auf 0 zurückgesetzt.

Abschlusscodes

Code	Bedeutung
0205 hex	Antwort-Zeitüberschreitung (Dieser Fehler kann auftreten, wenn der Kommunikationsmodus auf den Host-Link-Modus eingestellt ist.)
0401 hex	Nicht definierter Befehl (Dieser Fehler kann auftreten, wenn der Kommunikationsmodus auf Protokollmakro, NT-Link, Echoback-Test oder den Seriell-Gateway-Modus eingestellt ist.)
1001 hex	Der Befehl ist zu lang.
1002 hex	Der Befehl ist zu kurz.
1003 hex	Die spezifizierte Anzahl von Datenelementen stimmt nicht mit der tatsächlichen Menge der Sendedaten überein.
1004 hex	Das Befehlsformat ist nicht korrekt.
110C hex	Sonstiger Parameterfehler
2201 hex	Die Funktion konnte nicht während der Funktion ausgeführt werden. (Funktion ist deaktiviert, da die Baugruppe mit Senden ausgelastet ist.)
2202 hex	Die Funktion konnte nicht während gestoppten Betriebs ausgeführt werden. (Funktion ist deaktiviert, da das Protokoll für die Baugruppe gewechselt wird.)

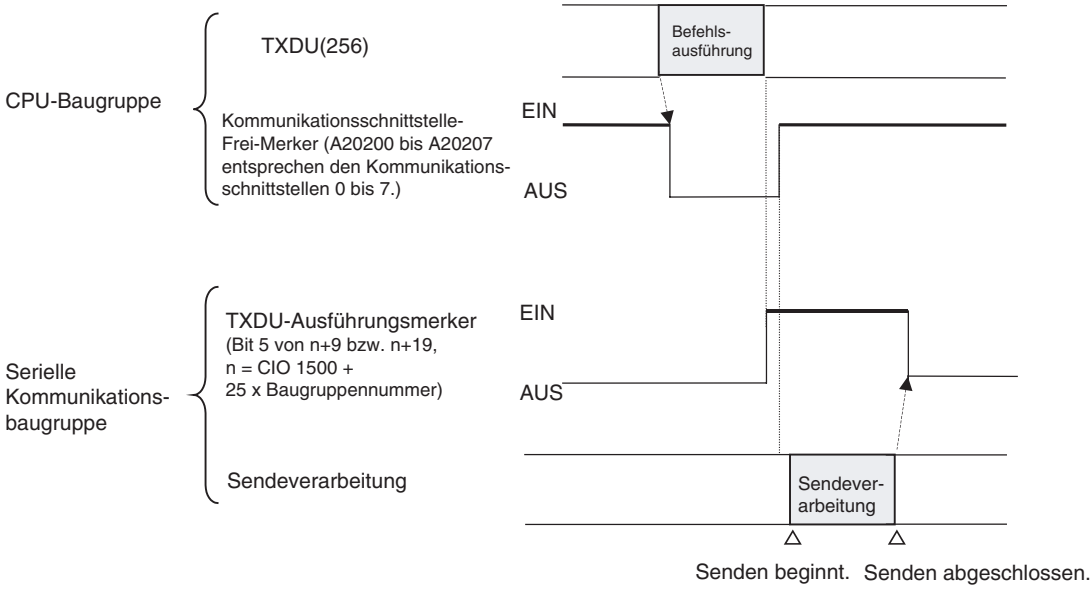
Zugehörige Merker im CPU-Bus-Baugruppenbereich

(n = CIO 1500 + 25 × Baugruppennummer)

Wort		Bit	Name	Status
Schnittstelle 1	Schnittstelle 2			
n+9	n+19	05	TXDU-Befehl-Ausführungsmerker	0: TXDU(256) wird nicht ausgeführt. 1: TXDU(256) wird ausgeführt.

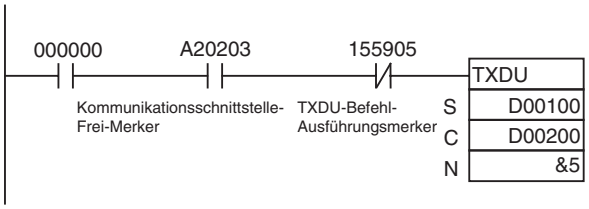
Beispiel: Merkerfunktion

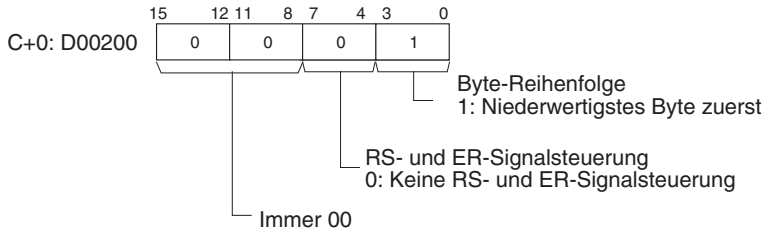
In der folgenden Abbildung sind die Funktionen des Kommunikationsschnittstelle-Frei-Merkers sowie des TXDU-Befehl-Ausführungsmerkers dargestellt.

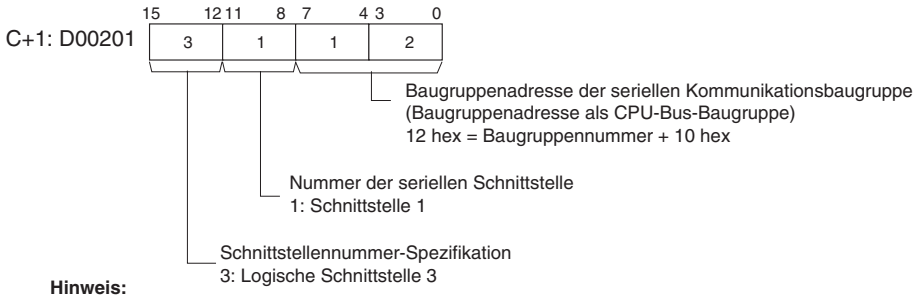


Beispiel: Senden von Daten

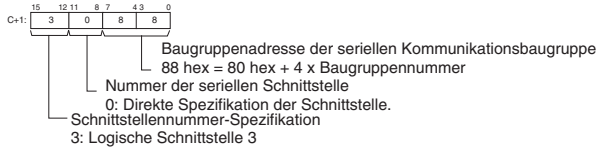
Wenn CIO 000000 auf EIN gesetzt ist, ist im folgenden Beispiel A20203 (der Kommunikationsschnittstelle-Frei-Merker) auf EIN und CIO 155905 (der TXDU-Befehl-Ausführungsmerker für Schnittstelle 1) auf AUS gesetzt. Der Befehl TXDU(256) gibt Daten über die serielle Schnittstelle 1 der seriellen Kommunikationsbaugruppe mit Baugruppennummer 2 aus. Die 5 Bytes Ausgabedaten werden aus dem DM-Bereich mit den niederwertigsten Byte von D00100 beginnend gelesen und über die logische Schnittstelle 3 an ein allgemeines Gerät (z. B. einen Drucker) ausgegeben.

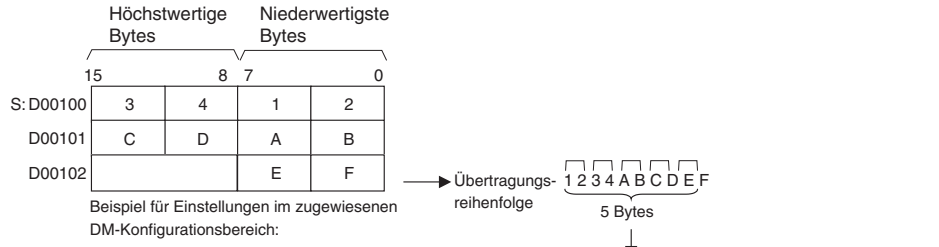






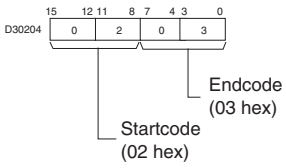
Hinweis:
Die Adresse der Baugruppe mit der seriellen Schnittstelle kann direkt spezifiziert werden, indem die Nummer der seriellen Schnittstelle auf 0 und die Baugruppenadresse der seriellen Kommunikationsbaugruppe auf die Adresse der Baugruppe mit der seriellen Schnittstelle eingestellt wird.
(Stellen Sie die Baugruppenadresse auf 80 hex +4 x Baugruppennummer für Schnittstelle 1 bzw. 81 hex +4 x Baugruppennummer für Schnittstelle 2 ein.)



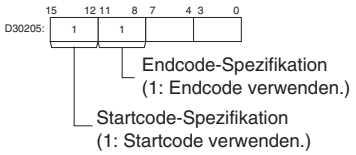


Beispiel für Einstellungen im zugewiesenen DM-Konfigurationsbereich:

Werte für Start- und Endcode



Spezifikationen für Start- und Endcode



In diesem Beispiel wurden ein Start- und ein Endcode im zugewiesenen DM-Konfigurationsbereich spezifiziert.

ST	12	34	AB	CD	EF	ED

ST: Startcode (z. B. 02 hex)
ED: Endcode (z. B. 03 hex)

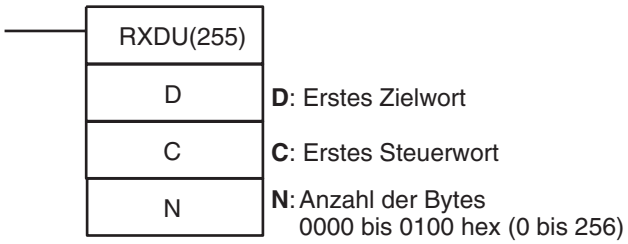
Daten gesendet.

3-24-6 RECEIVE VIA SERIAL COMMUNICATIONS UNIT: RXDU(255)

Zweck

List die spezifizierte Anzahl von Datenbytes von einer der seriellen Schnittstellen der seriellen Kommunikationsbaugruppe. (Die serielle Kommunikationsbaugruppe muss mindestens die Version 1.2 haben.)

Kontaktplansymbol



Variationen

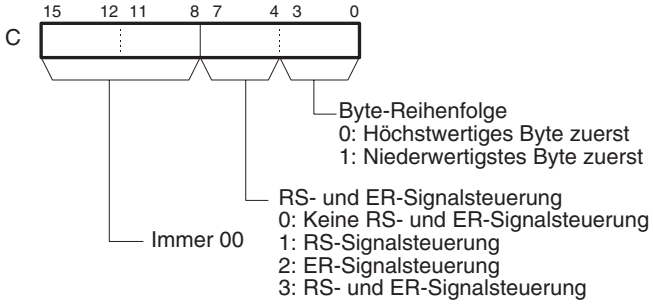
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	RXDU(255)
	Ausführung einmal bei steigender Flanke	@RXDU(255)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

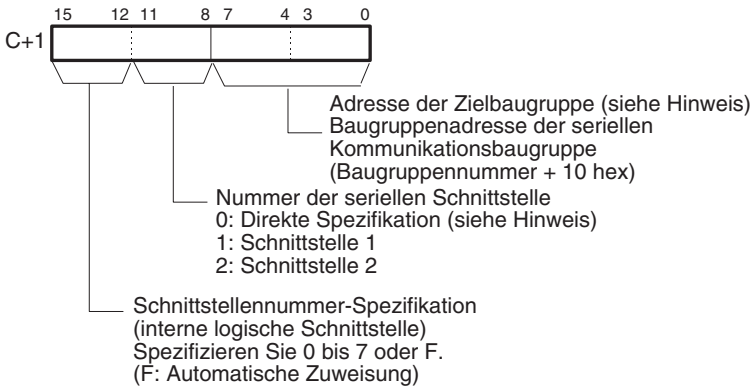
Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Der Inhalt der Steuerworte C und C+1 ist nachfolgend beschrieben.





Hinweis Die Adresse der Baugruppe mit der seriellen Schnittstelle kann direkt spezifiziert werden, indem die Nummer der seriellen Schnittstelle auf 0 und die Adresse der Zielbaugruppe auf die Adresse der Baugruppe mit der seriellen Schnittstelle eingestellt wird. (Stellen Sie die Adresse der Zielbaugruppe auf 80 hex + 4 × Baugruppennummer für Schnittstelle 1 bzw. 81 hex + 4 × Baugruppennummer für Schnittstelle 2 ein.)

Operanden-Spezifikationen

Bereich	D	C	D
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6143
Arbeitsbereich	W000 bis W511	W000 bis W510	W000 bis W511
Haftmerkerbereich	H000 bis H511	H000 bis H510	H000 bis H511
Zusatz-Systembereich	A000 bis A959	A000 bis A958	A000 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4094	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4094	C0000 bis C4095
DM-Bereich	D00000 bis D32767	D00000 bis D32766	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32766	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	Nur spezifizierte Werte	#0000 bis #0100 (binär) oder &0 bis &256 (dezimal)
Datenregister	---	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl RXDU(255) liest die Daten, die im protokollfreien Modus von der seriellen Kommunikationsbaugruppe mit der in den Bits 0 bis 7 von C+1 Baugruppenadresse über die in den Bits 8 bis 11 von C+1 spezifizierte Schnittstelle empfangen wurden, ein und speichert diese Daten in den mit D beginnenden Worten. Wenn weniger als N Bytes Daten über die Schnittstelle empfangen wurden, werden nur die empfangenen Daten gespeichert. Die Nummer der logischen Schnittstelle kann auf einen beliebigen Wert zwischen 0 und 7 eingestellt werden. Sie wird in den Bits 12 bis 15 von C+1 spezifiziert.

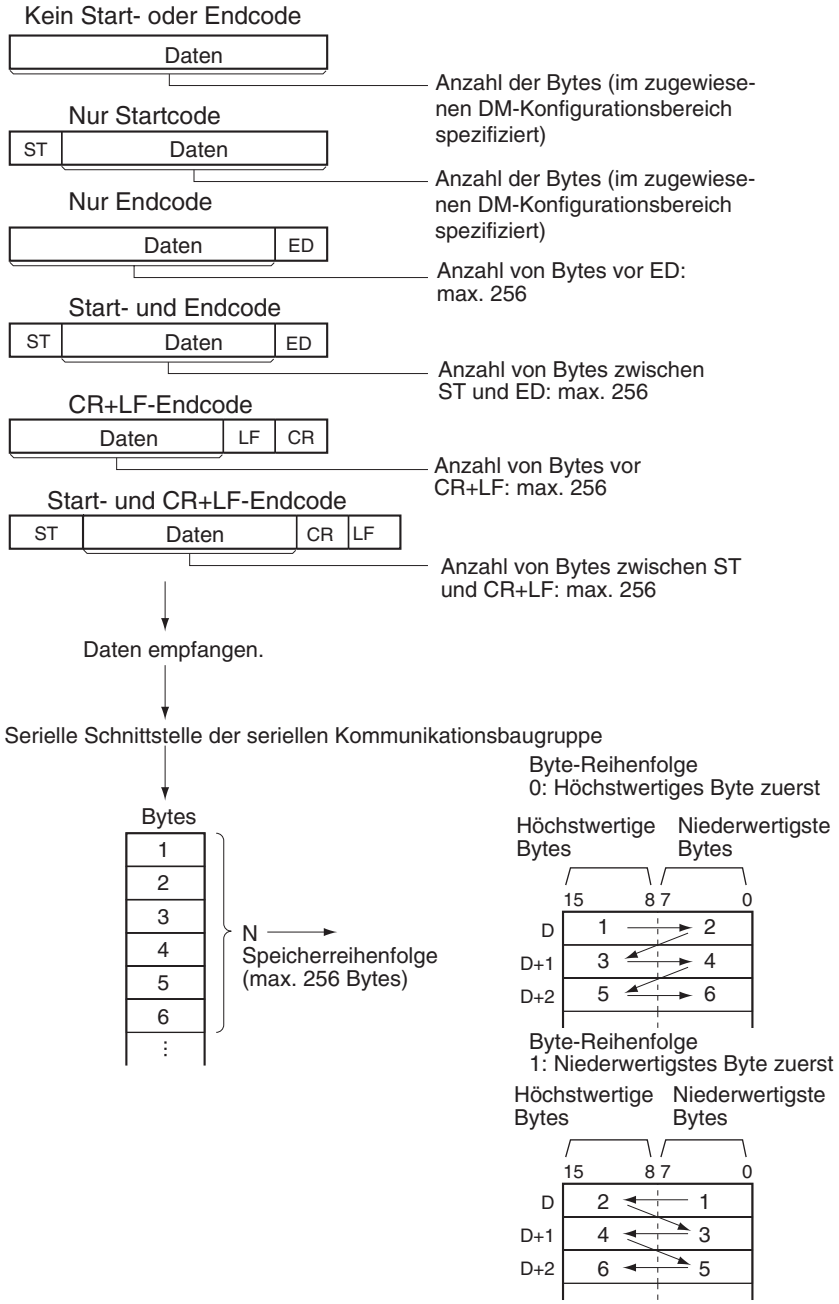
Führen Sie den Befehl RXDU(255) aus, um die empfangenen Daten aus dem Empfangspuffer zu lesen, wenn der Empfang-Abgeschlossen-Merker (im zugewiesenen DM-Konfigurationsbereich) auf EIN gesetzt ist.

Es können bis zu 259 Bytes empfangen werden, dazu gehören die Empfangsdaten (N = maximal 256 Bytes), der Startcode und der Endcode.

In der folgenden Abbildung ist die Reihenfolge dargestellt, in der die Daten empfangen werden, sowie der Inhalt des Empfangs-Frames bei verschiedenen Einstellungen.

Hinweis Die logische Schnittstellennummer kann auch automatisch zugewiesen werden, indem die Bits 12 bis 15 von C+1 auf F gesetzt werden. Weitere Informationen dazu finden Sie unter *Automatische Zuweisung von Kommunikationsschnittstellen* auf Seite 979.

In der folgenden Abbildung ist die Reihenfolge dargestellt, in der Daten gesendet werden, sowie der Inhalt des Sende-Frames bei verschiedenen Start- und Endcode-Einstellungen.



Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn bei Ausführung des Befehls alle logischen Schnittstellen belegt sind oder der Kommunikationsschnittstelle-Frei-Merker für die spezifizierte logische Schnittstelle auf AUS gesetzt ist. EIN, wenn der Wert von C außerhalb des gültigen Bereichs liegt. EIN, wenn der Wert von N nicht zwischen 0000 und 0100 hex liegt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Befehl RXDU(255) kann nur für eine serielle Schnittstelle einer seriellen Kommunikationsbaugruppe verwendet werden, die auf den protokollfreien Modus eingestellt ist.

Die folgenden Frame-Formate für Empfangsmeldungen können im zugewiesenen DM-Konfigurationsbereich eingestellt werden.

- Startcode: Keiner oder 00 bis FF hex.
- Endcode: Keiner, CR+LF oder 00 bis FF hex. Wenn keine Endcode spezifiziert ist, kann die Anzahl für zu empfangene Bytes auf 00 bis FF hex (1 bis 256 dezimal; 00 spezifiziert 256 Bytes) eingestellt werden.

Der Empfang-Abgeschlossen-Merker (Hinweis 1) wird auf EIN gesetzt, wenn die im zugewiesenen DM-Konfigurationsbereich spezifizierte Anzahl von Bytes empfangen wurde. Wenn der Empfang-Abgeschlossen-Merker auf EIN gesetzt wird, hat die Anzahl der Bytes im Empfangszähler (Hinweis 2) den gleichen Wert wie die Anzahl der im zugewiesenen DM-Konfigurationsbereich spezifizierten Empfangsbytes. Werden mehr Bytes als spezifiziert empfangen, so wird der Empfangsüberlauf-Merker (Hinweis 3) auf EIN gesetzt.

Wenn im zugewiesenen DM-Konfigurationsbereich ein Endcode spezifiziert ist, wird der Empfang-Abgeschlossen-Merker (Hinweis 1) auf EIN gesetzt, wenn der Endcode empfangen wird oder wenn 256 Bytes Daten empfangen wurden. Wenn weitere Daten empfangen werden, nachdem der Empfang-Abgeschlossen-Merker (Hinweis 1) auf EIN gesetzt ist, wird der Empfangsüberlauf-Merker (Hinweis 3) auf EIN gesetzt.

Der Empfang wird gestoppt, nachdem 259 Bytes Daten empfangen wurden. Wenn danach weitere Daten eingegeben werden, werden der Überlauf-Fehlermerker (Hinweis 4) und der Übertragungs-Fehlermerker (Hinweis 5) auf EIN gesetzt.

Wenn mehr Daten als in N spezifiziert an der seriellen Schnittstelle der seriellen Kommunikationsbaugruppe eingegeben werden, werden diese Daten bei der Ausführung des nächsten RXDU(255)-Befehls verworfen.

Wenn der Befehl RXDU(255) ausgeführt wird, werden die Daten in den mit D beginnenden Worten gespeichert, der Empfang-Abgeschlossen-Merker (Hinweis 1) wird auf AUS gesetzt (selbst falls der Empfangsüberlauf-Merker (Hinweis 3) auf EIN gesetzt ist) und der Empfangszähler (Hinweis 2) wird auf 0 zurückgesetzt.

Die Daten werden in der in C spezifizierten Reihenfolge im Speicher gespeichert.

Ist für N der Wert 0 spezifiziert, wird der Empfang-Abgeschlossen-Merker (Hinweis 1) auf AUS gesetzt, der Empfangsüberlauf-Merker (Hinweis 3) wird auf AUS gesetzt, der Empfangszähler (Hinweis 2) wird auf 0 zurückgesetzt und es werden keine Daten in den Speicher geschrieben.

Wenn die CS-Signalüberwachung in C spezifiziert ist, wird der Status des CS-Signals in Bit 15 von D gespeichert.

Wenn die DR-Signalüberwachung in C spezifiziert ist, wird der Status des DR-Signals in Bit 14 von D gespeichert.

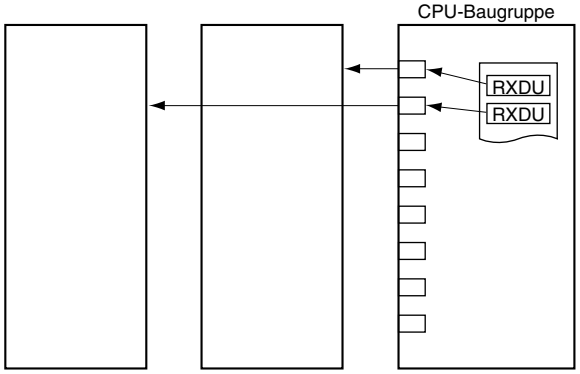
Wenn die CS- und DR-Signalüberwachung in C spezifiziert ist, wird der Status des CS-Signals in Bit 15 von D und der Status des DR-Signals in Bit 14 von D gespeichert.

Falls CS- oder DR-Signalüberwachung spezifiziert ist, werden keine Daten gespeichert.

Wenn für die RS- und DR-Signalsteuerung in C der Wert 1, 2 oder 3 hex spezifiziert ist, wird der Befehl RXDU(255) ungeachtet des Status des Empfang-Abgeschlossen-Merkers (Hinweis 1) ausgeführt.

Der Befehl RXDU(255) verwendet eine logische Schnittstelle (da er einen internen FINS-Befehl sendet) zur Ausgabe eines Empfangssequenz-Befehls an eine serielle Kommunikationsbaugruppe oder ein serielles Kommunikationsmodul der CS-Serie. Da die Befehle SEND(090), RECV(098), CMND(490), PMCR(260) und TXDU(256) ebenfalls die logischen Schnittstellen 0 bis 7 verwenden, kann der Befehl RXDU(255) nicht für eine logische Schnittstelle ausgeführt werden, wenn diese logische Schnittstelle bereits von einem dieser Befehle oder einem anderen RXDU(255)-Befehl verwendet wird.

Um sicherzustellen, dass der Befehl RXDU(255) nicht ausgeführt wird, wenn die spezifizierte logische Schnittstelle belegt ist, muss der Kommunikationsschnittstellen-Frei-Merker (A20200 bis A20207) als Schließerbedingung programmiert werden.



Der Befehl RXDU(255) kann nicht ausgeführt werden, während der Empfang-Abgeschlossen-Merker (Bit 6 von n+9 bzw. n+19, wobei n = CIO 1500 + 25 × Baugruppennummer) auf EIN gesetzt ist. Programmieren Sie den Empfang-Abgeschlossen-Merker als Schließer-Ausführungsbedingung für RXDU(255).

In den folgenden Fällen tritt ein Fehler auf und der Fehlermerker wird auf EIN gesetzt.

- Der Kommunikationsschnittstellen-Frei-Merker für die spezifizierte logische Schnittstelle ist bei der Ausführung von RXDU(255) auf AUS gesetzt.
- Der Wert von C liegt außerhalb des gültigen Bereichs.
- Der Wert von N liegt nicht zwischen 0000 und 0100 hex.

Hinweis

1. Empfang-Abgeschlossen-Merker
   (n = CIO 1500 + 25 × Baugruppennummer)
   Schnittstelle 1: Bit 6 von n+9
   Schnittstelle 2: Bit 6 von n+19
2. Empfangszähler (n = CIO 1500 + 25 × Baugruppennummer)
   Schnittstelle 1: n+10
   Schnittstelle 2: n+20
3. Empfangsüberlauf-Merker (n = CIO 1500 + 25 × Baugruppennummer)
   Schnittstelle 1: Bit 7 von n+9
   Schnittstelle 2: Bit 7 von n+19
4. Überlauf-Fehlermerker (n = CIO 1500 + 25 × Baugruppennummer)
   Schnittstelle 1: Bit 4 von n+8
   Schnittstelle 2: Bit 4 von n+18
5. Übertragungs-Fehlermerker (n = CIO 1500 + 25 × Baugruppennummer)
   Schnittstelle 1: Bit 15 von n+8
   Schnittstelle 2: Bit 15 von n+18
6. Es können keine weiteren Daten empfangen werden, bis die Empfangsdaten durch den Befehl RXDU(255) aus dem Empfangspuffer ausgelesen sind. Wenn der Empfang-Abgeschlossen-Merker auf EIN gesetzt wird, sollten die Daten unverzüglich durch RXDU(255) ausgelesen werden, bevor weitere Daten über die Schnittstelle eingegeben werden.
7. Wenn der Befehl RXDU(255) verwendet wird, um die Daten zu lesen, die über eine der Schnittstellen der seriellen Kommunikationsbaugruppe empfangen wurden, wird der Empfangspuffer der Schnittstelle nach der Ausführung von RXDU(255) gelöscht. Demzufolge kann der Befehl RXDU(255) **nicht** wiederholt ausgeführt werden, um einen Datenblock stückweise zu lesen.

Zugehörige Merker und Worte

Die folgenden Merker stehen in Beziehung zur Funktion des Befehls RXDU(255).

Einstellungen im DM-Konfigurationsbereich

(m = D30000 + 100 × Baugruppennummer)

Konfigurations-bereichs-Wort		Bit	Name	Einstellungen
Schnitt-stelle 1	Schnitt-stelle 2			
m+4	m+14	8 bis 15	Startcode für protokollfreien Modus	00 bis FF hex
		0 bis 7	Endcode für protokollfreien Modus	00 bis FF hex
m+5	m+15	12 bis 15	Startcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Startcode verwenden.
		8 bis 11	Endcode-Spezifikation für protokollfreien Modus	0: Ohne 1: Endcode verwenden. 2: CR+LF verwenden.

Zusatz-Systembereich

Name	Adresse	Beschreibung
Kommunikationsschnittstellen-Frei-Merker	A20200 bis A20207	EIN, wenn ein Kommunikationsbefehl (einschließlich RXDU(255)) mit der entsprechenden Schnittstellennummer ausgeführt werden kann. Die Bits 00 bis 07 entsprechen den Kommunikationsschnittstellen 0 bis 7. Der Merker ist auf AUS gesetzt, während ein Kommunikationsbefehl ausgeführt wird, und auf EIN gesetzt, wenn die Ausführung abgeschlossen ist (normaler Abschluss oder Abschluss mit Fehler).
Kommunikationsschnittstellen-Abschluss-Codes	A203 bis A210	Diese Wörter enthalten den Abschluss-Code für die entsprechenden Schnittstellennummern bei Ausführung von Kommunikationsbefehlen. Die Worte A203 bis A210 entsprechen den Kommunikationsschnittstellen 0 bis 7. Während der Befehl ausgeführt wird, ist der Code auf 00 gesetzt. Nach Abschluss der Ausführung enthält er den relevanten Code. Diese Worte werden bei Betriebsstart der SPS auf 0000 gelöscht.
Kommunikationsschnittstellen-Fehlermerker	A219	EIN, wenn bei der Ausführung eines Kommunikationsbefehls ein Fehler auftritt. Wenn ein Merker auf EIN gesetzt ist, muss der Abschlusscode in A203 bis A210 zwecks Fehlersuche geprüft werden. AUS, wenn die Ausführung normal abgeschlossen wurde. Die Bits 00 bis 07 entsprechen den Kommunikationsschnittstellen 0 bis 7. Der Merkerstatus wird bis zur Ausführung des nächsten Kommunikationsbefehls beibehalten. Selbst wenn ein Fehler aufgetreten ist, wird ein Merker bei der nächsten Ausführung eines Kommunikationsbefehls für betreffende Schnittstelle auf 0 zurückgesetzt.

Abschlusscodes

Code	Bedeutung
0205 hex	Antwort-Zeitüberschreitung (Dieser Fehler kann auftreten, wenn der Kommunikationsmodus auf den Host-Link-Modus eingestellt ist.)
0401 hex	Nicht definierter Befehl (Dieser Fehler kann auftreten, wenn der Kommunikationsmodus auf Protokollmakro, NT-Link, Echoback-Test oder den Seriell-Gateway-Modus eingestellt ist.)
1001 hex	Der Befehl ist zu lang.
1002 hex	Der Befehl ist zu kurz.
1004 hex	Das Befehlsformat ist nicht korrekt.

Code	Bedeutung
110C hex	Sonstiger Parameterfehler
2201 hex	Die Funktion konnte nicht während der Funktion ausgeführt werden. (Funktion ist deaktiviert, da die Baugruppe mit Senden ausgelastet ist.)
2202 hex	Die Funktion konnte nicht während gestoppten Betriebs ausgeführt werden. (Funktion ist deaktiviert, da das Protokoll für die Baugruppe gewechselt wird.)

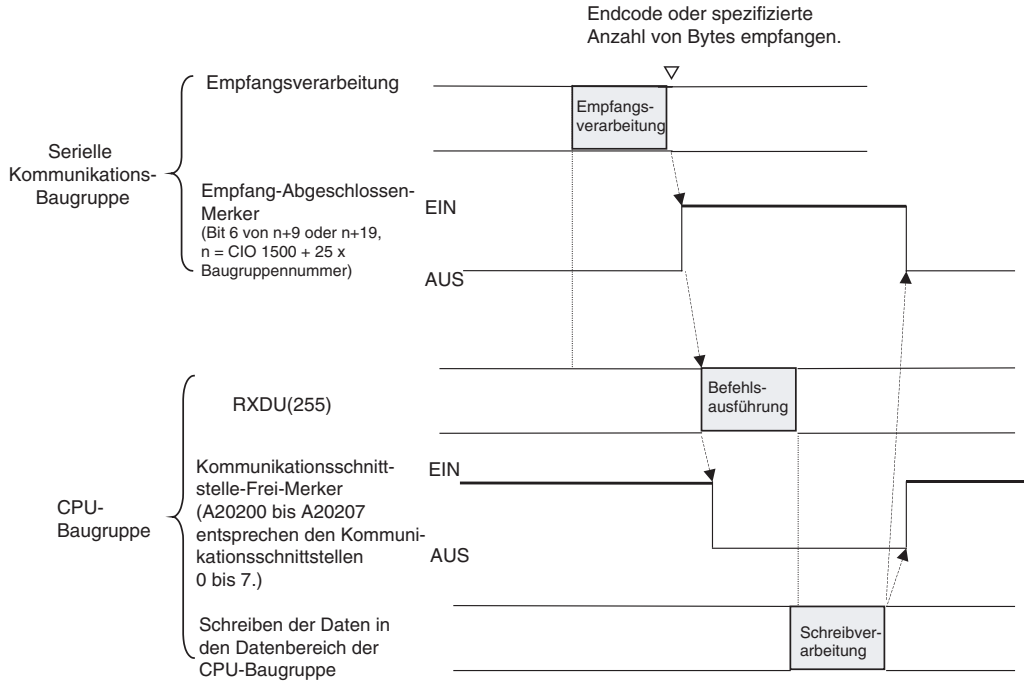
Zugehörige Merker im CPU-Bus-Baugruppenbereich

(n = CIO 1500 + 25 × Baugruppennummer)

Wort		Bit	Funktion
Schnitt-stelle 1	Schnitt-stelle 2		
n+8	n+18	04	Überlauf-Fehlermerker 1: Der Empfangspuffer enthielt vor der Ausführung des Befehls RXDU(255) mehr als 259 Bytes Daten. Note: Sobald dieser Fehlermerker auf EIN gesetzt ist, kann er nur wieder auf AUS gesetzt werden, indem die Spannungsversorgung aus- und wieder eingeschaltet wird oder das Modul neu gestartet wird.
n+9	n+19	06	Empfang-Abgeschlossen-Merker 0: Keine Daten empfangen oder es werden derzeit Daten empfangen 1: Empfang abgeschlossen 0 → 1: Das Modul bzw. die Baugruppe hat die spezifizierte Anzahl von Bytes empfangen. 1 → 0: RXD(235) oder RXDU(255) wurde ausgeführt, um die Daten aus dem Empfangspuffer in einen Datenbereich der CPU-Baugruppe zu schreiben.
n+9	n+19	07	Empfangsüberlauf-Merker 0: Das Modul bzw. die Baugruppe hat nicht mehr als die spezifizierte Anzahl von Bytes empfangen. 1: Das Modul bzw. die Baugruppe hat mehr als die spezifizierte Anzahl von Bytes empfangen. 0 → 1: Das Modul bzw. die Baugruppe hat nach Abschluss des Datenempfangs weitere Daten empfangen. 1 → 0: RXD(235) oder RXDU(255) wurde ausgeführt, um die Daten aus dem Empfangspuffer in einen Datenbereich der CPU-Baugruppe zu schreiben.
n+10	n+20	05	Empfangszähler Gibt die Anzahl der empfangenen Bytes als Hexadezimalzahl zwischen 0000 und 0100 hex (0 bis 256 dezimal) an.

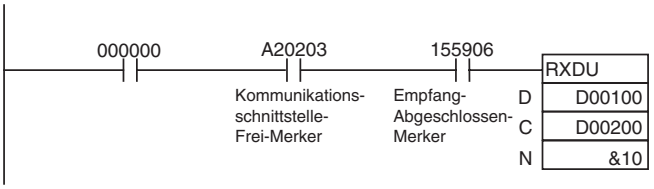
Beispiel: Merkerfunktion

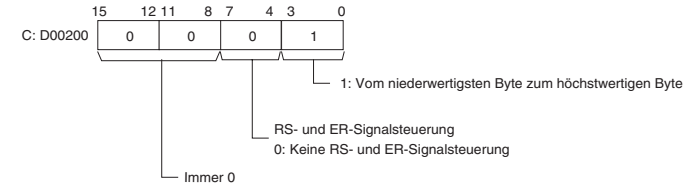
In der folgenden Abbildung ist die Funktion des Befehls RXDU(255) und die der zugehörigen Merker dargestellt.

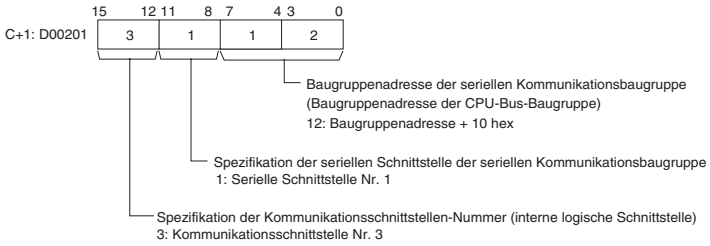


Beispiel: Empfangen von Daten

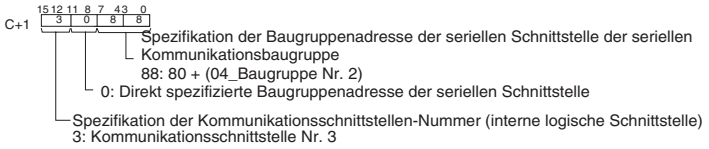
Wenn CIO 000000 auf EIN gesetzt ist, ist im folgenden Beispiel A20203 (der Kommunikationsschnittstelle-Frei-Merker) auf EIN und CIO 155906 (der Empfang-Abgeschlossen-Merker für Schnittstelle 1) auf AUS gesetzt. Der Befehl RXDU(255) liest die Daten, über die serielle Schnittstelle 1 der seriellen Kommunikationsbaugruppe mit Baugruppennummer 2 empfangen wurden. (Die logische Kommunikationsschnittstelle Nummer 3 wird zum Empfangen von Daten von einem allgemeinen Gerät, wie einem Barcode-Leser, verwendet.) Die 10 Bytes empfangener Daten werden mit dem niederwertigsten Byte von D00100 beginnend in den DM-Bereich geschrieben.

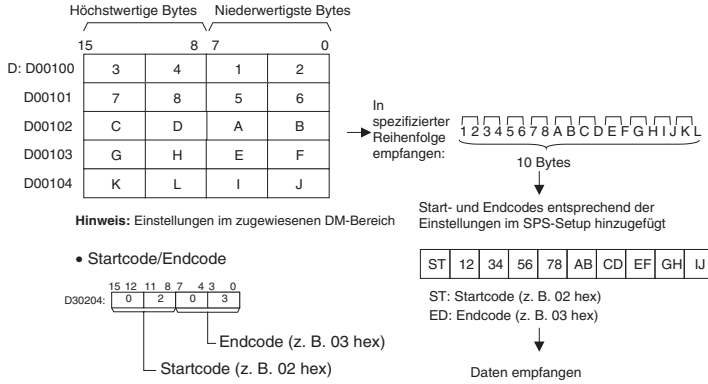




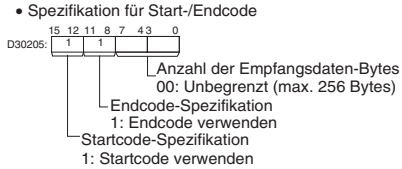


Hinweis: Die Baugruppenadresse der seriellen Schnittstelle der seriellen Kommunikationsbaugruppe kann auch direkt in C+1 spezifiziert werden.





Hinweis: Einstellungen im zugewiesenen DM-Bereich

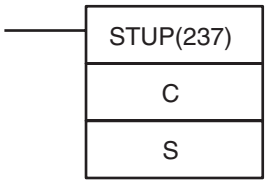


3-24-7 CHANGE SERIAL PORT SETUP: STUP(237)

Zweck

Ändert die Kommunikationsparameter einer seriellen Schnittstelle der CPU-Baugruppe, eines seriellen Kommunikationsmoduls (nur CS-Serie) oder einer seriellen Kommunikationsbaugruppe (CPU-Bus-Baugruppe). Damit ermöglicht STUP(237) die Änderung des Protokollmodus während des SPS-Betriebs.

Kontaktplansymbol



C: Steuerwort (Schnittstelle)

S: Erstes Quellenwort

Variationen

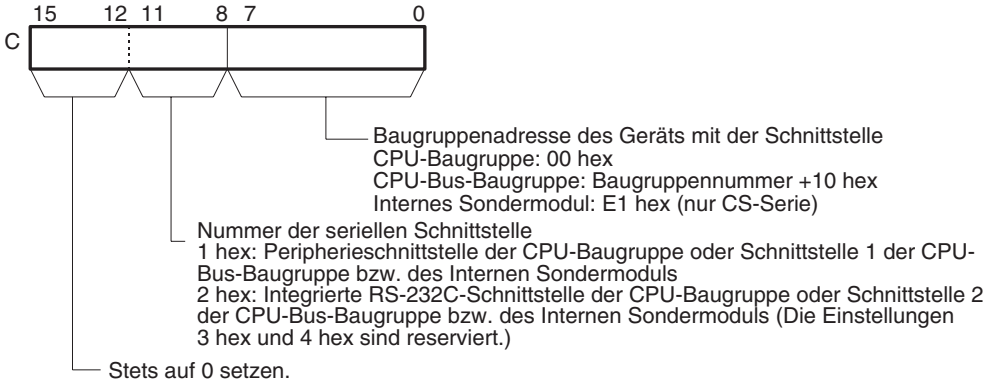
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	STUP(237)
	Ausführung einmal bei steigender Flanke	@STUP(237)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit auf Programmbereiche

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	Nicht zulässig

Operanden

Die Inhalte des Steuerworts C sind nachfolgend beschrieben.



Operanden-Spezifikationen

Bereich	C	S
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6134
Arbeitsbereich	W000 bis W511	W000 bis W502
Haftmerkerbereich	H000 bis H511	H000 bis H502
Zusatz-Systembereich	A000 bis A438 A448 bis A959	A000 bis A438 A448 bis A950
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4086
Zählerbereich	C0000 bis C4095	C0000 bis C4086
DM-Bereich	D00000 bis D32767	D00000 bis D32758
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32758
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32758 (n = 0 bis C)
Indirekte DM/EM-Binäradresse	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adresse in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	

Bereich	C	S
Konstanten	Nur spezifizierte Werte	#0000
Datenregister	DR0 bis DR15	---
Indexregister	---	
Indirektes Adressieren mittels Index-Registern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl STUP(237) schreibt die 10 Worte Daten von S bis S+9 in den Kommunikations-Konfigurationsbereich der Baugruppe mit der spezifizierten Baugruppenadresse (siehe folgende Tabelle). Wenn für S die Konstante #0000 spezifiziert wird, werden die Kommunikationseinstellungen der entsprechenden Schnittstelle auf die Werkseinstellungen gesetzt.

Baugruppen-adresse	Baugruppe	Schnitt-stellen-nummer	Serielle Schnittstelle	Kommunikations-Konfigurationsbereich der seriellen Schnittstelle
00 hex	CPU-Baugruppe	1 hex	Schnittstelle 1	Kommunikationsparameter für die Peripherieschnittstelle im SPS-Setup
		2 hex	Schnittstelle 2	Kommunikationsparameter für die RS-232C-Schnittstelle im SPS-Setup
Baugruppen-Nr. + 10 hex	Serielle Kommunikationsbaugruppe (CPU Bus-Baugruppe)	1 hex	Schnittstelle 1	10 Worte, beginnend mit D30000 + 100 x Baugruppen-Nr.
		2 hex	Schnittstelle 2	10 Worte, beginnend mit D30000 + 100 x Baugruppen-Nr. + 10
E1 hex	Serielle Kommunikationsmodul (internes Sondermodul) (nur CS-Serie)	1 hex	Schnittstelle 1	10 Worte, beginnend mit D32000
		2 hex	Schnittstelle 2	10 Worte, beginnend mit D32010

Wenn der Befehl STUP(237) ausgeführt wird, wird der entsprechende Schnittstellenparameter-Änderungsmerker (A61901, A61902 oder A619 bis A636) auf EIN gesetzt. Dieser Merker bleibt auf EIN gesetzt, bis die Änderung der Parameter abgeschlossen ist.

Verwenden Sie den Befehl STUP(237) zur Änderung der Kommunikationsparameter einer Schnittstelle während des laufenden Betriebs auf Grundlage spezifizierter Bedingungen. Beispielsweise kann der Befehl STUP(237) verwendet werden, um zur Host-Link-Kommunikation zu wechseln, damit eine Überwachung und Programmierung von einem Host-Computer möglich ist, wenn während der Ausführung einer Kommunikationssequenz für eine Modemverbindung spezifizierte Bedingungen erfüllt sind.

Unterschiede zwischen CPU-Baugruppen

Wenn die SPS nach Verwendung des Befehls STUP(237) zum Ändern der Kommunikationsparameter aus- und wieder eingeschaltet wird, ist es von der jeweiligen CPU-Baugruppe abhängig, ob die neuen Parameter erhalten bleiben oder eine Rückkehr zu den vorherigen Parametern stattfindet.

CPU-Baugruppe	Status der Kommunikationsparameter
CS1-H, CJ1-H, CJ1M oder CS1D	Wenn die SPS aus- und wieder eingeschaltet wird, kehren die Kommunikationsparameter wieder zu den Einstellungen zurück, die vor der Änderung durch STUP(237) vorhanden waren.
CS1	Wenn die SPS aus- und wieder eingeschaltet wird, bleiben die durch STUP(237) eingestellten Kommunikationsparameter erhalten.

Merker

Name	Bezeichnung	Funktion
Fehlermerker	ER	EIN, wenn die Werte in C außerhalb des gültigen Bereichs liegen. EIN, wenn der Befehl STUP(237) für eine Schnittstelle ausgeführt wird, deren Kommunikationsparameter-Änderungsmerker bereits auf EIN gesetzt ist. EIN, wenn STUP(237) in einer Interrupt-Task ausgeführt wird. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die Kommunikationsparameter bestehen aus Protokollmodus, Baudrate, Datenformat (Protokollmakro-Übertragungsmethode und maximale Protokollmakro-Kommunikationsdatenlänge) sowie weiteren Parametern. Detaillierte Informationen zu der Schnittstelle, deren Parameter eingestellt werden sollen, finden Sie im *CS/CJ-Serie Programmierbare Steuerungen Bedienerhandbuch (W339)* oder im *CS/CJ-Serie Serielle Kommunikationsmodule und serielle Kommunikationsbaugruppen Bedienerhandbuch (W336)*.

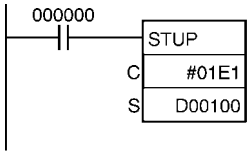
Zugehörige Merker und Worte

Die folgenden Merker können bei Bedarf bei der Ausführung von STUP(237) verwendet werden. Diese Merker befinden sich im Zusatz-Systembereich.

Name	Adresse	Inhalt
Peripherieschnittstellen-Parameter-Änderungsmerker	A61901	EIN, während die Kommunikationsparameter für die Peripherieschnittstelle geändert werden.
RS-232C-Schnittstellenparameter-Änderungsmerker	A61902	EIN, während die Kommunikationsparameter für die RS-232C-Schnittstelle geändert werden.
Schnittstellenparameter-Änderungsmerker für die Schnittstellen 1 bis 4 der seriellen Kommunikationsbaugruppen 1 bis 15.	A620 Bit 01 bis Bit 04 bis A635 Bit 01 bis Bit 04	EIN, während die Kommunikationsparameter für eine Schnittstelle einer seriellen Kommunikationsbaugruppe geändert werden.
Schnittstellenparameter-Änderungsmerker für die Schnittstellen 1 bis 4 des seriellen Kommunikationsmoduls (nur CS-Serie)	A63601 bis A63604	EIN, während die Kommunikationsparameter für eine Schnittstelle eines seriellen Kommunikationsmoduls geändert werden.

Beispiele

Wenn CIO 000000 in den folgenden Beispiel auf EIN gesetzt wird, werden die Kommunikationsparameter für die serielle Schnittstelle 1 des seriellen Kommunikationsmoduls (internes Sondermodul) auf die Einstellungen geändert, die in den 10 Worten von D00100 bis D00109 enthalten sind. In diesem Beispiel werden die Einstellungen für den Protokollmodus auf den Protokollmakro-Modus geändert.



S: D00100	0	6	0	0	Schnittstelleneinstellung: Vorgabe, Protokollmodus: 6 hex (Protokollmakro).
S+1: D00101	0	0	0	0	Baudrate: Vorgabe (9.600 Bit/s)
S+2: D00102					
bis		bis			
S+9: D00199					

Übertragung

DM-Worte, die der Kommunikationskonfiguration des seriellen Kommunikationsmoduls zugewiesen sind.

D32000	0	6	0	0
D32001	0	0	0	0
D32002				
bis		bis		
D32009				

3-25 Netzwerkbefehle

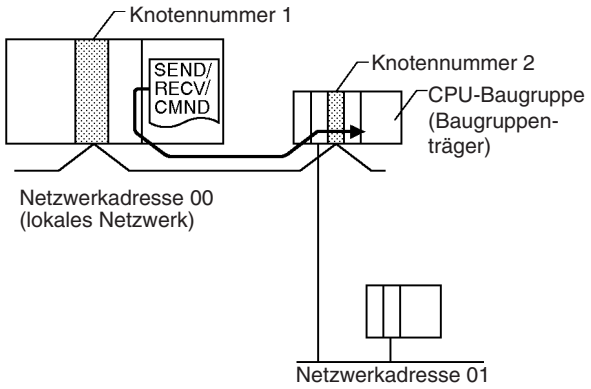
3-25-1 Informationen zur Funktionsweise von SYSMAC NET Link/ SYSMAC LINK

Man unterscheidet zwei Typen von Netzwerkbefehlen, SEND(090)/ RECV(098) und CMND(490). Diese Befehle werden zwischen Baugruppen (CPU-Baugruppen, CPU-Bus-Baugruppen und Computern) in einem Netzwerk gesendet, um Daten und Steuerungsfunktionen zu übertragen, wie z. B. das Ändern der Betriebsart.

Befehl	Inhalt der Meldungen	Funktion
SEND(090)/ RECV(098)	Befehle zum Senden/ Empfangen von Daten (FINS-Befehl)	CPU-Baugruppe: SEND(090) oder RECV(098) ⇄ Datenübertragung / Datenempfang ⇄ Anderes Gerät: CPU-Baugruppe, CS1 CPU-Bus-Baugruppe oder Computer
CMND(490)	Beliebige Befehle (FINS-Befehl)	CPU-Baugruppe: CMND(490) ⇄ Befehl gesendet / Antwort zurückgegeben ⇄ Anderes Gerät: CPU-Baugruppe, CS1 CPU-Bus-Baugruppe oder Computer

Die vom Netzwerk ausgeführten Befehle sind als „FINS-Befehle“ bekannt und werden für die Kommunikation zwischen Automations-Steuergeräten verwendet. (Detaillierte Informationen zu FINS-Befehlen finden Sie im *CS/CJ-Serie Kommunikationsbefehle Referenzhandbuch*.) Mit FINS-Befehlen ist die Kommunikation (mittels Befehls-/Antwortformat) mit einer beliebigen Baugruppe im Netzwerk oder im CPU-Baugruppenträger selbst möglich, indem nur die Adresse des Netzwerks, Knotennummer und Baugruppennummer der Zielbaugruppe spezifiziert werden müssen.

Im folgenden Beispiel wird ein FINS-Befehl über Knotennummer 2 im Netzwerk mit der Adresse 00 an die CPU-Baugruppe gesendet.



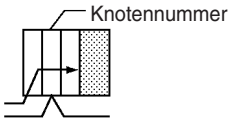
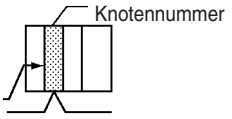
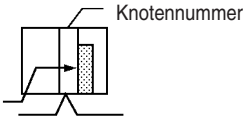
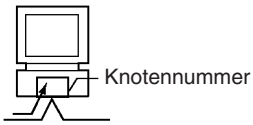
1,2,3... 1. Adresse des Netzwerks:
Adresse des Netzwerks (lokales Netzwerk = 00)

2. Knotennummer
Logische Adresse im Netzwerk

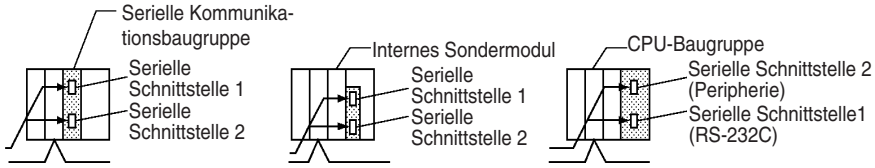
3. Baugruppennummer
Baugruppennummer der Zielbaugruppe
   a) CPU-Baugruppe:00
   b) CPU-Bus-Baugruppe:Baugruppennummer + 10 hexadezimal
   c) Spezial-E/A-Baugruppe (außer Spezial-E/A-Baugruppen der Serie C200H):
   Baugruppennummer + 20 hexadezimal

d) Internes Sondermodul (nur CS-Serie):
E1 hexadezimal

e) Computer: 01

Baugruppennummer (hexadezimal)	Zielgerät
00	
Baugruppennummer + 10	
E1	
01	

Hinweis Es kann auch innerhalb des Zielgeräts direkt eine serielle Schnittstelle (Baugruppenadresse) spezifiziert werden.



Baugruppenadressen der seriellen Schnittstelle:

- Schnittstellen der seriellen Kommunikationsbaugruppe

  Schnittstelle 1: 80 hex + 4 × Baugruppennummer

Baugruppennummer	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Hexadezimal	80	84	88	8C	90	94	98	9C	A0	A4	A8	AC	B0	B4	B8	BC
Dezimal	128	132	136	140	144	148	152	156	160	164	168	172	176	180	184	188

  Schnittstelle 2: 81 hex + 4 × Baugruppennummer

Baugruppennummer	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Hexadezimal	81	85	89	8D	91	95	99	9D	A1	A5	A9	AD	B1	B5	B9	BD
Dezimal	129	133	137	141	145	149	153	157	161	165	169	173	177	181	185	189

- Schnittstellen am seriellen Kommunikationsmodul

  Schnittstelle 1: E4 hex (228 dezimal)
  Schnittstelle 2: E5 hex (229 dezimal)

- Schnittstellen der CPU-Baugruppe

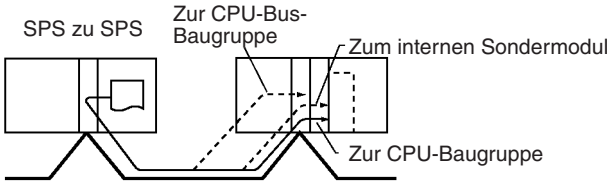
  Peripherieschnittstelle: FD hex (253 dezimal)
  RS-232C-Schnittstelle: FC hex (252 dezimal)

Netzwerk-Kommunikationsschemata Die folgenden Beispiele zeigen drei Arten der Netzwerkkommunikation: Kommunikation einer SPS mit anderen Geräten in einem Netzwerk, Kommunika-

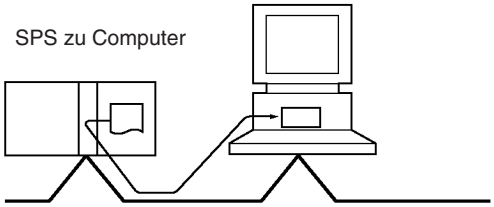
tion einer SPS mit seriellen Schnittstellen in einem Netzwerk und Kommunikation mit einem Host-Computer über einen Host-Link.

Kommunikation mit einem anderen Gerät im Netzwerk

Das folgende Beispiel zeigt die Kommunikation einer SPS mit Geräten in einer anderen SPS (CPU-Baugruppe, CPU-Bus-Baugruppe oder internes Sondermodul). Weitere Informationen sind dem Bedienerhandbuch für das jeweilige Netzwerk (Controller Link oder Ethernet) zu entnehmen.

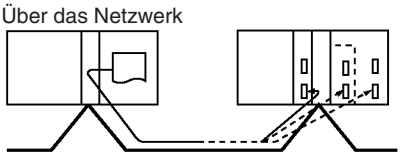


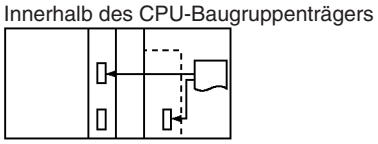
Dieses Beispiel zeigt die Kommunikation einer SPS mit einem PC.



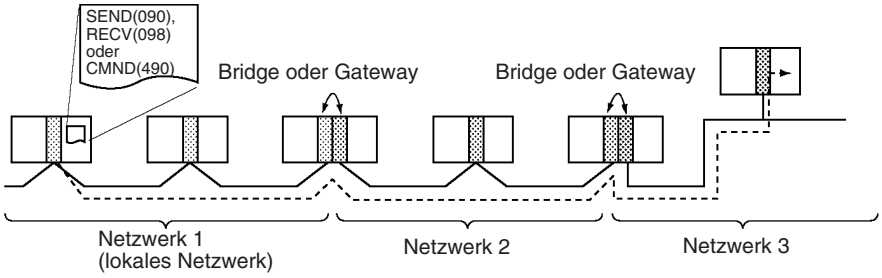
Kommunikation mit einer seriellen Schnittstelle im Netzwerk

Diese Beispiele zeigen die Kommunikation einer SPS mit seriellen Schnittstellen von Geräten im Netzwerk. Das erste Beispiel zeigt die Kommunikation mit seriellen Schnittstellen in Geräten einer anderen SPS (CPU-Baugruppe, CPU-Bus-Baugruppe oder internes Sondermodul), das zweite Beispiel zeigt die Kommunikation mit einer seriellen Schnittstelle eines Geräts innerhalb des gleichen CPU-Baugruppenträgers.





Hinweis Die Kommunikation kann bis zu 8 Netzwerkebenen umfassen, einschließlich des lokalen Netzwerks. (Das lokale Netzwerk ist das Netzwerk, in dem die Kommunikation ihren Ursprung hat.)



Um über das Netzwerks zu kommunizieren, ist die Registrierung einer Routing-Tabelle in der CPU-Baugruppe jeder SPS erforderlich, die die Route bestimmt, über die die Daten an den gewünschten Knoten übertragen wer-

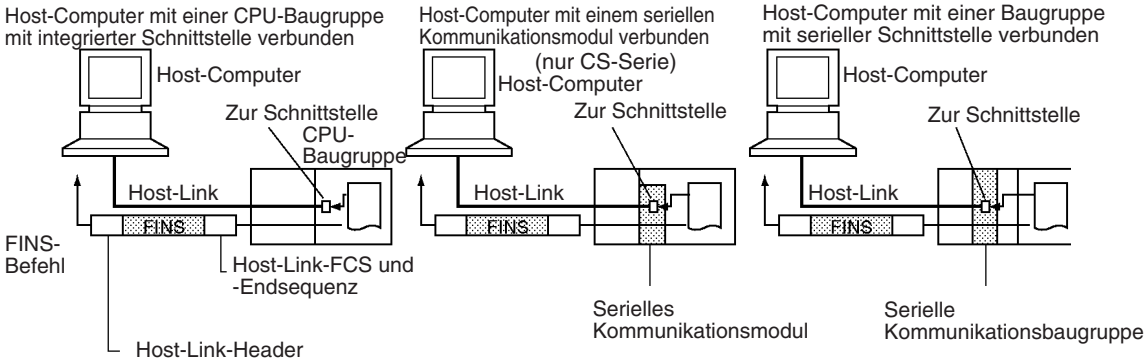
den. Jede Routing-Tabelle besteht aus einer lokalen Netzwerktabelle und einer Brücken-Netzwerktabelle.

1,2,3...

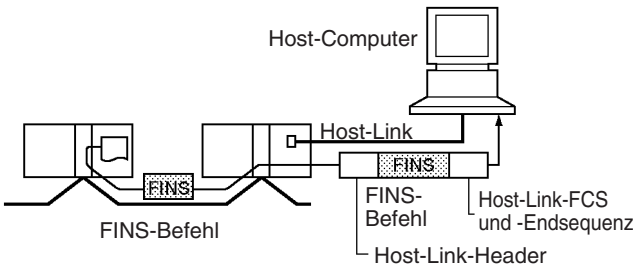
1. Lokale Netzwerktabelle
   Diese Tabelle enthält die Baugruppennummern und Netzwerkadressen der an die lokale SPS angeschlossenen Knoten.
2. Brücken-Netzwerktabelle
   Diese Tabelle enthält die Knotennummern und Netzwerkadressen der ersten Brückenknoten der Zielnetzwerke, die nicht mit der lokalen SPS verbunden sind.

Kommunikation mit einem Host-Computer (Host-Link)

Durch Ausgabe eines SEND(090)-, RECV(098)- oder CMND(490)-Befehls an eine auf den Host-Link-Modus eingestellte serielle Schnittstelle werden der erforderliche Host-Link-Header und die Endsequenz an den FINS-Befehl angehängt. Anschließend wird der Befehl an den Host-Computer gesendet.



Hinweis Host-Link-Kommunkation kann über das Netzwerk gesendet werden. In diesem Fall wird der FINS-Befehl normal über das Netzwerk übertragen. Wenn der Befehl das Host-Link-System erreicht, werden der erforderliche Host-Link-Header und die Endsequenz an den FINS-Befehl angehängt und der Befehl wird an den Host-Computer gesendet.

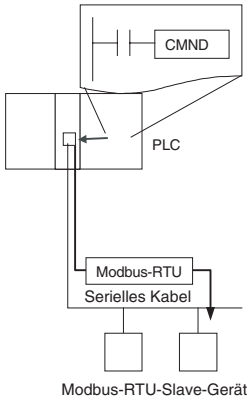


Seriell-Gateway-Kommunikation mit einer Komponente oder einem Host-Link-Slave

FINS-Befehle (oder Sende-/Empfangsdaten) können an eine Komponente oder einen Host-Link-Slave gesendet werden, die/der mit Hilfe der Seriell-Gateway-Funktion über seine serielle Schnittstelle mit der SPS verbunden ist.

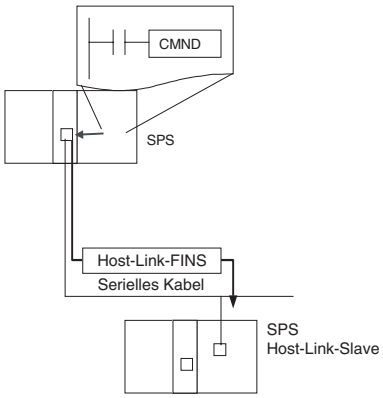
- Senden an eine Komponente

  Wenn ein CMND(490)-Befehl für eine serielle Schnittstelle ausgeführt wird, die die Seriell-Gateway-Funktion unterstützt, konvertiert die Seriell-Gateway-Funktion diesen Befehl in einen CompoWay/F-, Modbus-RTU- oder Modbus-ASCII-Befehl.



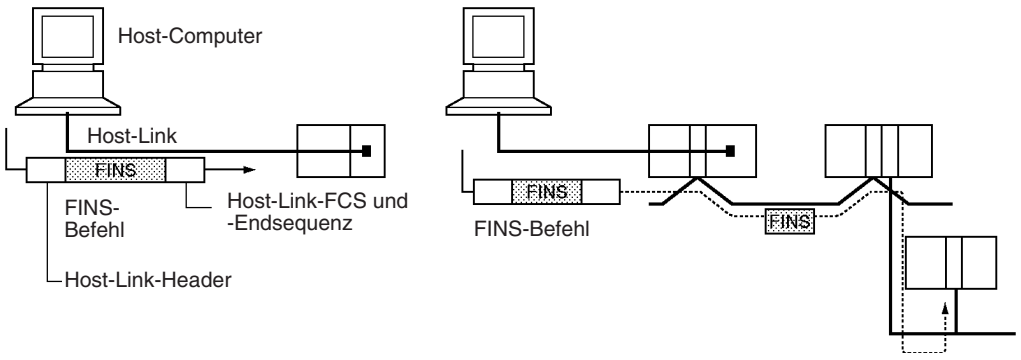
- Senden an eine als Host-Link-Slave arbeitende SPS

  Wenn ein CMND(490)-, SEND(090)- oder RECV(098)-Befehl für eine serielle Schnittstelle ausgeführt wird, die die Seriell-Gateway-Funktion unterstützt, kann die Seriell-Gateway-Funktion jeden beliebigen FINS-Befehl senden oder Daten senden/empfangen.



Kommunikation von einem Host-Computer (Host-Link)

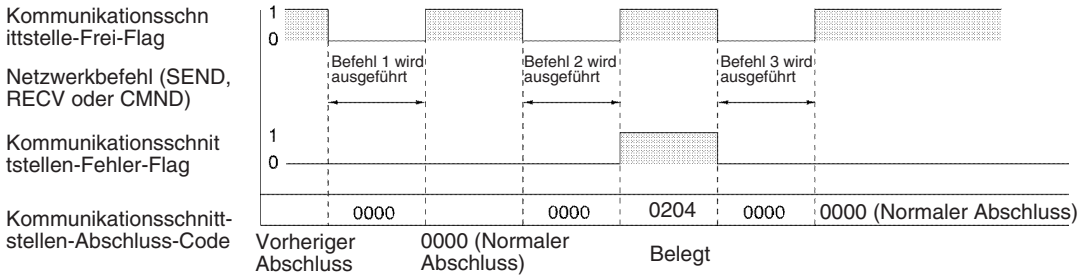
FINS-Befehle können von einem Host-Computer an die SPS gesendet werden, mit der er verbunden ist, ebenso aber auch an andere Geräte im Netzwerk (CPU-Baugruppen, Spezial- E/A-Baugruppen, Computer usw.). In diesem Fall müssen der erforderlichen Host-Link-Header und die Endsequenz beim Senden an den FINS-Befehl angehängt werden.



Kommunikations-Flags

Die Funktion der Kommunikations-Flags wird nachstehend erläutert.

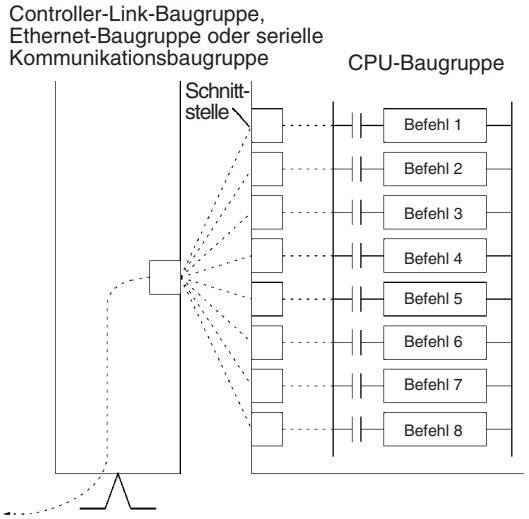
- Das Kommunikationsschnittstellen-Frei-Flag wird auf 0 zurückgesetzt, wenn eine Kommunikation ausgeführt wird und auf 1 gesetzt, wenn die Kommunikation beendet ist (normal oder fehlerhaft).
- Der Status des Kommunikationsschnittstellen-Fehler-Flags wird bis zum nächsten Senden oder Empfangen von Daten beibehalten.
- Das Kommunikationsschnittstellen-Fehler-Flag wird beim nächsten Senden oder Empfangen von Daten auf 0 zurückgesetzt, auch wenn beim vorherigen Vorgang ein Fehler aufgetreten ist.



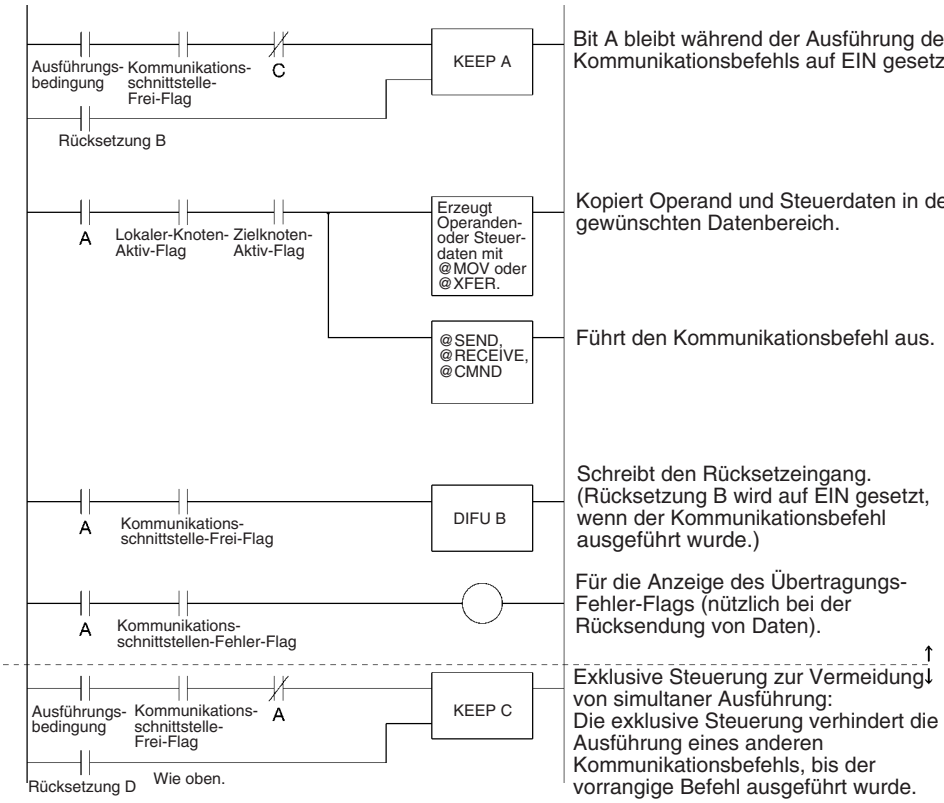
Informationen zu Kommunikationsschnittstellen-Nummern

Es sind 8 logische Kommunikationsschnittstellen vorhanden, sodass 8 Kommunikationsbefehle gleichzeitig ausgeführt werden können. Pro Kommunikationsschnittstelle kann nur jeweils ein Befehl ausgeführt werden. Bei Ausführung von mehr als 8 Befehlen muss die exklusive Steuerung verwendet werden.

Diese 8 Kommunikationsschnittstellen-Nummern werden von den Netzwerkbefehlen (SEND(090), RECV(098), und CMND(490)), den seriellen Kommunikationsbefehlen (TXDU(256) und RXDU(255)) und dem Befehl PROTOCOL MACRO (PMCR(260)) verwendet. Stellen Sie sicher, dass nicht für zwei Befehle gleichzeitig dieselbe Schnittstellennummer spezifiziert wird.



In der folgenden Abbildung ist ein Beispiel für exklusive Steuerung dargestellt.



Automatische Zuweisung von Kommunikationsschnittstellen

■ Übersicht

Die folgenden Befehle verwenden alle eine Kommunikationsschnittstelle (logische Schnittstelle) der Schnittstellen 0 bis 7.

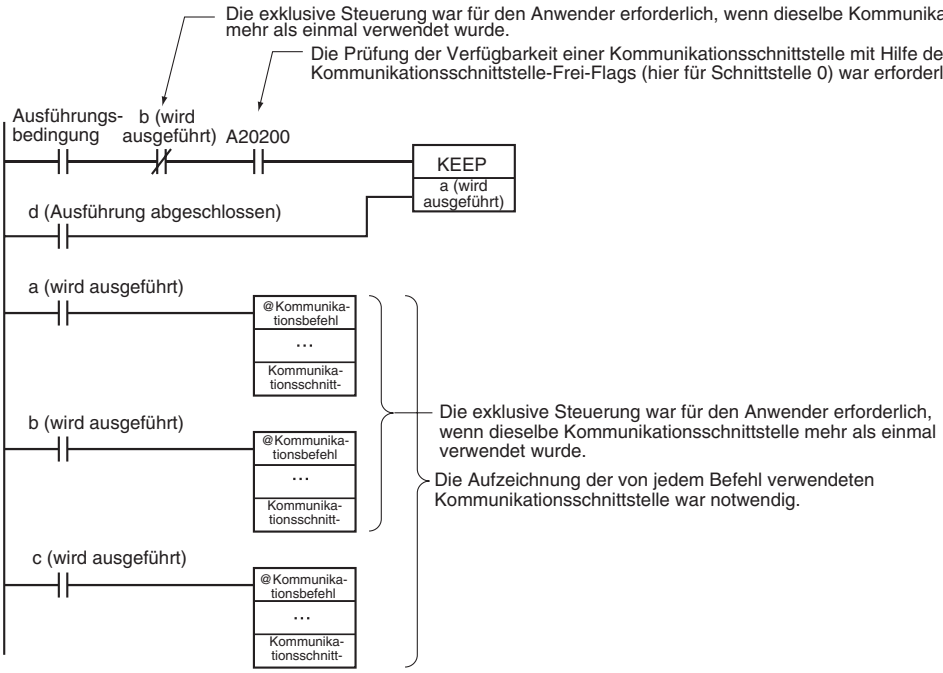
- Netzwerk-Kommunikationsbefehle: SEND(090), RECV(098) oder CMND(490)
- Befehle für die serielle Kommunikation: PMCR(260), TXDU(256) und RXDU(255)

In diesem Abschnitt werden mit „Kommunikationsbefehle“ alle oben genannte Befehle bezeichnet.

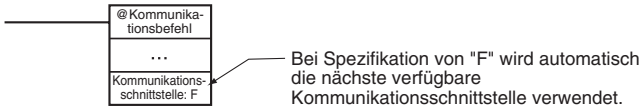
Jede Kommunikationsschnittstelle kann nur von jeweils einem Befehl verwendet werden. Zur Verwendung der Kommunikationsschnittstellen mussten vorher folgende Schritte ausgeführt werden.

- Beim Programmieren mussten die verwendeten Kommunikationsschnittstellen verfolgt werden, um sie in den Operanden spezifizieren zu können.
- Im Kontaktplanprogramm war es erforderlich, die Verfügbarkeit von Kommunikationsschnittstellen vor deren Verwendung zu prüfen.

Beispiele von vorherigen Programmierungserfordernissen



Jetzt kann bei CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D mit der Lot-Nummer ab 020601 (hergestellt ab 1. Juni 2002) die Schnittstellennummer anstatt zwischen 0 und 7 als „F“ spezifiziert werden, um die Kommunikationsschnittstelle automatisch zuzuweisen. Somit wird automatisch die nächste freie Kommunikationsschnittstelle verwendet.



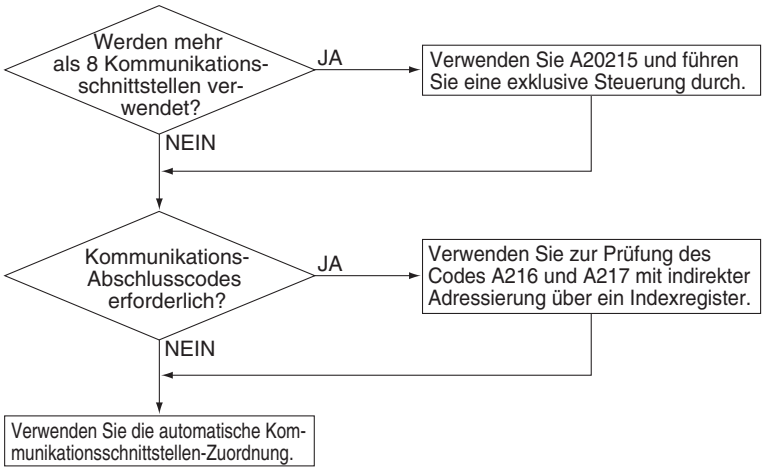
Dadurch muss der Programmierer die Nutzung der Kommunikationsschnittstellen während des Programmierens nicht nachhalten. Die Unterschiede zwischen der Zuweisung von bestimmten Schnittstellennummern und der automatischen Zuweisung von Schnittstellennummern sind in der nachfolgenden Tabelle zusammengestellt.

Gegenstand	Zuweisung bestimmter Nummern	Automatische Zuweisung
Spezifikation der Kommunikationsschnitt-stellen-Nummer in den Steuerdaten	0 bis 7	F
Exklusive Steuerung	Erforderlich	Nicht erforderlich, es sei denn, es wer-den mehr als 8 Kommuni-kationsschnittstellen gleichzeitig benötigt.
Anwendungen von Flags	LD oder LD NOT, verwendet mit einem der spezifizierten Kommunikationssch nittstelle ent-sprechendem Flag.	TST(350) oder TSTN(351) mit A218 (Nummer der verwendeten Kommunikationsschnittstelle) verwendet.
Netzwerkkommuni-kations-Abschlusscodes	Es erfolgt ein Zugriff auf den Abschlus-scode für eine vom Anwender spezifi-zierte Kommunika-tionsschnittstelle.	Auf Abschlusscodes wird durch Verwendung der in A216 und A217 gespeicherten E/A-Speicheradresse (Netzwerkkommunikations-Abschlusscode-Speicheradresse) sowie durch indirektes Adressieren mittels Indexregistern zugegriffen.

■ Bei der automatischen Zuordnung von Kommunikationsschnittstellen verwendete Bits und Wörter im Systembereich

Adresse	Bits	Name	Beschreibung
A202	15	Flag für Netzwerk-Kommunikationsschnittstellen-Zuordnung möglich	EIN, wenn eine Kommunikationsschnittstelle für die automatische Zuordnung verfügbar ist. Dieses Flag kann vor der Ausführung von Kommunikationsbefehlen zur Prüfung verwendet werden, ob bereits alle acht Kommunikationsschnittstellen zugeordnet wurden.
A214	00 bis 07	Erster Zyklus nach Abschluss der Netzwerkkommunikation	Jedes Bit wird für einen Zyklus nach Abschluss der Netzwerkkommunikation auf EIN gesetzt. Bits 00 bis 07 entsprechen den Schnittstellen 0 bis 7. Benutzen Sie die in A218 gespeicherte Nummer der verwendeten Kommunikationsschnittstelle, um zu bestimmen, auf welches Bit zugegriffen werden soll. **Hinweis:** Diese Bits sind erst im nächsten Zyklus nach der Ausführung des Kommunikationsbefehls gültig. Der Zugriff auf diese Bits muss daher um mindestens einen Zyklus verzögert erfolgen.
	08 bis 15	Nicht verwenden.	
A215	00 bis 07	Erster Zyklus nach Netzwerkkommunikationsfehler	Diese Bits werden nur für einen Zyklus nach Auftreten eines Netzwerkkommunikationsfehlers auf EIN gesetzt. Bits 00 bis 07 entsprechen den Schnittstellen 0 bis 7. Benutzen Sie die in A218 gespeicherte Nummer der verwendeten Kommunikationsschnittstelle, um zu bestimmen, auf welches Bit zugegriffen werden soll. **Hinweis:** Diese Bits sind erst im nächsten Zyklus nach der Ausführung des Kommunikationsbefehls gültig. Der Zugriff auf diese Bits muss daher um mindestens einen Zyklus verzögert erfolgen.
	08 bis 15	Nicht verwenden.	
A216 und A217	---	Speicheradresse des Netzwerkkommunikations-Abschlusscodes	Der Abschlusscode eines Kommunikationsbefehls wird automatisch in der in diesen Worten eingetragenen Speicheradresse gespeichert. Zum Auslesen des Kommunikations-Abschlusscodes laden Sie diese Adresse in ein Indexregister und verwenden dieses für eine indirekte Adressierung.
A218	---	Nummer der verwendeten Kommunikationsschnittstelle	Wenn ein Kommunikationsbefehl ausgeführt wird, wird die Nummer der verwendeten Kommunikationsschnittstelle in diesem Wort gespeichert. Die Werte 0000 bis 0007 hex entsprechen den Kommunikationsschnittstellen 0 bis 7.

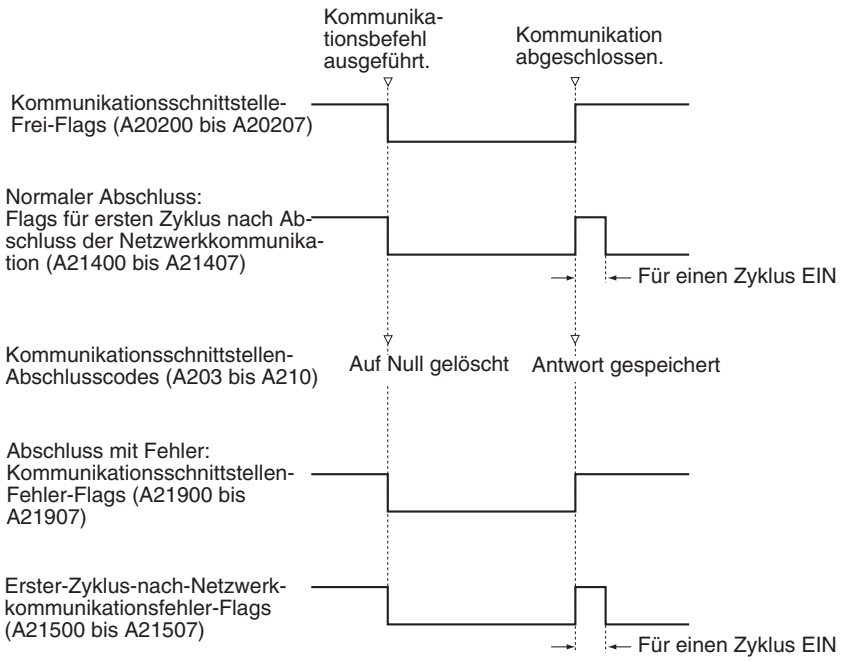
Hinweis 1. Verwenden Sie das nachfolgende Ablaufschema zur Bestimmung, ob das Flag für Netzwerk-Kommunikationsschnittstellen-Zuordnung möglich (A20215) und die Speicheradresse des Netzwerkkommunikations-Abschlusscodes (A216 und A217) verwendet werden müssen.



2. Die für durch den Anwender definierte Kommunikationsschnittstellen verwendeten Bits und Wörter im Systembereich sind in der nachfolgenden Tabelle aufgelistet.

Adresse	Bits	Name	Beschreibung
A202	00 bis 07	Kommunikationsschnittstelle-Frei-Flags	EIN, wenn ein Kommunikationsbefehl mit der entsprechenden Schnittstellennummer ausgeführt werden kann. Die Bits 00 bis 07 entsprechen den Kommunikationsschnittstellen 0 bis 7. Der Abschluss der Kommunikation kann bestätigt werden, indem überwacht wird, wann ein Flag auf EIN gesetzt wird. Das Flag wird bei Beginn der Ausführung eines Kommunikationsbefehls auf AUS gesetzt.
A203 bis A210	---	Kommunikationsschnittstellen-Abschlusscodes	Diese Wörter enthalten nach der Ausführung von Kommunikationsbefehlen den Abschlusscode für die entsprechenden Schnittstellennummern. Die Worte A203 bis A210 entsprechen den Kommunikationsschnittstellen 0 bis 7.
A219	00 bis 07	Kommunikationsschnittstellen-Fehler-Flags	EIN, wenn bei der Ausführung eines Kommunikationsbefehls ein Fehler aufgetreten ist. Wenn ein Flag auf EIN gesetzt ist, muss der Abschlusscode in A203 bis A210 zwecks Fehlersuche geprüft werden. Nach fehlerfreier Ausführung wird das Flag auf AUS gesetzt. Die Bits 00 bis 07 entsprechen den Kommunikationsschnittstellen 0 bis 7.

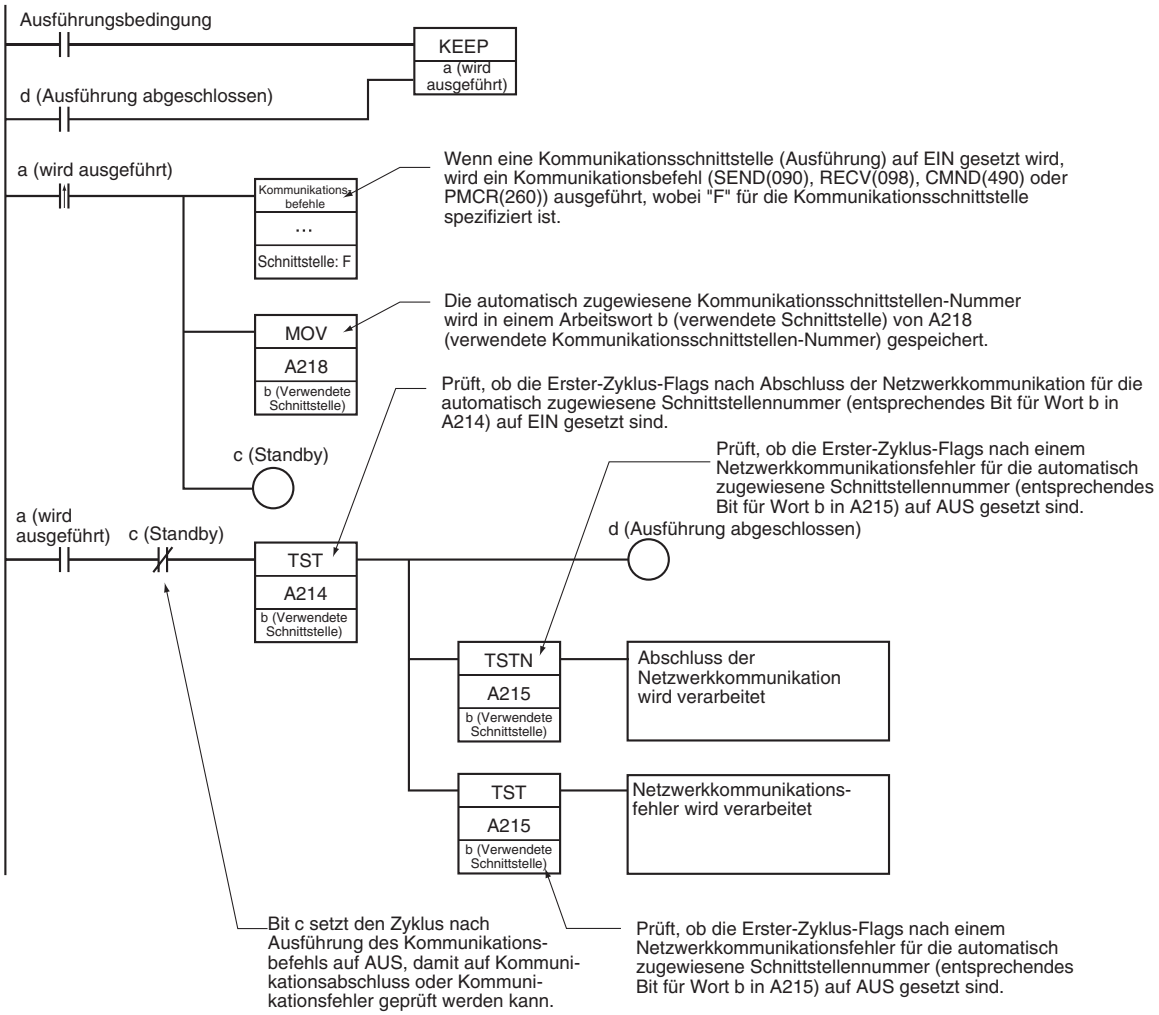
Bit-/Wortfunktionen



■ **Anwendungsmethoden**

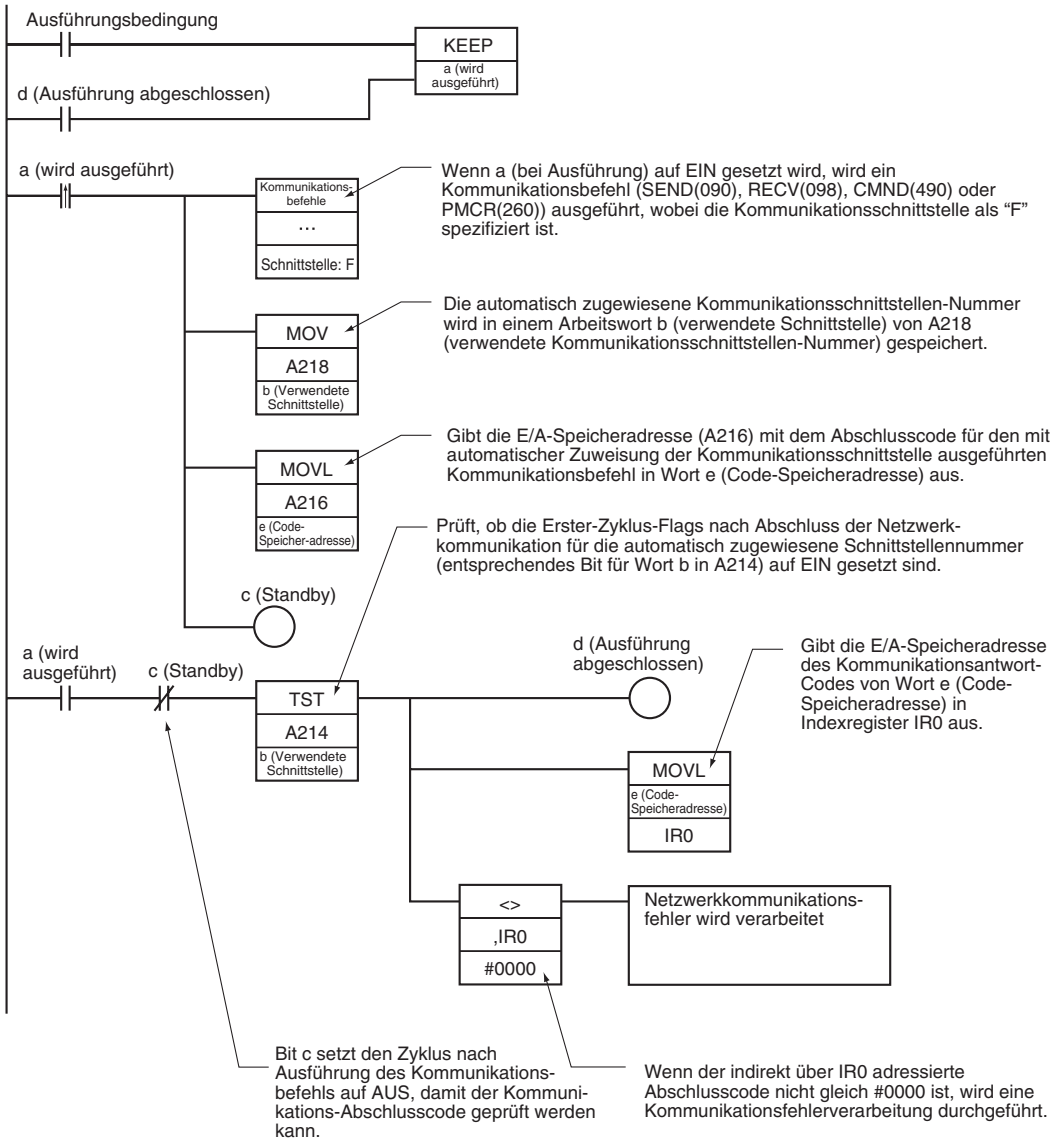
Um die automatische Kommunikationsschnittstellen-Zuweisung zu verwenden, muss die Kommunikationsschnittstellen-Nummer auf „F“ gesetzt und anschließend die unten dargestellte Programmierung vorgenommen werden.

Abschluss und Fehlerverarbeitung nach Ausführung von Kommunikationsbefehlen



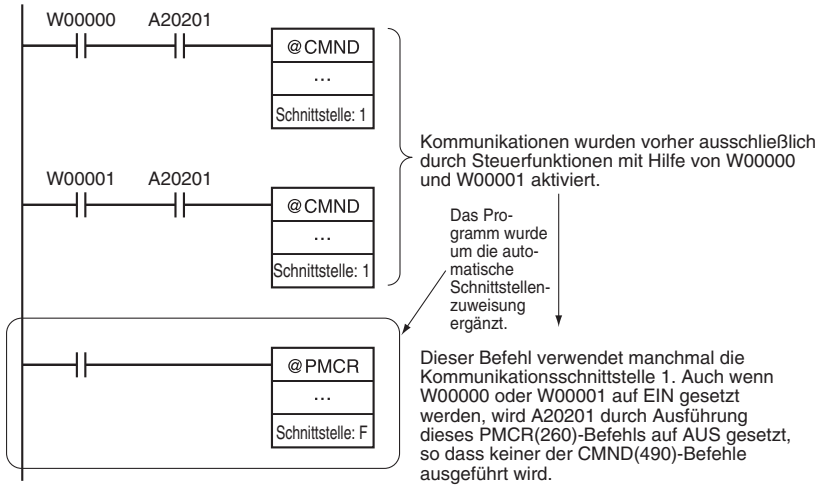
Zugriff auf den Abschlusscode nach Ausführung von Kommunikationsbefehlen

Die Abschlusscodes werden im Allgemeinen zur Fehlersuche verwendet. Ein Abschlusscode von 0000 hex kann dennoch als Bestätigung für eine normal abgeschlossene Kommunikation verwendet werden.



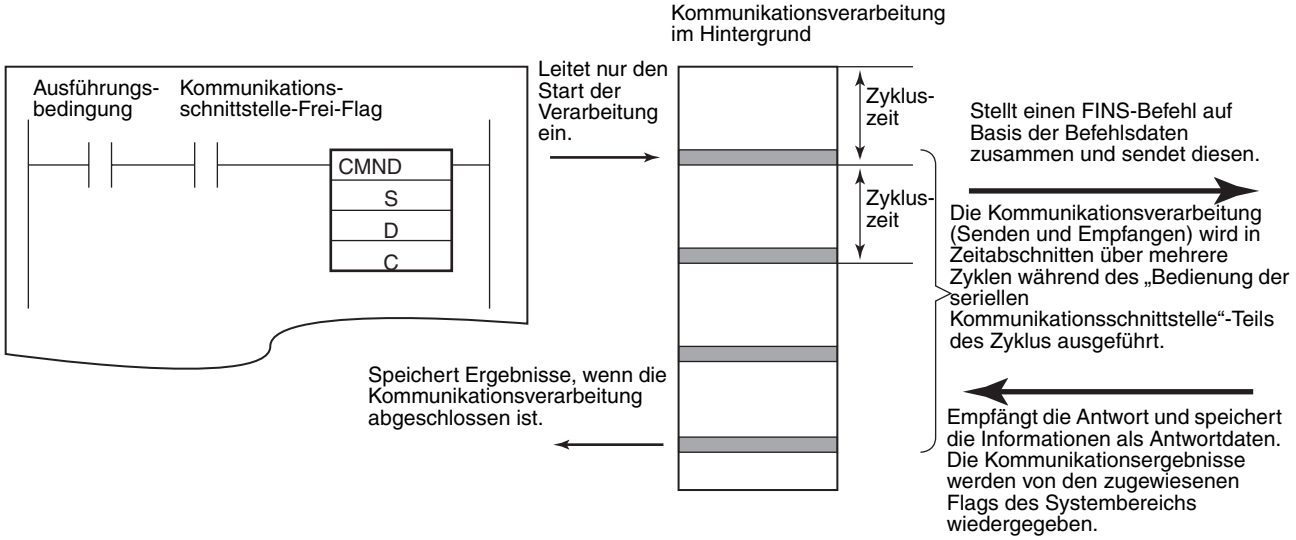
Hinweis Durch den Anwender spezifizierte Kommunikationsschnittstellen-Nummern und automatisch spezifizierte Kommunikationsschnittstellen-Nummern können im selben Programm verwendet werden. Es ist dennoch möglich, die vom Anwender spezifizierten Kommunikationsschnittstellen-Nummern für die automatische Zuweisung zu verwenden. Es ist allerdings wichtig, das Programm bei Hinzufügen von Kommunikationsbefehlen, die eine automatische Kommunikationsschnittstellen-Zuweisung verwenden, zu einem vorhandenen Programm sorgfältig zu prüfen, wie im nachstehenden Beispiel ersichtlich.

Programmierbeispiel



Zeitlicher Ablauf der Ausführung von Netzwerkbefehlen

Ein Netzwerkbefehl startet die Kommunikationsverarbeitung, wenn seine Ausführungsbedingung erfüllt wird. Die tatsächliche Kommunikationsverarbeitung wird im Hintergrund im „Bedienung der seriellen Kommunikationsschnittstelle“-Teil der Bedienung der Peripheriegeräte ausgeführt.

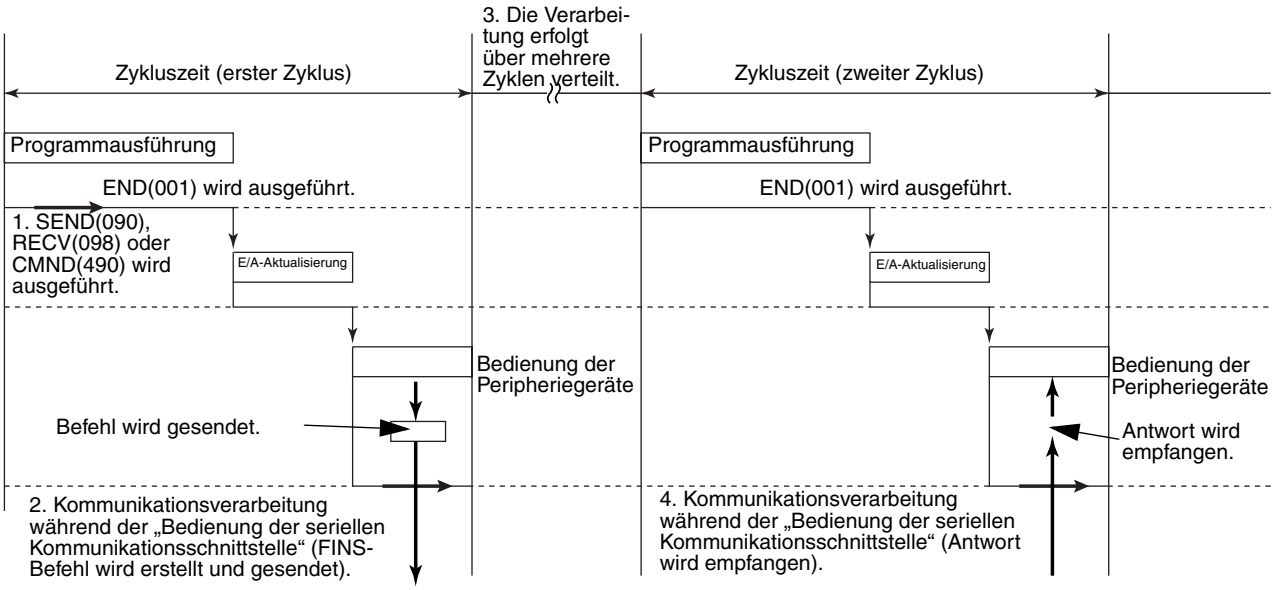


Die Kommunikationsverarbeitung findet wie folgt statt:

1. Wenn das entsprechende Kommunikationsschnittstellen-Frei-Flag (A20200 bis A20207) bei Erfüllung der Ausführungsbedingung auf EIN gesetzt ist, führt das System folgende Prozesse aus:
   - Das Kommunikationsschnittstelle-Frei-Flag und das Kommunikationsschnittstellen-Fehlerflag (A21900 bis A21907) der Schnittstelle werden auf AUS gesetzt.
   - Der Kommunikationsschnittstellen-Abschlusscode (A203 bis A210) der Schnittstelle wird auf 0000 gesetzt.
   - Die Steuerworte (beginnend bei C) werden gelesen und die Kommunikationsverarbeitung (Senden eines FINS-Befehl oder Empfangen einer Antwort) wird gestartet.
2. Im „Bedienung der seriellen Kommunikationsschnittstelle“-Teil des Zyklus stellt das System einen FINS-Befehl auf Basis der Operanden zusammen (siehe Hinweis) und sendet den FINS-Befehl an die Kommunikationsbaugruppe bzw. den Zielknoten.

Hinweis Bei Ausführung von SEND(090) werden die Inhalte von S und D gelesen und ein FINS-Befehl für die Datenübertragung erstellt.
Bei Ausführung von RECV(098) wird der Inhalt von S gelesen und ein FINS-Befehl für den Datenempfang erstellt.
Bei Ausführung von CMND(490) wird der Inhalt von S gelesen und der entsprechende FINS-Befehl wird erstellt.

3. Wenn der Sendevorgang nicht in der für die „Bedienung der seriellen Kommunikationsschnittstelle“ verfügbaren Zeit abgeschlossen werden kann, wird der Vorgang während der Bedienung der seriellen Kommunikationsschnittstelle im nächsten Zyklus fortgesetzt.
4. Wenn eine Antwort zurückgegeben wird, führt das System die folgenden Prozesse aus:
   - Die im Netzwerkbefehl spezifizierten Zielworte werden mit den Antwortdaten aktualisiert.
   - Das Kommunikationsschnittstelle-Frei-Flag wird auf EIN gesetzt.
   - Das Kommunikationsschnittstellen-Fehlerflag (A21900 bis A21907) sowie der Kommunikationsschnittstellen-Abschlusscode (A203 bis A210) der Schnittstelle werden aktualisiert.



3-25-2 Informationen zu Explizite-Meldung-Befehlen

Methoden für die Verwendung der Kommunikation mittels expliziter Meldungen

Es gibt zwei Methoden, die am besten für das Senden von expliziten Meldungen durch eine SPS geeignet sind.

- Verwenden des Befehls CMND(490) zum Senden eines FINS-Befehlscodes von 2801 hex (EXPLICIT MESSAGE SEND).
- Verwenden der folgenden Explizite-Meldung-Befehle. (Siehe Hinweis.)

  Hinweis Diese Befehle werden nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Explizite-Meldung-Befehle

Die folgenden Befehle, die speziell für explizite Meldungen verwendet werden, werden als Explizite-Meldung-Befehle bezeichnet.

Befehl	Name	Beschreibung
EXPLT(720)	EXPLICIT MESSAGE SEND	Sendet eine explizite Meldung mit einem beliebigen Service-Code. Hinweis: Funktionell entspricht dieser Befehl dem Senden von CMND(490) mit einem FINS-Befehlscode von 2801 hex.
EGATR(721)	EXPLICIT GET ATTRIBUTE	Sendet eine explizite Meldung mit einem Service-Code von 0E hex (GET ATTRIBUTE SINGLE).
ESATR(721)	EXPLICIT SET ATTRIBUTE	Sendet eine explizite Meldung mit einem Service-Code von 10E hex (SET ATTRIBUTE SINGLE).
EGATR(721)	EXPLICIT WORD READ	Verwendet eine explizite Meldung zum Lesen von Daten aus einer CPU-Baugruppe.
EGATR(721)	EXPLICIT WORD WRITE	Verwendet eine explizite Meldung zum Schreiben von Daten in eine CPU-Baugruppe.

Merkmale von Explizite-Meldung-Befehlen

- Explizite-Meldung-Befehle erfordern keinen FINS-Befehl „2801 hex“ und sind viel einfacher zu programmieren als CMND(490).
- Bei den EXPLICIT GET/SET ATTRIBUTE-Befehlen ist eine Eingabe des Service-Codes nicht erforderlich und es müssen nur Informationen ab der Klassen-ID eingegeben werden.
- Bei den EXPLICIT WORD READ/WRITE-Befehlen kann die E/A-Speicheradresse in lokalen und dezentralen CPU-Baugruppen direkt spezifiziert werden.

  Code-Spezifikationen für Bereichstypen sowie hexadezimale Wortadressen sind nicht erforderlich. (Diese sind bei CMND(490)-Befehlen mit Service-Code 1E hex (Wortdaten lesen) oder 1F hex (Wortdaten schreiben) erforderlich.)
  Dies ermöglicht ein einfaches Lesen und Schreiben von Daten zwischen CPU-Baugruppen durch Kommunikation mittels expliziter Meldungen (wie durch SEND/RECV-Befehle bei FINS-Befehlen).

Funktion

Anhand des Explizite-Kommunikation-Fehlerflags kann festgestellt werden, ob die Kommunikation normal oder mit einem Fehler abgeschlossen wurde.

Bei einem Abschluss mit Fehler (also wenn das Flag auf EIN gesetzt ist) wird das Kommunikationsschnittstellen-Fehlerflag für FINS-Befehle zur Bestimmung verwendet, ob die explizite Meldung nicht gesendet wurde (d.h., das Flag auf EIN gesetzt ist) oder ob in der gesendeten expliziten Meldung ein Fehler enthalten war (d.h., das Flag auf AUS gesetzt ist).

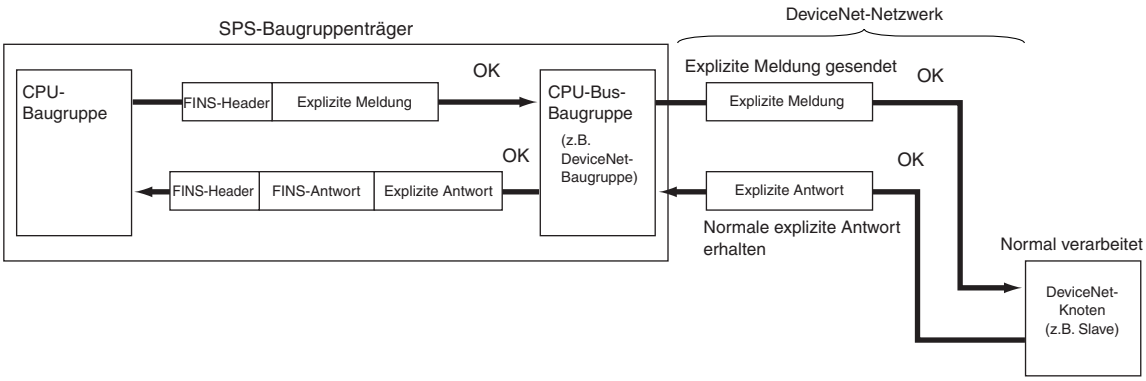
Der Kommunikationsschnittstellen-Abschlusscode hat nach einem normalen Abschluss der Kommunikation den Inhalt 0000 hex, nach Abschluss mit einem Explizite-Kommunikation-Fehler einen Explizite-Meldung-Fehlercode und nach einem Abschluss mit FINS-Fehler einen FINS-Meldung-Abschlusscode.

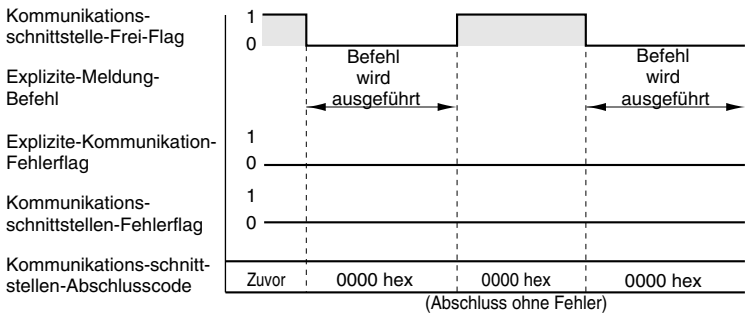
Zustand		Explizite-Kommunikation-Fehlerflag (A21300 bis A21307: Kommunikationsschnittstellen Nr. 0 bis 7)	Kommunikationsschnittstellen-Fehlerflag (A21900 bis A21907: Kommunikationsschnittstellen Nr. 0 bis 7)	Kommunikationsschnittstellen-Abschlusscode (A203 bis A210: Kommunikationsschnittstellen Nr. 0 bis 7)
1) Normaler Abschluss (ohne Fehler)		AUS	AUS	0000 hex
2) Abschluss mit Fehler	a) Wenn die explizite Meldung nicht gesendet werden konnte	EIN	EIN	FINS-Meldungen-Abschlusscode
	b) Wenn die explizite Meldung gesendet wurde, aber eine explizite Fehlerantwort zurückgegeben wurde		AUS	Explizite-Meldung-Fehlercode

1) Normaler Abschluss (ohne Fehler)

Eine explizite Meldung wurde gesendet und eine normale Antwort zurückgegeben.

Das entsprechende Explizite-Kommunikation-Fehlerflag (A21300 bis A21307: Kommunikationsschnittstelle Nr. 0 bis 7) wird auf AUS gesetzt und der Netzwerkkommunikations-Antwortcode (A203 bis A210: Kommunikationsschnittstelle Nr. 0 bis 7) enthält den normalen Explizite-Meldung-Antwortcode von 0000 hex.





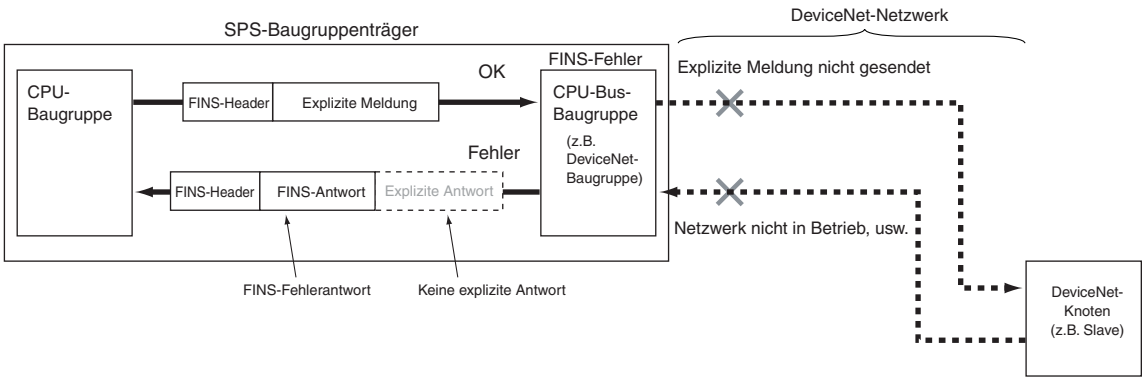
2) Abschluss mit Fehler

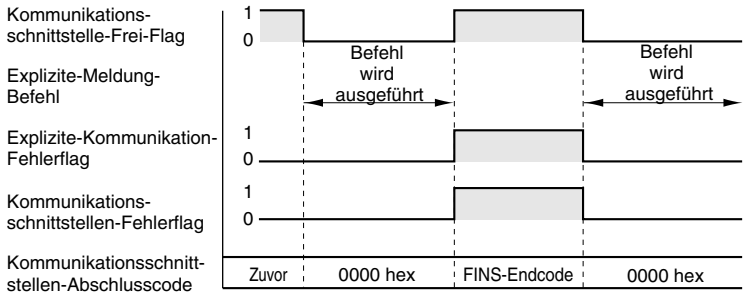
In den nächsten beiden Unterabschnitten werden die zwei Möglichkeiten eines Abschlusses mit Fehler erläutert.

a) Wenn die explizite Meldung nicht gesendet werden konnte

In diesem Fall ist die explizite Meldung nicht über das Netzwerk gesendet worden, beispielsweise weil das Netzwerk nicht in Betrieb war. Daher werden hier beide, das Explizite-Kommunikation-Fehlerflag (A21300 bis A21307: Kommunikationsschnittstelle Nr. 0 bis 7) und das Kommunikationsschnittstellen-Fehlerflag (A21900 bis A21907: Kommunikationsschnittstelle Nr. 0 bis 7) auf EIN gesetzt.

Nach Abschluss enthält der Kommunikationsschnittstellen-Abschlusscode (A203 bis A210: Kommunikationsschnittstelle Nr. 0 bis 7) den FINS-Meldung-Fehlercode.

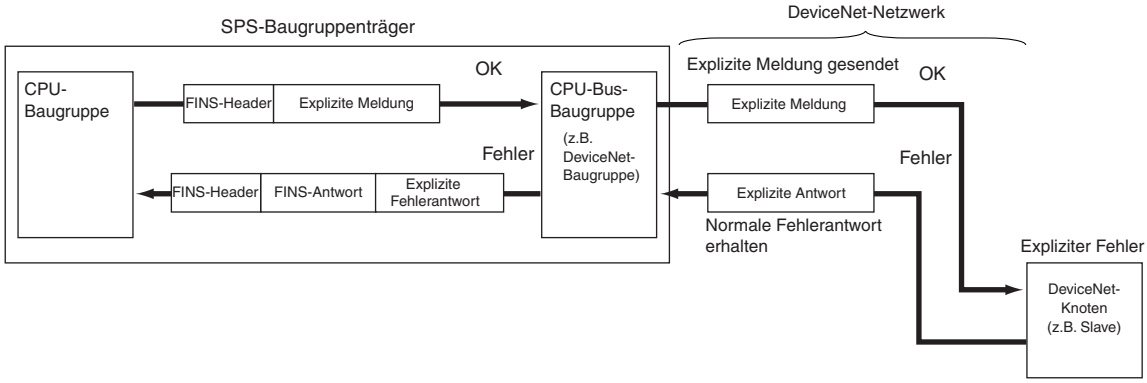


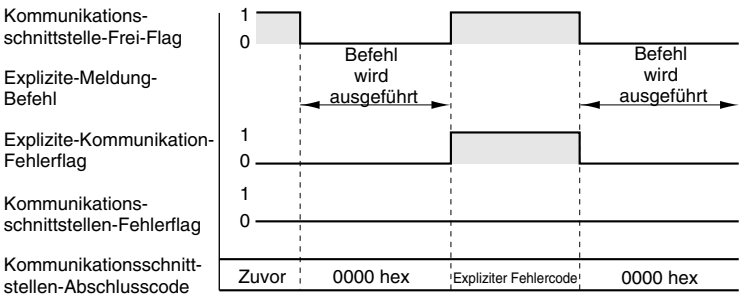


b) Wenn die explizite Meldung gesendet wurde, aber eine explizite Fehlerantwort zurückgegeben wurde

In diesem Fall wurde die explizite Meldung gesendet, aber im Befehls-Frame der expliziten Meldung war ein Fehler vorhanden (Code nicht unterstützt, unzulässige Größe, usw.). Hier wird das Explizite-Kommunikation-Fehlerflag (A21300 bis A21307: Kommunikationsschnittstelle Nr. 0 bis 7) auf EIN gesetzt und das Netzwerkkommunikations-Fehlerflag (A21900 bis A21907: Kommunikationsschnittstelle Nr. 0 bis 7) bleibt auf AUS gesetzt.

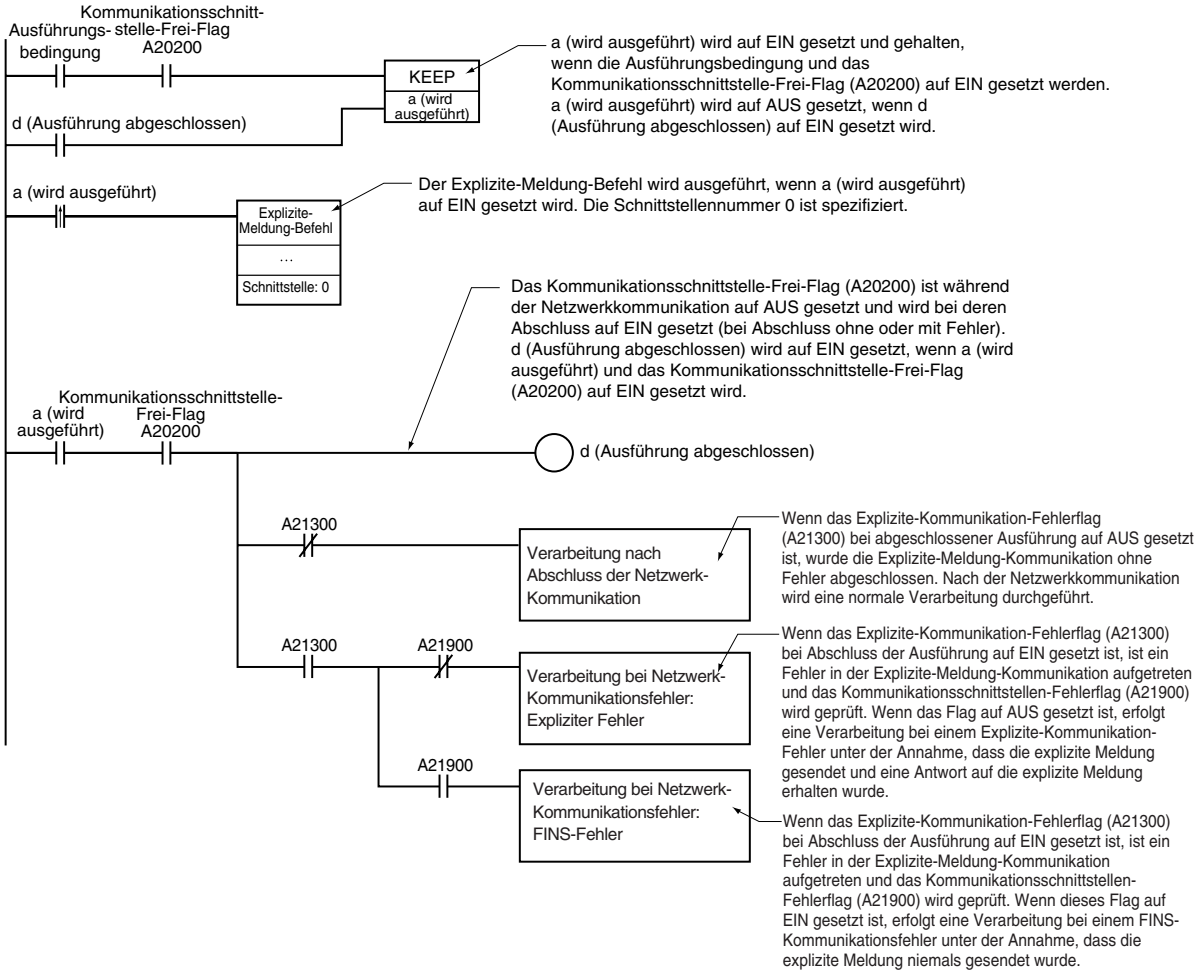
Nach Abschluss enthält der Netzwerkkommunikations-Antwortcode (A203 bis A210: Kommunikationsschnittstelle Nr. 0 bis 7) den Explizite-Meldung-Fehlercode.



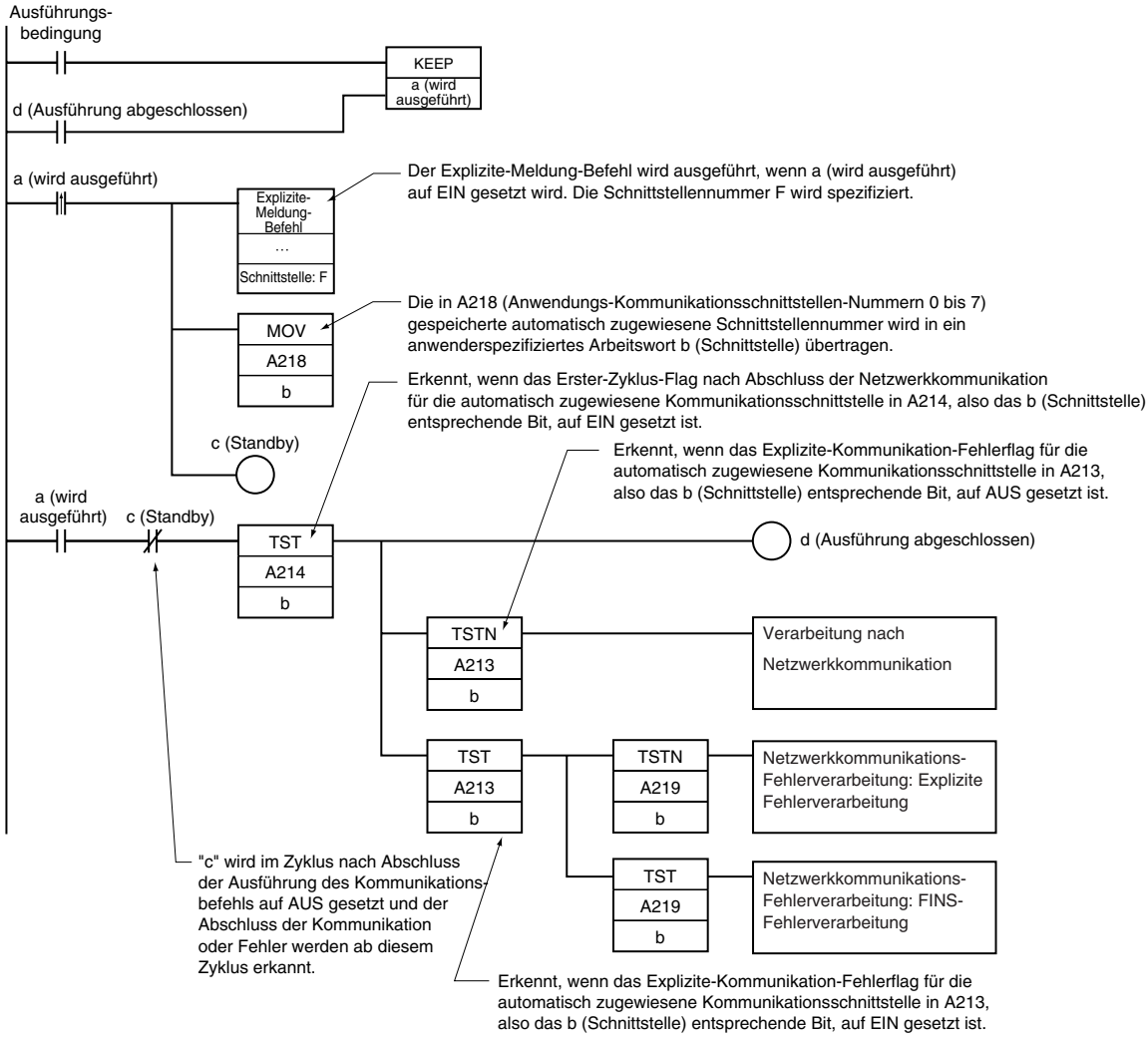


Beispiele für Kontaktplan-programmierung

Beispiel 1: Spezifikation der Kommunikationsschnittstellen-Nummer durch den Anwender



Beispiel 2: Automatische Zuweisung der Kommunikationsschnittstellen-Nummer

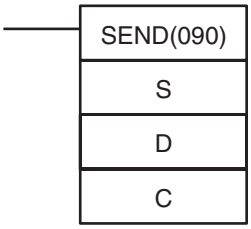


3-25-3 NETWORK SEND: SEND(090)

Funktion

Sendet Daten zu einem Knoten im Netzwerk.

Kontaktplansymbol



S: Erstes Quellwort (lokaler

D: Erstes Zielwort (dezentraler Knoten)

C: Erstes Steuerwort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	SEND(090)
	Ausführung einmal bei steigender Flanke	@SEND(090)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Erstes Steuerwort

Die fünf Steuerworte C bis C+4 spezifizieren die Anzahl der zu sendenden Worte, das Ziel sowie andere, in der nachfolgenden Tabelle aufgeführte Einstellungen.

Wort	Bits 00 bis 07	Bits 08 bis 15
C	Anzahl der Worte: 0001 bis zur maximal zulässigen[1] (vierstellig hexadezimal)	
C+1	Zielnetzwerkadresse: 00 bis 7F (0 bis 127)[2, 4]	Bits 08 bis 11: Nummer der seriellen Schnittstelle[3] (physische Schnittstelle) 1 hex: Schnittstelle 1 2 hex: Schnittstelle 2 (nicht auf 0, 3 oder 4 einstellen) Bits 12 bis 15: Immer 0.
C+2	Adresse der Zielbaugruppe: 00 bis FE[5]	Zielknotenadresse: 00 bis zur maximal zulässigen[6]
C+3	Anzahl der Wiederholversuche: 00 bis 0F (0 bis 15)	Bits 08 bis 11: Kommunikationsschnittstellen-Nummer (interne logische Schnittstelle): 0 bis 7, automatische Zuweisung: F[7] Bits 12 bis 15: Einstellung für Antwort 0: Antwort wird angefordert. 8: Antwort wird nicht angefordert.[8]
C+4	Antwort-Überwachungszeit: 0001 bis FFFF (0,1 bis 6553,5 Sekunden) (Die Standardeinstellung von 0000 bewirkt eine Überwachungszeit von 2 Sekunden.)	

Hinweis

1. Die maximal zulässige Anzahl von Worten hängt vom verwendeten Netzwerk ab. Bei einen Controller-Link erstreckt sich der zulässige Bereich von 0001 bis 03DE (1 bis 990 Worte).
2. Stellen Sie die Zielnetzwerkadresse für die Übermittlung innerhalb des lokalen Netzwerks auf 00 ein. Wenn zwei oder mehr CPU-Bus-Baugruppen installiert sind, ist die Netzwerkadresse die Baugruppennummer der Baugruppe mit der niedrigsten Baugruppennummer.
3. Die folgenden zwei Methoden können bei der Initiierung der Kommunikation durch die SPS zum Senden von Daten an den Host-Computer über eine serielle Schnittstelle mit dem Host-Link verwendet werden.
   a) Stellen Sie die Zielbaugruppenadresse (Bits 00 bis 07 von C+2) auf die Baugruppenadresse der CPU-Baugruppe, der seriellen Kommunikationsbaugruppe oder des seriellen Kommunikationsmoduls ein und die Nummer der seriellen Schnittstelle (Bits 08 bis 11 von C+1) für Schnittstelle 1 auf 1 bzw. für Schnittstelle 2 auf 2 ein.

Baugruppe nadresse (C+2, Bits 00 bis 07)	Baugruppe	Nummer der seriellen Schnitt-stelle (C+1, Bits 08 bis 11)	Serielle Schnittstelle
00 hex	CPU-Baugruppe	1 hex	Integrierte RS-232C-Schnittstelle
		2 hex	Peripherieschnittstelle
10 hex + Baugruppen-nummer	Serielle Kommunik-ationsbaugruppe (CPU-Bus-Baugruppe)	1 hex	Schnittstelle 1
		2 hex	Schnittstelle 2
E1 hex	Serielles Kommunikations-modul (internes Sonder-modul) (nur CS-Serie)	1 hex	Schnittstelle 1
		2 hex	Schnittstelle 2

b) Stellen Sie die Adresse der Zielbaugruppe direkt in den Bits 00 bis 07 von C+2 ein. In diesem Fall stellen Sie die Nummer der seriellen Schnittstelle zur direkten Spezifikation in den Bits 08 bis 11 von C+1 auf 0 ein.

Schnittstellen der seriellen Kommunikationsbaugruppe

Schnitt-stelle	Baugruppenadresse der Schnittstelle	Beispiel: Baugruppennummer = 1
Schnitt-stelle 1	80 hex + 4 × Baugruppennummer	80 + 4 × 1 = 84 hex (132 dezimal)
Schnitt-stelle 2	81 hex + 4 × Baugruppennummer	81 + 4 × 1 = 85 hex (133 dezimal)

Schnittstellen des seriellen Kommunikationsmoduls

Schnittstelle	Baugruppenadresse der Schnittstelle
Schnittstelle 1	E4 hex (228 dezimal)
Schnittstelle 2	E5 hex (229 dezimal)

Schnittstellen der CPU-Baugruppe

Schnittstelle	Baugruppenadresse der Schnittstelle
Peripherie-schnittstelle	FD hex (253 dezimal)
RS-232C	FC hex (252 dezimal)

4. Bei der Spezifikation der seriellen Schnittstelle ohne Routing-Tabelle für die Seriell-Gateway-Funktion (Konvertierung zu Host-Link-FINS) muss die Baugruppenadresse der seriellen Schnittstelle im Zielnetzwerk-Adressbyte eingestellt werden.
5. Die Baugruppenadresse spezifiziert die Baugruppe, wie in der nachstehenden Tabelle gezeigt.

Baugruppe	Einstellung der Baugruppenadresse
CPU-Baugruppe	00 hex
CPU-Bus-Baugruppe	10 hex + Baugruppennummer
Spezial-E/A-Baugruppe (außer Spezial-E/A-Baugruppen der Serie C200H)	20 hex + Baugruppennummer
Internes Sondermodul (nur CS-Serie)	E1 hex
Computer	01 hex
Mit dem Netzwerk verbundene Baugruppe (Spezifizierung der Baugruppe nicht erforderlich)	FE hex
Direkte Spezifikation der Baugruppenadresse der seriellen Schnittstelle	Schnittstellen der seriellen Kommunikationsbaugruppe Schnittstelle 1: 80 hex + 4 × Baugruppennummer Schnittstelle 2: 81 hex + 4 × Baugruppennummer Schnittstellen des seriellen Kommunikationsmoduls Schnittstelle 1: E4 hex (228 dezimal) Schnittstelle 2: E5 hex (229 dezimal) Schnittstellen der CPU-Baugruppe Peripherieschnittstelle: FD hex (253 dezimal) RS-232C-Schnittstelle: FC hex (252 dezimal)

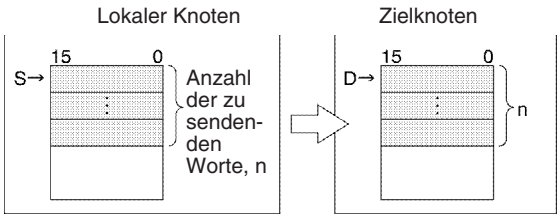
6. Die maximale Knotennummer hängt vom verwendeten Netzwerk ab. Bei einem kabelgebundenen Controller-Link erstreckt sich der zulässige Bereich von 00 bis 20 hexadezimal (0 bis 32 dezimal). Stellen Sie die Zielknotennummer auf FF ein, um zu allen Knoten zu senden; stellen Sie sie auf 00 ein, um innerhalb des lokalen Knotens zu senden.
7. Einzelheiten über die Verwendung der automatischen Zuweisung der Kommunikationsschnittstellen-Nummer (logische Schnittstelle) finden Sie unter *Automatische Zuweisung von Kommunikationsschnittstellen* auf Seite 979.
8. Wenn die Zielknotennummer auf FF (Senden an alle Knoten) eingestellt ist, erfolgt keine Antwort, auch wenn die Bits 12 bis 15 auf 0 gesetzt sind.

Operanden-Spezifikationen

Bereich	S	D	C
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6139
Arbeitsbereich nicht remanent	W000 bis W511		W000 to W507
Arbeitsbereich remanent	H000 bis H511		H000 to H507
Systembereich	A000 bis A959		A000 bis A955
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4091
Zählerbereich	C0000 bis C4095		C0000 bis C4091
DM-Bereich	D00000 bis D32767		D00000 bis D32763
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32763
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32763 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

SEND(090) überträgt die mit dem Wort S beginnenden Daten über den CPU-Bus der SPS oder über ein Netzwerk in die mit D beginnenden Adressen im spezifizierten Gerät. Die Anzahl der zu übertragenden Worte wird in C spezifiziert.



Wenn die Zielknotennummer auf FF eingestellt ist, werden die Daten an alle Knoten im spezifizierten Netzwerk gesendet. Das wird auch als „Broadcast-Senden“ bezeichnet.

Wenn eine Antwort angefordert wird (Bits 12 bis 15 von C+3 auf 0 eingestellt), aber innerhalb der Antwort-Überwachungszeit keine Antwort empfangen wurde, werden die Daten bis zu 15 Mal erneut gesendet (die Anzahl der Wiederholversuche wird in den Bits 0 bis 3 von C+3 festgelegt). Bei Broadcast-Sendungen gibt es keine Antworten oder Wiederholungen.

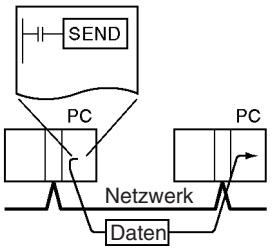
Der Befehl SEND(090) kann zum Senden von Daten an eine bestimmte serielle Schnittstelle im Zielgerät sowie im Gerät selbst verwendet werden.

Daten können an einen mit der seriellen Schnittstelle der SPS verbundenen Host-Computer (falls auf Host-Link-Modus eingestellt) gesendet werden. Ebenso ist das Senden an eine SPS oder einen Computer möglich, zu der/dem eine Verbindung über ein Controller-Link- oder Ethernet-Netzwerk besteht.

Wenn das Kommunikationsschnittstelle-Frei-Flag für die in C+3 spezifizierte Kommunikationsschnittstelle bei Ausführung von SEND(090) auf EIN gesetzt ist, wird das entsprechende Kommunikationsschnittstelle-Frei-Flag (Schnittstellen 00 bis 07: A20200 bis A20207) und das Kommunikationsschnittstellen-Fehlerflag (Schnittstellen 00 bis 07: A21900 bis A21907) auf AUS gesetzt und in das Wort, das den Abschlusscode enthält (Schnittstellen 00 bis 07: A203 bis A210) wird 0000 geschrieben. Sobald die Flags gesetzt sind, werden Daten an den Zielknoten gesendet.

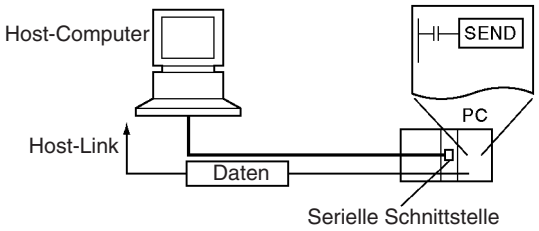
Übertragung über das Netzwerk

Der Befehl SEND(090) kann zum Senden von Daten von der SPS in den spezifizierten Datenbereich einer SPS oder eines Computers verwendet werden, mit der/dem eine Verbindung über ein Controller-Link- oder Ethernet-Netzwerk besteht.



Senden über Host-Link

Wenn die integrierte serielle Schnittstelle der CPU-Baugruppe, ein serielles Kommunikationsmodul (nur CS-Serie) oder eine serielle Kommunikationsbaugruppe auf den Host-Link-Modus eingestellt ist und eins-zu-eins mit einem Host-Computer verbunden ist, kann der Befehl SEND(090) ausgeführt werden, um Daten von der SPS zum Host-Computer zu senden, wenn die SPS das nächste Mal das Senderecht besitzt. Außerdem ist es möglich, Daten an andere Host-Computer zu senden, die mit anderen SPS-Systemen an anderer Stelle im Netzwerk verbunden sind.



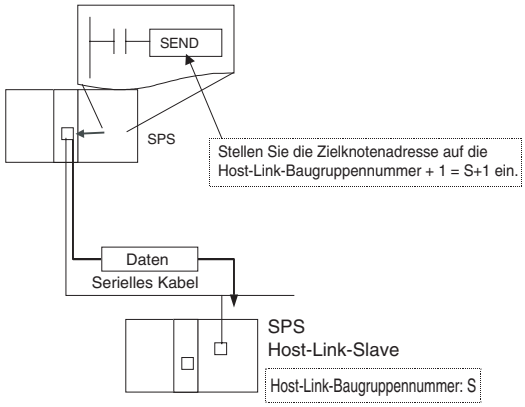
Wenn der Befehl SEND(090) an die serielle Schnittstelle der CPU-Baugruppe, an ein serielles Kommunikationsmodul (nur CS-Serie) oder an eine serielle Kommunikationsbaugruppe gesendet wird, wird ein Befehl von der seriellen Schnittstelle an den Host-Computer gesendet. Der Befehl ist eine FINS-Meldung, die zwischen Host-Link-Header und -Endsequenz eingeschlossen ist. Der FINS-Befehl ist ein MEMORY AREA WRITE-Befehl (Befehlscode 0102) und der Host-Link-Headercode lautet 0F hex.

Im Host-Computer muss ein Programm erstellt werden, um den empfangenen Befehl (den zwischen Host-Link-Header und -Endsequenz eingeschlossenen FINS-Befehl) zu verarbeiten.

Wenn sich die serielle Zielschnittstelle in der lokalen SPS befindet, muss die Netzwerkadresse in C+1 auf 00 (lokales Netzwerk), die Knotenadresse in C+2 auf 00 (lokale SPS) und die Baugruppenadresse auf 00 (CPU-Baugruppe), E1 (internes Sondermodul (nur CS-Serie)) oder Baugruppennummer + 10 hexadezimal (serielle Kommunikationsbaugruppe) eingestellt werden.

Senden von Daten an eine über Seriell-Gateway verbundene Host-Link-Slave-SPS.

Die Seriell-Gateway-Funktion kann zum Senden von Daten an eine SPS verwendet werden, welche als Host-Link-Slave mit einem seriellen Kommunikationsmodul oder einer seriellen Kommunikationsbaugruppe verbunden ist. In diesem Fall muss die Zielknotenadresse auf die Host-Link-Baugruppennummer + 1 eingestellt werden.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn die Nummer der in C+1 spezifizierten serielle Schnittstellennummer nicht innerhalb des Bereichs von 00 bis 04 liegt. EIN, wenn das Kommunikationsschnittstelle-Frei-Flag für die in C+3 spezifizierte Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

In der folgenden Tabelle sind die relevanten Bits und Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Kommunikationsschnittstelle-Frei-Flag	A20200 bis A20207	Diese Flags werden auf EIN gesetzt, um anzugeben, dass Netzwerkbefehle (einschließlich PMCR(260)) für die entsprechenden Schnittstellen (00 bis 07) ausgeführt werden können. Ein Flag wird auf AUS gesetzt, wenn ein Netzwerkbefehl für die entsprechende Schnittstelle ausgeführt wird, und wieder auf EIN gesetzt, wenn die Ausführung des Befehls abgeschlossen ist.
Kommunikationsschnittstellen-Fehlerflag	A21900 bis A21907	Diese Flags werden auf EIN gesetzt, um anzugeben, dass an den entsprechenden Schnittstellen (00 bis 07) während der Ausführung eines Netzwerkbefehls ein Fehler aufgetreten ist. Der Flag-Status wird bis zur Ausführung des nächsten Netzwerkbefehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunikationsschnittstellen-Abschlusscodes	A203 bis A210	Diese Worte enthalten nach der Ausführung eines Netzwerkbefehls die Abschlusscodes für die entsprechenden Schnittstellen (00 bis 07). Das entsprechende Wort hat während Ausführung des Netzwerkbefehls den Inhalt 0000. Nach Abschluss der Befehlsausführung wird der Abschlusscode in dieses Wort geschrieben. Diese Worte werden bei Ausführung eines Befehls gelöscht.

Vorsichtsmaßnahmen

Wenn das Kommunikationsschnittstelle-Frei-Flag für die in C+3 spezifizierte Schnittstellennummer auf AUS gesetzt ist, wird der Befehl als NOP(000) behandelt und nicht ausgeführt. In diesem Fall wird das Fehler-Flag auf EIN gesetzt.

Wenn für D eine Adresse in der aktuellen Bank des EM-Bereichs spezifiziert ist, werden die gesendeten Daten in die aktuelle EM-Bank des Zielknotens geschrieben.

Wenn Daten über die Grenzen des lokalen Netzwerks hinaus gesendet werden sollen, muss der Anwender Routing-Tabellen in den CPU-Baugruppen der SPS-Systeme jedes Netzwerks registrieren. (Routing-Tabellen bestimmen die Routen zu anderen Netzwerken, in die die Zielknoten eingebunden sind.)

Weitere Informationen über Abschlusscodes bei der Netzwerkkommunikation finden Sie im *CS/CJ-Serie Kommunikationsbefehle-Referenzhandbuch* (W342) unter FINS-Befehl-Antwortcodes.

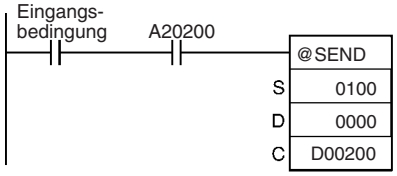
Für eine Kommunikationsschnittstelle kann nur jeweils ein Netzwerkbefehl ausgeführt werden. Programmieren Sie das Kommunikationsschnittstelle-Frei-Flag (A20200 bis A20207) als Schließerbedingung, um sicherzustellen, dass der Befehl SEND(090) nicht ausgeführt wird, wenn die jeweilige Schnittstelle belegt ist.

Kommunikationsschnittstellen-Nummern 00 bis 07 werden von den Netzwerkbefehlen und PMCR(260) gemeinsam genutzt, so dass SEND(090) nicht gleichzeitig mit PMCR(260) ausgeführt werden kann, wenn die Befehle dieselben Schnittstellennummer nutzen.

Störungen und andere Faktoren können dazu führen, dass die Sendung oder Antwort fehlerhaft ist oder verloren geht. Deshalb empfehlen wir, die Anzahl der Wiederholversuche auf einen Wert größer als Null zu setzen, wodurch SEND(090) erneut ausgeführt wird, wenn die Antwort nicht innerhalb der Antwortüberwachungszeit empfangen wird.

Beispiel 1

Wenn die Eingangsbedingung und A20200 (das Kommunikationsschnittstelle-Frei-Flag für Schnittstelle 0) im folgenden Beispiel auf EIN gesetzt sind, werden die zehn Worte von CIO 100 bis CIO 109 an den Host-Computer gesendet, der mit Schnittstelle 1 der seriellen Kommunikationsbaugruppe mit der Baugruppenadresse 10 (hex) an Knotennummer 3 im Netzwerk 0 verbunden ist.

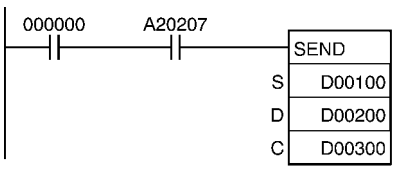


C	D00200	0 0 0 A	Anzahl der zu sendenden Worte: 10 Worte
C+1	D00201	0 1 0 0	An Netzwerk 0 und Schnittstelle 1 des seriellen Kommunikationsmoduls senden
C+2	D00202	0 0 1 0	Knotennummer 0, Baugruppenadresse 10
C+3	D00203	0 0 0 0	Antwort angefordert, Schnittstellennummer 0, keine Wiederholversuche
C+4	D00204	0 0 0 0	Antwort-Überwachungszeit: 2 Sekunden (0000: Standardwert)

Im Host-Computer muss ein entsprechendes Programm vorhanden sein, damit er die Daten empfangen und eine Antwort senden kann.

Beispiel 2

Wenn CIO 000000 und A20207 (das Kommunikationsschnittstelle-Frei-Flag für Schnittstelle 07) im folgenden Beispiel auf EIN gesetzt sind, werden die zehn Worte von D00100 bis D00109 an Knoten Nummer 3 im lokalen Netzwerk gesendet, wo sie in die zehn Worte von D00200 bis D00209 geschrieben werden. Die Daten werden bis zu dreimal wiederholt gesendet, wenn nicht innerhalb von zehn Sekunden eine Antwort empfangen wird.



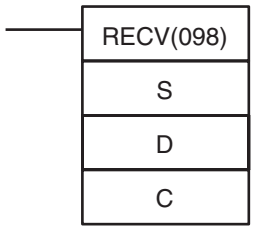
C: D00300	0 0 0 A	Anzahl der zu sendenden Worte: 10 Worte
C+1: D00301	0 0 0 0	An das lokale Netzwerk und das Gerät selbst senden
C+2: D00302	0 3 0 0	Knotennummer 3, Baugruppenadresse 00 (CPU-Baugruppe)
C+3: D00303	0 7 0 3	Antwort angefordert, Schnittstellennummer 7, 3 Wiederholversuche
C+4: D00304	0 0 6 4	Antwort-Überwachungszeit: 0064 hexadezimal (10 Sekunden)

3-25-4 NETWORK RECEIVE: RECV(098)

Funktion

Fordert Daten von einem Knoten im Netzwerk an und empfängt diese Daten.

Kontaktplansymbol



S: Erstes Quellwort (dezentraler Knoten)
D: Erstes Zielwort (lokaler Knoten)
C: Erstes Steuerwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	RECV(098)
	Ausführung einmal bei steigender Flanke	@RECV(098)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Erstes Steuerwort
Die fünf Steuerworte C bis C+4 spezifizieren die Anzahl der zu empfangenen Worte, die Quelle der Übertragung sowie andere, in der nachfolgenden Tabelle aufgeführte Einstellungen.

Wort	Bits 00 bis 07	Bits 08 bis 15
C	Anzahl der Worte: 0001 bis zur maximal zulässigen[1] (vierstellig hexadezimal)	
C+1	Quellnetzwerkadresse: 00 bis 7F (0 bis 127)[2, 4]	Bits 08 bis 11: Nummer der seriellen Schnittstelle (physische Schnittstelle) 1 hex: Schnittstelle 1 2 hex: Schnittstelle 2 (nicht auf 0, 3 oder 4 einstellen) Bits 12 bis 15: Immer 0.
C+2	Adresse der Quellbaugruppe[5]	Quellknotenadresse: 00 bis zur maximal zulässigen[6]
C+3	Anzahl der Wiederholversuche: 00 bis 0F (0 bis 15)	Schnittstellen-Nr.: 00 bis 07 (F: Automatische Zuweisung)[7] Antwort ist auf „erforderlich“ festgelegt.
C+4	Antwort-Überwachungszeit: 0001 bis FFFF (0,1 bis 6553,5 Sekunden) (Die Standardeinstellung von 0000 bewirkt eine Überwachungszeit von 2 Sekunden.)	

Hinweis

1. Die maximal zulässige Anzahl von Worten hängt vom verwendeten Netzwerk ab. Bei einen Controller-Link erstreckt sich der zulässige Bereich von 0001 bis 03DE (1 bis 990 Worte).
2. Stellen Sie die Quellnetzwerkadresse auf 00 ein, um ein Quelle innerhalb des lokalen Netzwerks zu spezifizieren. Wenn zwei oder mehr CPU-Bus-Baugruppen installiert sind, ist die Netzwerkadresse die Baugruppennummer der Baugruppe mit der niedrigsten Baugruppennummer.
3. Bei der Initiierung der Kommunikation von der SPS aus können die folgenden zwei Methoden zum Empfangen von Daten von einem Host-Computer über eine serielle Schnittstelle mittels Host-Link verwendet werden.

a) Stellen Sie die Adresse der Quellbaugruppe (Bits 00 bis 07 von C+2) auf die Baugruppenadresse der CPU-Baugruppe oder der seriellen Kommunikationsbaugruppe/des seriellen Kommunikationsmoduls ein und die Nummer der seriellen Schnittstelle (Bits 08 bis 11 von C+1) für Schnittstelle 1 auf 1 bzw. für Schnittstelle 2 auf 2 ein.

Baugruppenadresse (C+2, Bits 00 bis 07)	Baugruppe	Nummer der seriellen Schnittstelle (C+1, Bits 08 bis 11)	Serielle Schnittstelle
00 hex	CPU-Baugruppe	1 hex	Integrierte RS-232C-Schnittstelle
		2 hex	Peripheries chnittstelle
10 hex + Baugruppennummer	Serielle Kommunikationsbaugruppe (CPU-Bus-Baugruppe)	1 hex	Schnittstelle 1
		2 hex	Schnittstelle 2
E1 hex	Serielles Kommunikationsmodul (internes Sondermodul) (nur CS-Serie)	1 hex	Schnittstelle 1
		2 hex	Schnittstelle 2

b) Stellen Sie die Adresse der Quellbaugruppe direkt in den Bits 00 bis 07 von C+2 ein. In diesem Fall stellen Sie die Nummer der seriellen Schnittstelle in den Bits 08 bis 11 von C+1 für die direkte Spezifikation auf 0 ein.

Schnittstellen der seriellen Kommunikationsbaugruppe

Schnittstelle	Baugruppenadresse der Schnittstelle	Beispiel: Baugruppennummer = 1
Schnittstelle 1	80 hex + 4 × Baugruppennummer	80 + 4 × 1 = 84 hex (132 dezimal)
Schnittstelle 2	81 hex + 4 × Baugruppennummer	81 + 4 × 1 = 85 hex (133 dezimal)

Schnittstellen des seriellen Kommunikationsmoduls

Schnittstelle	Baugruppenadresse der Schnittstelle
Schnittstelle 1	E4 hex (228 dezimal)
Schnittstelle 2	E5 hex (229 dezimal)

Schnittstellen der CPU-Baugruppe

Schnittstelle	Baugruppenadresse der Schnittstelle
Peripheries-chnittstelle	FD hex (253 dezimal)
RS-232C	FC hex (252 dezimal)

4. Bei Festlegung der seriellen Schnittstelle ohne Routing-Tabelle für die Seriell-Gateway-Funktion (Konvertierung zu Host-Link-FINS) muss die Baugruppenadresse der seriellen Schnittstelle im Quellnetzwerk-Adressbyte eingestellt werden.
5. Die Baugruppenadresse spezifiziert die Baugruppe, wie in der nachstehenden Tabelle gezeigt.

Baugruppe	Einstellung der Baugruppenadresse
CPU-Baugruppe	00 hex
CPU-Bus-Baugruppe	10 hex + Baugruppennummer
Spezial-E/A-Baugruppe (außer Spezial-E/A-Baugruppen der Serie C200H)	20 hex + Baugruppennummer
Internes Sondermodul (nur CS-Serie)	E1 hex

Baugruppe	Einstellung der Baugruppenadresse
Computer	01 hex
Mit dem Netzwerk verbundene Baugruppe (Spezifizierung der Baugruppe nicht erforderlich)	FE hex
Direkte Spezifikation der Baugruppenadresse der seriellen Schnittstelle	Schnittstellen der seriellen Kommunikationsbaugruppe Schnittstelle 1: 80 hex + 4 × Baugruppennummer Schnittstelle 2: 81 hex + 4 × Baugruppennummer Schnittstellen des seriellen Kommunikationsmoduls Schnittstelle 1: E4 hex (228 dezimal) Schnittstelle 2: E5 hex (229 dezimal) Schnittstellen der CPU-Baugruppe Peripherieschnittstelle: FD hex (253 dezimal) RS-232C-Schnittstelle: FC hex (252 dezimal)

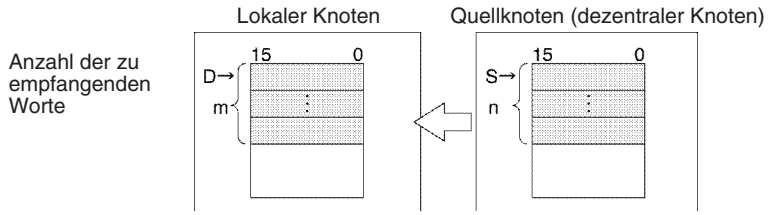
6. Die maximale Knotennummer hängt vom verwendeten Netzwerk ab. Bei einem kabelgebundenen Controller-Link erstreckt sich der zulässige Bereich von 00 bis 20 hexadezimal (0 bis 32 dezimal). Für die Übertragung innerhalb des lokalen Knotens setzen Sie die Quellknotennummer auf 00.
7. Einzelheiten über die Verwendung der automatischen Zuweisung der Kommunikationsschnittstellen-Nummer (logische Schnittstelle) finden Sie unter *Automatische Zuweisung von Kommunikationsschnittstellen* auf Seite 979.

Operanden-Spezifikationen

Bereich	S	D	C
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6139
Arbeitsbereich nicht remanent	W000 bis W511		W000 to W507
Arbeitsbereich remanent	H000 bis H511		H000 to H507
Systembereich	A000 bis A447 A448 bis A959	A448 bis A959	A000 bis A443 A448 bis A955
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4091
Zählerbereich	C0000 bis C4095		C0000 bis C4091
DM-Bereich	D00000 bis D32767		D00000 bis D32763
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32763
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32763 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl RECV(098) fordert vom spezifizierten Gerät das Senden der in C spezifizierten Anzahl von Worten, mit dem Wort S beginnend, an die lokale SPS an. Die Daten werden über den CPU-Bus der SPS oder über das Netzwerk empfangen und bei D beginnend in den Datenbereich der SPS geschrieben.



Für RECV(098) ist eine Antwort erforderlich, da die Antwort die zu empfangenden Daten enthält. Wenn die Antwort nicht innerhalb der in C+4 festgelegten Antwort-Überwachungszeit empfangen wurde, wird die Anforderung zur Datenübertragung bis zu 15 Mal wiederholt (die Anzahl der Wiederholversuche wird durch die Bits 0 bis 3 von C+3 festgelegt).

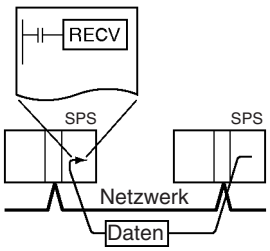
SEND(098) kann zur Anforderung von Daten von einer bestimmten seriellen Schnittstelle im Quellgerät sowie im Gerät selbst verwendet werden.

Daten können von einem mit der seriellen Schnittstelle der SPS (falls auf Host-Link-Modus eingestellt) verbundenen Host-Computer empfangen werden sowie von einer SPS oder einem Computer, mit der/dem eine Verbindung über ein Controller-Link- oder Ethernet-Netzwerk besteht.

Wenn das Kommunikationsschnittstelle-Frei-Flag für die in C+3 spezifizierte Kommunikationsschnittstelle bei Ausführung von SEND(090) auf EIN gesetzt ist, wird das entsprechende Kommunikationsschnittstelle-Frei-Flag (Schnittstellen 00 bis 07: A20200 bis A20207) und das Kommunikationsschnittstellen-Fehlerflag (Schnittstellen 00 bis 07: A21900 bis A21907) auf AUS gesetzt und in das Wort, das den Abschlusscode enthält (Schnittstellen 00 bis 07: A203 bis A210) wird 0000 geschrieben. Nachdem die Flags gesetzt sind, werden Daten vom Zielknoten empfangen.

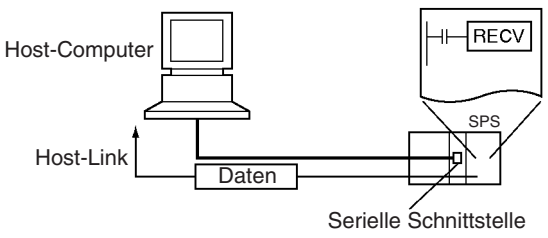
Übertragung über das Netzwerk

Der Befehl RECV(098) kann zum Empfang von Daten verwendet werden, die aus dem spezifizierten Datenbereich einer SPS oder eines Computers gesendet werden, mit der/dem eine Verbindung über ein Controller-Link- oder Ethernet-Netzwerk besteht. Die empfangenen Daten werden in den spezifizierten Datenbereich der lokalen SPS geschrieben.



Übertragung über Host-Link

Wenn die integrierte serielle Schnittstelle der CPU-Baugruppe, ein serielles Kommunikationsmodul (nur CS-Serie) oder eine serielle Kommunikationsbaugruppe auf den Host-Link-Modus eingestellt ist und eins-zu-eins mit einem Host-Computer verbunden ist, kann der Befehl RECV(098) ausgeführt werden, um Daten vom Host-Computer zu empfangen, wenn die SPS das nächste Mal das Recht zum Senden von Befehlen besitzt. Darüber hinaus ist es auch möglich, Daten von anderen Host-Computern zu empfangen, die mit anderen SPS an anderer Stelle im Netzwerk verbunden sind.



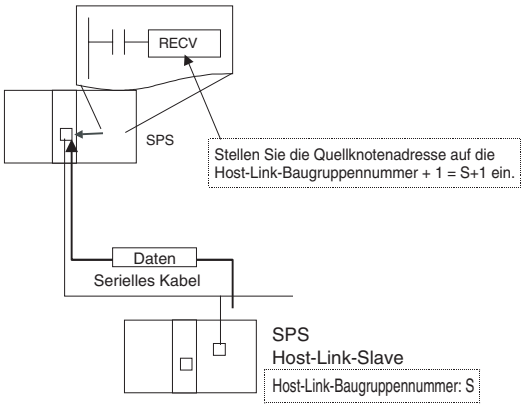
Wenn RECV(098) für die serielle Schnittstelle der CPU-Baugruppe, ein serielles Kommunikationsmodul (nur CS-Serie) oder eine serielle Kommunikationsbaugruppe ausgeführt wird, wird ein Befehl über die serielle Schnittstelle an den Host-Computer gesendet. Der Befehl ist eine FINS-Meldung, die zwischen Host-Link-Header und -Endsequenz eingeschlossen ist. Der FINS-Befehl ist ein MEMORY AREA READ-Befehl (Befehlscode 0101) und der Host-Link-Headercode lautet 0F hexadezimal.

Im Host-Computer muss ein Programm erstellt werden, um den gesendeten Befehl (den zwischen Host-Link-Header und -Endsequenz eingeschlossenen FINS-Befehl) zu verarbeiten.

Wenn sich die serielle Zielschnittstelle in der lokalen SPS befindet, muss die Netzwerkadresse in C+1 auf 00 (lokales Netzwerk), die Knotenadresse in C+2 auf 00 (lokale SPS) und die Baugruppenadresse auf 00 (CPU-Baugruppe), E1 (internes Sondermodul, nur CS-Serie) oder Baugruppennummer + 10 hexadezimal (serielle Kommunikationsbaugruppe) eingestellt werden.

Empfangen von Daten von einer über Seriell-Gateway verbundenen Host-Link-Slave-SPS

Die Seriell-Gateway-Funktion kann zum Empfangen von Daten von einer SPS verwendet werden, die als Host-Link-Slave mit einem seriellen Kommunikationsmodul oder einer seriellen Kommunikationsbaugruppe verbunden ist. In diesem Fall muss die Quellknotenadresse auf die Host-Link-Baugruppennummer + 1 eingestellt werden.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn die Nummer der in C+1 spezifizierten serielle Schnittstellennummer nicht innerhalb des Bereichs von 00 bis 04 liegt. EIN, wenn das Kommunikationsschnittstelle-Frei-Flag für die in C+3 spezifizierte Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

In der folgenden Tabelle sind die relevanten Bits und Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Kommunikationsschnittstelle-Frei-Flag	A20200 bis A20207	Diese Flags werden auf EIN gesetzt, um anzugeben, dass Netzwerkbefehle (einschließlich PMCR(260)) für die entsprechenden Schnittstellen (00 bis 07) ausgeführt werden können. Ein Flag wird auf AUS gesetzt, wenn ein Netzwerkbefehl für die entsprechende Schnittstelle ausgeführt wird, und wieder auf EIN gesetzt, wenn die Ausführung des Befehls abgeschlossen ist.

Name	Adresse	Funktion
Kommunikations-schnittstellen-Fehlerflag	A21900 bis A21907	Diese Flags werden auf EIN gesetzt, um anzugeben, dass an den entsprechenden Schnittstellen (00 bis 07) während der Ausführung eines Netzwerkbefehls ein Fehler aufgetreten ist. Der Flag-Status wird bis zur Ausführung des nächsten Netzwerkbefehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunikations-schnittstellen-Abschlusscodes	A203 bis A210	Diese Worte enthalten nach der Ausführung eines Netzwerkbefehls die Abschlusscodes für die entsprechenden Schnittstellen (00 bis 07). Das entsprechende Wort hat während Ausführung des Netzwerkbefehls den Inhalt 0000. Nach Abschluss der Befehlsausführung wird der Abschlusscode in dieses Wort geschrieben. Diese Worte werden zu Beginn der Programmausführung gelöscht.

Vorsichtsmaßnahmen

Wenn das Kommunikationsschnittstelle-Frei-Flag für die in C+3 spezifizierte Schnittstellennummer auf AUS gesetzt ist, wird der Befehl als NOP(000) behandelt und nicht ausgeführt. In diesem Fall wird das Fehler-Flag auf EIN gesetzt.

Wenn für D eine Adresse in der aktuellen Bank des EM-Bereichs spezifiziert ist, werden die gesendeten Daten in die aktuelle EM-Bank des Zielknotens geschrieben.

Wenn Daten über die Grenzen des lokalen Netzwerks hinaus gesendet werden sollen, muss der Anwender Routing-Tabellen in den CPU-Baugruppen der SPS-Systeme jedes Netzwerks registrieren. (Routing-Tabellen bestimmen die Routen zu anderen Netzwerken, in die die Zielknoten eingebunden sind.)

Weitere Informationen über Abschlusscodes bei der Netzwerkkommunikation finden Sie im *CS/CJ-Serie Kommunikationsbefehle-Referenzhandbuch* (W342) unter FINS-Befehl-Antwortcodes.

Für eine Kommunikationsschnittstelle kann nur jeweils ein Netzwerkbefehl ausgeführt werden. Programmieren Sie das Kommunikationsschnittstelle-Frei-Flag (A20200 bis A20207) als Schließerbedingung, um sicherzustellen, dass der Befehl RECV(098) nicht ausgeführt wird, wenn die jeweilige Schnittstelle belegt ist.

Die Kommunikationsschnittstellen-Nummern 00 bis 07 werden von den Netzwerkbefehlen und PMCR(260) gemeinsam genutzt, sodass RECV(098) nicht gleichzeitig mit PMCR(260) ausgeführt werden kann, wenn von den Befehlen dieselbe Schnittstellennummer verwendet wird.

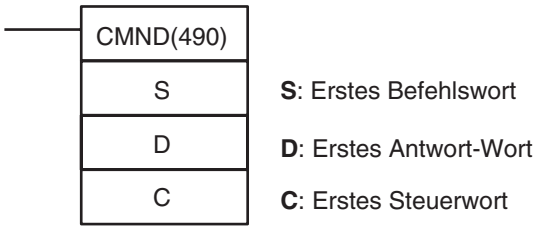
Störungen und andere Faktoren können dazu führen, dass die Sendung oder Antwort fehlerhaft ist oder verloren geht. Deshalb empfehlen wir, die Anzahl der Wiederholversuche auf einen Wert größer als Null zu setzen, wodurch RECV(098) erneut ausgeführt wird, wenn die Antwort nicht innerhalb der Antwortüberwachungszeit empfangen wird.

3-25-5 DELIVER COMMAND: CMND(490)

Funktion

Sendet einen FINS-Befehl und empfängt die Antwort. Detaillierte Informationen zu FINS-Befehlen finden Sie im *CS/CJ-Serie Kommunikationsbefehle Referenzhandbuch*.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CMND(490)
	Ausführung einmal bei steigender Flanke	@CMND(490)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Erstes Steuerwort

Die sechs Steuerworte C bis C+5 spezifizieren die Anzahl Bytes der Befehls- und Antwortdaten, das Ziel sowie andere, in der nachfolgenden Tabelle aufgeführte Einstellungen.

Wort	Bits 00 bis 07	Bits 08 bis 15
C	Anzahl Bytes von Befehlsdaten: 0002 bis zur maximal zulässigen[1] (vierstellig hexadezimal)	
C+1	Anzahl Bytes von Antwortdaten: 0000 bis zur maximal zulässigen[1 bis 3] (vierstellig hexadezimal)	
C+2	Zielnetzwerkadresse: 00 bis 07[4, 6]	Bits 08 bis 11: Nummer der seriellen Schnittstelle (physische Schnittstelle) 1 hex: Schnittstelle 1 2 hex: Schnittstelle 2 (nicht auf 0, 3 oder 4 einstellen) Bits 12 bis 15: Immer 0.
C+3	Adresse der Zielbaugruppe: 00 bis FE[5, 7, 9]	Zielknotennummer: 00 bis zur maximal zulässigen[8]
C+4	Anzahl der Wiederholversuche: 00 bis 0F (0 bis 15)	Bits 08 bis 11: Schnittstellen-Nummer (interne logische Schnittstelle): 0 bis 7 (F: Automatische Zuweisung)[10] Bits 12 bis 15: Einstellung für Antwort 0: Antwort wird angefordert. 8: Antwort wird nicht angefordert.[11]
C+5	Antwort-Überwachungszeit: 0001 bis FFFF (0,1 bis 6553,5 Sekunden) (Die Standardeinstellung von 0000 bewirkt eine Überwachungszeit von 2 Sekunden.)	

Hinweis

1. Die Anzahl der Bytes von Befehlsdaten in C kann zwischen 0002 und der maximalen Datenlänge in hexadezimal eingestellt werden. Beispielsweise erstreckt sich der zulässige Bereich für die Byte-Anzahl bei Controller-Link-Systemen von 0002 bis 07C6 hex (2 bis 1.990 Bytes). Die Byte-Anzahl für die lokale CPU-Baugruppe beträgt 07C6 hex (1.990 Bytes). Die Byte-Anzahl der Befehlsdaten ist vom Netzwerk abhängig.
2. Die Byte-Anzahl der Antwortdaten in C+1 beträgt 0000 bis zur maximalen Datenlänge in hexadezimal. Beispielsweise erstreckt sich der zulässige Bereich für die Byte-Anzahl bei Controller-Link-Systemen von 0000 bis 07C6 hex (0 bis 1.990 Bytes). Die Byte-Anzahl für die lokale CPU-Baugruppe beträgt 07C6 hex (1.990 Bytes). Die Byte-Anzahl der Antwortdaten ist vom Netzwerk abhängig.
3. Weitere Informationen über die maximale Datenlänge der Befehlsdaten und Antwortdaten finden Sie im Bedienerhandbuch für das jeweilige Netzwerk. Für FINS-Befehle, die mehrere Netzwerke durchlaufen, wird die maximale Datenlänge der Befehls- und Antwortdaten durch das Netzwerk mit der kleinsten maximalen Datenlänge bestimmt.
4. Stellen Sie die Zielnetzwerkadresse für die Übermittlung innerhalb des lokalen Netzwerks auf 00 ein. Wenn zwei oder mehr CPU-Bus-Baugruppen installiert sind, ist die Netzwerkadresse die Baugruppennummer der Baugruppe mit der niedrigsten Baugruppennummer.

5. Die folgenden beiden Methoden können zum Senden eines FINS-Befehls an einen Host-Computer über eine serielle Schnittstelle im Host-Link-Modus verwendet werden, wenn die Kommunikation von der SPS oder der Seriell-Gateway-Funktion initiiert wird (konvertiert in CompoWay/F, Modbus-RTU oder Modbus-ASCII).

   a) Stellen Sie die Adresse der Zielbaugruppe (Bits 00 bis 07 von C+3) auf die Adresse der CPU-Baugruppe oder der seriellen Kommunikationsbaugruppe/des seriellen Kommunikationsmoduls ein und die Nummer der seriellen Schnittstelle (Bits 08 bis 11 von C+2) für Schnittstelle 1 auf 1 bzw. für Schnittstelle 2 auf 2 ein.

Baugruppenadresse (C+3, Bits 00 bis 07)	Baugruppe	Nummer der seriellen Schnittstelle (C+2, Bits 08 bis 11)	Serielle Schnittstelle
00 hex	CPU-Baugruppe	1 hex	Integrierte RS-232C-Schnittstelle
		2 hex	Peripheries chnittstelle
10 hex + Baugruppennummer	Serielle Kommunikationsbaugruppe (CPU-Bus-Baugruppe)	1 hex	Schnittstelle 1
		2 hex	Schnittstelle 2
E1 hex	Serielles Kommunikationsmodul (internes Sondermodul) (nur CS-Serie)	1 hex	Schnittstelle 1
		2 hex	Schnittstelle 2

   b) Stellen Sie die Adresse der Zielbaugruppe direkt in den Bits 00 bis 07 von C+3 ein. In diesem Fall stellen Sie die Nummer der seriellen Schnittstelle zur direkten Spezifikation in den Bits 08 bis 11 von C+2 auf 0 ein.

   Schnittstellen der seriellen Kommunikationsbaugruppe

Schnittstelle	Baugruppenadresse der Schnittstelle	Beispiel: Baugruppennummer = 1
Schnittstelle 1	80 hex + 4 × Baugruppennummer	80 + 4 × 1 = 84 hex (132 dezimal)
Schnittstelle 2	81 hex + 4 × Baugruppennummer	81 + 4 × 1 = 85 hex (133 dezimal)

   Schnittstellen des seriellen Kommunikationsmoduls

Schnittstelle	Baugruppenadresse der Schnittstelle
Schnittstelle 1	E4 hex (228 dezimal)
Schnittstelle 2	E5 hex (229 dezimal)

   Schnittstellen der CPU-Baugruppe

Schnittstelle	Baugruppenadresse der Schnittstelle
Peripheriesch nittstelle	FD hex (253 dezimal)
RS-232C	FC hex (252 dezimal)

6. Bei der Spezifikation der seriellen Schnittstelle ohne Routing-Tabelle für die Seriell-Gateway-Funktion (Konvertierung zu Host-Link-FINS) muss die Baugruppenadresse der seriellen Schnittstelle im Zielnetzwerk-Adressbyte eingestellt werden.

7. Die Baugruppenadresse spezifiziert die Baugruppe, wie in der nachstehenden Tabelle gezeigt.

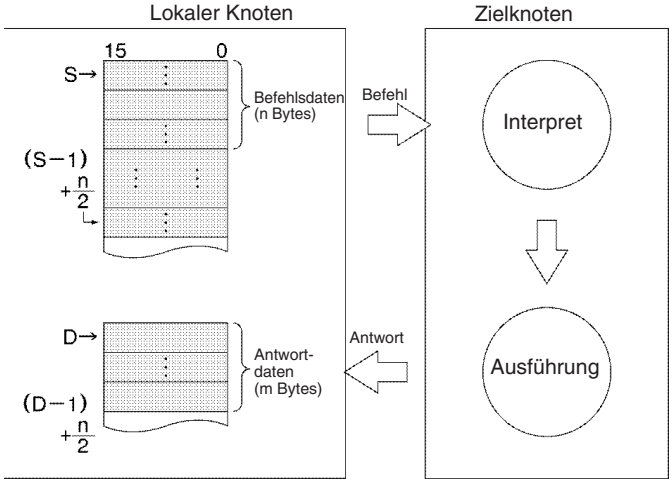
Baugruppe	Einstellung der Baugruppenadresse
CPU-Baugruppe	00 hex
CPU-Bus-Baugruppe	10 hex + Baugruppennummer
Spezial-E/A-Baugruppe (außer Spezial-E/A-Baugruppen der Serie C200H)	20 hex + Baugruppennummer
Internes Sondermodul (nur CS-Serie)	E1 hex
Computer	01 hex
Mit dem Netzwerk verbundene Baugruppe (Spezifizierung der Baugruppe nicht erforderlich)	FE hex
Direkte Spezifikation der Baugruppenadresse der seriellen Schnittstelle	Schnittstellen der seriellen Kommunikationsbaugruppe Schnittstelle 1: 80 hex + 4 × Baugruppennummer Schnittstelle 2: 81 hex + 4 × Baugruppennummer Schnittstellen des seriellen Kommunikationsmoduls Schnittstelle 1: E4 hex (228 dezimal) Schnittstelle 2: E5 hex (229 dezimal) Schnittstellen der CPU-Baugruppe Peripherieschnittstelle: FD hex (253 dezimal) RS-232C-Schnittstelle: FC hex (252 dezimal)

8. Die maximale Knotennummer hängt vom verwendeten Netzwerk ab. Bei einem kabelgebundenen Controller-Link erstreckt sich der zulässige Bereich von 00 bis 20 hexadezimal (0 bis 32 dezimal). Stellen Sie die Zielknotennummer auf FF ein, um an alle Knoten zu senden; stellen Sie sie auf 00 ein, um innerhalb des lokalen Knotens zu senden.
9. Bei der Spezifizierung der seriellen Schnittstelle in der Seriell-Gateway-Funktion (Konvertierung in Host-Link-FINS) muss die Zielbaugruppenadresse auf die Host-Link-Baugruppennummer der Ziel-SPS + 1 eingestellt werden (Einstellbereich: 1 bis 32).
10. Einzelheiten über die Verwendung der automatischen Zuweisung der Kommunikationsschnittstellen-Nummer (logische Schnittstelle) finden Sie unter *Automatische Zuweisung von Kommunikationsschnittstellen* auf Seite 979.
11. Wenn die Zielknotennummer auf FF (Senden an alle Knoten) eingestellt ist, erfolgt keine Antwort, auch wenn die Bits 12 bis 15 auf 0 gesetzt sind.

Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6138
Arbeitsbereich nicht remanent	W000 bis W511		W000 bis W506
Arbeitsbereich remanent	H000 bis H511		H000 bis H506
Systembereich	A000 bis A447 A448 bis A959	A448 bis A959	A000 bis A442 A448 bis A954
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4090
Zählerbereich	C0000 bis C4095		C0000 bis C4090
DM-Bereich	D00000 bis D32767		D00000 bis D32762
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32762
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32763 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl CMND(490) sendet die spezifizierte Anzahl Bytes von FINS-Befehlsdaten (beginnend mit dem Wort S) über den CPU-Bus der SPS oder über ein Netzwerk an das spezifizierte Gerät. Die Antwort wird bei Wort D beginnend in den Speicher geschrieben.



CMND(490) kann zum Senden von Befehlsdaten an eine bestimmte serielle Schnittstelle im Zielgerät sowie an das Gerät selbst verwendet werden. Die Funktion von CMND(490) gleicht der von SEND(090), wenn der FINS-Befehlscode 0102 (MEMORY AREA WRITE) lautet und der von RECV(098), wenn der Code 0101 (MEMORY AREA READ) lautet.

Die CPU-Baugruppe, in der der Befehl CMND(490) ausgeführt wird, kann einen FINS-Befehl an sich selbst senden (außer CS1 CPU-Baugruppen der CS-Serie vor V1□). Verwenden Sie die folgenden Steuerdateneinstellungen, um dies zu erreichen.

- Zielnetzwerkadresse (Bits 00 bis 07 von C+2): 00 hex (lokales Netzwerk)
- Nummer der seriellen Schnittstelle (Bits 08 bis 11 von C+2): 0 hex (nicht verwendet)
- Adresse der Zielbaugruppe (Bits 00 bis 07 von C+3): 00 hex (CPU-Baugruppe)
- Zielknotenadresse (Bits 08 bis 15 von C+3): 00 hex (lokaler Knoten)
- Anzahl der Wiederholversuche (Bits 00 bis 03 von C+4): 0 hex (diese Einstellung ist ungültig; auf 0 einstellen)
- Antwort-Überwachungszeit: (Bits 00 bis 15 von C+5): 0000 bis FFFF hex (allerdings spezifiziert 0000 hier 6553,5 s und nicht 2 s wie normal)

Wenn die Zielknotennummer auf FF eingestellt ist, werden die Befehlsdaten an alle Knoten im spezifizierten Netzwerk gesendet. Das wird auch als „Broadcast-Senden“ bezeichnet.

Wenn eine Antwort angefordert wurde (Bits 12 bis 15 von C+4 auf 0 eingestellt), aber innerhalb der Antwort-Überwachungszeit keine Antwort empfangen wurde, werden die Befehlsdaten bis zu 15 Mal erneut gesendet (die Anzahl der Wiederholversuche wird in den Bits 0 bis 3 von C+3 spezifiziert). Für das Broadcast-Senden gibt es keine Antworten und keine Wiederholversuche. Stellen Sie die Antworteinstellung für Befehle, die keine Antwort benötigen, auf „nicht erforderlich“.

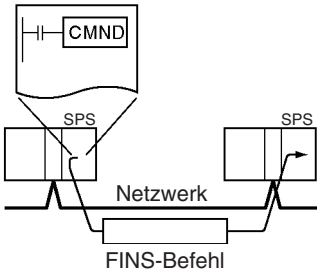
Wenn die Menge der Antwortdaten die für die Antwortdaten in C+1 eingestellte Byte-Anzahl überschreitet, tritt ein Fehler auf.

FINS-Befehlsdaten können an einen Host-Computer gesendet werden, der mit einer seriellen Schnittstelle (bei Einstellung auf den Host-Link-Modus) der SPS verbunden ist sowie an eine SPS (CPU-Baugruppe, internes Sondermodul (nur CS-Serie) oder CPU-Bus-Baugruppe) oder an einen Computer, mit der/dem eine Verbindung über ein Controller-Link- oder Ethernet-Netzwerk besteht.

Wenn das Kommunikationsschnittstelle-Frei-Flag für die in C+3 spezifizierten Kommunikationsschnittstelle bei Ausführung von CMND(490) auf EIN gesetzt ist, werden das entsprechende Kommunikationsschnittstelle-Frei-Flag (Schnittstellen 00 bis 07: A20200 bis A20207) und das Kommunikationsschnittstellen-Fehlerflag (Schnittstellen 00 bis 07: A21900 bis A21907) auf AUS gesetzt und in das Wort, das den Abschlusscode enthält (Schnittstellen 00 bis 07: A203 bis A210) wird 0000 geschrieben. Nachdem die Flags gesetzt sind, werden die Befehlsdaten an den/die Zielknoten gesendet.

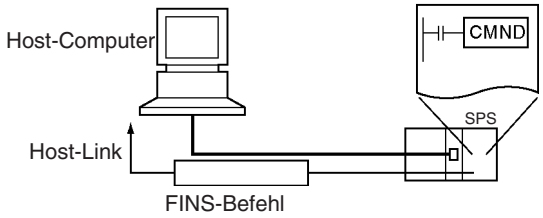
Übertragung über das Netzwerk

Der Befehl CMND(490) kann für das Senden von FINS-Befehlen an einen PC oder eine SPS (CPU-Baugruppe, internes Sondermodul (nur CS-Serie) oder CPU-Bus-Baugruppe) verwendet werden, mit dem/der eine Verbindung über ein Controller-Link- oder Ethernet-Netzwerk besteht.



Übertragung über Host-Link

Wenn die integrierte serielle Schnittstelle der CPU-Baugruppe, ein serielles Kommunikationsmodul (nur CS-Serie) oder eine serielle Kommunikationsbaugruppe auf den Host-Link-Modus eingestellt ist und eins-zu-eins mit einem Host-Computer verbunden ist, kann der Befehl CMND(490) ausgeführt werden, um einen FINS-Befehl von der SPS zum Host-Computer zu senden, wenn die SPS das nächste Mal das Senderecht besitzt. Darüber hinaus ist auch ein Senden an andere Host-Computer möglich, die mit anderen SPS-Systemen an anderer Stelle im Netzwerk verbunden sind.



Der Befehl CMND(490) kann für eine beliebige Schnittstelle der CPU-Baugruppe, eines seriellen Kommunikationsmoduls (nur CS-Serie) oder einer seriellen Kommunikationsbaugruppe ausgeführt werden, um einen Befehl an den verbundenen Host-Computer zu senden. (Spezifizieren Sie die serielle Schnittstelle in den Bits 08 bis 11 von C+2 als 1 hex bzw. 2 hex.) Der Befehl ist eine FINS-Meldung, die zwischen Host-Link-Header und -Endsequenz eingeschlossen ist. Es kann ein beliebiger FINS-Befehl gesendet werden; der Host-Link-Header-Code lautet 0F hexadezimal.

Im Host-Computer muss ein Programm erstellt werden, um den empfangenen Befehl (den zwischen Host-Link-Header und -Endsequenz eingeschlossenen FINS-Befehl) zu verarbeiten.

Wenn sich die serielle Zielschnittstelle in der lokalen SPS befindet, muss die Netzwerkadresse in C+2 auf 00 (lokales Netzwerk), die Knotenadresse in C+3 auf 00 (lokale SPS) und die Baugruppenadresse auf 00 (CPU-Baugruppe), E1 (internes Sondermodul, nur CS-Serie) oder Baugruppennummer + 10 hexadezimal (serielle Schnittstellenbaugruppe) eingestellt werden.

Seriell-Gateway-Kommunikation mit einer Komponente oder einem Host-Link-Slave

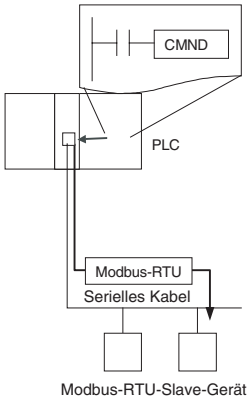
FINS-Befehle (oder Sende-/Empfangsdaten) können an eine Komponente oder einen Host-Link-Slave gesendet werden, die/der mit der Seriell-Gateway-Funktion über seine serielle Schnittstelle mit der SPS verbunden ist.

- Senden an eine Komponente
  (Konvertierung in CompoWay/F, Modbus-RTU oder Modbus-ASCII)

  Die Seriell-Gateway-Funktion kann die folgenden FINS-Befehle in CompoWay/F-, Modbus-RTU- oder Modbus-ASCII-Befehle konvertieren, wenn der FINS-Befehl an eine serielle Schnittstelle eines seriellen Kommunikationsmoduls, einer seriellen Kommunikationsbaugruppe oder der CPU-Baugruppe (Peripherie- oder RS-232C-Schnittstelle) gesendet wird.

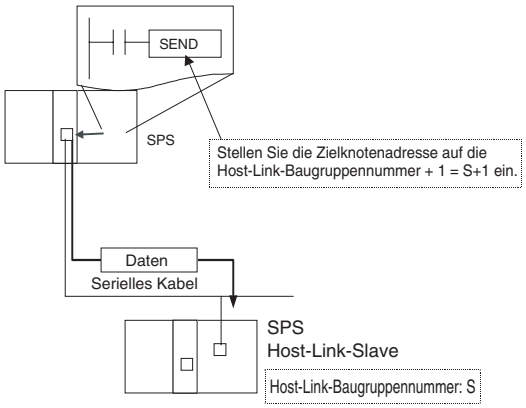
  Konvertierung in CompoWay/F-Befehl: 2803 hex
  Konvertierung in Modbus-RTU-Befehl: 2804 hex (siehe Hinweis)
  Konvertierung in Modbus-ASCII-Befehl: 2805 hex (siehe Hinweis)

  Hinweis Modbus-RTU- und Modbus-ASCII-Befehle können nicht an die seriellen Schnittstellen der CPU-Baugruppe gesendet werden.



- Senden an eine als Host-Link-Slave arbeitende SPS

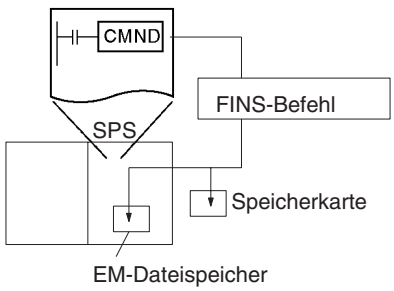
  Die Seriell-Gateway-Funktion kann zum Senden von FINS-Befehlen an eine SPS verwendet werden, die als Host-Link-Slave über eine serielle Schnittstelle eines seriellen Kommunikationsmoduls oder einer seriellen Kommunikationsbaugruppe verbunden ist. In diesem Fall muss die Zielknotenadresse auf die Host-Link-Baugruppennummer + 1 eingestellt werden.



Senden eines FINS-Befehls an die CPU-Baugruppe, in der der CMND(490)-Befehl ausgeführt wird (außer CS1 CPU-Baugruppen der CS-Serie vor V1)

Die den Befehl CMND(490) ausführende CPU-Baugruppe kann einen FINS-Befehl an sich selbst senden (außer CS1 CPU-Baugruppen der CS-Serie ohne den Suffix -V□). Beispielsweise können Datenspeicherbefehle (Befehlscodes 22□□ hex) zum Formatieren von Datenspeicher, zum Löschen oder Kopieren von Dateien oder zur Durchführung anderer Funktionen gesendet werden. Weitere Informationen finden Sie unter *5-2 Dateifunktionen* im *CS/CJ-Serie CPU-Baugruppen Programmierhandbuch*.

Das Dateispeicherfunktions-Flag (A34313) wird auf EIN gesetzt, wenn ein FINS-Befehl an die lokale CPU-Baugruppe gesendet wird (auch bei FINS-Befehlen, die sich nicht auf den Dateispeicher beziehen). Verwenden Sie A34313 immer als Schließer-Eingangsbedingung für CMND(490), um sicherzustellen, dass nur jeweils ein FINS-Befehl für die CPU-Baugruppe ausgeführt wird.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn die Nummer der in C+2 spezifizierten serielle Schnittstellennummer nicht innerhalb des Bereichs von 00 bis 04 liegt. EIN, wenn das Kommunikationsschnittstelle-Frei-Flag für die in C+4 spezifizierte Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. EIN, wenn ein FINS-Befehl an die lokale CPU-Baugruppe gesendet wird, während das Dateispeicherfunktions-Flag (A34313) auf EIN gesetzt ist. In allen anderen Fällen AUS.

In der folgenden Tabelle sind die relevanten Bits und Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Kommunika-tionsschnitt-stelle-Frei-Flag	A20200 bis A20207	Diese Flags werden auf EIN gesetzt, um anzugeben, dass Netzwerkbefehle (einschließlich PMCR(260)) für die entsprechenden Schnittstellen (00 bis 07) ausgeführt werden können. Ein Flag wird auf AUS gesetzt, wenn ein Netzwerkbefehl für die entsprechende Schnittstelle ausgeführt wird, und wieder auf EIN gesetzt, wenn die Ausführung des Befehls abgeschlossen ist.
Kommunika-tionsschnitt-stellen-Fehlerflag	A21900 bis A21907	Diese Flags werden auf EIN gesetzt, um anzugeben, dass an den entsprechenden Schnittstellen (00 bis 07) während der Ausführung eines Netzwerkbefehls ein Fehler aufgetreten ist. Der Flag-Status wird bis zur Ausführung des nächsten Netzwerkbefehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunika-tionsschnitt-stellen-Abschlus-scodes	A203 bis A210	Diese Worte enthalten nach der Ausführung eines Netzwerkbefehls die Abschlusscodes für die entsprechenden Schnittstellen (00 bis 07). Das entsprechende Wort hat während Ausführung des Netzwerkbefehls den Inhalt 0000. Nach Abschluss der Befehlsausführung wird der Abschlusscode in dieses Wort geschrieben. Diese Worte werden zu Beginn der Programmausführung gelöscht.
Dateispeicher-funktions-Flag	A34313	EIN, wenn ein FINS-Befehl an die lokale CPU-Baugruppe gesendet wird (auch bei FINS-Befehlen, die sich nicht auf den Dateispeicher beziehen) oder wenn einer/eine der folgenden Befehle oder Funktionen für den Dateispeicher ausgeführt wird. FREAD(700) oder FWRIT(701) Überschreiben des Programms mit Steuerbit im Speicher Easy-Backup-Sicherungsvorgang

Vorsichtsmaßnahmen

Wenn das Kommunikationsschnittstelle-Frei-Flag für die in C+4 spezifizierte Schnittstellennummer auf AUS gesetzt ist, wird der Befehl als NOP(000) behandelt und nicht ausgeführt. In diesem Fall wird das Fehler-Flag auf EIN gesetzt.

Wenn Daten über die Grenzen des lokalen Netzwerks hinaus gesendet werden sollen, muss der Anwender Routing-Tabellen in den CPU-Baugruppen der SPS-Systeme jedes Netzwerks registrieren. (Routing-Tabellen bestimmen die Routen zu anderen Netzwerken, in die die Zielknoten eingebunden sind.)

Weitere Informationen über Abschlusscodes bei der Netzwerkkommunikation finden Sie im *CS/CJ-Serie Kommunikationsbefehle-Referenzhandbuch* (W342) unter FINS-Befehl-Antwortcodes.

Die Kommunikationsschnittstellen-Nummern 00 bis 07 werden vom Netzwerk und den seriellen Kommunikationsbefehlen (SEND(090), RECV(098), CMND(490), PMCR(260), TXDU(256) oder RXDU(255)) gemeinsam verwen-

det, so dass nur jeweils einer dieser Befehle für eine Kommunikationsschnittstelle ausgeführt werden darf. Programmieren Sie das Kommunikationsschnittstelle-Frei-Flag (A20200 bis A20207) als Schließerbedingung, um sicherzustellen, dass der Befehl CMND(490) nicht ausgeführt wird, wenn die jeweilige Schnittstelle belegt ist.

Verwenden Sie beim Senden eines FINS-Befehls an die lokale CPU-Baugruppe immer eines der Kommunikationsschnittstelle-Frei-Flags (A20200 bis A20207) in einer Schließer-Eingangsbedingung und das Dateispeicherfunktions-Flag (A34313) in einer Öffner-Eingangsbedingung für CMND(490).

Störungen und andere Faktoren können dazu führen, dass die Sendung oder Antwort fehlerhaft ist oder verloren geht. Deshalb empfehlen wir, die Anzahl der Wiederholversuche auf einen Wert größer als Null zu setzen, wodurch CMND(490) erneut ausgeführt wird, wenn die Antwort nicht innerhalb der Antwort-Überwachungszeit empfangen wird.

Beispiele

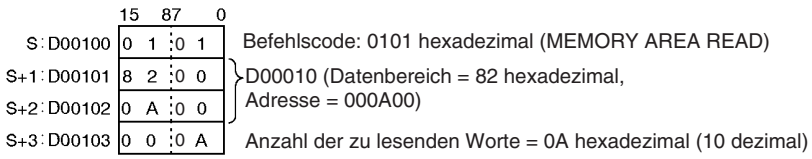
Der folgende Programmabschnitt zeigt ein Beispiel für das Senden eines FINS-Befehls an eine andere CPU-Baugruppe.

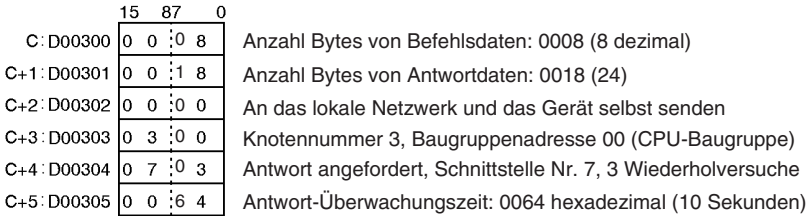
Wenn CIO 000000 und A20207 (das Kommunikationsschnittstelle-Frei-Flag für Schnittstelle 07) auf EIN gesetzt sind, sendet CMND(490) den FINS-Befehl 0101 (MEMORY AREA READ) an Knoten Nummer 3. Die Antwort wird in D00200 bis D00211 gespeichert.

Der Befehl MEMORY AREA READ liest 10 Worte von D00010 bis D00019. Die Antwort enthält den 2-Byte-Befehlscode (0101), den 2-Byte-Abschlusscode und dann die 10 Datenworte, besteht also aus insgesamt 12 Worten oder 24 Bytes.

Die Daten werden bis zu dreimal wiederholt gesendet, wenn nicht innerhalb von zehn Sekunden eine Antwort empfangen wird.

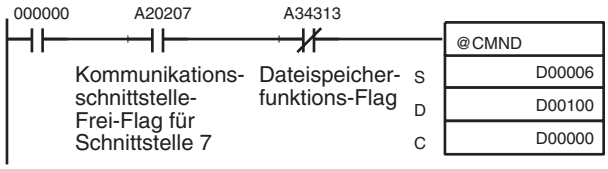






Der folgende Programmabschnitt zeigt ein Beispiel für das Senden eines FINS-Befehls an die lokale CPU-Baugruppe.

Wenn CIO 000000 und A20207 (das Kommunikationsschnittstelle-Frei-Flag für Schnittstelle 07) auf EIN und A34313 (das Dateispeicherfunktions-Flag) auf AUS gesetzt sind, sendet CMND(490) den FINS-Befehl 2215 (CREATE/ DELETE DIRECTORY) an die lokale CPU-Baugruppe. Die Antwort wird in D00100 bis D00101 gespeichert. Hier erstellt der FINS-Befehl im Verzeichnis „OMRON“ ein Verzeichnis mit der Bezeichnung „CS/CJ“. Als Antwort werden der Befehlscode (2 Bytes) und der Abschlusscode (2 Bytes) zurückgegeben und gespeichert.



		15 8 7 0	
S:	D00006	2 2 1 5	Befehlscode: 2215 hex (CREATE/DELETE DIRECTORY)
S+1:	D00007	8 0 0 0	Datenträger-Nr.: 8000 hex (Speicherkarte)
S+2:	D00008	0 0 0 0	Parameter: 0000 hex (Verzeichnis erstellen)
S+3:	D00009	4 3 5 3	Name des Unterverzeichnisses: CS1□□□□□. □□□ (□= Leerzeichen)
S+4:	D00010	3 1 2 0	
S+5:	D00011	2 0 2 0	
S+6:	D00012	2 0 2 0	
S+7:	D00013	2 E 2 0	
S+8:	D00014	2 0 2 0	
S+9:	D00015	0 0 0 6	Verzeichnisnamenlänge: 0006 (6 Zeichen)
S+10:	D00016	5 C 4 F	Absoluter Verzeichnispfad: \OMRON
S+11:	D00017	4 D 5 2	
S+12:	D00018	4 F 4 E	

		15 8 7 0	
S:	D00000	0 0 1 A	Anzahl Bytes von Befehlsdaten: 001A (26 dezimal)
S+1:	D00001	0 0 0 4	Anzahl Bytes von Antwortdaten: 0004 (4)
S+2:	D00002	0 0 0 0	Zielnetzwerkadresse: 00 hex (lokales Netzwerk)
S+3:	D00003	0 0 0 0	Adresse der Zielbaugruppe: 00 hex, Zielknotennummer: 00 hex (CPU-Baugruppe an lokalem Knoten)
S+4:	D00004	0 7 0 0	Antwort angefordert, Schnittstelle Nr. 7, 0 Wiederholversuche
S+5:	D00005	0 0 0 0	Antwort-Überwachungszeit: 0000 hex (6553,5 Sekunden)

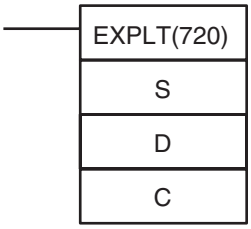
3-25-6 EXPLICIT MESSAGE SEND: EXPLT(720)

Funktion

Sendet eine explizite Meldung mit einem beliebigen Service-Code.

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



S: Erstes Wort der zu sendenden Meldung
D: Erstes Wort der empfangenen Meldung
C: Erstes Steuerwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	EXPLT(720)
	Ausführung einmal bei steigender Flanke	@EXPLT(720)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

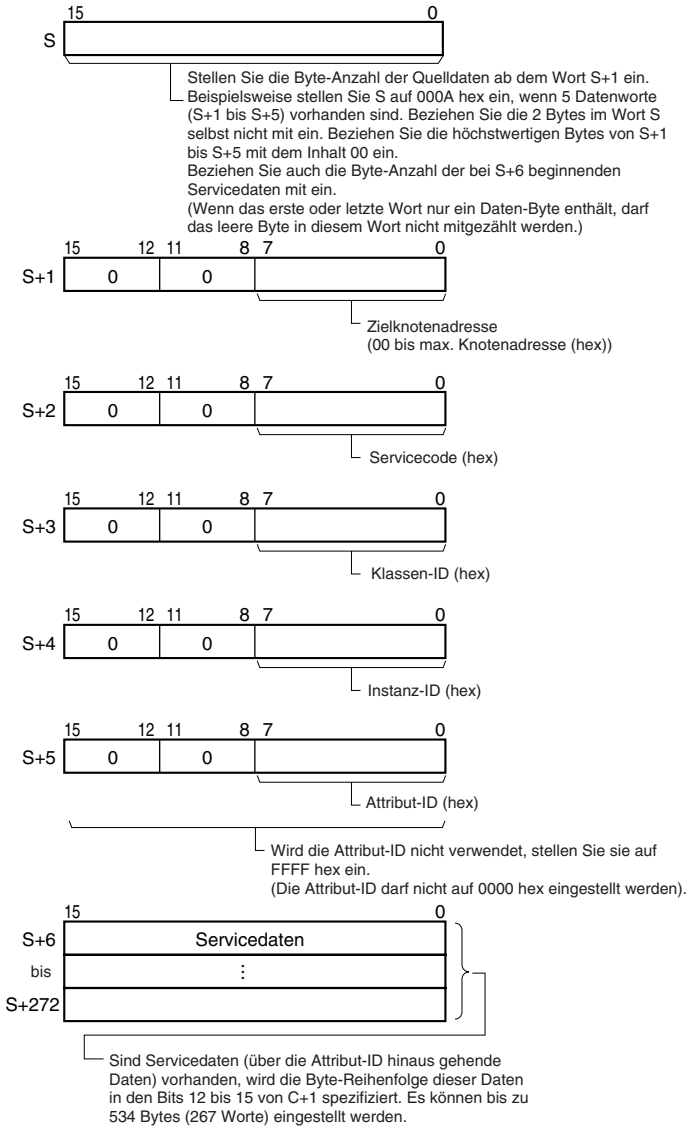
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

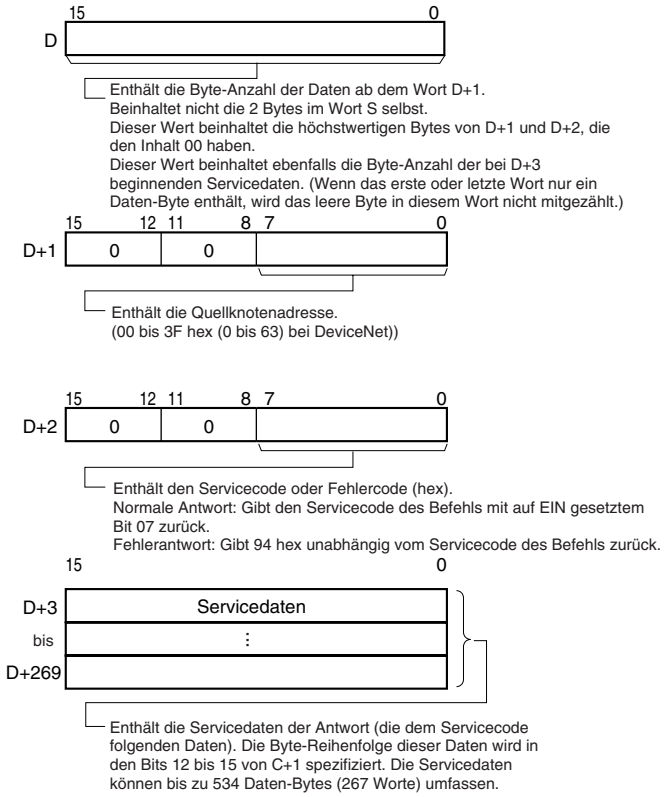
S: Erstes Wort der zu sendenden Meldung

Spezifiziert das erste Wort der zu sendenden Meldung (S bis max. S+272).



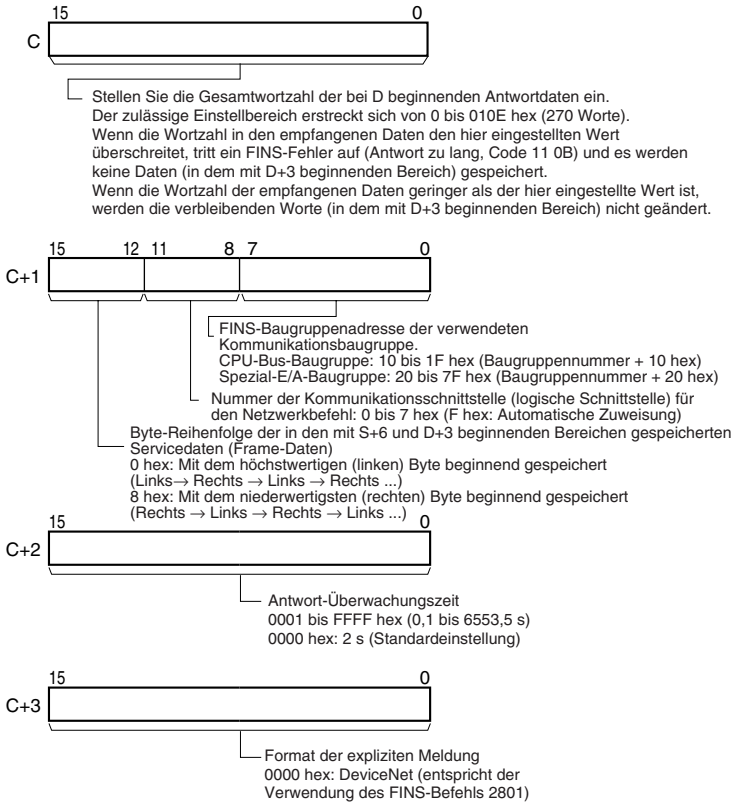
D: Erstes Wort der empfangenen Meldung

Spezifiziert das erste Wort der empfangenen Meldung (D bis max. D+269).



C: Erstes Steuerwort

Spezifiziert das erste von vier Steuerworten (C bis C+3).



Operanden-Spezifikationen

Bereich	S	D	C
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6140
Arbeitsbereich nicht remanent	W000 bis W511		W000 bis W508
Arbeitsbereich remanent	H000 bis H511		H000 bis H508
Systembereich	A000 bis A959	A448 bis A959	A000 bis A956
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4092
Zählerbereich	C0000 bis C4095		C0000 bis C4092
DM-Bereich	D00000 bis D32767		D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

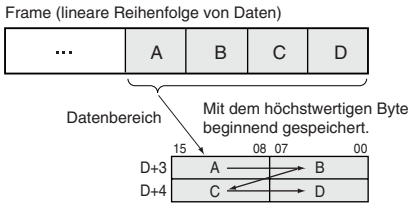
Sendet den Explizite-Meldung-Befehl (im mit S+2 beginnenden Wortbereich gespeichert) an die in S+1 spezifizierte Knotenadresse über die Kommunikationsbaugruppe mit der in den Bits 00 bis 07 von C+1 spezifizierten FINS-Baugruppenadresse. Wenn die Antwort auf die expliziten Meldung empfangen wird, wird diese im mit D+2 beginnenden Wortbereich gespeichert.

Einstellungen für die Byte-Anzahl

Die Byte-Anzahl der gesendeten Daten in S beinhaltet die 10 Bytes in S+1 bis S+5 ebenso wie die Byte-Anzahl der bei S+6 beginnenden Servicedaten. (Bei z.B. 1 Byte Servicedaten beträgt die Gesamtsumme aller Daten 11 Bytes. Aus diesem Grund muss S auf 000B hex eingestellt werden.)

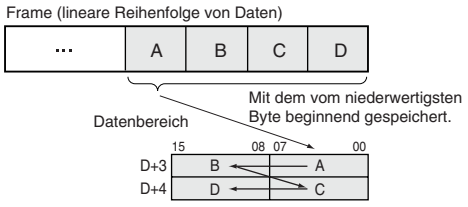
Die Byte-Anzahl der empfangenen Daten in D beinhaltet die 4 Bytes in D+1 und D+2 ebenso wie die Byte-Anzahl der mit D+3 beginnenden Servicedaten. (Bei z.B. 1 Byte Servicedaten beträgt die Gesamtsumme aller Daten 5 Bytes und D enthält 0005 hex.)
Die Einstellung in Bits 12 bis 15 von C+1 (0 oder 8 hex) bestimmt die Byte-Reihenfolge der in S+6 und D+3 gespeicherten Servicedaten.

- Speichern von Daten beginnend mit dem höchstwertigen Byte. Stellen Sie die Bits 12 bis 15 von C+1 auf 0 hex ein.



Hinweis: A, B, C und D stellen Daten-Bytes dar.

- Speichern von Daten beginnend mit dem niederwertigsten Byte. Stellen Sie die Bits 12 bis 15 von C+1 auf 8 hex ein.



Hinweis: A, B, C und D stellen Daten-Bytes dar.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das Kommunikationsschnittstelle-Frei-Flag für die Kommunikationsschnittstelle mit der in C spezifizierten Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

Das entsprechende Explizite-Kommunikation-Fehlerflag wird auf AUS gesetzt, wenn die Ausführung des Befehls normal abgeschlossen wird, oder auf EIN gesetzt, wenn ein Fehler aufgetreten ist.

Bei Auftreten eines Fehlers (entsprechendes Flag in A213 auf EIN gesetzt) kann das entsprechende Kommunikationsschnittstellen-Fehlerflag zur Ermittlung verwendet werden, ob die explizite Meldung selbst nicht gesendet wurde (entsprechendes Flag in A219 auf EIN gesetzt) oder ob die Meldung gesendet wurde, aber einen Fehler enthielt (entsprechendes Flag in A219 auf AUS gesetzt).

Der entsprechende Kommunikationsschnittstellen-Abschlusscode (A203 bis A210) enthält 0000 hex, wenn die Ausführung des Befehls normal abgeschlossen wurde, einen Explizite-Meldung-Fehlercode, wenn ein Fehler bei der expliziten Meldung aufgetreten ist, oder einen FINS-Fehlercode, wenn ein FINS-Fehler aufgetreten ist.

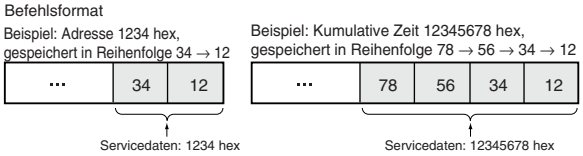
Weitere Informationen über die allgemeinen Funktionen der Befehle für explizite Meldungen sind unter *3-25-2 Informationen zu Explizite-Meldung-Befehlen* zu finden.

In der folgenden Tabelle sind die relevanten Bits und Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Kommunikationsschnittstelle-Frei-Flag	A20200 bis A20207	Diese Flags werden auf EIN gesetzt, um anzugeben, dass Netzwerkbefehle (einschließlich PMCR(260)) für die entsprechenden Schnittstellen (00 bis 07) ausgeführt werden können. Ein Flag wird auf AUS gesetzt, wenn ein Netzwerkbefehl für die entsprechende Schnittstelle ausgeführt wird, und wieder auf EIN gesetzt, wenn die Ausführung des Befehls abgeschlossen ist.
Explizite-Kommunikation-Fehlerflag	A21300 bis A21307	Diese Flags werden auf EIN gesetzt, um anzugeben, dass während der Ausführung der Kommunikation mittels expliziter Meldungen an den entsprechenden Schnittstellen (00 bis 07) ein Fehler aufgetreten ist. Die Flags werden auf EIN gesetzt, wenn die explizite Meldung nicht gesendet wurde oder die Meldung gesendet wurde, aber eine Fehlermeldung zurückgegeben wurde. Der Flag-Status wird bis zur Ausführung des nächsten Explizite-Meldung-Befehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunikationsschnittstellen-Fehlerflag	A21900 bis A21907	Diese Flags werden auf EIN gesetzt, um anzugeben, dass die explizite Meldung während der Ausführung eines Explizite-Meldung-Befehl nicht über die entsprechende Schnittstelle (00 bis 07) gesendet wurde. Der Flag-Status wird bis zur Ausführung des nächsten Netzwerkbefehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunikationsschnittstellen-Abschlusscodes	A203 bis A210	Diese Worte enthalten nach der Ausführung eines Netzwerkbefehls die Abschlusscodes für die entsprechenden Schnittstellen (00 bis 07). Das entsprechende Wort hat den Inhalt 0000, solange das Explizite-Kommunikation-Fehlerflag auf AUS gesetzt ist. Das entsprechende Wort enthält einen FINS-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle beide auf EIN gesetzt sind. Das entsprechende Wort enthält den entsprechenden Explizite-Meldung-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag dieser Schnittstelle auf EIN gesetzt ist und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle auf AUS gesetzt ist. Das entsprechende Wort hat während Ausführung des Netzwerkbefehls den Inhalt 0000. Nach Abschluss der Befehlsausführung wird der Abschlusscode in dieses Wort geschrieben. Diese Worte werden zu Beginn der Programmausführung gelöscht.

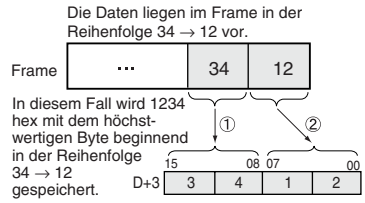
Vorsichtsmaßnahmen

Achten Sie darauf, dass die Reihenfolge der Bytes in den Quelldaten der Reihenfolge im Frame der expliziten Meldung (lineare Reihenfolge der Daten) entspricht. Wenn z.B. die Servicedaten in 2-Byte- oder 4-Byte-Einheiten vorliegen, liegen die Daten im Frame in Zweierpaaren, jeweils in der Reihenfolge vom höchstwertigen Byte zum niederwertigsten Byte, vor (siehe nachfolgende Abbildung).

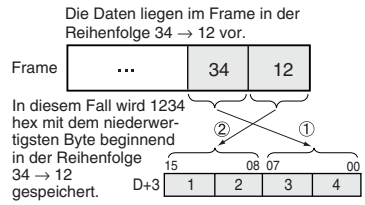


In den folgenden Abbildungen ist dargestellt, in welcher Form die Daten in den Datenbereichen gespeichert werden, wenn die Servicedaten in 2-Byte- oder 4-Byte-Einheiten vorliegen.

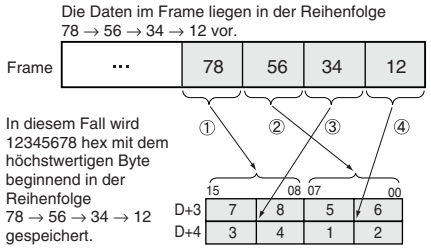
1. Daten in 2-Byte-Einheiten
   - Speichern von Daten mit dem höchstwertigen Byte beginnend (Bits 12 bis 15 von C = 0 hex)
     Beispiel: Speichern des Werts 1234 hex in D+3



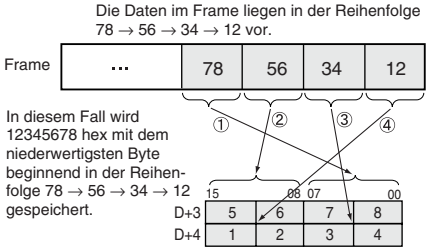
   - Speichern von Daten mit dem niederwertigsten Byte beginnend (Bits 12 bis 15 von C = 8 hex)
     Beispiel: Speichern des Werts 1234 hex in D+3



2. Daten in 4-Byte-Einheiten
   - Speichern von Daten mit dem höchstwertigen Byte beginnend (Bits 12 bis 15 von C = 0 hex)
     Beispiel: Speichern des Werts 12345678 hex in D+3 und D+4

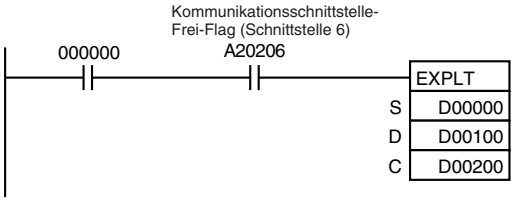


   - Speichern von Daten mit dem niederwertigsten Byte beginnend (Bits 12 bis 15 von C = 8 hex)
     Beispiel: Speichern des Werts 12345678 hex in D+3 und D+4



Hinweis Die obigen Beispiele zeigen die Speicherung von empfangenen Daten in D+3, gesendete Daten werden auf dieselbe Weise in S+6 gespeichert.

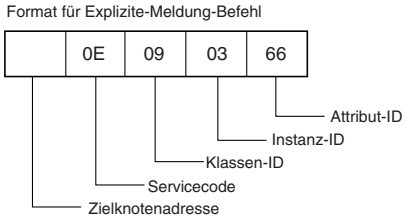
Beispiel In diesem Beispiel wird der Befehl EXPLT(720) zum Auslesen der Gesamteinschaltzeit oder der Anzahl der Schaltspiele von einem DRT2-Slave (E/A-Modul) verwendet.

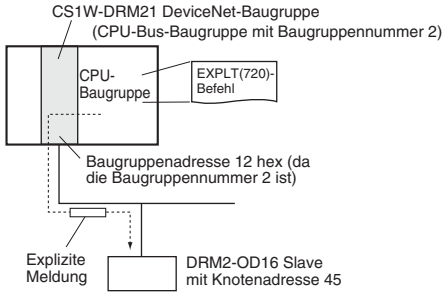


Wenn CIO 000000 und A20206 (das Kommunikationsschnittstelle-Frei-Flag für Schnittstelle 06) auf EIN gesetzt sind, liest EXPLT(720) die Gesamteinschaltzeit (in s) oder die Anzahl der Schaltspiele von einem DRT2-Slave (E/A-Modul). In diesem Fall wird die Gesamteinschaltzeit oder die Anzahl der Schaltspiele für Eingang 3 gelesen.

Servicecode = 0E hex, Klassen-ID = 09 hex, Instanz-ID = 03 hex und Attribut-ID = 66 hex.

In diesem Beispiel wird der Wert 2.752.039 s als Antwort für die Gesamteinschaltzeit zurückgegeben.





S:	D00000	0	0	0	A	Anzahl der Datenbytes: S+1 bis S+5 = 5 Worte = 10 Bytes = 0A hex
S+1:	D00001	0	0	2	D	Slave-Knotenadresse = 45 = 2D hex
S+2:	D00002	0	0	0	E	Servicecode = 0E hex
S+3:	D00003	0	0	0	9	Klassen-ID = 09 hex
S+4:	D00004	0	0	0	3	Instanz-ID = 03 hex (Eingang 3)
S+5:	D00005	0	0	6	6	Attribut-ID = 66 hex

D:	D00100	0	0	0	8	Enthält 08 hex für 8 Bytes im Antwort-Frame empfangener Daten.
D+1:	D00101	0	0	2	D	Gibt die Slave-Knotenadresse zurück = 45 = 2D hex.
D+2:	D00102	0	0	8	E	Service-Code = 8E hex (normaler Abschluss)
D+3:	D00103	2	7	F	E	
D+4:	D00104	2	9	0	0	Service-Daten = 0029FE27 hex (2.752.039 s dezimal)

C:	D00200	0	0	0	4	5 Worte einstellen = 0005 hex, da sich 5 Worte in D bis D+5 befinden.
C+1:	D00201	0	6	1	2	Byte-Reihenfolge = 0 hex (mit höchstwertigem Byte beginnen), Kommunikationsschnittstelle = 6 hex (Schnittstelle 6) und die DeviceNet-Baugruppenadresse = 12 hex
C+2:	D00202	0	0	0	0	Antwort-Überwachungszeit = 0000 hex (2 s)
C+3:	D00203	0	0	0	0	Formattyp für explizite Meldung = 0000 hex (DeviceNet-Format)

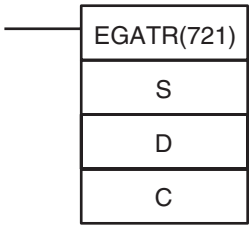
3-25-7 EXPLICIT GET ATTRIBUTE: EGATR(721)

Funktion

Sendet einen Informations-/Status-Lesebefehl in einer expliziten Meldung (Einzelnes Attribut abrufen, Service-Code: 0E hex).

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



S: Erstes Wort der zu sendenden Meldung
D: Erstes Wort der empfangenen Meldung
C: Erstes Steuerwort

Variationen

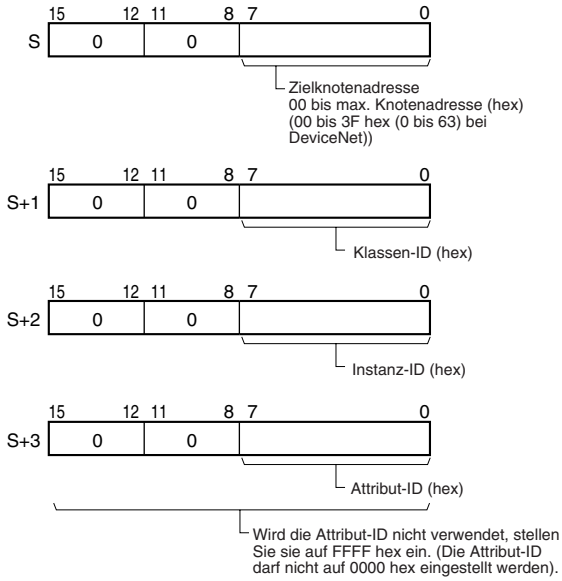
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	EGATR(721)
	Ausführung einmal bei steigender Flanke	@EGATR(721)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

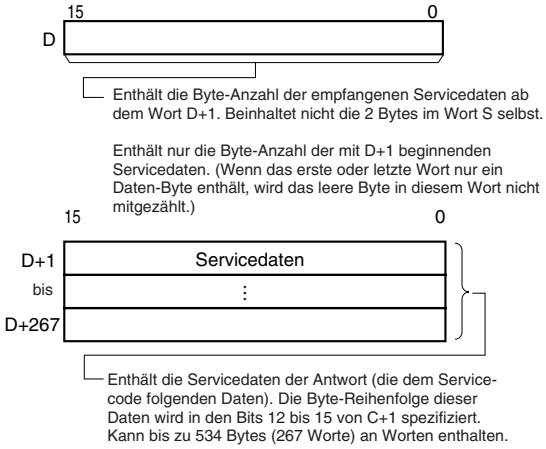
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

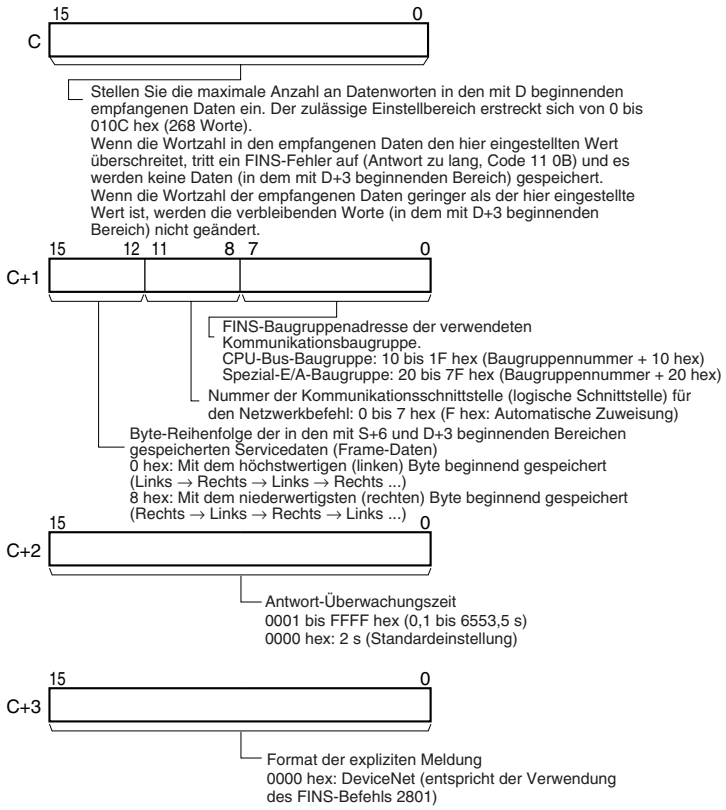
S: Erstes Wort der zu sendenden Meldung
Spezifiziert das erste Wort der zu sendenden Meldung (S bis S+3).



D: Erstes Wort der empfangenen Meldung
Spezifiziert das erste Wort der empfangenen Meldung (D bis max. D+267).



C: Erstes Steuerwort
Spezifiziert das erste von vier Steuerworten (C bis C+3).



Operanden-Spezifikationen

Bereich	S	D	C
CIO-Bereich	CIO 0000 bis CIO 6140	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6140
Arbeitsbereich nicht remanent	W000 bis W508	W000 bis W511	W000 bis W508
Arbeitsbereich remanent	H000 bis H508	H000 bis H511	H000 bis H508
Systembereich	A000 bis A956	A000 bis A959	A000 bis A956
Zeitgeberbereich	T0000 bis T4092	T0000 bis T4095	T0000 bis T4092

Bereich	S	D	C
Zählerbereich	C0000 bis C4092	C0000 bis C4095	C0000 bis C4092
DM-Bereich	D00000 bis D32764	D00000 bis D32767	D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32764	E00000 bis E32767	E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

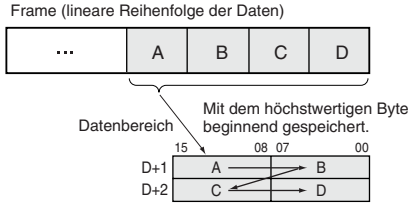
Beschreibung

Sendet den Explizite-Meldung-Befehl „Information/Status lesen“ (gespeichert in den Worten S+1 bis S+3) an die in S spezifizierte Knotenadresse über die Kommunikationsbaugruppe mit der in den Bits 00 bis 07 von C+1 spezifizierten FINS-Baugruppenadresse.

Wenn die Antwort auf die explizite Meldung empfangen wird, werden die Servicedaten der Antwort (die dem Servicecode folgenden Daten) in dem mit D+1 beginnenden Wortebereich gespeichert.

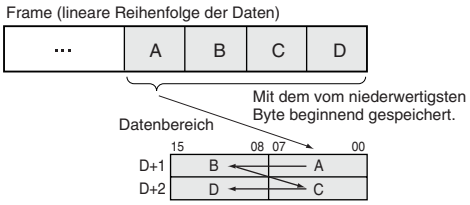
Die in D angegebene Byte-Anzahl der empfangenen Daten ist die Byte-Anzahl der Servicedaten. (Wenn z.B. 1 Byte Servicedaten vorhanden ist, hat D den Inhalt 0001 hex. D enthält 0001 hex, unabhängig von der Einstellung für die Byte-Reihenfolge, also unabhängig davon, ob das Byte im niederwertigsten oder höchstwertigen Byte von D gespeichert ist.)
Die Einstellung in den Bits 12 bis 15 von C+1 (0 oder 8 hex) bestimmt die Byte-Reihenfolge der in S+6 und D+3 gespeicherten Servicedaten.

- Speichern von Daten mit dem höchstwertigen Byte beginnend
  Stellen Sie die Bits 12 bis 15 von C+1 auf 0 hex ein.



Hinweis: A, B, C und D stellen Daten-Bytes dar.

- Speichern von Daten mit dem niederwertigsten Byte beginnend
  Stellen Sie die Bits 12 bis 15 von C+1 auf 8 hex ein.



Hinweis: A, B, C und D stellen Daten-Bytes dar.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das Kommunikationsschnittstelle-Frei-Flag für die Kommunikationsschnittstelle mit der in C spezifizierten Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

Das entsprechende Explizite-Kommunikation-Fehlerflag wird auf AUS gesetzt, wenn die Ausführung des Befehls normal abgeschlossen wird, oder auf EIN gesetzt, wenn ein Fehler aufgetreten ist.

Bei Auftreten eines Fehlers (entsprechendes Flag in A213 auf EIN gesetzt) kann das entsprechende Kommunikationsschnittstellen-Fehlerflag zur Ermittlung verwendet werden, ob die explizite Meldung selbst nicht gesendet wurde (entsprechendes Flag in A219 auf EIN gesetzt) oder ob die Meldung gesendet wurde, aber einen Fehler enthielt (entsprechendes Flag in A219 auf AUS gesetzt).

Der entsprechende Kommunikationsschnittstellen-Abschlusscode (A203 bis A210) enthält 0000 hex, wenn die Ausführung des Befehls normal abgeschlossen wurde, einen Explizite-Meldung-Fehlercode, wenn ein Fehler bei der expliziten Meldung aufgetreten ist, oder einen FINS-Fehlercode, wenn ein FINS-Fehler aufgetreten ist.

Weitere Informationen über die allgemeinen Funktionen der Befehle für explizite Meldungen sind unter *3-25-2 Informationen zu Explizite-Meldung-Befehlen* zu finden.

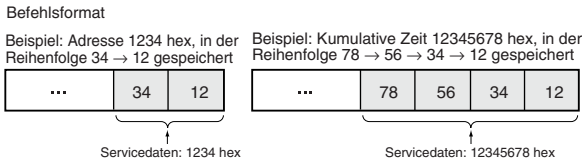
In der folgenden Tabelle sind die relevanten Bits und Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Kommunikationsschnittstelle-Frei-Flag	A20200 bis A20207	Diese Flags werden auf EIN gesetzt, um anzugeben, dass Netzwerkbefehle (einschließlich PMCR(260)) für die entsprechenden Schnittstellen (00 bis 07) ausgeführt werden können. Ein Flag wird auf AUS gesetzt, wenn ein Netzwerkbefehl für die entsprechende Schnittstelle ausgeführt wird, und wieder auf EIN gesetzt, wenn die Ausführung des Befehls abgeschlossen ist.
Explizite-Kommunikation-Fehlerflag	A21300 bis A21307	Diese Flags werden auf EIN gesetzt, um anzugeben, dass während der Ausführung der Kommunikation mittels expliziter Meldungen an den entsprechenden Schnittstellen (00 bis 07) ein Fehler aufgetreten ist. Die Flags werden auf EIN gesetzt, wenn die explizite Meldung nicht gesendet wurde oder die Meldung gesendet wurde, aber eine Fehlermeldung zurückgegeben wurde. Der Flag-Status wird bis zur Ausführung des nächsten Explizite-Meldung-Befehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.

Name	Adresse	Funktion
Kommunikationsschnittstellen-Fehlerflag	A21900 bis A21907	Diese Flags werden auf EIN gesetzt, um anzugeben, dass die explizite Meldung während der Ausführung eines Explizite-Meldung-Befehl nicht über die entsprechende Schnittstelle (00 bis 07) gesendet wurde. Der Flag-Status wird bis zur Ausführung des nächsten Netzwerkbefehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunikationsschnittstellen-Abschlusscodes	A203 bis A210	Diese Worte enthalten nach der Ausführung eines Netzwerkbefehls die Abschlusscodes für die entsprechenden Schnittstellen (00 bis 07). Das entsprechende Wort hat den Inhalt 0000, solange das Explizite-Kommunikation-Fehlerflag auf AUS gesetzt ist. Das entsprechende Wort enthält einen FINS-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle beide auf EIN gesetzt sind. Das entsprechende Wort enthält den entsprechenden Explizite-Meldung-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag dieser Schnittstelle auf EIN gesetzt ist und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle auf AUS gesetzt ist. Das entsprechende Wort hat während Ausführung des Netzwerkbefehls den Inhalt 0000. Nach Abschluss der Befehlsausführung wird der Abschlusscode in dieses Wort geschrieben. Diese Worte werden zu Beginn der Programmausführung gelöscht.

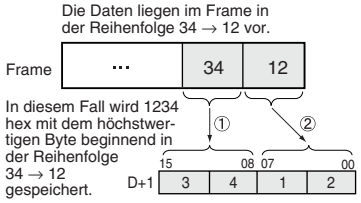
Vorsichtsmaßnahmen

Achten Sie darauf, dass die Reihenfolge der Bytes in den Quelldaten der Reihenfolge im Frame der expliziten Meldung (lineare Reihenfolge der Daten) entspricht. Wenn z.B. die Servicedaten in 2-Byte- oder 4-Byte-Einheiten vorliegen, liegen die Daten im Frame in Zweierpaaren, jeweils in der Reihenfolge vom höchstwertigen Byte zum niederwertigsten Byte, vor (siehe nachfolgende Abbildung).

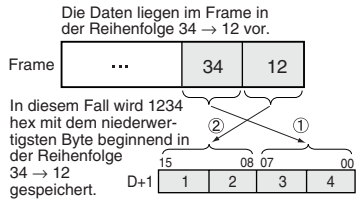


In den folgenden Abbildungen ist dargestellt, in welcher Form die Daten in den Datenbereichen gespeichert werden, wenn die Servicedaten in 2-Byte- oder 4-Byte-Einheiten vorliegen.

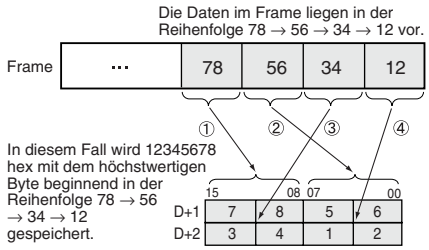
1. Daten in 2-Byte-Einheiten
   - Speichern von Daten mit dem höchstwertigen Byte beginnend (Bits 12 bis 15 von C = 0 hex)
     Beispiel: Speichern des Werts 1234 hex in D+1



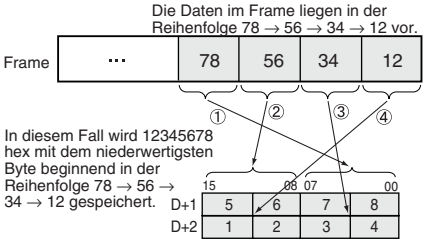
- Speichern von Daten mit dem niederwertigsten Byte beginnend (Bits 12 bis 15 von C = 8 hex)
  Beispiel: Speichern des Werts 1234 hex in D+1



2. Daten in 4-Byte-Einheiten
   - Speichern von Daten mit dem höchstwertigen Byte beginnend (Bits 12 bis 15 von C = 0 hex)
     Beispiel: Speichern des Werts 12345678 hex in D+1 und D+2

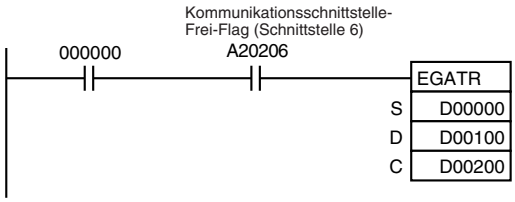


   - Speichern von Daten mit dem niederwertigsten Byte beginnend (Bits 12 bis 15 von C = 8 hex)
     Beispiel: Speichern des Werts 12345678 hex in D+1 und D+2



Beispiel

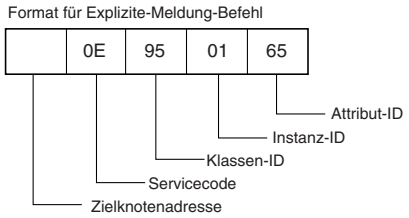
In diesem Beispiel wird EGATR(721) zum Lesen des allgemeinen Status eines DRT2-Slaves (E/A-Modul) verwendet.

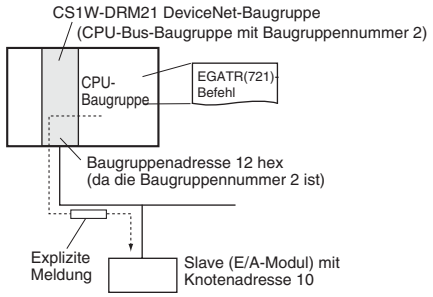


Wenn CIO 000000 und A20206 (das Kommunikationsschnittstelle-Frei-Flag für Schnittstelle 06) auf EIN gesetzt sind, liest EGATR(721) den allgemeinen Status des DRT2-Slaves (E/A-Modul). In diesem Fall wird die Gesamteinschaltzeit oder die Anzahl der Schaltspiele für Eingang 3 gelesen.

Servicecode = 0E hex, Klassen-ID = 95 hex, Instanz-ID = 01 hex und Attribut-ID = 65 hex.

Der allgemeine Status wird in 1 Byte zurückgegeben.

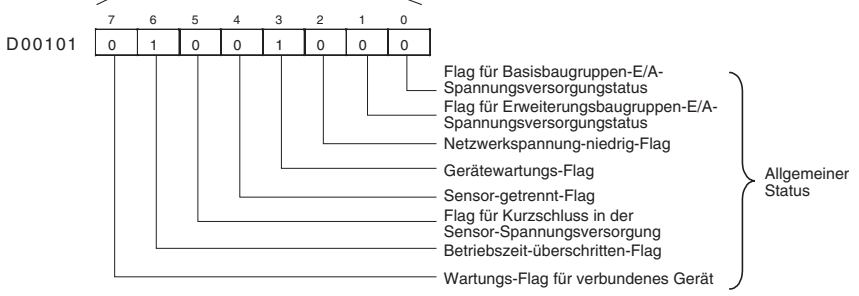




S:	D00000	0	0	0	A	Slave-Knotenadresse = 10 = 0A hex
S+1:	D00001	0	0	9	5	Klassen-ID = 95 hex
S+2:	D00002	0	0	0	1	Instanz-ID = 01 hex
S+3:	D00003	0	0	6	5	Attribut-ID = 65 hex

C:	D00200	0	0	0	2	Stellen Sie 2 Worte = 0002 hex ein, da sich 2 Worte in D bis D+1 befinden.
C+1:	D00201	8	6	1	2	Byte-Reihenfolge = 8 hex (mit dem niederwertigsten Byte beginnend), Kommunikationsschnittstelle = 6 hex (Schnittstelle 6) und die DeviceNet-Baugruppenadresse = 12 hex
C+2:	D00202	0	0	0	0	Antwort-Überwachungszeit = 0000 hex (2 s)
C+3:	D00203	0	0	0	0	Formattyp für explizite Meldung = 0000 hex (DeviceNet-Format)

D:	D00100	0	0	0	1	D enthält 0 hex für das 1 Datenbyte, das in das niederwertigste Byte von D+1 zurückgegeben wurde. Der allgemeine Status des Slaves wird in die Bits 00 bis 07 zurückgegeben. (Die Daten werden in den Bits 00 bis 07 gespeichert, da die Byte-Reihenfolge in den Bits 12 bis 15 von C+1 auf 8 hex (mit dem niederwertigsten Byte beginnend) eingestellt ist.
D+1:	D00101	0	0	4	8	



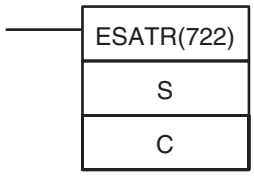
3-25-8 EXPLICIT SET ATTRIBUTE: ESATR(722)

Funktion

Sendet einen Informations-Schreibbefehl in einer expliziten Meldung (Einzelnes Attribut setzen, Servicecode: 10 hex).

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



S: Erstes Wort der zu sendenden Meldung

C: Erstes Steuerwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	ESATR(722)
	Ausführung einmal bei steigender Flanke	@ESATR(722)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

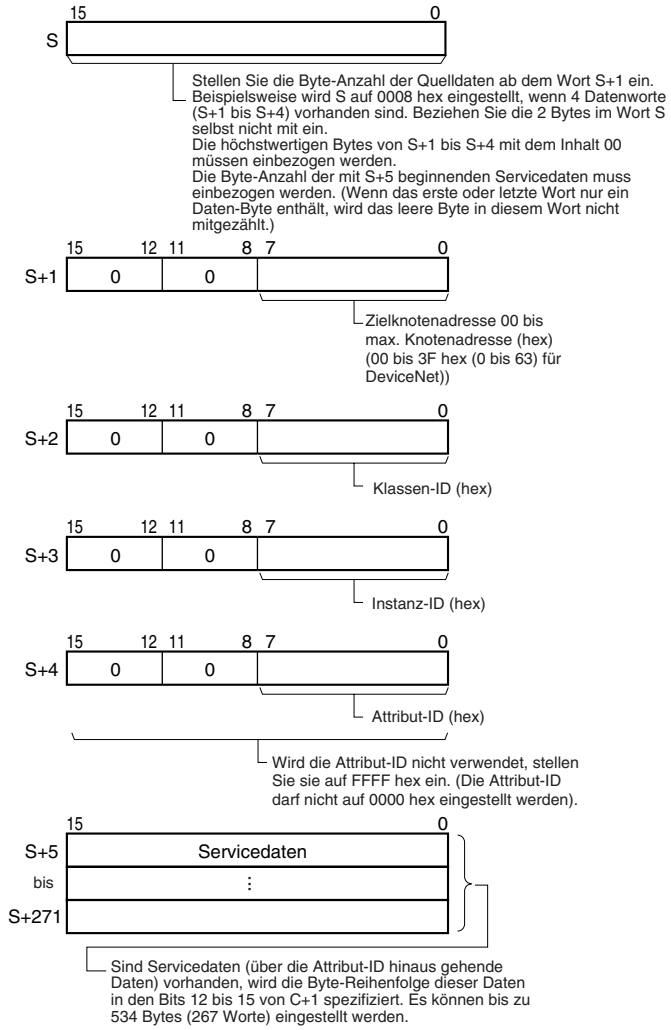
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

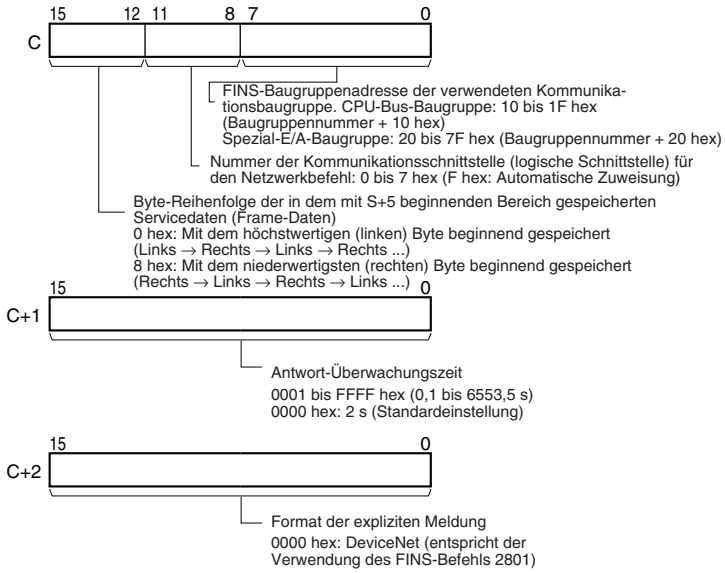
S: Erstes Wort der zu sendenden Meldung

Spezifiziert das erste Wort der zu sendenden Meldung (S bis max. S+271).



C: Erstes Steuerwort

Spezifiziert das erste von drei Steuerworten (C bis C+2).



Operanden-Spezifikationen

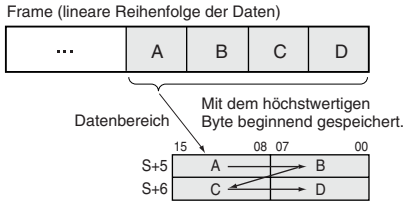
Bereich	S	C
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6141
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W509
Arbeitsbereich remanent	H000 bis H511	H000 bis H509
Systembereich	A000 bis A959	A000 bis A957
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4093
Zählerbereich	C0000 bis C4095	C0000 bis C4093
DM-Bereich	D00000 bis D32767	D00000 bis D32765
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32765
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32765 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Sendet den Explizite-Meldung-Befehl mit dem Servicecode 10 hex (im mit S+2 beginnenden Wortebereich gespeichert) über die Kommunikationsbaugruppe mit der in den Bits 00 bis 07 von C spezifizierten FINS-Baugruppenadresse an den in S+1 spezifizierten Knoten. Wenn die Antwort auf die explizite Meldung empfangen wird, wird diese im mit D+2 beginnenden Wortebereich gespeichert.

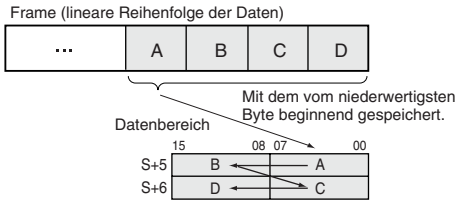
Die Einstellung in den Bits 12 bis 15 von C (0 oder 8 hex) bestimmt die Byte-Reihenfolge der in S+5 gespeicherten Servicedaten.

- Speichern von Daten mit dem höchstwertigen Byte beginnend
  Stellen Sie die Bits 12 bis 15 von C auf 0 hex ein.



Hinweis: A, B, C und D stellen Daten-Bytes dar.

- Speichern von Daten mit dem niederwertigsten Byte beginnend
  Stellen Sie die Bits 12 bis 15 von C auf 8 hex ein.



Hinweis: A, B, C und D stellen Daten-Bytes dar.

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn das Kommunikationsschnittstelle-Frei-Flag für die Kommunikationsschnittstelle mit der in C spezifizierten Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

Das entsprechende Explizite-Kommunikation-Fehlerflag wird auf AUS gesetzt, wenn die Ausführung des Befehls normal abgeschlossen wird, oder auf EIN gesetzt, wenn ein Fehler aufgetreten ist.

Bei Auftreten eines Fehlers (entsprechendes Flag in A213 auf EIN gesetzt) kann das entsprechende Kommunikationsschnittstellen-Fehlerflag zur Ermittlung verwendet werden, ob die explizite Meldung selbst nicht gesendet wurde (entsprechendes Flag in A219 auf EIN gesetzt) oder ob die Meldung gesendet wurde, aber einen Fehler enthielt (entsprechendes Flag in A219 auf AUS gesetzt).

Der entsprechende Kommunikationsschnittstellen-Abschlusscode (A203 bis A210) enthält 0000 hex, wenn die Ausführung des Befehls normal abgeschlossen wurde, einen Explizite-Meldung-Fehlercode, wenn ein Fehler bei der expliziten Meldung aufgetreten ist, oder einen FINS-Fehlercode, wenn ein FINS-Fehler aufgetreten ist.

Weitere Informationen über die allgemeinen Funktionen der Befehle für explizite Meldungen sind unter *3-25-2 Informationen zu Explizite-Meldung-Befehlen* zu finden.

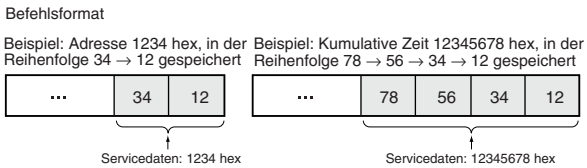
In der folgenden Tabelle sind die relevanten Bits und Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Kommunikationsschnittstelle-Frei-Flag	A20200 bis A20207	Diese Flags werden auf EIN gesetzt, um anzugeben, dass Netzwerkbefehle (einschließlich PMCR(260)) für die entsprechenden Schnittstellen (00 bis 07) ausgeführt werden können. Ein Flag wird auf AUS gesetzt, wenn ein Netzwerkbefehl für die entsprechende Schnittstelle ausgeführt wird, und wieder auf EIN gesetzt, wenn die Ausführung des Befehls abgeschlossen ist.
Explizite-Kommunikation-Fehlerflag	A21300 bis A21307	Diese Flags werden auf EIN gesetzt, um anzugeben, dass während der Ausführung der Kommunikation mittels expliziter Meldungen an den entsprechenden Schnittstellen (00 bis 07) ein Fehler aufgetreten ist. Die Flags werden auf EIN gesetzt, wenn die explizite Meldung nicht gesendet wurde oder die Meldung gesendet wurde, aber eine Fehlermeldung zurückgegeben wurde. Der Flag-Status wird bis zur Ausführung des nächsten Explizite-Meldung-Befehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.

Name	Adresse	Funktion
Kommunikationsschnittstellen-Fehlerflag	A21900 bis A21907	Diese Flags werden auf EIN gesetzt, um anzugeben, dass die explizite Meldung während der Ausführung eines Explizite-Meldung-Befehl nicht über die entsprechende Schnittstelle (00 bis 07) gesendet wurde. Der Flag-Status wird bis zur Ausführung des nächsten Netzwerkbefehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunikationsschnittstellen-Abschlusscodes	A203 bis A210	Diese Worte enthalten nach der Ausführung eines Netzwerkbefehls die Abschlusscodes für die entsprechenden Schnittstellen (00 bis 07). Das entsprechende Wort hat den Inhalt 0000, solange das Explizite-Kommunikation-Fehlerflag auf AUS gesetzt ist. Das entsprechende Wort enthält einen FINS-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle beide auf EIN gesetzt sind. Das entsprechende Wort enthält den entsprechenden Explizite-Meldung-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag dieser Schnittstelle auf EIN gesetzt ist und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle auf AUS gesetzt ist. Das entsprechende Wort hat während Ausführung des Netzwerkbefehls den Inhalt 0000. Nach Abschluss der Befehlsausführung wird der Abschlusscode in dieses Wort geschrieben. Diese Worte werden zu Beginn der Programmausführung gelöscht.

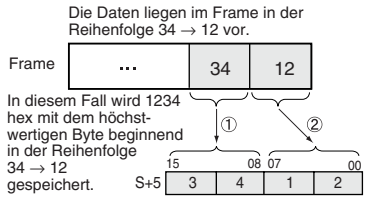
Vorsichtsmaßnahmen

Achten Sie darauf, dass die Reihenfolge der Bytes in den Quelldaten der Reihenfolge im Frame der expliziten Meldung (lineare Reihenfolge der Daten) entspricht. Wenn z.B. die Servicedaten in 2-Byte- oder 4-Byte-Einheiten vorliegen, liegen die Daten im Frame in Zweierpaaren, jeweils in der Reihenfolge vom höchstwertigen Byte zum niederwertigsten Byte, vor (siehe nachfolgende Abbildung).

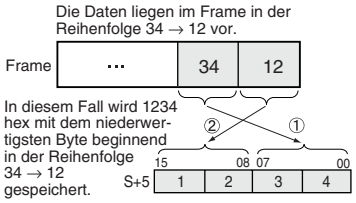


In den folgenden Abbildungen ist dargestellt, in welcher Form die Daten in den Datenbereichen gespeichert werden, wenn die Servicedaten in 2-Byte- oder 4-Byte-Einheiten vorliegen.

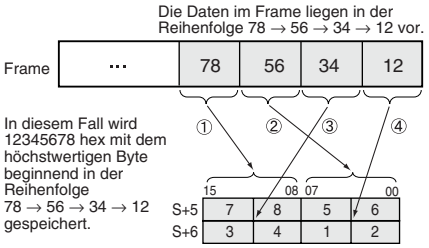
1. Daten in 2-Byte-Einheiten
   - Speichern von Daten mit dem höchstwertigen Byte beginnend (Bits 12 bis 15 von C = 0 hex)
     Beispiel: Speichern des Werts 1234 hex in S+5



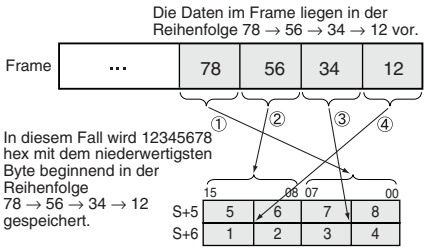
- Speichern von Daten mit dem niederwertigsten Byte beginnend (Bits 12 bis 15 von C = 8 hex)
  Beispiel: Speichern des Werts 1234 hex in S+5



2. Daten in 4-Byte-Einheiten
   - Speichern von Daten mit dem höchstwertigen Byte beginnend (Bits 12 bis 15 von C = 0 hex)
     Beispiel: Speichern des Werts 12345678 hex in S+5 und S+6

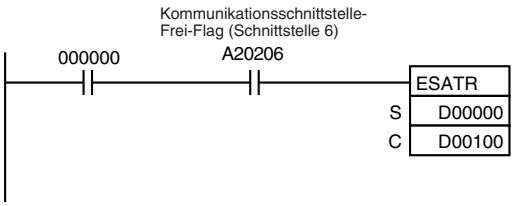


   - Speichern von Daten mit dem niederwertigsten Byte beginnend (Bits 12 bis 15 von C = 8 hex)
     Beispiel: Speichern des Werts 12345678 hex in S+5 und S+6



Beispiel

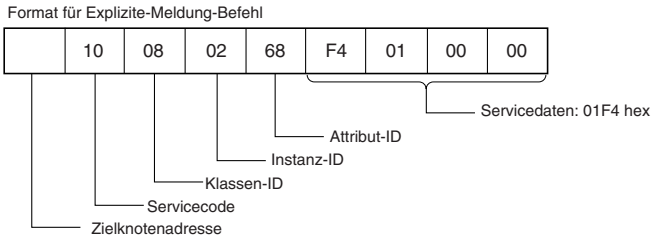
In diesem Beispiel wird der Befehl ESATR(722) zum Überschreiben des Sollwerts für die Anzahl der Schaltspiele in einem DRT2-Slave (E/A-Modul) verwendet.

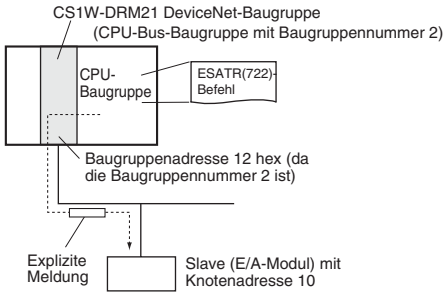


Wenn CIO 000000 und A20206 (das Kommunikationsschnittstelle-Frei-Flag für Schnittstelle 06) auf EIN gesetzt sind, schreibt EXPLT(720) den Sollwert für die Anzahl der Schaltspiele von Eingang 2 in einen DRT2-Slave (E/A-Modul).

(Servicecode = 10 hex, Klassen-ID = 08 hex, Instanz-ID = 02 hex und Attribut-ID = 68 hex.

In diesem Fall wird der Sollwert für die Anzahl der Schaltspiele auf 500 (1F4 hex) eingestellt, also müssen die Servicedaten auf 000001F4 eingestellt werden.





S	D00000	0	0	0	C	Anzahl der Datenbytes: S+1 bis S+6 = 6 Worte = 12 Bytes = 0C hex
S:+1	D00001	0	0	0	A	Slave-Knotenadresse = 10 = 0A hex
S+2:	D00002	0	0	0	8	Klassen-ID = 08 hex
S+3:	D00003	0	0	0	2	Instanz-ID = 02 hex
S+4:	D00004	0	0	6	8	Attribut-ID = 68 hex
S+5:	D00005	0	1	F	4	Servicedaten = F401 hex
S+6:	D00006	0	0	0	0	

C:	D00201	8	6	1	2	Byte-Reihenfolge = 8 hex (mit dem niederwertigsten Byte beginnend), Kommunikationsschnittstelle = 6 hex (Schnittstelle 6) und die DeviceNet-Baugruppenadresse = 12 hex
C+1:	D00202	0	0	0	0	Antwort-Überwachungszeit = 0000 hex (2 s)
C+2:	D00203	0	0	0	0	Formattyp für explizite Meldung = 0000 hex (DeviceNet-Format)

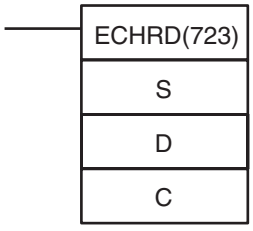
3-25-9 EXPLICIT WORD READ: ECHRD(723)

Funktion

Liest Daten von einer anderen CPU-Baugruppe im Netzwerk speichert sie in der lokalen CPU-Baugruppe. (Die dezentrale CPU-Baugruppe muss explizite Meldungen unterstützten).

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



S: Erstes Quellwort in dezentraler CPU-Baugruppe

D: Erstes Zielwort in lokaler CPU-Baugruppe

C: Erstes Steuerwort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	ECHRD(723)
	Ausführung einmal bei steigender Flanke	@ECHRD(723)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

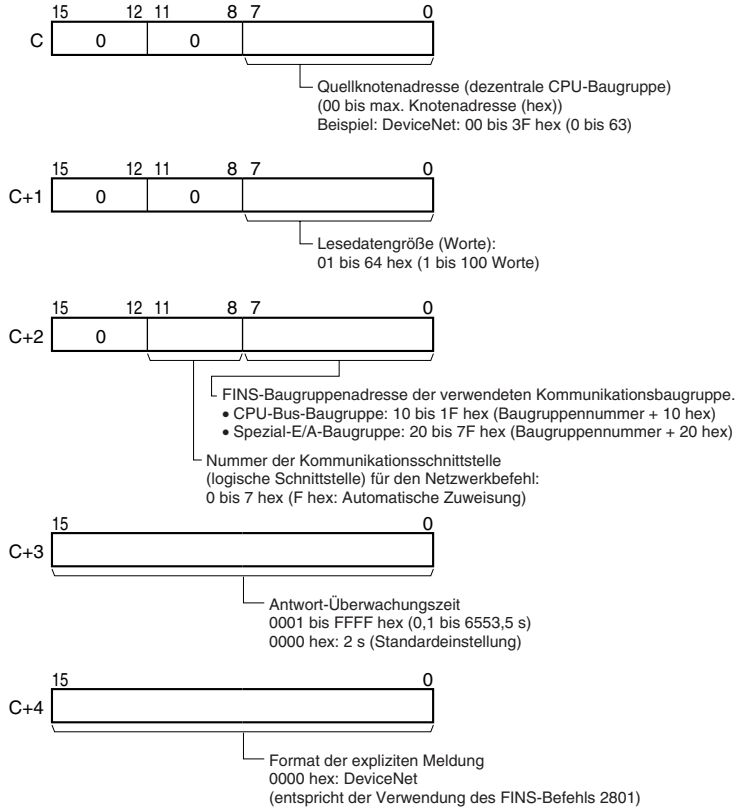
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Erstes Quellwort in dezentraler CPU-Baugruppe
Spezifiziert die Adresse des ersten der Worte, die die Daten enthalten, die von der dezentralen CPU-Baugruppe gelesen werden sollen.

D: Erstes Zielwort in lokaler CPU-Baugruppe
Spezifiziert die Adresse des ersten der Worte, in denen die gelesenen Daten in der lokalen CPU-Baugruppe gespeichert werden sollen.

C: Erstes Steuerwort
Spezifiziert das erste von fünf Steuerworten (C bis C+4).



Operanden-Spezifikationen

Bereich	S	D	C
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6139
Arbeitsbereich nicht remanent	W000 bis W511		W000 to W507
Arbeitsbereich remanent	H000 bis H511		H000 to H507
Systembereich	A000 bis A959	A448 bis A959	A000 bis A955
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4091
Zählerbereich	C0000 bis C4095		C0000 bis C4091
DM-Bereich	D00000 bis D32767		D00000 bis D32763
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32763
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32763 (n = 0 bis C)

Bereich	S	D	C
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Liest die spezifizierte Anzahl von Worten mit dem ersten Lesewort (spezifiziert in S) beginnend in der dezentralen CPU-Baugruppe mit der in C spezifizierten Knotenadresse, und speichert die Daten in den mit D beginnenden Speicherworten der lokalen CPU-Baugruppe.

Hinweis

Der Befehl ECHRD(723) sendet eine explizite Meldung mit dem Servicecode 1C hex (Bytedaten lesen).

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das Kommunikationsschnittstelle-Frei-Flag für die Kommunikationsschnittstelle mit der in C spezifizierten Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

Das entsprechende Explizite-Kommunikation-Fehlerflag wird auf AUS gesetzt, wenn die Ausführung des Befehls normal abgeschlossen wird, oder auf EIN gesetzt, wenn ein Fehler aufgetreten ist.

Bei Auftreten eines Fehlers (entsprechendes Flag in A213 auf EIN gesetzt) kann das entsprechende Kommunikationsschnittstellen-Fehlerflag zur Ermittlung verwendet werden, ob die explizite Meldung selbst nicht gesendet wurde (entsprechendes Flag in A219 auf EIN gesetzt) oder ob die Meldung gesendet wurde, aber einen Fehler enthielt (entsprechendes Flag in A219 auf AUS gesetzt).

Der entsprechende Kommunikationsschnittstellen-Abschlusscode (A203 bis A210) enthält 0000 hex, wenn die Ausführung des Befehls normal abgeschlossen wurde, einen Explizite-Meldung-Fehlercode, wenn ein Fehler bei der expliziten Meldung aufgetreten ist, oder einen FINS-Fehlercode, wenn ein FINS-Fehler aufgetreten ist.

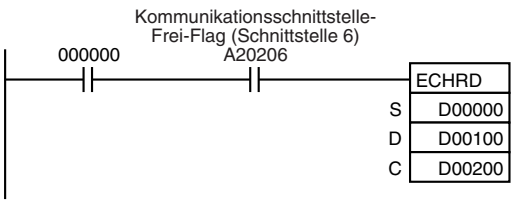
Weitere Informationen über die allgemeinen Funktionen von Netzwerkbefehlen sind unter *3-25-2 Informationen zu Explizite-Meldung-Befehlen* zu finden.

In der folgenden Tabelle sind die relevanten Bits und Flags im Systembereich aufgeführt.

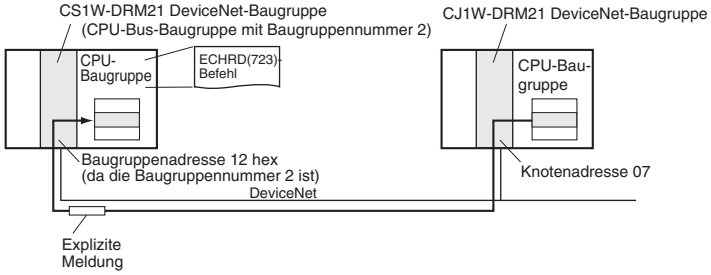
Name	Adresse	Funktion
Kommunikationsschnittstelle-Frei-Flag	A20200 bis A20207	Diese Flags werden auf EIN gesetzt, um anzugeben, dass Netzwerkbefehle (einschließlich PMCR(260)) für die entsprechenden Schnittstellen (00 bis 07) ausgeführt werden können. Ein Flag wird auf AUS gesetzt, wenn ein Netzwerkbefehl für die entsprechende Schnittstelle ausgeführt wird, und wieder auf EIN gesetzt, wenn die Ausführung des Befehls abgeschlossen ist.
Explizite-Kommunikation-Fehlerflag	A21300 bis A21307	Diese Flags werden auf EIN gesetzt, um anzugeben, dass während der Ausführung der Kommunikation mittels expliziter Meldungen an den entsprechenden Schnittstellen (00 bis 07) ein Fehler aufgetreten ist. Die Flags werden auf EIN gesetzt, wenn die explizite Meldung nicht gesendet wurde oder die Meldung gesendet wurde, aber eine Fehlermeldung zurückgegeben wurde. Der Flag-Status wird bis zur Ausführung des nächsten Explizite-Meldung-Befehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunikationsschnittstellen-Fehlerflag	A21900 bis A21907	Diese Flags werden auf EIN gesetzt, um anzugeben, dass die explizite Meldung während der Ausführung eines Explizite-Meldung-Befehl nicht über die entsprechende Schnittstelle (00 bis 07) gesendet wurde. Der Flag-Status wird bis zur Ausführung des nächsten Netzwerkbefehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunikationsschnittstellen-Abschlusscodes	A203 bis A210	Diese Worte enthalten nach der Ausführung eines Netzwerkbefehls die Abschlusscodes für die entsprechenden Schnittstellen (00 bis 07). Das entsprechende Wort hat den Inhalt 0000, solange das Explizite-Kommunikation-Fehlerflag auf AUS gesetzt ist. Das entsprechende Wort enthält einen FINS-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle beide auf EIN gesetzt sind. Das entsprechende Wort enthält den entsprechenden Explizite-Meldung-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag dieser Schnittstelle auf EIN gesetzt ist und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle auf AUS gesetzt ist. Das entsprechende Wort hat während Ausführung des Netzwerkbefehls den Inhalt 0000. Nach Abschluss der Befehlsausführung wird der Abschlusscode in dieses Wort geschrieben. Diese Worte werden zu Beginn der Programmausführung gelöscht.

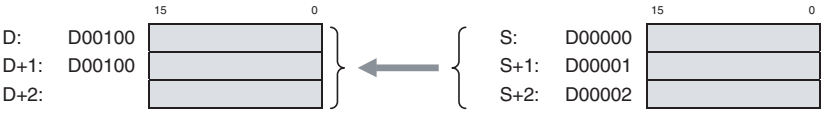
Beispiel

In diesem Beispiel wird ECHRD(723) zum Lesen des E/A-Speichers der CJ-Serie CPU-Baugruppe im DeviceNet-Netzwerk sowie zum Speichern der Daten im E/A-Speicher der lokalen CPU-Baugruppe verwendet.



Wenn CIO 000000 und A20206 (das Kommunikationsschnittstelle-Frei-Flag für Schnittstelle 06) auf EIN gesetzt sind, liest ECHRD(723) die Worte D00000 bis D00002 aus dem E/A-Speicher der CJ-Serie CPU-Baugruppe mit der Knotenadresse 07 im DeviceNet-Netzwerk und speichert die Daten in den Worten D00100 bis D00102 der lokalen CPU-Baugruppe.





		15–8	7–0	
C:	D00200	00	07	Knotenadresse der dezentralen CPU-Baugruppe, von der die Daten gelesen werden sollen = 07 hex (Knoten 07)
C+1:	D00201	00	03	Lesedatengröße (Anzahl Worte) = 3 hex
C+2:	D00202	06	12	Kommunikationsschnittstelle = 6 hex (Schnittstelle 6) und die Baugruppenadresse der DeviceNet-Baugruppe = 12 hex
C+3:	D00203	00	00	Antwort-Überwachungszeit = 0000 hex (2 s)
C+4:	D00204	00	00	Formattyp für explizite Meldung = 0000 hex (DeviceNet-Format)

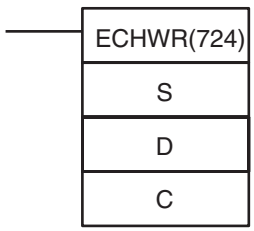
3-25-10 EXPLICIT WORD WRITE: ECHWR(724)

Funktion

Schreibt Daten von der lokalen CPU-Baugruppe in eine andere CPU-Baugruppe im Netzwerk. (Die dezentrale CPU-Baugruppe muss explizite Meldungen unterstützen.)

Dieser Befehl wird nur von CPU-Baugruppen der CS/CJ-Serie ab Version 2.0 unterstützt.

Kontaktplansymbol



S: Erstes Quellwort in lokaler CPU-Baugruppe

D: Erstes Zielwort in dezentraler CPU-Baugruppe

C: Erstes Steuerwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	ECHWR(724)
	Ausführung einmal bei steigender Flanke	@ECHWR(724)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

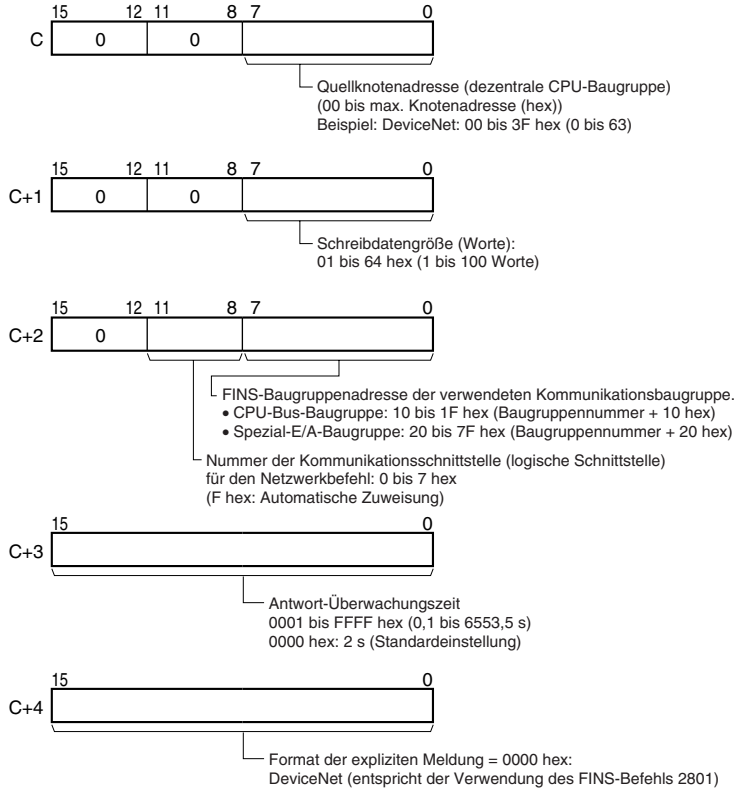
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Erstes Quellwort in lokaler CPU-Baugruppe
Spezifiziert die Adresse des ersten der Worte in der lokalen CPU-Baugruppe, die die zu schreibenden Daten enthalten.

D: Erstes Zielwort in dezentraler CPU-Baugruppe
Spezifiziert die Adresse des ersten Schreibziel-Worts in der dezentralen CPU-Baugruppe.

C: Erstes Steuerwort
Spezifiziert das erste von fünf Steuerworten (C bis C+4).



Operanden-Spezifikationen

Bereich	S	D	C
CIO-Bereich	CIO 0000 bis CIO 6143		CIO 0000 bis CIO 6139
Arbeitsbereich nicht remanent	W000 bis W511		W000 to W507
Arbeitsbereich remanent	H000 bis H511		H000 to H507
Systembereich	A000 bis A959	A448 bis A959	A000 bis A955
Zeitgeberbereich	T0000 bis T4095		T0000 bis T4091
Zählerbereich	C0000 bis C4095		C0000 bis C4091
DM-Bereich	D00000 bis D32767		D00000 bis D32763
EM-Bereich ohne Bank	E00000 bis E32767		E00000 bis E32763
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		En_00000 bis En_32763 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		

Bereich	S	D	C
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Schreibt die spezifizierte Anzahl von Worten bei S beginnend von der lokalen CPU-Baugruppe in das bei D beginnende Schreibziel der dezentralen CPU-Baugruppe mit der in C spezifizierten Knotenadresse.

Hinweis ECHWR(724) sendet eine explizite Meldung mit dem Servicecode 1E hex (Bytedaten schreiben).

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das Kommunikationsschnittstelle-Frei-Flag für die Kommunikationsschnittstelle mit der in C spezifizierten Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

Das entsprechende Explizite-Kommunikation-Fehlerflag wird auf AUS gesetzt, wenn die Ausführung des Befehls normal abgeschlossen wird, oder auf EIN gesetzt, wenn ein Fehler aufgetreten ist.

Bei Auftreten eines Fehlers (entsprechendes Flag in A213 auf EIN gesetzt) kann das entsprechende Kommunikationsschnittstellen-Fehlerflag zur Ermittlung verwendet werden, ob die explizite Meldung selbst nicht gesendet wurde (entsprechendes Flag in A219 auf EIN gesetzt) oder ob die Meldung gesendet wurde, aber einen Fehler enthielt (entsprechendes Flag in A219 auf AUS gesetzt).

Der entsprechende Kommunikationsschnittstellen-Abschlusscode (A203 bis A210) enthält 0000 hex, wenn die Ausführung des Befehls normal abgeschlossen wurde, einen Explizite-Meldung-Fehlercode, wenn ein Fehler bei der expliziten Meldung aufgetreten ist, oder einen FINS-Fehlercode, wenn ein FINS-Fehler aufgetreten ist.

Weitere Informationen über die allgemeinen Funktionen der Befehle für explizite Meldungen sind unter *3-25-2 Informationen zu Explizite-Meldung-Befehlen* zu finden.

In der folgenden Tabelle sind die relevanten Bits und Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Kommunikationsschnittstelle-Frei-Flag	A20200 bis A20207	Diese Flags werden auf EIN gesetzt, um anzugeben, dass Netzwerkbefehle (einschließlich PMCR(260)) für die entsprechenden Schnittstellen (00 bis 07) ausgeführt werden können. Ein Flag wird auf AUS gesetzt, wenn ein Netzwerkbefehl für die entsprechende Schnittstelle ausgeführt wird, und wieder auf EIN gesetzt, wenn die Ausführung des Befehls abgeschlossen ist.
Explizite-Kommunikation-Fehlerflag	A21300 bis A21307	Diese Flags werden auf EIN gesetzt, um anzugeben, dass während der Ausführung der Kommunikation mittels expliziter Meldungen an den entsprechenden Schnittstellen (00 bis 07) ein Fehler aufgetreten ist. Die Flags werden auf EIN gesetzt, wenn die explizite Meldung nicht gesendet wurde oder die Meldung gesendet wurde, aber eine Fehlermeldung zurückgegeben wurde. Der Flag-Status wird bis zur Ausführung des nächsten Explizite-Meldung-Befehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.

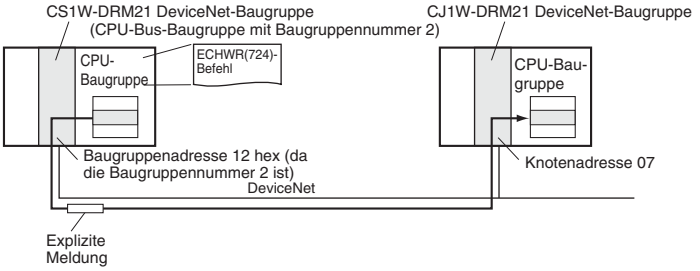
Name	Adresse	Funktion
Kommunikationsschnittstellen-Fehlerflag	A21900 bis A21907	Diese Flags werden auf EIN gesetzt, um anzugeben, dass die explizite Meldung während der Ausführung eines Explizite-Meldung-Befehl nicht über die entsprechende Schnittstelle (00 bis 07) gesendet wurde. Der Flag-Status wird bis zur Ausführung des nächsten Netzwerkbefehls beibehalten. Das Flag wird bei Ausführung des nächsten Befehls wieder auf AUS gesetzt, auch wenn vorher ein Fehler aufgetreten ist.
Kommunikationsschnittstellen-Abschlusscodes	A203 bis A210	Diese Worte enthalten nach der Ausführung eines Netzwerkbefehls die Abschlusscodes für die entsprechenden Schnittstellen (00 bis 07). Das entsprechende Wort hat den Inhalt 0000, solange das Explizite-Kommunikation-Fehlerflag auf AUS gesetzt ist. Das entsprechende Wort enthält einen FINS-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle beide auf EIN gesetzt sind. Das entsprechende Wort enthält den entsprechenden Explizite-Meldung-Fehlercode, wenn das Explizite-Kommunikation-Fehlerflag dieser Schnittstelle auf EIN gesetzt ist und das Kommunikationsschnittstellen-Fehlerflag dieser Schnittstelle auf AUS gesetzt ist. Das entsprechende Wort hat während Ausführung des Netzwerkbefehls den Inhalt 0000. Nach Abschluss der Befehlsausführung wird der Abschlusscode in dieses Wort geschrieben. Diese Worte werden zu Beginn der Programmausführung gelöscht.

Beispiel

In diesem Beispiel wird der Befehl ECHWR(724) zum Schreiben von Daten aus dem E/A-Speicher der lokalen CPU-Baugruppe in den E/A-Speicher einer CJ-Serie CPU-Baugruppe im DeviceNet-Netzwerk verwendet.



Wenn CIO 000000 und A20206 (Kommunikationsschnittstelle-Frei-Flag für Schnittstelle 06) auf EIN gesetzt sind, liest ECHWR(724) die Worte D00000 bis D00002 aus dem E/A-Speicher der lokalen CPU-Baugruppe und speichert die Daten in den Worten D00100 bis D00102 der CJ-Serie CPU-Baugruppe mit der Knotenadresse 07 im DeviceNet-Netzwerk.





		15 8	7 0	
C:	D00200	00	07	Knotenadresse der dezentralen CPU-Baugruppe, in die geschrieben werden soll = 07 hex (Knoten 07)
C+1:	D00201	00	03	Schreibdatengröße (Anzahl Worte) = 3 hex
C+2:	D00202	06	12	Kommunikationsschnittstelle = 6 hex (Schnittstelle 6) und die Baugruppenadresse der DeviceNet-Baugruppe = 12 hex
C+3:	D00203	00	00	Antwort-Überwachungszeit = 0000 hex (2 s)
C+4:	D00204	00	00	Formattyp für explizite Meldung = 0000 hex (DeviceNet-Format)

3-26 Dateispeicherbefehle

In diesem Abschnitt werden Befehle beschrieben, die für den Dateispeicher (EM-Bereich oder Speicherkarten) verwendet werden.

Hinweis Der Dateispeicherfunktionen können auch durch Ausführung von CMND(490) zum Senden eines FINS-Befehls an die lokale CPU-Baugruppe durchgeführt werden. Detaillierte Informationen hierzu finden Sie im *CS/CJ-Serie SPS Bedienerhandbuch*.

Befehl	AWL	Funktionscode	Seite
READ DATA FILE	FREAD	700	1045
WRITE DATA FILE	FWRIT	701	1052

3-26-1 Vorsichtsmaßnahmen bei der Verwendung von Speicherkarten

Prüfen Sie vor der Verwendung einer Speicherkarte Folgendes:

Formatierung

Speicherkarten werden vor der Auslieferung formatiert. Nach dem Erwerb muss keine Formatierung erfolgen. Wenn Sie Speicherkarten formatieren möchten, nachdem diese bereits verwendet wurden, führen Sie den Vorgang immer in der CPU-Baugruppe mit Hilfe von CX-Programmer oder einer Programmierkonsole durch.

Wenn eine Speicherkarte direkt in einem Notebook-Computer oder einem anderen Computer formatiert wird, kann sie von der CPU-Baugruppe möglicherweise nicht erkannt werden. Sie können die Speicherkarte in diesem Fall nicht mehr verwenden, selbst dann nicht, wenn Sie in der CPU-Baugruppe neu formatiert wird.

Anzahl der Dateien im Stammverzeichnis

Die Anzahl der Dateien, die im Stammverzeichnis einer Speicherkarte gespeichert werden können, ist begrenzt (wie es auch bei Festplatten eine Begrenzung gilt). Die Begrenzung ist zwar vom Typ und Format der Speicherkarte abhängig, jedoch liegt die Anzahl der Dateien zwischen 128 und 512. Bei Anwendungen, die in bestimmten Abständen Protokolldateien oder andere Dateien schreiben, sollten die Dateien in ein Unterverzeichnis und nicht in das Stammverzeichnis geschrieben werden.

Unterverzeichnisse können auf einem Computer oder mit Hilfe des Befehls CMND(490) erstellt werden. Ein spezifisches Beispiel unter Verwendung von CMND(490) finden Sie unter *3-25-5 DELIVER COMMAND: CMND(490)*.

Anzahl der Schreibvorgänge

Es gibt im Allgemeinen keine Begrenzung hinsichtlich der Anzahl der Schreibvorgänge, die bei einem Flash-Speicher ausgeführt werden können. Für Speicherkarten wurde aus Gewährleistungsgründen jedoch eine Begrenzung auf 100.000 Schreibvorgänge festgelegt. Wenn beispielsweise alle 10 Minuten auf eine Speicherkarte geschrieben wird, werden innerhalb von 2 Jahren mehr als 100.000 Schreibvorgänge ausgeführt.

Mindestdateigröße

Wenn auf einer Speicherkarte viele kleine Dateien, wie die nur wenige Worte beinhaltenden Dateien der Daten im DM-Bereich, gespeichert werden, ist es nicht möglich, die gesamte Kapazität der Speicherkarte zu nutzen. Wenn beispielsweise eine Speicherkarte mit einer Zuordnungseinheitsgröße von 4.096 Byte verwendet wird, werden unabhängig von der Größe der Datei mindestens 4.096 Byte Speicherplatz je Datei verwendet. Wenn Sie 10 Worte Daten des DM-Bereichs auf der Speicherkarte speichern, werden 4.096 Byte Speicherplatz in Anspruch genommen, obwohl die tatsächliche Dateigröße lediglich 68 Byte beträgt. Die Verwendung kleiner Dateigrößen reduziert die Ausnutzung der Speicherkarte erheblich. Wenn die Größe der Zuordnungseinheit jedoch reduziert wird, um die Ausnutzung zu erhöhen, verringert sich dadurch die Zugriffsgeschwindigkeit.

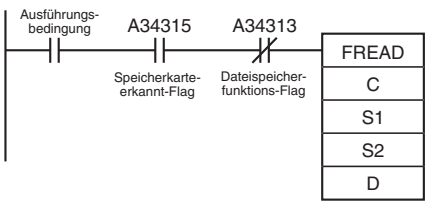
Die Größe der Zuordnungseinheit auf der Speicherkarte kann in der DOS-Eingabeaufforderung mittels des Befehls CHKDSK überprüft werden. Die genaue Vorgehensweise wird hier nicht ausgeführt. Weitere Informationen zur Größe von Zuordnungseinheiten finden Sie in den allgemeinen Computerunterlagen.

Vorsichtsmaßnahmen zum Speicherkartenzugriff

Beim Zugriff der SPS auf die Speicherkarte leuchtet die BUSY-Anzeige auf der CPU-Baugruppe. Beachten Sie die folgenden Vorsichtsmaßnahmen.

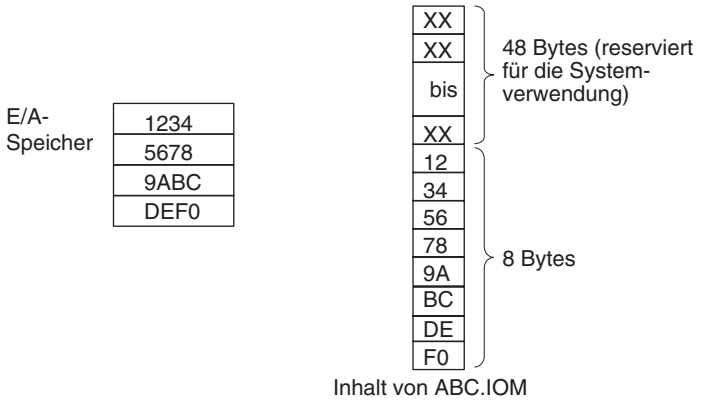
1,2,3...

1. Schalten Sie die Spannungsversorgung der CPU-Baugruppe niemals AUS, während die BUSY-Anzeige leuchtet. Die Speicherkarte kann dadurch unbrauchbar werden.
2. Entnehmen Sie die Speicherkarte niemals aus der CPU-Baugruppe, während die BUSY-Anzeige leuchtet. Drücken Sie zunächst die Speicherkartentaste der CPU-Baugruppe, und warten Sie, bis die Anzeige BUSY erloschen ist, bevor Sie die Speicherkarte entnehmen. Die Speicherkarte kann unbrauchbar werden, wenn Sie nicht in dieser Weise vorgehen.
3. Setzen Sie die Speicherkarte so in den Speicherkartensteckplatz ein, dass die beschriftete Seite der Speicherkarte nach rechts weist. Versuchen Sie nicht, die Karte in einer anderen Ausrichtung einzuführen. Die Speicherkarte und die CPU-Baugruppe könnten beschädigt werden.
4. Nach dem Einsetzen der Speicherkarte dauert es einige Sekunden, bis sie von der CPU-Baugruppe erkannt wird. Wenn Sie auf eine Speicherkarte sofort nach dem Einschalten der Spannungsversorgung oder nach dem Einsetzen der Karte zugreifen, müssen Sie eine Schließer-Bedingung für das Speicherkarte-erkannt-Flag (A34315) als Eingangsbedingung programmieren, wie im Folgenden dargestellt.

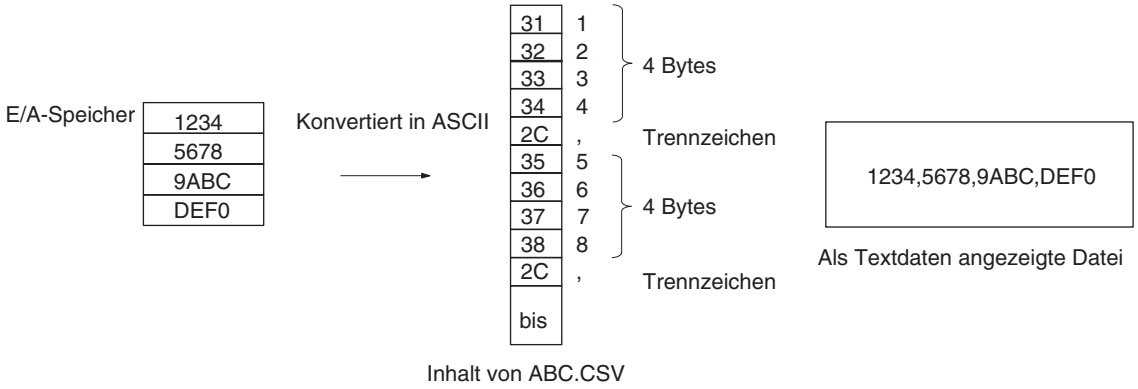


Hinweis Nachstehend ist die Struktur der Datendateien dargestellt.

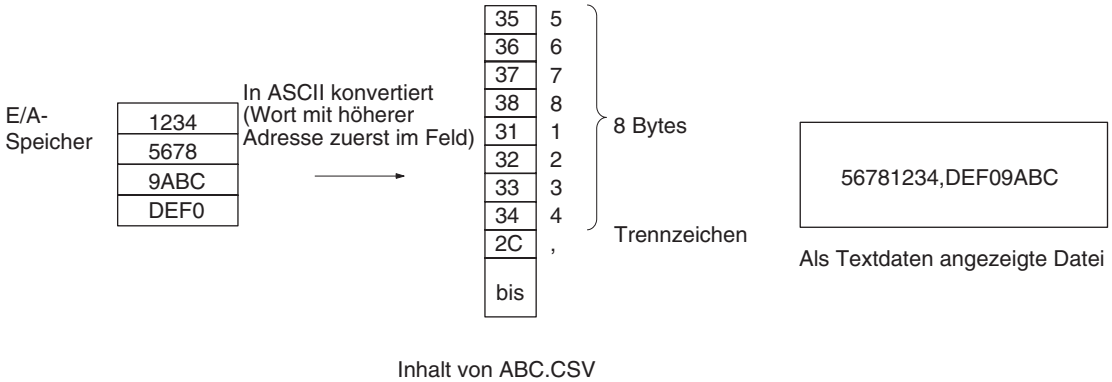
Im Binärformat (.IOM) sehen die Daten wie folgt aus, wenn 1234 hex, 5678 hex, 9ABC hex und DEF0 hex in der Datei ABC.IOM gespeichert werden (obwohl sich der Anwender gewöhnlich um diese Struktur keine Gedanken zu machen braucht):



Im Wort-CSV-Format (.CSV) sehen die Daten wie folgt aus, wenn 1234 hex, 5678 hex, 9ABC hex und DEF0 hex in der Datei ABC.CSV gespeichert werden (die Basisstruktur wäre die gleiche wie für Textdaten (.TXT):

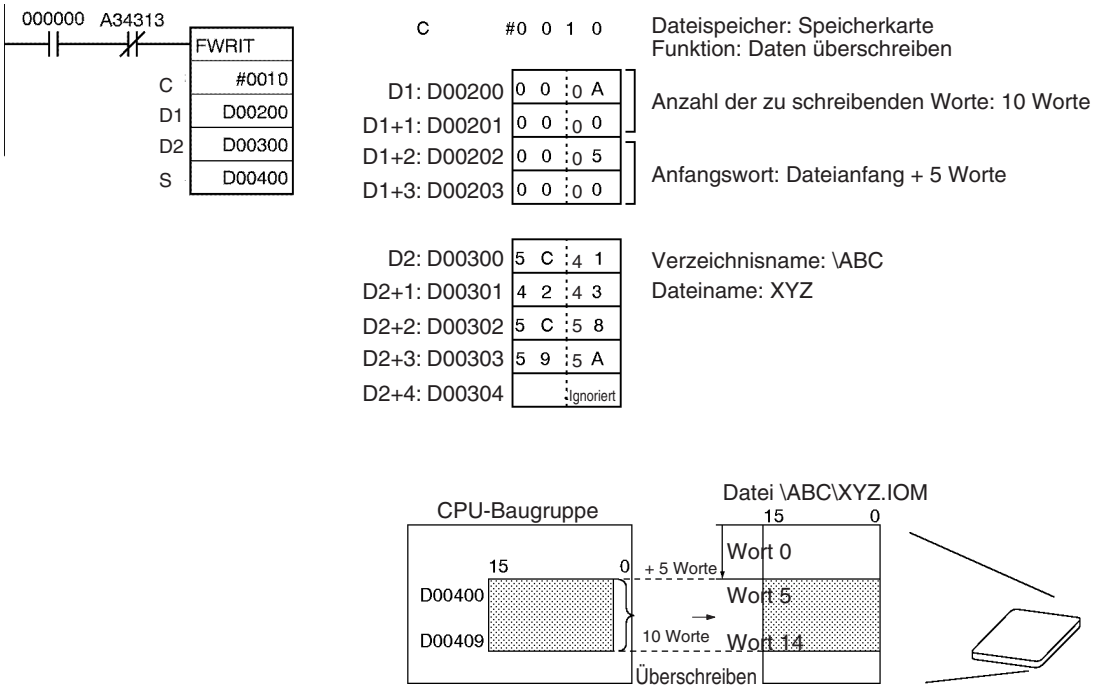


Im Langwort-CSV-Format (.CSV) sehen die Daten wie folgt aus, wenn 1234 hex, 5678 hex, 9ABC hex und DEF0 hex in der Datei ABC.CSV gespeichert werden (die Basisstruktur wäre die gleiche wie für Textdaten (.TXT):



Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, liest FWRIT(701) 10 Datenworte aus D00400 bis D00409 und verwendet diese Daten zum Überschreiben von 10 Worten in der Datei \ABC\XYZ.IOM, beginnend mit dem Anfang der Datei + 5 Worten.

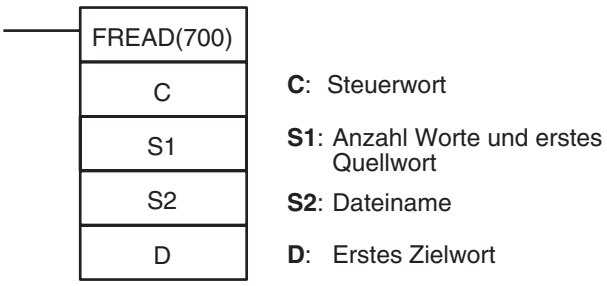


3-26-2 READ DATA FILE: FREAD(700)

Funktion

Liest die spezifizierten Daten bzw. die spezifizierte Datenmenge aus der spezifizierten Datendatei im Dateispeicher in den spezifizierten Datenbereich der CPU-Baugruppe ein.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FREAD(700)
	Ausführung einmal bei steigender Flanke	@FREAD(700)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

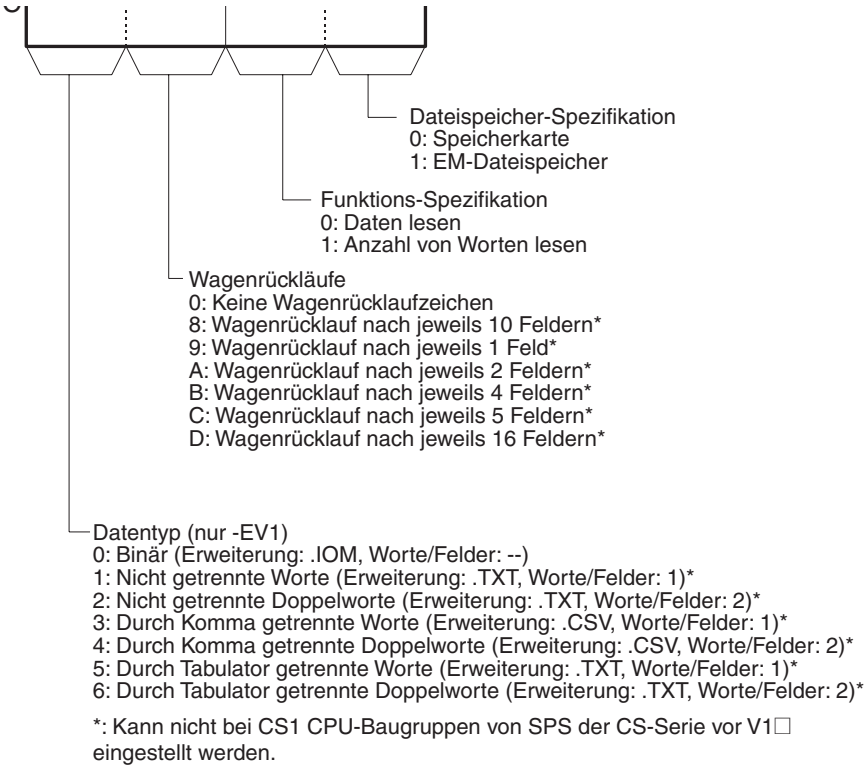
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

Wie in der folgenden Abbildung veranschaulicht, spezifiziert die erste Stelle, ob sich die Quelldatei auf der Speicherkarte oder im EM-Dateispeicher befindet. Die zweite Stelle des Steuerworts spezifiziert, ob die tatsächlichen Daten oder eine Anzahl von Datenworten gelesen werden sollen, die dritte Stelle spezifiziert das Vorhandensein von Wagenrückläufen und die vierte Stelle den Datentyp.



Hinweis

1. Jedes Feld enthält bei Wortdatentypen 1 Wort des E/A-Speichers und bei Doppelwort-Datentypen 2 Worte des E/A-Speichers.
2. Beim Lesen von Daten mit Wagenrücklaufzeichen müssen die Bits 00 bis 11 von C auf einen Wert zwischen 8 und D hex eingestellt werden.
3. Bei Doppelworten wird das erste Datenwort in der höheren Speicheradresse gespeichert, z.B. würde 12345678 mit 1234 in D00001 und 5678 in D00000 gespeichert.

S1 und S1+1: Anzahl der zu lesenden Elemente

Der 8-stellige Hexadezimalwert in S1 und S1+1 spezifiziert, wie viele Worte oder Felder aus dem Dateispeicher gelesen werden sollen. Wenn die spezifizierte Anzahl von Worten oder Feldern die Anzahl der Worte in der Datendatei überschreitet, werden die Daten in der Datei normal und ohne Fehler übertragen.



S1+1 enthält die höchstwertigen vier Stellen und S1 enthält die niederwertigsten vier Stellen.

Datentyp	Bits 12 bis 15 von C	Inhalte von S1 und S1+1
Binär	0 hex (binär)	Anzahl der aus dem Dateispeicher zu lesenden Worte. 00000000 bis 3FFFFFFF hex
Wort	1 hex (nicht getrennt), 3 hex (durch Komma getrennt) oder 5 hex (durch Tabulator getrennt)	Anzahl der aus dem Dateispeicher zu lesenden Felder, d. h., die Anzahl der aus dem Dateispeicher zu lesenden Worte. 00000000 bis 1FFFFFFF hex
Doppelwort	2 hex (nicht getrennt), 4 hex (durch Komma getrennt) oder 6 hex (durch Tabulator getrennt)	Anzahl der aus dem Dateispeicher zu lesenden Felder, d. h., die halbe Anzahl der aus dem Dateispeicher zu lesenden Worte. 00000000 bis 0FFFFFFF hex

S1+2 und S1+3: Erstes Quellwort

Der achtstellige Hexadezimalwert in S1+2 und S1+3 spezifiziert das Anfangswort für das Lesen (vom Anfang der Datei aus gezählt).



S1+3 enthält die höchstwertigen 4 Stellen und S1+2 enthält die niederwertigsten 4 Stellen.

Datentyp	Bits 12 bis 15 von C	Inhalte von S1+2 und S1+3
Binär	0 hex (binär)	Das Wort, mit dem der Lesevorgang begonnen werden soll (vom Anfang des Dateispeichers aus gezählt). 00000000 bis 3FFFFFFF hex
Wort	1 hex (nicht getrennt), 3 hex (durch Komma getrennt) oder 5 hex (durch Tabulator getrennt)	Das Feld, mit dem der Lesevorgang begonnen werden soll (vom Anfang des Dateispeichers aus gezählt), also die Anzahl von Worten ab dem Anfang. 00000000 bis 1FFFFFFF hex
Doppelwort	2 hex (nicht getrennt), 4 hex (durch Komma getrennt) oder 6 hex (durch Tabulator getrennt)	Das Feld, mit dem der Lesevorgang begonnen werden soll (ab dem Anfang des Dateispeichers gezählt), also die halbe Anzahl von Worten ab dem Anfang. 00000000 bis 0FFFFFFF hex

Hinweis

1. S1+2 und S1+3 werden nur für Text- und CSV-Daten ohne Wagenrücklaufzeichen (Bits 08 bis 11 von C sind auf 0 hex gesetzt) oder für Binärdaten verwendet. Stellen Sie S1+2 und S1+3 beim Lesen von Daten mit Wagenrücklaufzeichen (Bits 08 bis 11 von C sind auf 8 bis D hex gesetzt) stets auf 00000000 hex ein.
2. S1 bis S1+3 müssen im gleichen Datenbereich liegen.
3. S1 bis S1+3 werden nur beim Lesen von Daten verwendet.
4. Wenn das spezifizierte Anfangswort die Anzahl Worte in der Datendatei überschreitet, wird das Dateilesefehler-Flag (A34310) auf EIN gesetzt und die Dateidaten werden nicht gelesen.

S2: Dateiname

S2 ist die Anfangsadresse der Worte, die den absoluten Pfad und den Dateinamen im ASCII-Format enthalten. Verwenden Sie die ASCII-Zeichen a bis z, A bis Z und 0 bis 9.

Der vollständige Pfadname für das Verzeichnis, welches die Datendatei enthält, darf einschließlich des Anfangs-Schrägstrichs (ASCII 5C) bis zu 65 Zeichen lang sein. Der Dateiname kann bis zu 8 Zeichen lang sein, jedoch sind Nullzeichen (ASCII 00) im Dateinamen nicht zulässig, da das Nullzeichen zur Kennzeichnung des Endes der Zeichenkette verwendet wird. Fügen Sie keine Dateinamenerweiterung hinzu; die Erweiterung .IOM wird automatisch hinzugefügt.

S2	F1	F2
S2+1	F3	F4
⋮	⋮	⋮
S2+38	F73	F74

Die Zeichenkette wird mit dem höchstwertigen Byte in S2 beginnend gespeichert. Der gesamte Pfad- und Dateiname darf bis zu 74 Zeichen (Bytes) lang sein, einschließlich des Anfangs-Schrägstrichs und der Null als Endzeichen.

Hinweis

1. Stellen Sie sicher, dass die Zeichenkette, die den Pfad- und Dateinamen enthält, das Ende des Datenbereichs nicht überschreitet.
2. Wenn die spezifizierten Dateien oder Verzeichnisse nicht vorhanden sind, wird das Datei-fehlt-Flag (A34311) auf EIN gesetzt und die Datendatei wird nicht gelesen.

Schrieben Sie den Pfad- und Dateinamen im ASCII-Format, mit dem höchstwertigen Byte von S2 beginnend. Siehe nachfolgendes Beispiel für \ABC\XYZ.IOM (die Erweiterung .IOM wird automatisch angehängt).

S2	"\"	"A"
S2+1	"B"	"C"
S2+2	"\"	"X"
S2+3	"Y"	"Z"
S2+4	NUL	

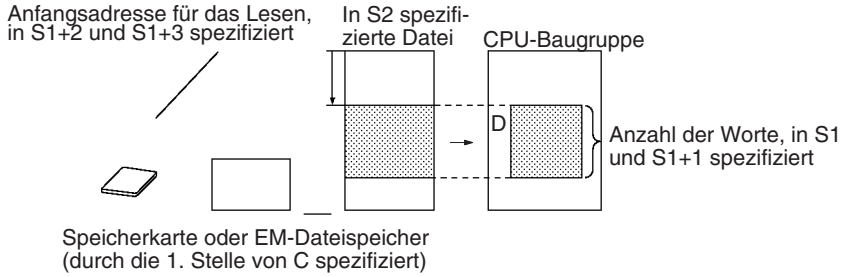
S2	5C	41
S2+1	42	43
S2+2	5C	58
S2+3	59	5A
S2+4	00	

D: Erstes Zielwort
Wenn die Daten gelesen werden, spezifiziert D die Anfangsadresse, ab der die aus dem Dateispeicher gelesenen Daten gespeichert werden.

Wenn die Anzahl der Datenworte gelesen wird, wird die Anzahl der Worte als 8-stellige Hexadezimalzahl (00000000 bis 7FFFFFF) in D und D+1 geschrieben. D enthält die niederwertigsten 4 Stellen und D+1 enthält die höchstwertigen 4 Stellen.

Beschreibung

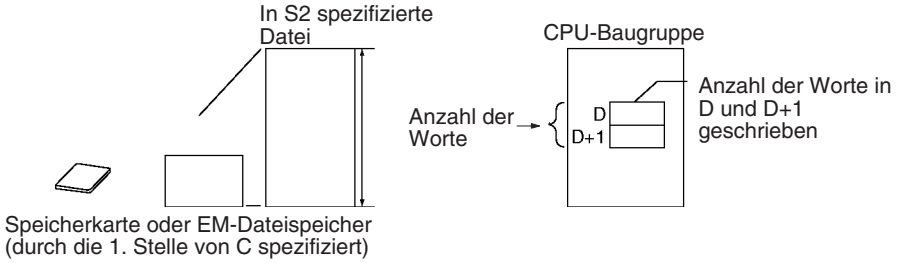
Lesen von Daten (dritte Stelle von C = 0)
FREAD(700) liest die in S1 und S1+1 spezifizierte Anzahl der Worte oder Felder aus der in S2 (mit Dateinamenerweiterung .IOM, .TXT oder .CSV) spezifizierten Datei mit der in S1+2 und S1+3 spezifizierten Adresse beginnend. Die Daten werden anschließend bei dem in D spezifizierten Wort beginnend in den Speicher geschrieben.



Hinweis Daten werden in der Reihenfolge der absoluten internen Speicheradressen gespeichert, sodass die geschriebenen Daten die Daten im nächsten Datenbereich überschreiben, wenn diese die Kapazität des in D spezifizierten Datenbereichs überschreiten. Nähere Einzelheiten finden Sie unter *Vorsichtsmaßnahmen*.

Bei Ausführung von FREAD(700) wird die in S1 und S1+1 spezifizierte Anzahl der Worte (oder Felder) in A346 und A347 (Anzahl der zu übertragenden Daten) geschrieben und dieser Wert wird bei jedem übertragenem Wort oder Feld um 1 dekrementiert. Der Inhalt dieser Worte kann zur Prüfung verwendet werden, ob die erwartete Anzahl von Worten oder Feldern übertragen wurde.

Lesen der Anzahl der Datenworte (dritte Stelle von C = 1)
FREAD(700) ermittelt die in Anzahl der Worte in der in S2 spezifizierten Datei (mit Dateinamenerweiterung .IOM) und schreibt diese als 8-stelligen Hexadezimalwert in D und D+1.



Operanden-Spezifikationen

Bereich	C	S1	S2	D
CIO-Bereich	CIO 0000 bis CIO6143	CIO 0000 bis CIO 6140	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W508	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	H000 bis 508	H000 bis W511	
Systembereich	A000 bis A959	A000 bis A444 A448 bis A956	A000 bis A447 A448 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4092	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	C0000 bis C4092	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	D00000 bis D32764	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32764	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32764 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	–	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	–	∗D00000 bis *D32767 ∗E00000 bis *E32767 ∗En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	Nur spezifizierte Werte	–		
Datenregister	–			
Indexregister	–			
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15			

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn der in C spezifizierte Dateispeicher nicht vorhanden ist. EIN, wenn die Einstellung in C nicht innerhalb des spezifizierten Bereichs liegt. EIN, wenn der in S2 spezifizierte Dateiname nicht den erforderlichen Bedingungen entspricht. EIN, wenn das Dateispeicherfunktions-Flag auf EIN gesetzt war. EIN, wenn für C keine Konstante spezifiziert wurde (nur bei CS1 CPU-Baugruppen von SPS der CS-Serie vor V1□). EIN, wenn die für S1 spezifizierten Daten außerhalb des gültigen Bereich liegen (alle CPU-Baugruppen außer CS1 CPU-Baugruppen von SPS der CS-Serie vor V1□). EIN, wenn für D ein unzulässiger Bereich spezifiziert wurde. Bei CS1D CPU-Baugruppen: EIN, wenn die aktive und die Standby-CPU-Baugruppe nicht synchronisiert werden konnten. In allen anderen Fällen AUS.

In der folgenden Tabelle sind die relevanten Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Speicherkartentyp	A34300 bis A34302	Enthält eine Binärzahl, die den Typ der Speicherkarte bezeichnet, sofern eine solche in die CPU-Baugruppe eingesetzt ist. (0: Keine, 4: Flash-ROM)
Speicherkarten-Formatierungsfehler-Flag	A34307	EIN, wenn die Speicherkarte nicht formatiert ist oder ein Formatierungsfehler aufgetreten ist.
Dateilesefehler-Flag	A34310	EIN, wenn eine Datei aufgrund beschädigter Daten oder des falschen Datentyps nicht gelesen werden konnte.
Datei-fehlt-Flag	A34311	EIN, wenn Daten nicht gelesen werden konnten, da die spezifizierte Datei nicht vorhanden ist.
Dateispeicher-funktions-Flag	A34313	EIN, falls eine der folgenden Bedingungen zutrifft: Die CPU-Baugruppe hat mit CMND(490) einen FINS-Befehl an sich selbst gesendet. FREAD(700) oder FWRIT(701) wird ausgeführt. Das Programm wird mittels eines Steuerbits im Speicher überschrieben. Ein Easy-Backup-Sicherungsvorgang wird ausgeführt.
Dateizugriffs-Flag	A34314	EIN, während ein Zugriff auf Dateidaten erfolgt. Verwenden Sie dieses Flag als Ausführungsbedingung, die die Ausführung eines Dateispeicherbefehls verhindert, solange ein anderer Befehl ausgeführt wird.
Speicherkarte-erkannt-Flag	A34315	EIN, wenn eine Speicherkarte erkannt wurde.
EM-Datei-Format-Anfangsbank	A344	Enthält die Anfangsbanknummer des EM-Bereichs, der zur Verwendung als EM-Dateispeicher formatiert wurde. Enthält FFFF, wenn kein EM-Bereich formatiert wurde. Um den EM-Bereich zur Verwendung als Dateispeicher zu formatieren, muss die EM-Dateispeichereinstellung im SPS-Setup auf 1 gesetzt und die EM-Dateispeicher-Anfangsbank (0 bis C) eingestellt werden. Alle EM-Bänke von dieser Anfangsbank bis zur letzten Bank werden dadurch als Dateispeicher formatiert.
EM-Dateispeicher-Formatierungsfehler-Flag	A34306	EIN, wenn ein Formatierungsfehler in der Anfangsbank des EM-Dateispeichers vorliegt.
Anzahl der zu übertragenden Daten	A346 bis A347	Der Inhalt dieser Worte gibt den Status der Datendateiübertragungen an. Wenn ein FREAD(700)- oder FWRIT(701)-Befehl ausgeführt wird, wird die Anzahl der zu übertragenden Worte oder Felder in diese Worte geschrieben. Der Wert wird bei der Übertragung jedes Worts oder Felds um 1 dekrementiert. A346 enthält die niederwertigsten 16 Bits und A347 enthält die höchstwertigen 16 Bits des 32-Bit-Binärwerts.

Vorsichtsmaßnahmen

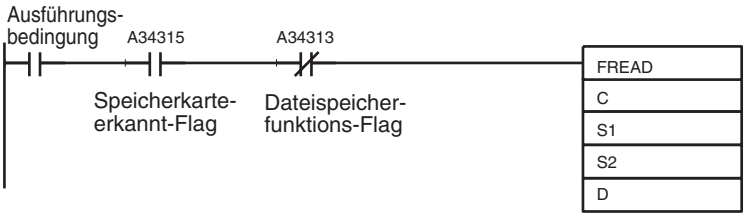
Während der normalen Befehlsverarbeitung wird FREAD(700) nur zum Starten des Dateispeicher-Lesevorgangs verwendet. Die am Ende dieses Handbuchs angegebenen Befehlsausführungszeiten sind deshalb die Zeiten, die zum Start des Lesevorgangs erforderlich sind, nicht zu dessen vollständiger Ausführung. Der tatsächliche Lesevorgang (Übertragung) wird durch die Dateizugriffsverarbeitung in der Peripheriebedienung ausgeführt. Deshalb wird der Lesevorgang ab der Ausführung von FREAD(700) kontinuierlich ausgeführt, sogar wenn die Ausführungsbedingung in den folgenden Zyklen auf AUS gesetzt ist. Wenn die Übertragung abgeschlossen ist, wird das Dateispeicherfunktions-Flag (A34313) auf AUS gesetzt. Dieses Flag kann für die exklusive Steuerung von Dateispeicherbefehlen verwendet werden.

Die erforderliche Gesamtzeit für die Durchführung der Datenübertragung, die durch FREAD(700) gestartet wird, hängt von der Menge der zu übertragenden Daten sowie von der für die Dateizugriffsverarbeitung zugewiesenen Servicezeit und weiteren Bedingungen ab. Als Richtlinie gilt, dass die Übertragungszeit bei einer Zykluszeit von 10 ms für eine Datei im Stammverzeichnis mit der Standard-Servicezeiteinstellung 0,92 s für 1024 Worte und 4,64 s für 9999 Worte beträgt.

Das Dateispeicherfunktions-Flag (A34313) wird bei Ausführung von FREAD(700) auf EIN gesetzt. Wenn A34313 bereits auf EIN gesetzt ist, tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

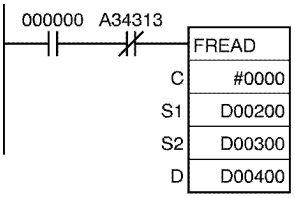
Das Dateilesefehler-Flag (A34310) wird auf EIN gesetzt, und der Befehl wird nicht ausgeführt, wenn die spezifizierte Datei den falschen Datentyp enthält oder die Dateidaten beschädigt sind. Bei Text- oder CSV-Dateien muss der Zeichencode aus hexadezimalen Daten bestehen, und die Trennzeichen müssen bei Wortdaten alle 4 Stellen und bei Doppelwortdaten alle 8 Stellen vorhanden sein. Die Daten werden bis zu dem Punkt gelesen, an dem ein ungültiges Zeichen erkannt wird.

Die CPU-Baugruppe benötigt einige Sekunden, um eine Speicherkarte nach dem Einsetzen zu erkennen. Wenn kurz nach dem Einschalten der Versorgungsspannung oder nach dem Einsetzen der Speicherkarte auf die Speicherkarte zugegriffen werden soll, muss das Speicherkarte-erkannt-Flag (A34315) in einer Schließer-Eingangsbedingung verwendet werden (wie nachstehend gezeigt), um sicherzustellen, dass die Speicherkarte erkannt wurde.

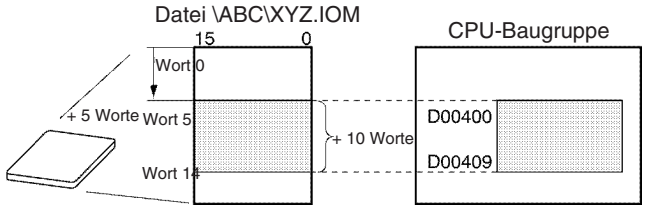


Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, liest FREAD(700) 10 Datenworte aus der Datei \ABC\XYZ.IOM, beginnend mit dem Anfang der Datei + 5 Worte. Diese 10 Datenworte werden in D00400 bis D00409 ausgegeben.





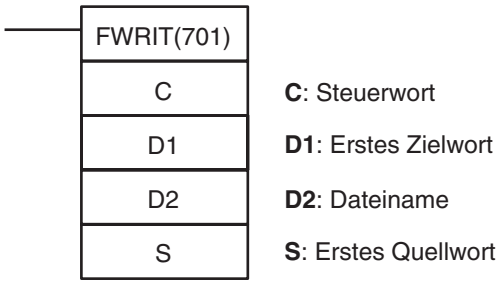


3-26-3 WRITE DATA FILE: FWRIT(701)

Funktion

Überschreibt Daten in der in der spezifizierten Datendatei im Dateispeicher mit den spezifizierten Daten aus dem Datenbereich der CPU-Baugruppe, oder hängt diese Daten an. Wenn die spezifizierte Datei nicht vorhanden ist, wird eine neue Datei mit diesem Dateinamen erstellt. Daten können im Binär-, Text- oder CS-Format geschrieben werden.

Kontaktplansymbol



Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	FWRIT(701)
	Ausführung einmal bei steigender Flanke	@FWRIT(701)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

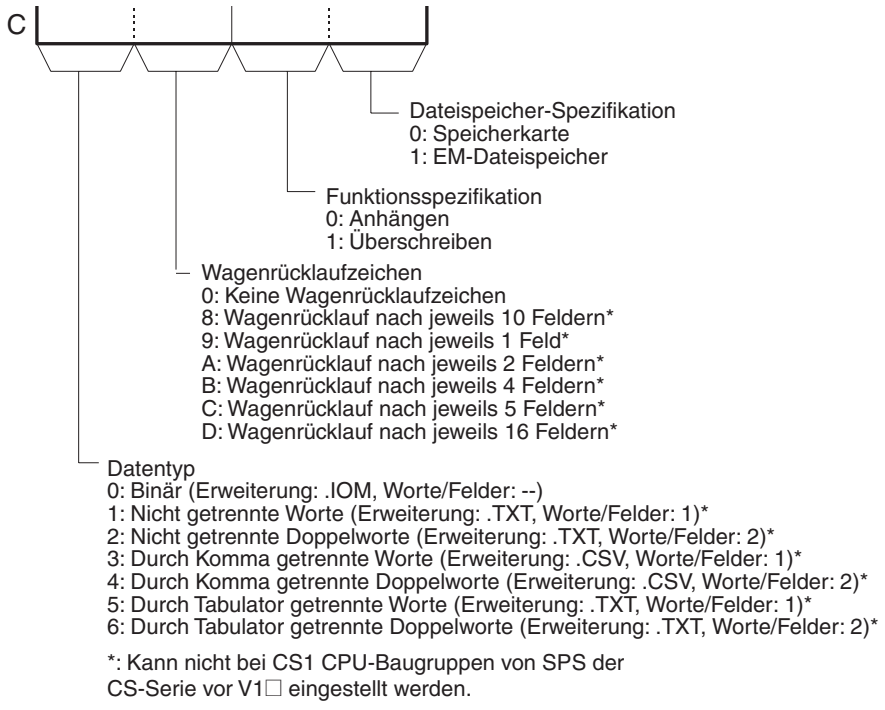
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

Wie in der folgenden Abbildung veranschaulicht, spezifiziert die dritte Stelle des Steuerworts, ob Daten in der Datendatei angehängt oder überschrieben werden sollen und die vierte Stelle spezifiziert, ob sich die Zieldatei auf der Speicherkarte oder im EM-Dateispeicher befindet.



Hinweis

1. Jedes Feld enthält bei Wortdatentypen 1 Wort des E/A-Speichers und bei Doppelwort-Datentypen 2 Worte des E/A-Speichers.
2. Bei Doppelworten wird das erste Datenwort aus der höheren Speicheradresse gelesen, beispielsweise würde 12345678 mit 1234 aus D00001 und 5678 aus D00000 geschrieben.
3. Wenn eine Trennung spezifiziert ist, wird bei Wort-Datentypen nach jedem Wort und bei Doppelwort-Datentypen nach jedem zweiten Wort ein Trennzeichen hinzugefügt. (Bei Kommatrennungen wird der Code für ein Komma und bei Tabulatortrennungen der Code für einen Tabulator hinzugefügt.)
4. Wenn nicht getrennte Worte oder Doppelworte spezifiziert sind, werden die Daten für alle Felder kontinuierlich ohne jegliche Trennzeichen geschrieben.
5. Wenn Wagenrückläufe spezifiziert sind, wird nach jedem Satz der spezifizierten Wortzahl ein Wagenrücklauf hinzugefügt. Sind keine Wagenrückläufe spezifiziert, werden die Daten kontinuierlich ohne Wagenrückläufe geschrieben.

D1 und D1+1: Anzahl der zu schreibenden Elemente

Der 8-stellige Hexadezimalwert in D1 und D1+1 spezifiziert, wie viele Worte oder Felder in den Dateispeicher geschrieben werden sollen.



D1+1 enthält die höchstwertigen 4 Stellen und D1 enthält die niederwertigsten 4 Stellen.

Datentyp	Bits 12 bis 15 von C	Inhalt von D1 und D1+1
Binär	0 hex (binär)	Anzahl der in den Dateispeicher zu schreibenden Worte. 00000000 bis 3FFFFFFF hex
Wort	1 hex (nicht getrennt), 3 hex (durch Komma getrennt) oder 5 hex (durch Tabulator getrennt)	Anzahl der in den Dateispeicher zu schreibenden Felder, also die Anzahl der in den Dateispeicher zu schreibenden Worte. 00000000 bis 1FFFFFFF hex
Doppelwort	2 hex (nicht getrennt), 4 hex (durch Komma getrennt) oder 6 hex (durch Tabulator getrennt)	Anzahl der in den Dateispeicher zu schreibenden Felder, also die halbe Anzahl der in den Dateispeicher zu schreibenden Worte. 00000000 bis 0FFFFFFF hex

D1+2 und D1+3: Erstes Zielwort

Der achtstellige Hexadezimalwert in D1+2 und D1+3 spezifiziert das Anfangswort für das Schreiben der Daten, vom Anfang der Datei aus gezählt.



D1+3 enthält die höchstwertigen 4 Stellen und D1+2 enthält die niederwertigsten 4 Stellen.

Datentyp	Bits 12 bis 15 von C	Inhalt von D1+2 und D1+3
Binär	0 hex (binär)	Das Wort, an dem mit dem Schreiben begonnen werden soll, vom Anfang des Dateispeichers aus gezählt. 00000000 bis 3FFFFFFF hex
Wort	1 hex (nicht getrennt), 3 hex (durch Komma getrennt) oder 5 hex (durch Tabulator getrennt)	Das Feld, an dem mit dem Schreiben begonnen werden soll, vom Anfang des Dateispeichers aus gezählt, d. h., die Anzahl von Worten ab dem Anfang. 00000000 bis 1FFFFFFF hex
Doppelwort	2 hex (nicht getrennt), 4 hex (durch Komma getrennt) oder 6 hex (durch Tabulator getrennt)	Das Feld, an dem mit dem Schreiben begonnen werden soll, vom Anfang des Dateispeichers aus gezählt, d. h., die halbe Anzahl von Worten ab dem Anfang. 00000000 bis 0FFFFFFF hex

Hinweis

1. D1+2 und D1+3 werden nur beim Überschreiben von Daten und nur bei 1.) Text und CSV-Daten ohne Wagenrücklaufzeichen (Bits 08 bis 11 von C auf 0 hex gesetzt) oder 2.) Binärdaten. Setzen Sie D1+2 und D1+3 beim

Schreiben von Daten mit Wagenrücklaufzeichen (Bits 08 bis 11 von C auf 8 bis D hex gesetzt) stets auf 00000000 hex.

2. D1 und D1+3 müssen im gleichen Datenbereich liegen.
3. Wenn das spezifizierte Anfangswort die Anzahl der Worte in der Datendatei überschreitet, wird das Dateischreibfehler-Flag (A34308) auf EIN gesetzt und die Daten werden nicht geschrieben.

D2: Dateiname

D2 ist die Anfangsadresse der Worte, die den absoluten Pfad- und Dateinamen im ASCII-Format enthalten. Verwenden Sie die ASCII-Zeichen a bis z, A bis Z und 0 bis 9.

Der vollständige Pfadname für das Verzeichnis, welches die Datendatei enthält, darf einschließlich des Anfangs-Schrägstrichs (ASCII 5C) bis zu 65 Zeichen lang sein. Der Dateiname kann bis zu 8 Zeichen lang sein, jedoch sind Nullzeichen (ASCII 00) im Dateinamen nicht zulässig, da das Nullzeichen zur Kennzeichnung des Endes der Zeichenkette verwendet wird. Fügen Sie keine Dateinamenerweiterung hinzu; die Erweiterungen .IOM, .TXT oder .CSV werden automatisch hinzugefügt.

D2	F1	F2
D2+1	F3	F4
⋮	⋮	⋮
D2+38	F73	F74

Speicherung der Zeichenkette mit dem höchstwertigen Byte in D2 beginnend. Der gesamte Pfad- und Dateiname darf bis zu 74 Zeichen (Bytes) lang sein, einschließlich des Anfangs-Schrägstrichs und der Null als Endzeichen.

Hinweis

1. Stellen Sie sicher, dass die Zeichenkette, die den Pfad- und Dateinamen enthält, das Ende des Datenbereichs nicht überschreitet.
2. Wenn das spezifizierte Verzeichnis nicht vorhanden ist, wird das Dateifehlt-Flag (A34311) auf EIN gesetzt und die Daten werden nicht geschrieben.

Schrieben Sie den Pfad- und Dateinamen im ASCII-Format, mit dem höchstwertigen Byte von D2 beginnend. Siehe nachfolgendes Beispiel für \ABC\XYZ.IOM (die Erweiterung .IOM wird automatisch angehängt).

D2	\ *	*A*	D2	5C	41
D2+1	*B*	*C*	→ D2+1	42	43
D2+2	\ *	*X*	D2+2	5C	58
D2+3	*Y*	*Z *	D2+3	59	5A
D2+4	NUL		→ D2+4	00	

S: Erstes Quellwort

S spezifiziert die Anfangsadresse der Worte, die die in den Dateispeicher zu schreibenden Daten enthalten. Die Daten werden von der absoluten SPS-Speicheradresse gelesen, sodass FWRIT(701) mit dem Lesen der Quelldaten aus dem nächsten Datenbereich fortfährt, wenn die Anzahl der zu lesenden Worte das Ende des in S spezifizierten Datenbereichs überschreitet.

Beschreibung

Während der normalen Befehlsverarbeitung wird FWRIT(701) nur zum Starten des Dateispeicher-Schreibvorgangs verwendet. Die am Ende dieses Handbuchs angegebenen Befehlsausführungszeiten sind daher die Zeiten, die zum Start des Schreibvorgangs erforderlich sind, nicht zu dessen vollständiger Ausführung. Der tatsächliche Schreibvorgang (Übertragung) wird durch den Dateizugriffsverarbeitung in der Peripheriebedienung ausgeführt. Dementsprechend wird der Schreibvorgang nach Ausführung von FWRIT(701) kontinuierlich ausgeführt, sogar wenn die Ausführungsbedingung in den folgenden Zyklen auf AUS gesetzt ist. Wenn die Übertragung abgeschlossen ist, wird das Dateispeicherfunktions-Flag (A34313) auf AUS gesetzt. Dieses Flag kann für die exklusive Steuerung von Dateispeicherbefehlen verwendet werden.

Die erforderliche Gesamtzeit für die Durchführung der Datenübertragung, die durch FWRIT(701) gestartet wird, hängt von der Menge der zu übertragenden Daten sowie von der für die Dateizugriffsverarbeitung zugewiesenen Service-

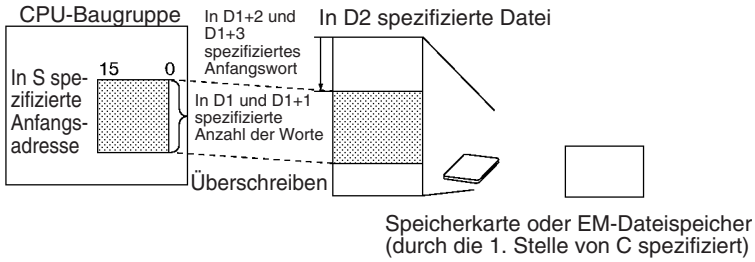
zeit und weiteren Bedingungen ab. Als Richtlinie gilt, dass die Übertragungszeit bei einer Zykluszeit von 10 ms für eine Datei im Stammverzeichnis mit der Standard-Servicezeiteinstellung für 1024 Worte 1,97 s (neue Datei) bzw. 1,33 s (vorhandene Datei) und für 9999 Worte 6,64 s (neue Datei) bzw. 6,12 s (vorhandene Datei) beträgt.

Die Quelldaten werden aus den absoluten internen Speicheradressen im RAM gelesen, sodass der gesamte Datenblock gelesen wird, auch wenn sich die Daten über zwei oder mehr Datenbereiche erstrecken. Wenn sich beispielsweise die erste Zieladresse im nicht remanenten Arbeitsbereich befindet, aber die Datenmenge die Kapazität dieses Bereichs überschreitet, fährt FWRIT(701) mit dem kontinuierlichen Lesen am Beginn des nächsten Bereichs (in diesem Fall der Zeitgeberbereich) fort. Eine Speicherübersicht, die die Datenbereiche im RAM darstellt, finden Sie in *Anhang D* des *CS/CJ-Serie Speicherprogrammierbare Steuerungen Bedienerhandbuchs (W339)*.

Bei Ausführung von FWRIT(701) wird die in D1 und D1+1 spezifizierte Anzahl von Worten oder Feldern in A346 und A347 (Anzahl der zu übertragenden Daten) geschrieben und dieser Wert wird bei jedem übertragenem Wort bzw. Feld um 1 dekrementiert. Der Inhalt dieser Worte kann zur Prüfung verwendet werden, ob die erwartete Anzahl von Worten oder Feldern übertragen wurde.

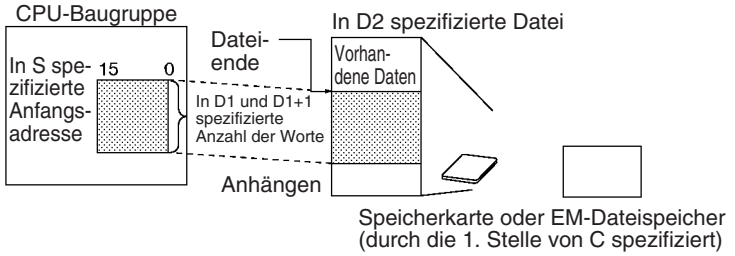
Überschreiben von Daten in einer vorhandenen Datei (dritte Stelle von C = 1)

FWRIT(701) verwendet die mit dem in S spezifizierten Wort beginnenden Datenbereichsdaten zum Überschreiben von Dateispeicherdaten des spezifizierten Datentyps. Der Befehl überschreibt die in D1 und D1+1 spezifizierte Anzahl von Worten oder Feldern in der in D2 spezifizierten Datei (mit Dateierweiterung .IOM, .TXT oder .CVS) mit der in D1+2 und D1+3 spezifizierten Adresse beginnend.



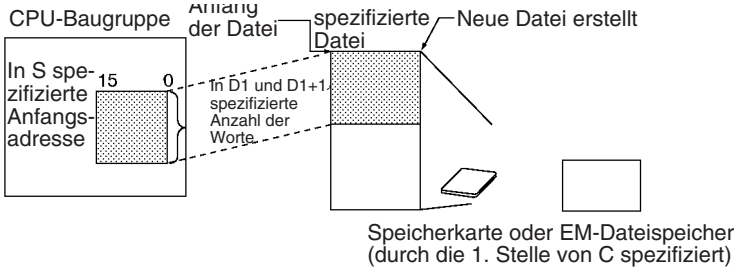
Anhängen von Daten an eine vorhandene Datei (dritte Stelle von C = 0)

FWRIT(701) hängt die mit dem in S spezifizierten Wort beginnenden Datenbereichdaten an eine Datendatei des spezifizierten Datentyps im Dateispeicher an. Der Befehl hängt die in D1 und D1+1 spezifizierte Anzahl von Worten oder Feldern an die in D2 spezifizierte Datei (mit Dateierweiterung .IOM, .TXT oder .CVS) an.



Erstellen einer neuen Datei mit den Quelldaten

Wenn die in D2 spezifizierte Datei nicht vorhanden ist, erstellt FWRIT(701) eine neue Datei mit diesem Namen und der Dateinamenerweiterung (.IOM, .TXT oder .CVS) und schreibt die spezifizierten Quelldaten des spezifizierten Datentyps am Anfang der Datei beginnend in diese Datei. In diesem Fall ist es unerheblich, ob das Anhängen oder Überschreiben von Daten spezifiziert wurde.



Operanden-Spezifikationen

Bereich	C	D1	D2	S
CIO-Bereich	CIO 0000 bis CIO 6143	CIO 0000 bis CIO 6140	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	W000 bis W508	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	H000 bis 508	H000 bis H511	
Systembereich	A000 bis A959	A000 bis A444 A448 bis A956	A000 bis A447 A448 bis A959	
Zeitgeberbereich	T0000 bis T4095	T0000 bis T4092	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	C0000 bis C4092	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	D00000 bis D32764	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	E00000 bis E32764	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	En_00000 bis En_32764 (n = 0 bis C)	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	–	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	–	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	Nur spezifizierte Werte	–		
Datenregister	–			
Indexregister	–			
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15			

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der in C spezifizierte Dateispeichertyp nicht vorhanden ist. EIN, wenn die Einstellung in C nicht innerhalb des spezifizierten Bereichs liegt. EIN, wenn der in D2 spezifizierte Dateiname nicht den erforderlichen Bedingungen entspricht. EIN, wenn das Dateispeicherfunktions-Flag auf EIN gesetzt war. EIN, wenn für C keine Konstante spezifiziert wurde (nur bei CS1 CPU-Baugruppen von SPS der CS-Serie vor V1). EIN, wenn die für D1 spezifizierten Daten außerhalb des gültigen Bereichs liegen (alle CPU-Baugruppen außer CS1 CPU-Baugruppen von SPS der CS-Serie vor V1). EIN, wenn für S ein unzulässiger Bereich spezifiziert wurde. Bei CS1D CPU-Baugruppen: EIN, wenn die aktive und die Standby-CPU-Baugruppe nicht synchronisiert werden konnten. In allen anderen Fällen AUS.

In der folgenden Tabelle sind die relevanten Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Speicherkartentyp	A34300 bis A34302	Enthält eine Binärzahl, die den Typ der Speicherkarte bezeichnet, sofern eine solche in die CPU-Baugruppe eingesetzt ist. (0: Keine, 4: Flash-ROM)
Speicherkarten-Formatierungsfehler-Flag	A34307	EIN, wenn die Speicherkarte nicht formatiert ist oder ein Formatierungsfehler aufgetreten ist.
Dateischreibfehler-Flag	A34308	EIN, wenn beim Schreiben in die Datei ein Fehler aufgetreten ist.
Schreiben-in-Datei-unmöglich-Flag	A34309	EIN, wenn die Daten nicht geschrieben werden konnten, da die Datei schreibgeschützt war oder nicht ausreichend freier Speicherplatz zur Verfügung stand.
Keine-Datei-Flag	A34311	EIN, wenn das spezifizierte Verzeichnis beim Schreiben einer Datei nicht vorhanden war.
Dateispeicherfunktions-Flag	A34313	EIN, falls eine der folgenden Bedingungen zutrifft: Die CPU-Baugruppe hat mit CMND(490) einen FINS-Befehl an sich selbst gesendet. FREAD(700) oder FWRIT(701) wird ausgeführt. Das Programm wird mittels eines Steuerbits im Speicher überschrieben. Ein Easy-Backup-Sicherungsvorgang wird ausgeführt.
Dateizugriffs-Flag	A34314	EIN, während ein Zugriff auf Dateidaten erfolgt. Verwenden Sie dieses Flag als Ausführungsbedingung, die die Ausführung eines Dateispeicherbefehls verhindert, solange ein anderer Befehl ausgeführt wird.
Speicherkarte-erkannt-Flag	A34315	EIN, wenn eine Speicherkarte erkannt wurde.
EM-Datei-Format-Anfangsbank	A344	Enthält die Anfangsbanknummer des EM-Bereichs, der zur Verwendung als EM-Dateispeicher formatiert wurde. Enthält FFFF, wenn kein EM-Bereich formatiert wurde. Um den EM-Bereich zur Verwendung als Dateispeicher zu formatieren, muss die EM-Dateispeichereinstellung im SPS-Setup auf 1 gesetzt und die EM-Dateispeicher-Anfangsbank (0 bis C) eingestellt werden. Alle EM-Bänke von dieser Anfangsbank bis zur letzten Bank werden dadurch als Dateispeicher formatiert.

Name	Adresse	Funktion
EM-Datei-speicher-Formatierungs-fehler-Flag	A34306	EIN, wenn ein Formatierungsfehler in der Anfangsbank des EM-Dateispeichers vorliegt.
Anzahl der zu übertragenden Daten	A346 bis A347	Der Inhalt dieser Worte gibt den Status der Datendateiübertragungen an. Bei Ausführung eines FWRIT(701)-Befehls wird die Anzahl der zu übertragenden Worte oder Felder in diese Worte geschrieben. Der Wert wird bei der Übertragung jedes Worts um 1 dekrementiert. A346 enthält die niederwertigsten 16 Bits und A347 enthält die höchstwertigen 16 Bits des 32-Bit-Binärwerts.

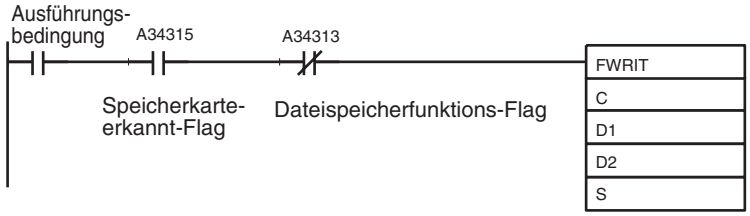
Vorsichtsmaßnahmen

Das Dateispeicherfunktions-Flag (A34313) wird bei Ausführung von FWRIT(701) auf EIN gesetzt. Wenn A34313 bereits auf EIN gesetzt ist, tritt ein Fehler auf und der Befehl wird nicht ausgeführt.

Das Schreiben-in-Datei-unmöglich-Flag (A34309) wird auf EIN gesetzt und der Befehl wird nicht ausgeführt, wenn die Daten aufgrund einer schreibgeschützten Datei oder wegen zu wenig freiem Speicherplatz nicht geschrieben werden konnten.

Das Dateischreibfehler-Flag (A34308) wird auf EIN gesetzt und der Befehl wird nicht ausgeführt, wenn die spezifizierte Datei nicht den richtigen Datentyp besitzt oder die Dateidaten beschädigt sind.

Die CPU-Baugruppe benötigt einige Sekunden, um eine Speicherkarte nach dem Einsetzen zu erkennen. Wenn kurz nach dem Einschalten der Versorgungsspannung oder nach dem Einsetzen der Speicherkarte auf die Speicherkarte zugegriffen werden soll, muss das Speicherkarte-erkannt-Flag (A34315) in einer Schließer-Eingangsbedingung verwendet werden (wie nachstehend gezeigt), um sicherzustellen, dass die Speicherkarte erkannt wurde.

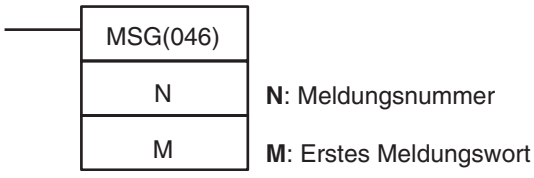


3-27 Anzeigebefehle: DISPLAY MESSAGE: MSG(046)

Funktion

Liest die spezifizierten sechzehn Worte der erweiterten ASCII-Daten und zeigt die Meldung auf einem Peripheriegerät, wie zum Beispiel einer Programmierkonsole, an.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MSG(046)
	Ausführung einmal bei steigender Flanke	@MSG(046)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Meldungsnummer
Die Meldungsnummer muss einen Wert von 0000 bis 0007 hexadezimal (oder 0 bis 7 dezimal) haben.

M: Erstes Meldungswort
Beim Anzeigen einer Meldung spezifiziert M die Adresse des ersten der Worte, die die ASCII-Meldung enthalten. Beim Löschen einer Meldung kann M eine beliebige hexadezimale Konstante (0000 bis FFFF) sein.

Operanden-Spezifikationen

Bereich	N	M
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	#0000 bis #0007 (binär) oder &0 bis &7	#0000 bis #FFFF (binär)
Datenregister	DR0 bis DR15	---
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Wenn die Ausführungsbedingung EIN ist, registriert MSG(046) die 16 Worte mit ASCII-Daten (bis zu 32 Zeichen, einschließlich des Nullzeichens) von M bis M+15 für die durch N spezifizierte Meldungsnummer. Wenn eine Meldung registriert wurde, kann eine Programmierkonsole angeschlossen werden und die Meldung wird nach evtl. vorhandenen Fehlermeldungen angezeigt.

Nach der Registrierung einer Meldung kann die Meldungsanzeige durch Überschreiben der Meldung im Meldungsspeicherbereich geändert werden.

Um die registrierte Meldung zu löschen, muss MSG(046) ausgeführt werden, wobei S auf die Meldungsnummer der zu löschenden Meldung und N auf eine Konstante (0000 bis FFFF) gesetzt sein muss.

Eine während der Programmausführung registrierte Meldung bleibt auch erhalten, wenn die Programmausführung unterbrochen wird. Bei Fortsetzung der Programmausführung werden jedoch alle Meldungen gelöscht.

Hinweis Eine Tabelle der erweiterten ASCII-Zeichen finden Sie in *Anhang A* im *CS/CJ-Serie Programmkonsolen-Bedienerhandbuch (W341)*.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn der Inhalt von S nicht 0000 bis 0007 hexadezimal ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

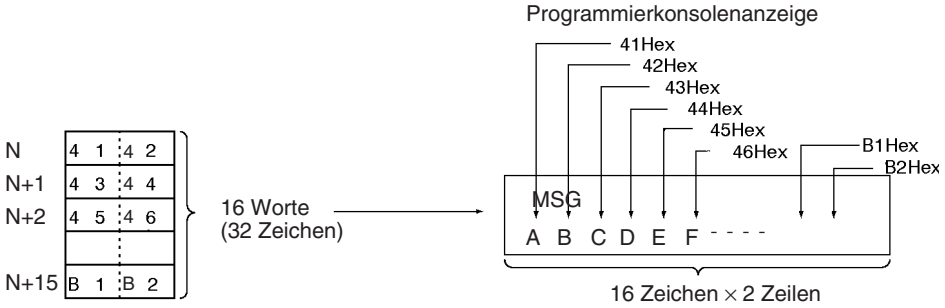
Registrierte Meldungen werden bei jeder Ausführung von MSG(046) aktualisiert.

Alle Meldungszeichen nach dem Nullzeichen (00) werden in auf dem Programmierkonsolen-Display in Leerstellen konvertiert.

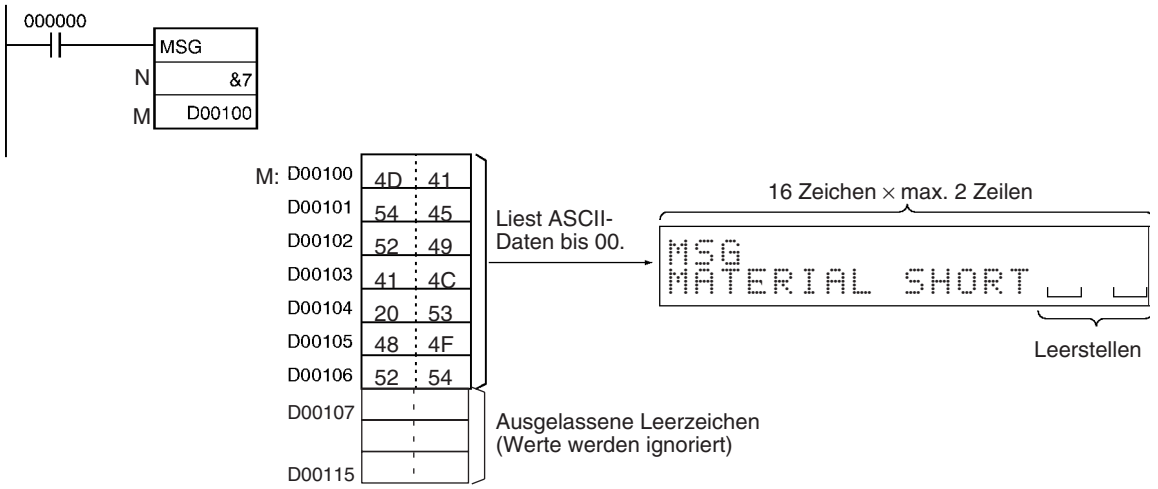
Das im höchstwertigen Byte gespeicherte Zeichen wird vor dem Zeichen im niederwertigsten Byte angezeigt.

Beispiele

In der folgenden Abbildung ist veranschaulicht, wie 16 Worte Hexadezimaldaten in eine auf der Programmierkonsole angezeigte Meldung konvertiert werden.



Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, werden die 16 Datenworte in D00100 bis D00115 als die 32 Zeichen der ASCII-Daten für Meldung Nummer 7 gelesen und auf dem Peripheriegerät angezeigt.



ASCII

Die vier niederwertigsten Bits \ Die vier höchstwertigen Bits	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	‘	p				ー	タ	ミ		
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			”	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			$	4	D	T	d	t			、	エ	ト	ヤ		
5			%	5	E	U	e	u			・	オ	ナ	ユ		
6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7			’	7	G	W	g	w			ァ	キ	ヌ	ラ		
8			(	8	H	X	h	x			ィ	ク	ネ	リ		
9			)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ェ	コ	ハ	レ		
B			+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C			,	<	L	¥	l	|			ャ	シ	フ	ワ		
D			−	=	M	]	m	}			ュ	ス	ヘ	ン		
E			.	>	N	^	n	~			ョ	セ	ホ	゛		
F			/	?	O	_	o				ッ	ソ	マ	°		

3-28 Datum- und Uhrzeitbefehle

In diesem Abschnitt werden die für Systemuhr verwendeten Befehle beschrieben.

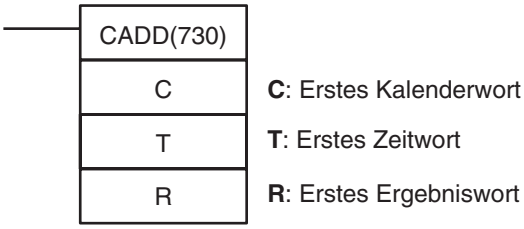
Befehl	AWL	Funktionscode	Seite
CALENDAR ADD	CADD	730	1061
CALENDAR SUBTRACT	CSUB	731	1065
HOURS TO SECONDS	SEC	065	1068
SECONDS TO HOURS	HMS	066	1070
CLOCK ADJUSTMENT	DATE	735	1073

3-28-1 CALENDAR ADD: CADD(730)

Funktion

Addiert Zeit zu den Kalenderdaten in den spezifizierten Worten.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CADD(730)
	Ausführung einmal bei steigender Flanke	@CADD(730)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

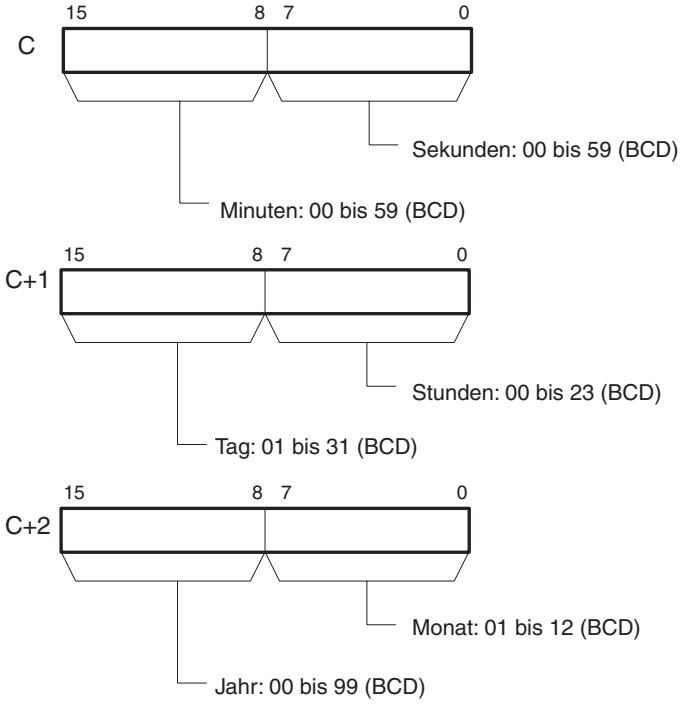
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C bis C+2: Kalenderdaten

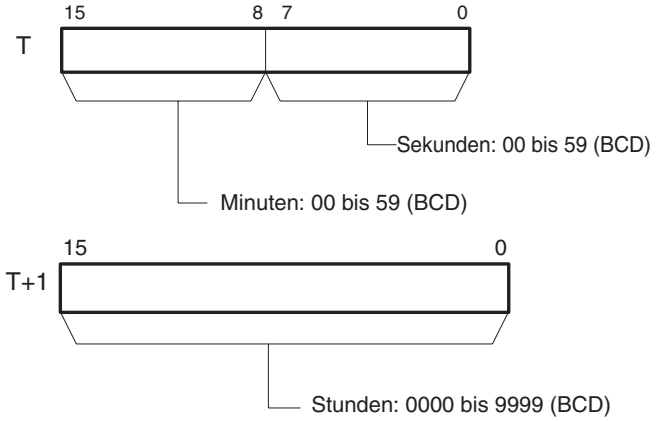
Die Kalenderdaten in C bis C+2 werden wie in der nachfolgenden Abbildung gezeigt eingestellt.

C bis C+2 müssen im gleichen Datenbereich liegen.



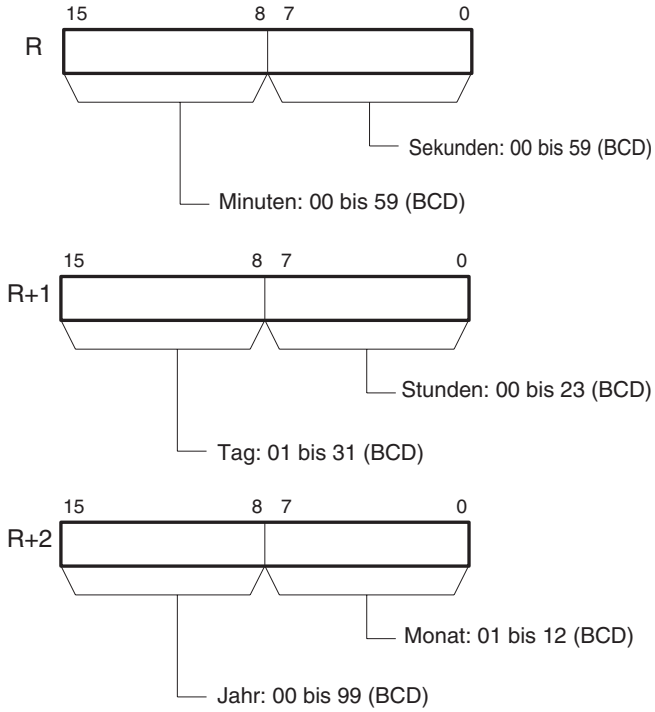
T und T+1: Zeitdaten

Die Zeitdaten in T und T+1 werden wie in der nachfolgenden Abbildung gezeigt eingestellt. T und T+1 müssen im gleichen Datenbereich liegen.



R bis R+2: Ergebnisdaten

R bis R+2 enthalten das Ergebnis der Addition. R bis R+2 müssen im gleichen Datenbereich liegen.



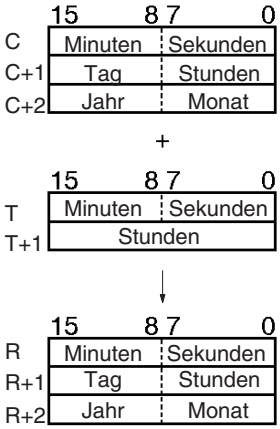
Operanden-Spezifikationen

Bereich	C	T	R
CIO-Bereich	CIO 0000 bis CIO 6141	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6141
Arbeitsbereich nicht remanent	W000 bis W509	W000 bis W510	W000 bis W509
Arbeitsbereich remanent	H000 bis H509	H000 bis H510	H000 bis H509
Systembereich	A000 bis A957	A000 bis A958	A448 bis A957
Zeitgeberbereich	T0000 bis T4093	T0000 bis T4094	T0000 bis T4093
Zählerbereich	C0000 bis C4093	C0000 bis C4094	C0000 bis C4093
DM-Bereich	D00000 bis D32765	D00000 bis D32766	D00000 bis D32765
EM-Bereich ohne Bank	E00000 bis E32765	E00000 bis E32766	E00000 bis E32765
EM-Bereich mit Bank	En_00000 bis En_32765 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32765 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	Nur spezifizierte Werte	---
Datenregister	---		

Bereich	C	T	R
Indexregister	–		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR005+(++) bis IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

CADD(730) addiert Kalenderdaten (Worte C bis C+2) zu den Zeitdaten (Worte T und T+1) und gibt die resultierenden Kalenderdaten in R bis R+2 aus.

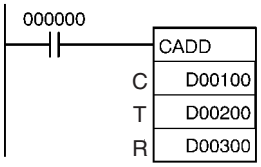


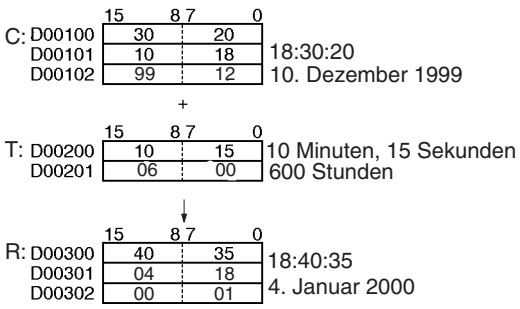
Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn die Kalenderdaten in C bis C+2 nicht innerhalb der spezifizierten Bereiche liegen. EIN, wenn die Zeitdaten T und T+1 nicht innerhalb der spezifizierten Bereiche liegen. In allen anderen Fällen AUS.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, werden die Kalenderdaten in D00100 bis D00102 (Jahr, Monat, Tag, Stunden, Minuten, Sekunden) zu den Zeitdaten in D00200 und D00201 (Stunden, Minuten, Sekunden) addiert und das Ergebnis in D00300 bis D00302 ausgegeben.



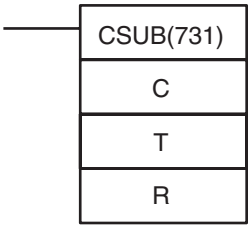


3-28-2 CALENDAR SUBTRACT: CSUB(731)

Funktion

Subtrahiert Zeit von den Kalenderdaten in den spezifizierten Worten.

Kontaktplansymbol



C: Erstes Kalenderwort

T: Erstes Zeitwort

R: Erstes Ergebniswort

Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	CSUB(731)
	Ausführung einmal bei steigender Flanke	@CSUB(731)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

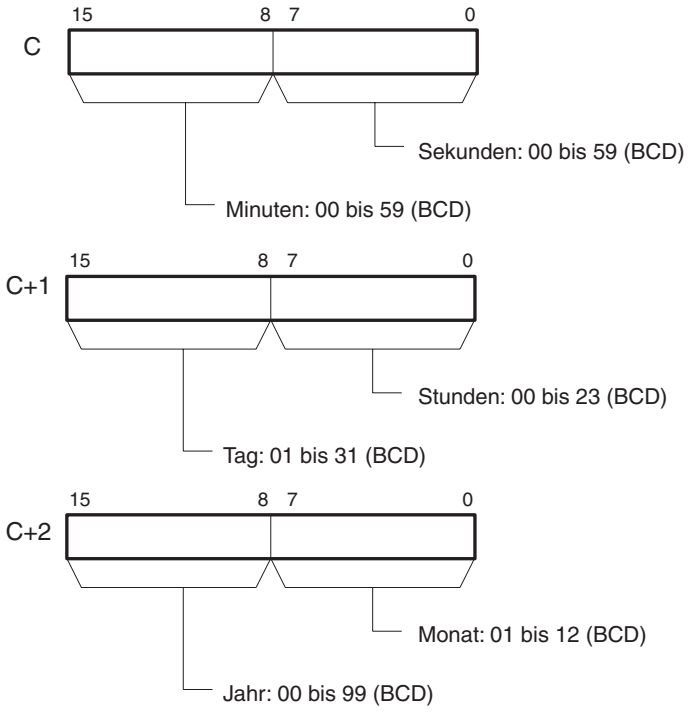
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C bis C+2: Kalenderdaten

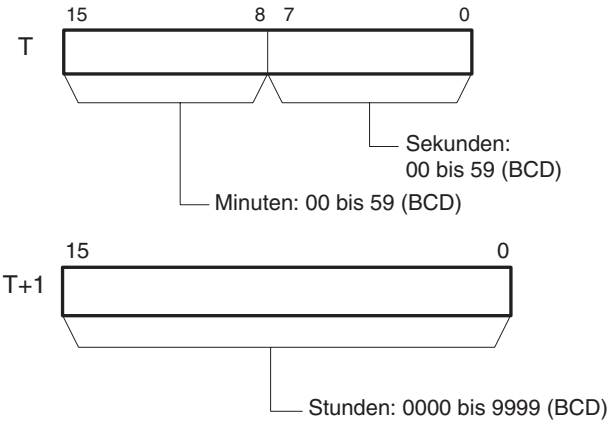
Die Kalenderdaten in C bis C+2 werden wie in der nachfolgenden Abbildung gezeigt eingestellt.

C bis C+2 müssen im gleichen Datenbereich liegen.



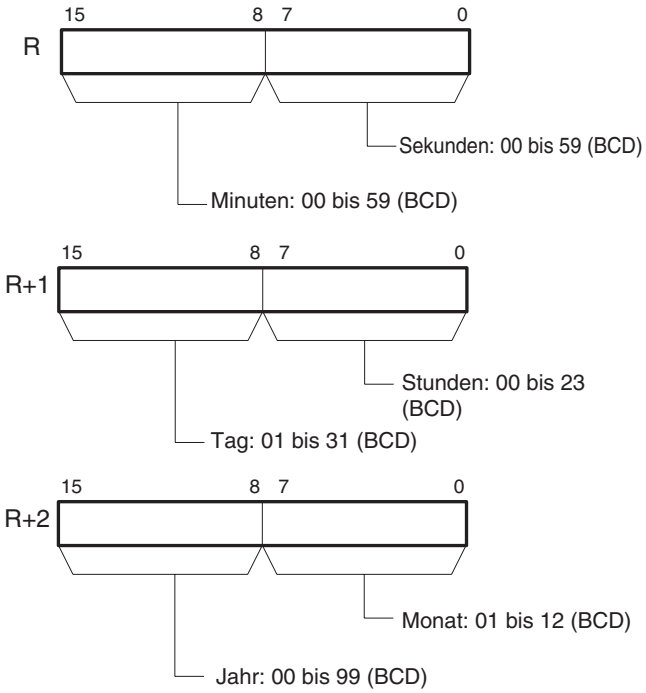
T und T+1: Zeitdaten

Die Zeitdaten in T und T+1 werden wie in der nachfolgenden Abbildung gezeigt eingestellt. T und T+1 müssen im gleichen Datenbereich liegen.



R bis R+2: Ergebnisdaten

R bis R+2 enthalten das Ergebnis der Subtraktion. R bis R+2 müssen im gleichen Datenbereich liegen.



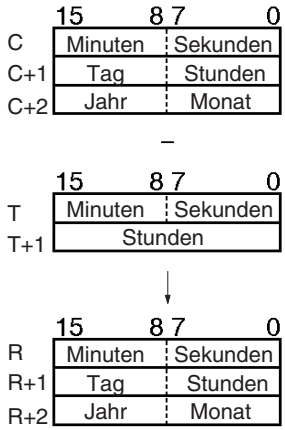
Operanden-Spezifikationen

Bereich	C	T	R
CIO-Bereich	CIO 0000 bis CIO 6141	CIO 0000 bis CIO 6142	CIO 0000 bis CIO 6141
Arbeitsbereich nicht remanent	W000 bis W509	W000 bis W510	W000 bis W509
Arbeitsbereich remanent	H000 bis H509	H000 bis H510	H000 bis H509
Systembereich	A000 bis A957	A000 bis A958	A448 bis A957
Zeitgeberbereich	T0000 bis T4093	T0000 bis T4094	T0000 bis T4093
Zählerbereich	C0000 bis C4093	C0000 bis C4094	C0000 bis C4093
DM-Bereich	D00000 bis D32765	D00000 bis D32766	D00000 bis D32765

Bereich	C	T	R
EM-Bereich ohne Bank	E00000 bis E32765	E00000 bis E32766	E00000 bis E32765
EM-Bereich mit Bank	En_00000 bis En_32765 (n = 0 bis C)	En_00000 bis En_32766 (n = 0 bis C)	En_00000 bis En_32765 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	Nur spezifizierte Werte	---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR005+(++) bis IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

CSUB(731) subtrahiert Zeitdaten (Worte T und T+1) von den Kalenderdaten (Worte C bis C+2) und gibt die resultierenden Kalenderdaten in R bis R+2 aus.

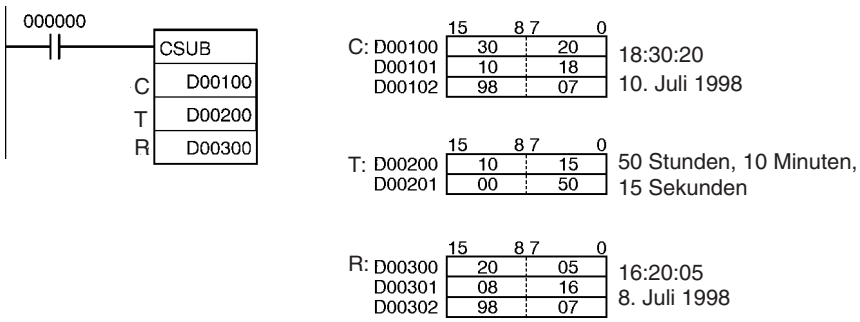


Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn die Kalenderdaten in C bis C+2 nicht innerhalb der spezifizierten Bereiche liegen. EIN, wenn die Zeitdaten T und T+1 nicht innerhalb der spezifizierten Bereiche liegen. In allen anderen Fällen AUS.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, werden die Zeitdaten in D00200 und D00201 (Stunden, Minuten, Sekunden) von den Kalenderdaten in D00100 bis D00102 (Jahr, Monat, Tag, Stunden, Minuten, Sekunden) subtrahiert und das Ergebnis in D00300 bis D00302 ausgegeben.

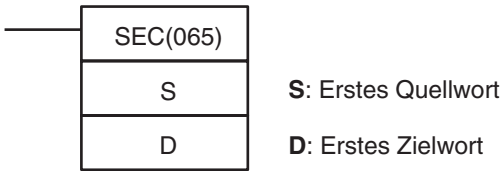


3-28-3 HOURS TO SECONDS: SEC(065)

Funktion

Konvertiert aus Stunden/Minuten/Sekunden bestehende Zeitdaten in Sekunden.

Kontaktplansymbol



S: Erstes Quellwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	SEC(065)
	Ausführung einmal bei steigender Flanke	@SEC(065)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

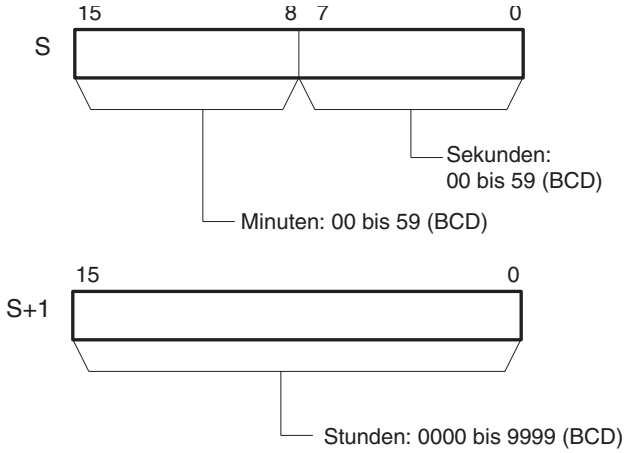
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

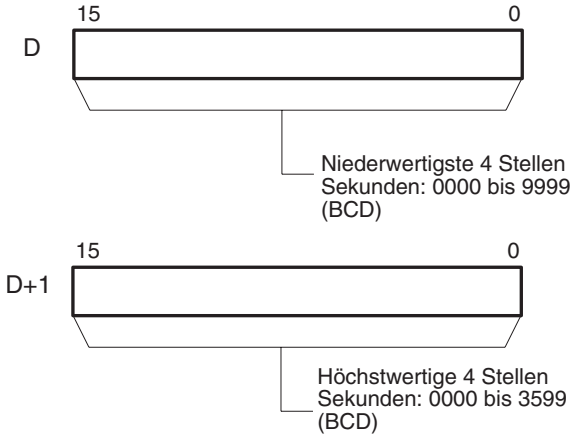
S und S+1: Quelldaten

Die Quelldaten in Stunden/Minuten/Sekunden in S und S+1 haben das in der nachfolgenden Abbildung gezeigte Format. S und S+1 müssen im gleichen Datenbereich liegen.



D und D+1: Ergebnisdaten

D und D+1 enthalten Ergebnisdaten ausschließlich im Sekundenformat. D und D+1 müssen im gleichen Datenbereich liegen.

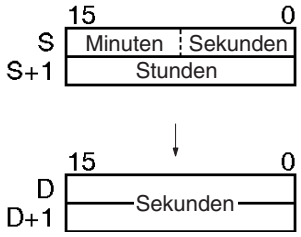


Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	Nur spezifizierte Werte	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

SEC(065) wandelt die im 8-stelligen BCD-Format vorliegenden Stunden/Minuten/Sekunden-Daten in S und S+1 in 8-stellige BCD-Sekundendaten um und gibt das Ergebnis in D und D+1 aus.



Flags

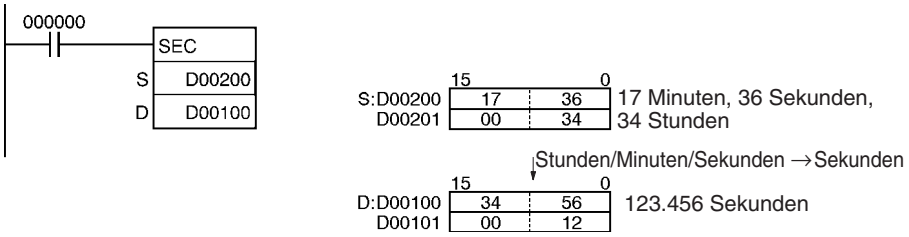
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn die Minutendaten in S (Bits 08 bis 15) nicht in BCD vorliegen und nicht im Bereich von 00 bis 59 liegen. EIN, wenn die Sekundendaten in S (Bits 00 bis 07) nicht in BCD vorliegen und nicht im Bereich von 00 bis 59 liegen. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn der Inhalt von D nach der Konvertierung den Wert 0000 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Maximalwert für die Quelldaten beträgt 9.999 Stunden, 59 Minuten und 59 Sekunden (35.999.999 Sekunden).

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, werden die Stunden/Minuten/Sekunden-Daten in D00200 und D00201 (34 Stunden, 17 Minuten und 36 Sekunden) in Sekundendaten konvertiert und das Ergebnis wird in D00100 und D00101 ausgegeben.

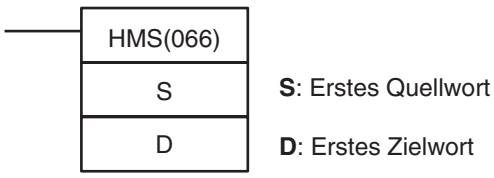


3-28-4 SECONDS TO HOURS: HMS(066)

Funktion

Wandelt Sekundendaten in die entsprechende Zeit in Stunden/Minuten/Sekunden um.

Kontaktplansymbol



S: Erstes Quellwort

D: Erstes Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	HMS(066)
	Ausführung einmal bei steigender Flanke	@HMS(066)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

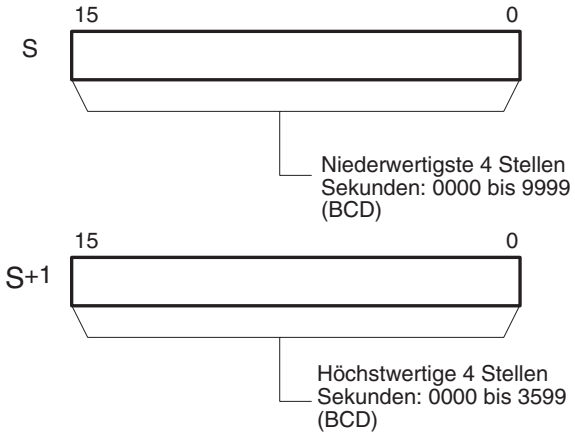
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

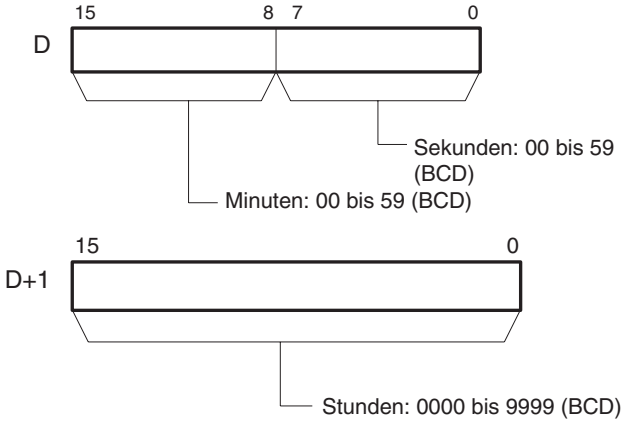
S und S+1: Quelldaten

Die Sekunden-Quelldaten in S und S+1 müssen wie in der nachfolgenden Abbildung ersichtlich eingestellt werden.
S und S+1 müssen im gleichen Datenbereich liegen.



D und D+1: Ergebnisdaten

D und D+1 enthalten die Ergebnisdaten im Stunden/Minuten/Sekunden-Format. D und D+1 müssen im gleichen Datenbereich liegen.

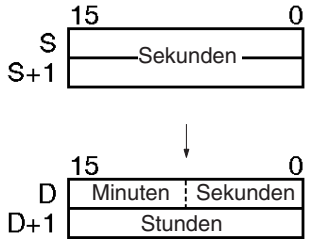


Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6142	
Arbeitsbereich nicht remanent	W000 bis W510	
Arbeitsbereich remanent	H000 bis H510	
Systembereich	A000 bis A958	A448 bis A958
Zeitgeberbereich	T0000 bis T4094	
Zählerbereich	C0000 bis C4094	
DM-Bereich	D00000 bis D32766	
EM-Bereich ohne Bank	E00000 bis E32766	
EM-Bereich mit Bank	En_00000 bis En_32766 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	

Bereich	S	D
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	00000000 bis 35999999 (BCD)	---
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

HMS(065) wandelt die als 8-stelliger BCD-Wert vorliegenden Sekunden-Daten in S und S+1 in 8-stellige BCD-Stunden/Minuten/Sekundendaten um und gibt das Ergebnis in D und D+1 aus.



Flags

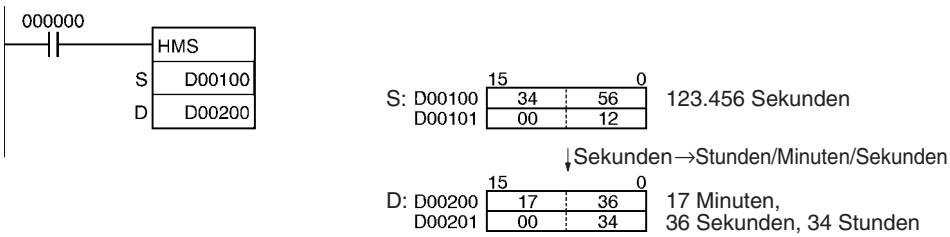
Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn die Sekundendaten in S und S+1 nicht in BCD vorliegen und nicht im Bereich von 0 bis 35.999.999 liegen. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn der Inhalt von D nach der Konvertierung den Wert 0000 hat. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Maximalwert für die Quelldaten beträgt 35.999.999 Sekunden (9.999 Stunden, 59 Minuten und 59 Sekunden).

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, werden die Sekundendaten in D00100 und D00101 (123.456 Sekunden) in Stunden/Minuten/Sekundendaten umgewandelt und das Ergebnis wird in D00200 und D00201 ausgegeben.



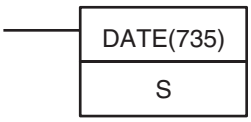
3-28-5 CLOCK ADJUSTMENT: DATE(735)

Funktion

Ändert die Einstellung der internen Uhr auf die Einstellung in den spezifizierten Quellworten.

Hinweis Die Einstellung der internen Uhr kann auch mit Hilfe eines Peripheriegeräts oder dem FINS-Befehl CLOCK WRITE (0702) geändert werden.

Kontaktplansymbol



S: Erstes Quellwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	DATE(735)
	Ausführung einmal bei steigender Flanke	@DATE(735)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

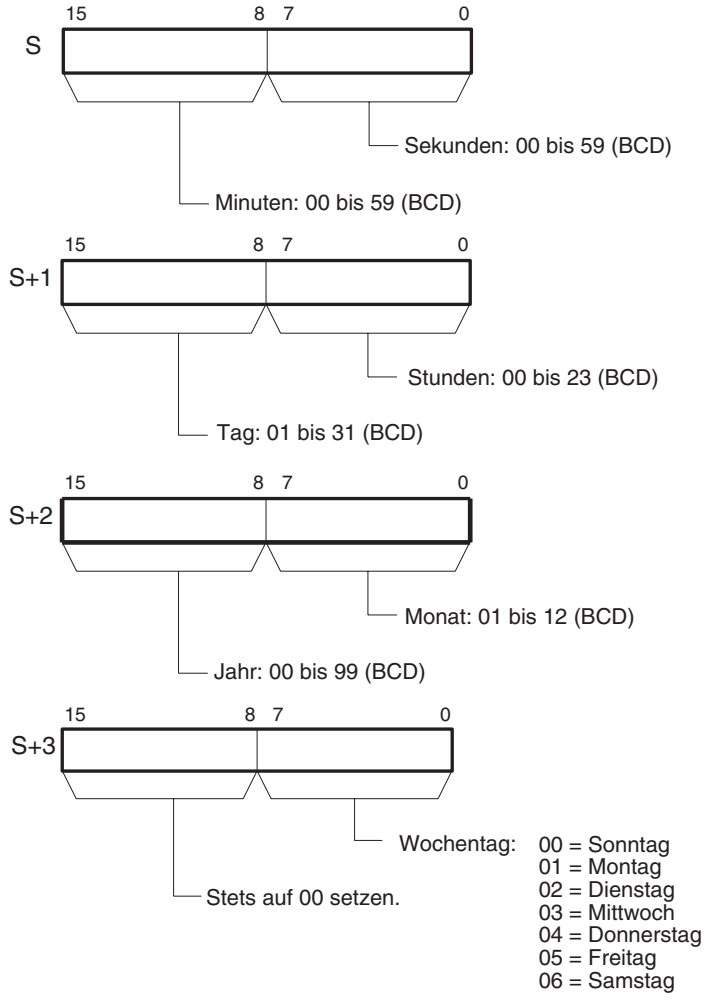
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S bis S+3: Neue Uhreinstellung

Einstellung der neuen Uhrzeit in S bis S+3, wie in der nachfolgenden Abbildung dargestellt.

S bis S+3 müssen im gleichen Datenbereich liegen.



In der folgenden Tabelle ist die Struktur des Kalender-/Uhrbereichs dargestellt.

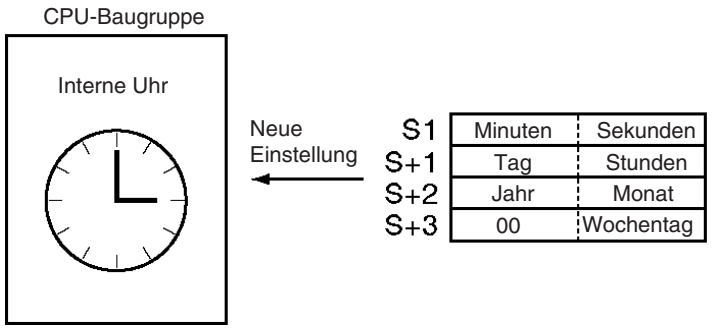
Adressen	Inhalt
A35100 bis A35107	Sekunden (00 bis 59, BCD)
A35108 bis A35115	Minuten (00 bis 59, BCD)
A35200 bis A35207	Stunden (00 bis 23, BCD)
A35208 bis A35215	Tag (01 bis 31, BCD)
A35300 bis A35307	Monat (01 bis 12, BCD)
A35308 bis A35315	Jahr (00 bis 99, BCD)
A35400 bis A35407	Wochentag (00 bis 06 = Sonntag bis Samstag, hexadezimal)
A35408 bis A35415	Stets auf 00 setzen.

Operanden-Spezifikationen

Bereich	S
CIO-Bereich	CIO 0000 bis CIO 6140
Arbeitsbereich nicht remanent	W000 bis W508
Arbeitsbereich remanent	H000 bis H508
Systembereich	A000 bis A956
Zeitgeberbereich	T0000 bis T4092
Zählerbereich	C0000 bis C4092
DM-Bereich	D00000 bis D32764
EM-Bereich ohne Bank	E00000 bis E32764
EM-Bereich mit Bank	En_00000 bis En_32764 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

DATE(735) ändert die Einstellung der internen Uhr auf die Uhrdaten in den vier Quellworten. Die Neueinstellung der internen Uhr spiegelt sich sofort im Kalender-/Uhrbereich (A351 bis A354) wider.



Flags

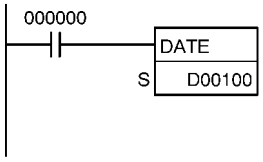
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn sich die neue Uhreinstellung in S bis S+3 nicht innerhalb des spezifizierten Bereichs befindet. In allen anderen Fällen AUS.

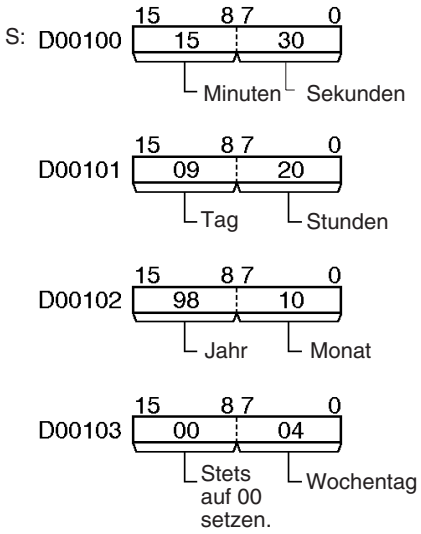
Vorsichtsmaßnahmen

Es wird kein Fehler generiert, auch wenn die interne Uhr auf ein nicht existentes Datum (wie z.B. den 31. November) eingestellt wird.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, wird die interne Uhr auf 20:15:30 am Donnerstag, 9. Oktober 1998 eingestellt.





3-29 Fehlersuchebefehle

3-29-1 TRACE MEMORY SAMPLING: TRSM(045)

Funktion

Wenn TRSM(045) ausgeführt wird, wird der Status eines vorausgewählten Bits oder Wortes erfasst und im Aufzeichnungsspeicher gespeichert. TRSM(045) kann überall im Programm beliebig oft verwendet werden.

Kontaktplansymbol



Variationen

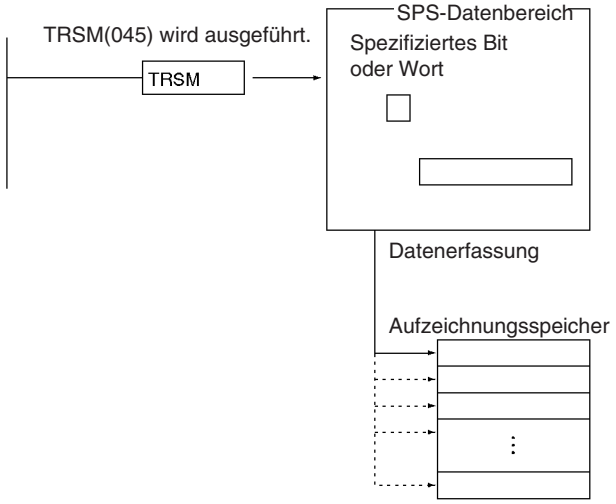
Variationen		
Variationen	**Ausführung bei jedem Zyklus**	TRSM(045)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Beschreibung

Vor der Ausführung von TRSM(045) muss das aufzuzeichnende Bit oder Wort mit einem Peripheriegerät spezifiziert werden. Bei jeder Ausführung von TRSM(045) wird der aktuelle Wert des spezifizierten Bits oder Worts erfasst und im Aufzeichnungsspeicher gespeichert. Die Erfassung wird beendet, wenn der Aufzeichnungsspeicher voll ist. Der Inhalt des Aufzeichnungsspeicher kann ggf. über ein Peripheriegerät überwacht werden.



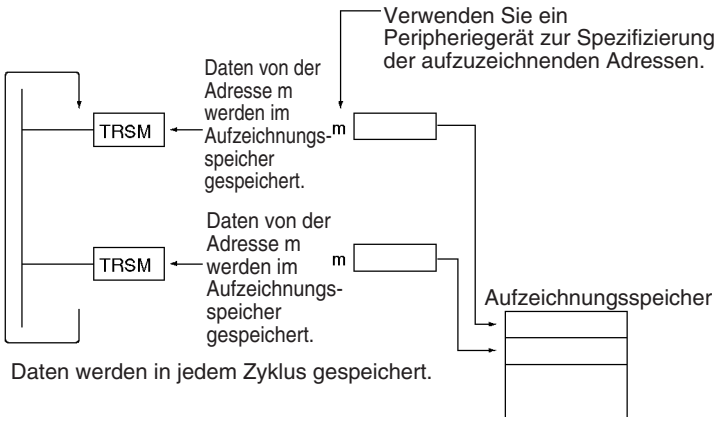
Die Erfassung wird beendet, wenn der Aufzeichnungsspeicher voll ist.

Dieser Befehl bestimmt nur den Zeitpunkt, an dem die spezifizierten Daten erfasst werden. Alle anderen Einstellungen und Datenaufzeichnungsfunktionen werden mit Hilfe eines Peripheriegeräts eingestellt. Die beiden anderen Wege zur Steuerung der Datenaufzeichnung sind die Erfassung am Ende jedes Zyklus und die Erfassung in einem festgelegten Intervall (unabhängig von der Zykluszeit).

TRSM(045) benötigt keine Ausführungsbedingung und wird stets so ausgeführt, als würde eine EIN-Ausführungsbedingung vorliegen. Verbinden Sie TRSM(045) direkt mit der linken Stromschiene.

Verwenden Sie den Befehl TRSM(045) zur Erfassung des Werts des spezifizierten Bits oder Worts an dem Punkt im Programm, an dem die Ausführungsbedingung des Befehls auf EIN gesetzt ist. Wenn die Bedingung zur Befehlsausführung in jedem Zyklus EIN ist, wird der Wert des spezifizierten Bits oder Worts in jedem Zyklus im Aufzeichnungsspeicher gespeichert.

Es können zwei oder mehr TRSM(045)-Befehle in einem Programm verwendet werden. In diesem Fall wird der Wert des gleichen spezifizierten Bits bzw. Worts bei jeder Ausführung eines der TRSM(045)-Befehle im Aufzeichnungsspeicher gespeichert.



Daten werden in jedem Zyklus gespeichert.

Hinweis Weitere Informationen über die Datenaufzeichnung finden Sie im Bedienerhandbuch des Peripheriegeräts.

Die mit dem Peripheriegerät durchgeführten Datenaufzeichnungsfunktionen sind in der folgenden Liste zusammengestellt.

1,2,3... 1. Stellen Sie die folgenden Parameter mit dem Peripheriegerät ein.
   a) Stellen Sie die Adresse des aufzuzeichnenden Bits bzw. Worts ein.
   b) Legen Sie die Triggerbedingung fest. Eine der drei folgenden Bedingungen kann steuern, wann die im Aufzeichnungsspeicher gespeicherten Daten gültig sind.
      i) Das Aufzeichnungsstartbit wechselt von AUS zu EIN.
      ii) Ein spezifiziertes Bit wechselt von AUS zu EIN.
      iii) Der Wert eines spezifizierten Worts entspricht dem Sollwert.
   c) Stellen Sie das Erfassungsintervall für die Erfassung bei Ausführung von TRSM(045) im Programm auf „TRSM“ ein.
   d) Stellen Sie die Verzögerung ein.
2. Wenn das Erfassungsstartbit mit dem Peripheriegerät von AUS auf EIN gesetzt wird, erfolgt ab diesem Zeitpunkt bei jeder Ausführung von TRSM(045) eine Erfassung der spezifizierten Daten und die erfassten Daten werden im Aufzeichnungsspeicher gespeichert. Das Aufzeichnung-Beschäftigt-Flag (A50813) wird gleichzeitig auf EIN gesetzt.
3. Wenn die Triggerbedingung (Aufzeichnungsstartbit EIN, spezifiziertes Bit EIN oder Wert des spezifizierten Worts entspricht dem Sollwert) erfüllt ist, werden die erfassten Daten bei der nächsten Erfassung, plus oder minus der für die Verzögerung eingestellten Anzahl von Erfassungen, gültig. Das Aufzeichnungstrigger-Überwachung-Flag (A50813) wird gleichzeitig auf EIN gesetzt.
4. Die Aufzeichnung wird beendet, wenn TRSM(045) so oft ausgeführt wurde, dass der Aufzeichnungsspeicher voll ist. Wenn die Aufzeichnung endet, wird das Aufzeichnung-Abgeschlossen-Flag (A50812) auf EIN und das Aufzeichnung-Beschäftigt-Flag (A50813) auf AUS gesetzt.
5. Lesen Sie den Inhalt des Aufzeichnungsspeichers mit Hilfe des Peripheriegeräts aus.

In der folgenden Tabelle sind die relevanten Bits und Flags im Systembereich aufgeführt. Nur A50814 und A50815 können vom Anwender gesteuert werden. A50815 darf nicht durch das Programm auf EIN gesetzt werden, sondern nur über ein Peripheriegerät.

Name	Adresse	Funktion
Aufzeichnung-strigger-Überwachung-Flag	A50811	Dieses Flag wird auf EIN gesetzt, wenn die Triggerbedingung mit dem Aufzeichnungsstartbit erfüllt wurde. Es wird auf AUS gesetzt, wenn die Erfassung für die nächste Aufzeichnung (durch das Erfassungsstartbit) gestartet wird.
Aufzeichnung-abgeschlossen-Flag	A50812	Dieses Flag wird auf EIN gesetzt, wenn der Aufzeichnungsspeicher mit aufgezeichneten Daten gefüllt ist. Es wird auf AUS gesetzt, wenn der Zustand des Erfassungsstartbits das nächste Mal von AUS auf EIN wechselt.
Aufzeichnung-beschäftigt-Flag	A50813	Dieses Flag wird auf EIN gesetzt, wenn das Erfassungsstartbit von AUS zu EIN wechselt. Es wird auf AUS gesetzt, wenn die Aufzeichnung abgeschlossen ist.

Name	Adresse	Funktion
Aufzeichnungs-startbit	A50814	Die Aufzeichnungstrigger-Bedingungen sind erfüllt, wenn der Status dieses Bits von AUS zu EIN wechselt. Messwerte werden nach der spezifizierten Verzögerung aufgezeichnet (positive Verzögerung) oder die spezifizierte Anzahl vorhandener Messwerte ist gültig (negative Verzögerung).
Erfassungs-startbit	A50815	Wenn dieses Bit über ein Peripheriegerät von AUS auf EIN gesetzt wird, beginnt die Speicherung von Erfassungsdaten im Aufzeichnungsspeicher mit Hilfe einer der drei folgenden Methoden, die zur Bestimmung der Erfassung verwendet werden: 1) Periodische Erfassung (alle 10 bis 2.550 ms) 2) Erfassung bei Ausführung des Befehls TRSM(045) 3) Erfassung am Ende jedes Zyklus Dieses Bit muss durch ein Peripheriegerät auf EIN und AUS gesetzt werden.

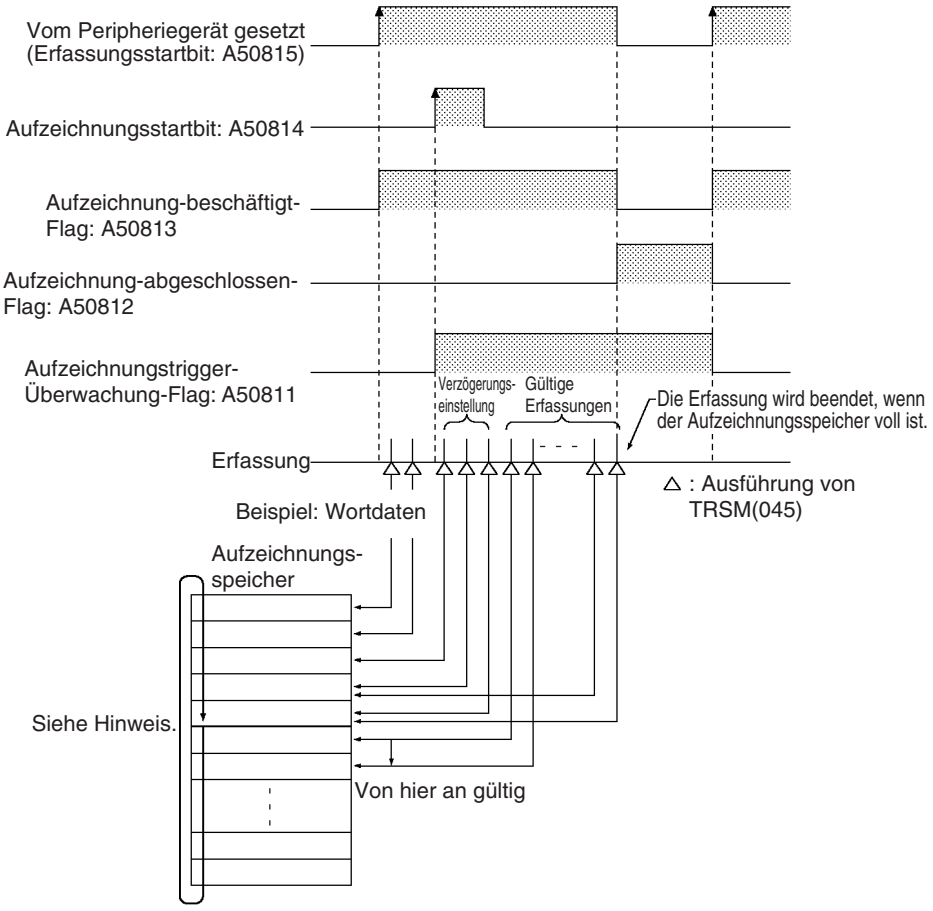
Vorsichtsmaßnahmen

TRSM(045) wird als NOP(000) verarbeitet, wenn keine Datenaufzeichnung stattfindet oder wenn das mit einem Peripheriegerät in den Parametern festgelegte Erfassungsintervall nicht auf die Erfassung bei Ausführung des Befehls TRSM(045) eingestellt ist.

Das Erfassungsstartbit (A50815) darf nicht vom Programm aus auf EIN oder AUS gesetzt werden. Dieses Bit muss durch ein Peripheriegerät auf EIN und AUS gesetzt werden.

Beispiel

Das folgende Beispiel zeigt den gesamten Datenaufzeichnungsvorgang.



Hinweis Der Aufzeichnungsspeicher ist ein Ringspeicher. Die Daten werden bis zum Ende des Aufzeichnungsspeicherbereichs gespeichert. Danach wird die Speicherung am Anfang des Bereichs fortgesetzt und endet unmittelbar vor der ersten gültigen Datenerfassung.

3-30 Fehlerdiagnosebefehle

In diesem Abschnitt werden die Befehle zur Ermittlung und Behandlung von Fehlern beschrieben.

Befehl	AWL	Funktionscode	Seite
FAILURE ALARM	FAL	006	1079
SEVERE FAILURE ALARM	FALS	007	1087
FAILURE POINT DETECTION	FPD	269	1095

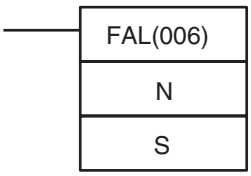
3-30-1 FAILURE ALARM: FAL(006)

Funktion

Erzeugt oder löscht anwenderdefinierte nicht schwerwiegende Fehler. Durch nicht schwerwiegende Fehler wird der SPS-Betrieb nicht angehalten.
Bei CS1-H, CJ1-H und CJ1M CPU-Baugruppen kann FAL(006) auch zum Erzeugen von nicht schwerwiegenden Systemfehlern verwendet werden.

Kontaktplansymbol

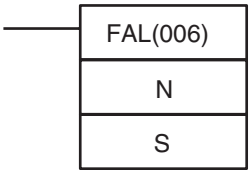
- Erzeugen oder Löschen von anwenderdefinierten, nicht schwerwiegenden Fehlern



N: FAL-Nummer

S: Erstes Meldungswort oder Konstante (0000 bis FFFF)

- Erzeugen von nicht schwerwiegenden Systemfehlern (nur CS1-H, CJ1-H, CJ1M und CS1D)



N: FAL-Nummer (Wert in A529)

S: Erstes der Worte, die Fehlercode und Fehlerdetails

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FAL(006)
	Ausführung einmal bei steigender Flanke	@FAL(006)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Die Funktion der Operanden bei Verwendung von FAL(006) zum Erzeugen/Löschen von anwenderdefinierten Fehlern unterscheidet sich leicht von der Funktion, wenn FAL(006) zum Erzeugen von Systemfehlern (nur CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen) verwendet wird.

Erzeugen oder Löschen von anwenderdefinierten, nicht schwerwiegenden Fehlern

Die folgende Tabelle zeigt die Funktion der Operanden.

Hinweis Der Wert von Operand N muss **verschieden** sein vom Inhalt von A529 (die vom System generierte FAL/FALS-Nummer).

N	S	Funktion
0	#0001 bis #01FF	Löscht den nicht schwerwiegenden Fehler mit der entsprechenden FAL-Nummer.
	#FFFF	Löscht alle nicht schwerwiegenden Fehler.
	Andere Einstellung*	Löscht den wichtigsten der nicht schwerwiegenden Fehler.
1 bis 511 (Diese FAL-Nummern werden mit FALS-Nummern gemeinsam genutzt).	#0000 bis #FFFF	Erzeugt einen nicht schwerwiegenden Fehler mit der entsprechenden FAL-Nummer (keine Meldung).
	Wortadresse	Erzeugt einen nicht schwerwiegenden Fehler mit der entsprechenden FAL-Nummer. Die in S bis S+7 enthaltenen 16 Zeichen umfassende ASCII-Meldung wird auf dem Programmiergerät angezeigt.

Hinweis *Andere Einstellungen wären die Konstanten #0200 bis #FFFE oder eine Wortadresse.

Erzeugen von nicht schwerwiegenden Systemfehlern (nur CS1-H, CJ1-H, CJ1M und CS1D)

In der folgenden Tabelle sind die Funktionen der Operanden beschrieben.

Hinweis Der Wert von Operand N muss **identisch mit** dem Inhalt von A529 (die vom System generierte FAL/FALS-Nummer) sein.

Operand	Funktion
N	1 bis 511 (Diese FAL-Nummern werden mit FALS-Nummern gemeinsam genutzt).
S	Generierter Fehlercode. (Siehe *Beschreibung* weiter unten.)
S+1	Generierter Fehlerdetailcode. (Siehe *Beschreibung* weiter unten).

Operanden-Spezifikationen

Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A959
Zeitgeberbereich	---	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	0 bis 511	#0000 bis #FFFF (binär)
Datenregister	---	

Bereich	N	S
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Die Funktion von FAL(006) hängt vom Wert von N ab. Stellen Sie N auf 0000 ein, um einen Fehler zu löschen und auf 0001 bis 01FF, um einen Fehler zu erzeugen. Wenn der Wert von N gleich dem Inhalt von A529 (nur CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen) ist, wird ein Systemfehler erzeugt.

Erzeugen von nicht schwerwiegenden, anwenderdefinierten Fehlern

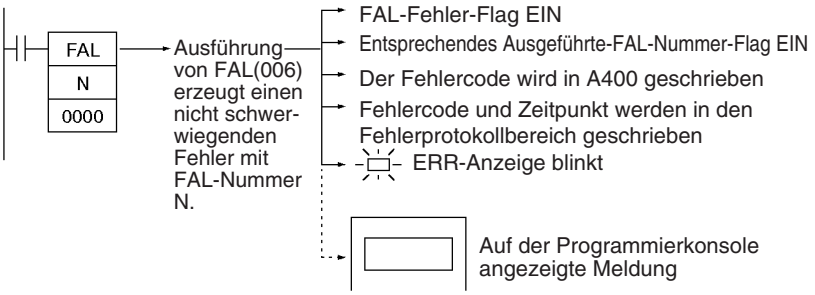
Wenn FAL(006) ausgeführt wird und N auf eine FAL-Nummer (&1 bis &511) eingestellt ist, die nicht gleich dem Inhalt von A529 (die vom System generierte FAL/FALS-Nummer) ist, wird ein nicht schwerwiegender Fehler mit dieser FAL-Nummer generiert und es erfolgt die folgende Verarbeitung:

1,2,3...

1. Das FAL-Fehler-Flag (A40215) wird auf EIN gesetzt. (Der Betrieb der SPS wird fortgesetzt.)
2. Das Ausgeführte-FAL-Nummer-Flag für die entsprechende FAL-Nummer wird auf EIN gesetzt. Die Flags A36001 bis A39115 entsprechen den FAL-Nummern 0001 bis 01FF (1 bis 511).
3. Der Fehlercode wird in A400 geschrieben. Die Fehlercodes 4101 bis 42FF entsprechen den FAL-Nummern 0001 bis 01FF (1 bis 511).

   Hinweis Wenn ein schwerwiegender Fehler oder ein ernsterer, aber nicht schwerwiegender Fehler gleichzeitig mit der Ausführung des FAL(006)-Befehls auftritt, wird der Fehlercode des ernsteren Fehlers in A400 geschrieben.
4. Der Fehlercode und der Zeitpunkt des Auftretens des Fehlers werden in den Fehlerprotokollbereich (A100 bis A199) geschrieben.

   Hinweis Bei CS1-H, CJ1-H und CJ1M CPU-Baugruppen wird die Fehleraufzeichnung nicht in den Fehlerprotokollbereich geschrieben, wenn das SPS-Setup darauf eingestellt wurde, dass von FAL(006) generierte Fehler nicht aufgezeichnet werden, d. h., wenn die Programmierkonsolen-Adresse 129 Bit 15 auf 1 gesetzt ist.
5. Die ERR-Anzeige auf der CPU-Baugruppe blinkt.
6. Wenn in S eine Wortadresse spezifiziert wurde, wird die mit S beginnende Meldung registriert (auf dem Programmiergerät angezeigt).



In der folgenden Tabelle sind die Fehlercodes und FAL-Fehler-Flags für FAL(006) aufgeführt.

FAL-Nummer	FAL-Fehlercodes	Ausgeführte-FAL-Nummer-Flags
1 bis 511 (dezimal)	4101 bis 42FF	A36001 bis A39115

Anzeigen von Meldungen bei nicht schwerwiegenden, anwenderdefinierten Fehlern

Wenn S eine Wortadresse ist und in S eine ASCII-Meldung gespeichert wurde, wird diese Meldung bei Ausführung von FAL(006) auf dem Peripheriegerät angezeigt. (Wenn eine Meldung nicht erforderlich ist, setzen Sie S auf eine Konstante.)

Die mit S beginnende Meldung wird bei Ausführung von FAL(006) registriert. Nach Registrierung der Meldung wird diese bei Anschluss einer Programmierkonsole auf dieser angezeigt.

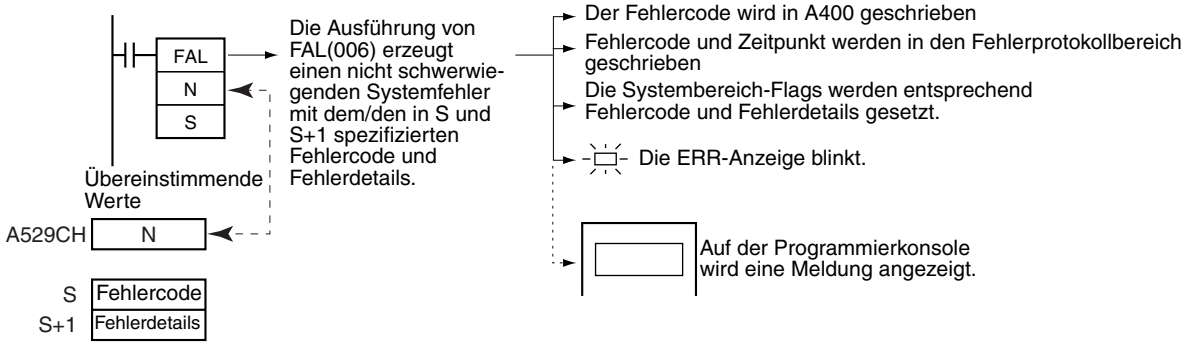
In S bis S+7 kann eine ASCII-Meldung mit einer Länge von bis zu 16 Zeichen gespeichert werden. Das höchstwertige Byte jedes Worts wird zuerst angezeigt.

Der Endcode für die Meldung ist das Nullzeichen (00 hexadezimal). Beim Auslassen des Nullzeichens werden alle 16 Zeichen in den Worten S bis S+7 angezeigt.

Wenn der Inhalt der Worte, die die Meldung enthalten, nach Ausführung von FAL(006) geändert wird, wird die Meldung entsprechend geändert.

Erzeugen von nicht schwerwiegenden Systemfehlern (nur CS1-H, CJ1-H, CJ1M und CS1D)

Wenn FAL(006) ausgeführt wird und N auf eine FAL-Nummer (&1 bis &511) eingestellt ist, die nicht gleich dem Inhalt von A529 (die vom System generierte FAL/FALS-Nummer) ist, wird ein nicht schwerwiegender Fehler mit dem in S und S+1 spezifizierten Fehlercode und Fehlerdetailcode generiert. Gleichzeitig erfolgt die folgende Verarbeitung:



1,2,3...

1. Der spezifizierte Fehlercode wird in A400 geschrieben.
2. Der Fehlercode und der Zeitpunkt des Auftretens des Fehlers werden in den Fehlerprotokollbereich (A100 bis A199) geschrieben.
3. Die entsprechenden Systembereich-Flags werden entsprechend Fehlercode und Fehlerdetails gesetzt.
4. Die ERR-Anzeige auf der CPU-Baugruppe blinkt und der SPS-Betrieb wird fortgesetzt.
5. Die Meldung für einen nicht schwerwiegenden Systemfehlers wird auf der Programmierkonsole angezeigt.

Hinweis

1. FAL(006) kann verwendet werden, um bei der Fehlersuche im Programm nicht schwerwiegende Fehler vom System erzeugen zu lassen. Beispielsweise kann ein Systemfehler absichtlich erzeugt werden, um zu prüfen, ob Fehlermeldungen auf einer Benutzeroberfläche, wie z.B. einem programmierbaren Bedienterminal, richtig angezeigt werden.
2. Der Wert von A529 (die vom System generierte FAL/FALS-Nummer) ist eine Dummy-FAL-Nummer (FAL-, FALS- und FPD-Nummern werden gemeinsam genutzt), die bei der absichtlichen Erzeugung eines nicht schwerwiegenden Fehlers durch das System verwendet wird. Da diese Nummer eine Dummy-FAL-Nummer ist, ändert sie den Status der Ausgeführte-FAL-Nummer-Flags (A36001 bis A39115) oder den Fehlercode nicht.
   Wenn die Erzeugung von zwei oder mehr Systemfehlern (schwerwiegende und/oder nicht schwerwiegende Fehler) erforderlich ist, können durch mehrmalige Ausführung der FAL/FALS/FPD-Befehle mit denselben Werten in A529 und N, aber unterschiedlichen Werten in S und S+1, verschiedene Fehler generiert werden.

3. Wenn ein ernster Fehler (einschließlich eines vom System generierten schwerwiegenden Fehlers oder FALS(007)-Fehlers) gleichzeitig mit der Ausführung des FAL(006)-Befehl auftritt, wird der Fehlercode des ernsteren Fehlers in A400 geschrieben.
4. Um einen durch FAL(006) erzeugten Systemfehler zu löschen, muss die SPS aus- und wieder eingeschaltet werden. Die SPS kann eingeschaltet bleiben, aber zum Löschen des Fehlers ist die gleiche Verarbeitung erforderlich, als wenn der spezifizierte Fehler tatsächlich aufgetreten wäre.

In der folgenden Tabelle ist dargestellt, wie Fehlercodes und Fehlerdetails in S und S+1 spezifiziert werden.

Fehlerbezeichnung	S	S+1
Interrupt-Task-Fehler	008B hex	• Bit 15 AUS: Interrupt-Task-Fehler Bits 00 bis 14: Task-Nummer der Interrupt-Task, in der der Fehler aufgetreten ist. • Bit 15 EIN: Konflikt bei der Ausführung der Interrupt-Task mit der Aktualisierung einer Spezial-E/A-Baugruppe Bits 00 bis 14: Baugruppennummer der Spezial-E/A-Baugruppe mit Aktualisierungskonflikt
E/A-Baugruppenfehler	009A hex	Baugruppenträger-Position der Baugruppe, in der der Fehler aufgetreten ist • Bits 08 bis 15: Baugruppenträger-Nummer (binär) des Baugruppenträgers, in dem die betroffene Baugruppe montiert ist • Bits 00 bis 07: Nummer des Steckplatzes (binär), in dem die betroffene Baugruppe montiert ist
SPS-Konfigurationsfehler	009B hex	Art des SPS-Konfigurationsfehlers
E/A-Tabellen-Verifizierungsfehler	00E7 hex	--- (nicht festgelegt)
Nicht schwerwiegender Fehler in internem Sondermodul	02F0 hex	Fehlerinformationen zum internen Sondermodul • Bits 00 bis 03: Ungültig • Bits 04 bis 15: Durch internes Sondermodul definierter Fehler
CS1 CPU-Bus-Baugruppenfehler	0200 hex	Baugruppenadresse der CS1 CPU-Bus-Baugruppe: 0000 bis 000F hex
Spezial-E/A-Baugruppenfehler	0300 hex	Baugruppennummer der Spezial-E/A-Baugruppe: 0000 bis 005F hex oder 00FF hex (Baugruppennummer unbestimmt)
SYSMAC BUS-Fehler	00A0 hex	Baugruppennummer der SYSMAC BUS-Master-Baugruppe: 0000 oder 0001 hex
Batteriefehler	00F7 hex	--- (nicht festgelegt)
CS1 CPU-Bus-Baugruppen-Konfigurationsfehler	0400 hex	Baugruppennummer der CS1 CPU-Bus-Baugruppe: 0000 bis 000F hex
Spezial-E/A-Baugruppen-Konfigurationsfehler	0500 hex	Baugruppennummer der Spezial-E/A-Baugruppe: 0000 bis 005F hex

Deaktivieren von Fehlerprotokolleinträgen für anwenderdefinierte Fehler (nur CS1-H, CJ1-H, CJ1M oder CS1D)

Wenn FAL(006) einen anwenderdefinierten Fehler erzeugt, werden Fehlercode und Zeitpunkt des Auftretens normalerweise in den Fehlerprotokollbereich (A100 bis A199) geschrieben. Das SPS-Setup kann jedoch so eingestellt werden, dass durch FAL(006) erzeugte anwenderdefinierte Fehler nicht im Fehlerprotokoll gespeichert werden.

Auch wenn der Fehler nicht im Fehlerprotokoll gespeichert wird, wird das FAL-Fehler-Flag (40215) auf EIN gesetzt, das entsprechende Flag in den Ausgeführte-FAL-Nummer-Flags (A36001 bis A39115) auf EIN gesetzt und der Fehlercode in A400 geschrieben.

Deaktivieren Sie die Fehlerprotokolleinträge für anwenderdefinierte FAL(006)-Fehler, wenn nur die vom System erzeugten Fehler aufgezeichnet werden sollen. Diese Funktion ist z.B. während der Fehlersuche hilfreich, wenn FAL(006)-Befehle in verschiedenen Anwendungen verwendet werden und

sich das Fehlerprotokoll mit anwenderdefinierten FAL(006)-Fehlern füllt. In der folgenden Tabelle ist die Einstellung des SPS-Setups dargestellt:

Parameter	Einstellung	
Einstellungsadresse der Programmierkonsole	Wort	129
	Bit	15
Name	FAL-Fehlerprotokoll-Registrierung	
Einstellungen	0: FAL-Fehler im Fehlerprotokoll aufzeichnen. 1: FAL-Fehler nicht im Fehlerprotokoll aufzeichnen.	
Standardeinstellung	0: FAL-Fehler im Fehlerprotokoll aufzeichnen.	
Zeitpunkt, an dem die SPS-Setup-Einstellung gelesen wird	Jeden Zyklus (bei Auftreten eines FAL-Fehlers)	

Auch wenn das SPS-Setup-Wort 129 Bit 15 auf 1 gesetzt ist (FAL-Fehler nicht im Fehlerprotokoll aufzeichnen), werden die folgenden Fehler aufgezeichnet:

- Durch FALS(007) erzeugte, schwerwiegende Fehler
- Nicht schwerwiegende Systemfehler
- Schwerwiegende Systemfehler
- Nicht schwerwiegende Systemfehler, die absichtlich mit FAL(006) oder FPD(269) erzeugt wurden
- Schwerwiegende Systemfehler, die absichtlich durch FALS(007) erzeugt wurden

Löschen von nicht schwerwiegenden Fehlern ohne Programmiergerät

1. Löschen von anwenderdefinierten, nicht schwerwiegenden Fehlern
   Wenn FAL(006) ausgeführt wird und N auf 0 gesetzt ist, können nicht schwerwiegende Fehler gelöscht werden. Der Wert von S bestimmt die Verarbeitung, wie in der folgenden Tabelle ersichtlich.

S	Prozess
&1 bis &511 (0001 bis 01FF hex)	Der FAL-Fehler mit der spezifizierten Nummer wird gelöscht.
FFFF hex	Alle nicht schwerwiegenden Fehler (einschließlich Systemfehler) werden gelöscht.
0200 bis FFFE hex oder Spezifikation eines Worts	Der ernsteste, nicht schwerwiegendste Fehler (auch wenn es sich um einen nicht schwerwiegenden Systemfehler handelt), der aufgetreten ist, wird gelöscht. Wenn mehr als ein FAL-Fehler aufgetreten ist, wird der FAL-Fehler mit der kleinsten FAL-Nummer gelöscht.

2. Löschen von nicht schwerwiegenden Systemfehlern (nur CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen)
   Es gibt zwei Wege, nicht schwerwiegende Systemfehler zu löschen, die mit FAL(006) erzeugt wurden.
   - Schalten Sie die Versorgungsspannung der SPS aus und wieder ein.
   - Soll die SPS eingeschaltet bleiben, muss der Systemfehler gelöscht werden, als ob der spezifizierte Fehler tatsächlich aufgetreten wäre.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N nicht im spezifizierten Bereich von 0 bis 511 dezimal liegt. EIN, wenn ein nicht schwerwiegender Systemfehler erzeugt wird (nur CS1-H/CJ1-H/CJ1M/CS1D), aber der spezifizierte Fehlercode oder Fehlerdetailcode falsch ist. In allen anderen Fällen AUS.

In den nachfolgende Tabellen sind die relevanten Worte und Flags im Systembereich ausgeführt.

- Systembereichs-Worte/-Flags nur für anwenderdefinierte Fehler

Name	Adresse	Funktion
FAL-Fehler-Flag	A40215	EIN, wenn ein Fehler mit FAL(006) erzeugt wurde.
Ausgeführte-FAL-Nummer-Flags	A36001 bis A39115	Wenn ein Fehler mit FAL(006) erzeugt wurde, wird das entsprechende Flag auf EIN gesetzt. Die Flags A36001 bis A39115 entsprechen den FAL-Nummern 0001 bis 01FF.

- Systembereichs-Worte/-Flags nur für Systemfehler (nur CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen)

Name	Adresse	Funktion
Vom System erzeugte FAL/FALS-Nummer	A529	Wenn ein Systemfehler mit FAL(006) erzeugt wird, wird eine Dummy-FAL/FALS-Nummer verwendet. Dieselbe Dummy-FAL/FALS-Nummer (0001 bis 01FF hex, 1 bis 511 dezimal) wird in dieses Wort gesetzt.

- Systembereichs-Worte/-Flags für anwenderdefinierte Fehler und Systemfehler

Name	Adresse	Funktion
Fehler-protokollbereich	A100 bis A199	Der Fehlerprotokollbereich enthält Fehlercodes und Zeit/Datum der 20 zuletzt aufgetretenen Fehler, einschließlich durch FAL(006) erzeugter Fehler.
Fehlercode	A400	Beim Auftreten eines Fehlers wird der entsprechende Fehlercode in A400 gespeichert. Die Fehlercodes für die FAL-Nummern 0001 bis 01FF lauten 4101 bis 42FF. Wenn zwei oder mehr Fehler gleichzeitig auftreten, wird der Fehlercode des schwersten Fehlers in A400 gespeichert.

Vorsichtsmaßnahmen

N muss einen Wert zwischen 0000 und 01FF haben. Wenn sich N außerhalb des spezifizierten Bereichs befindet, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Beispiele

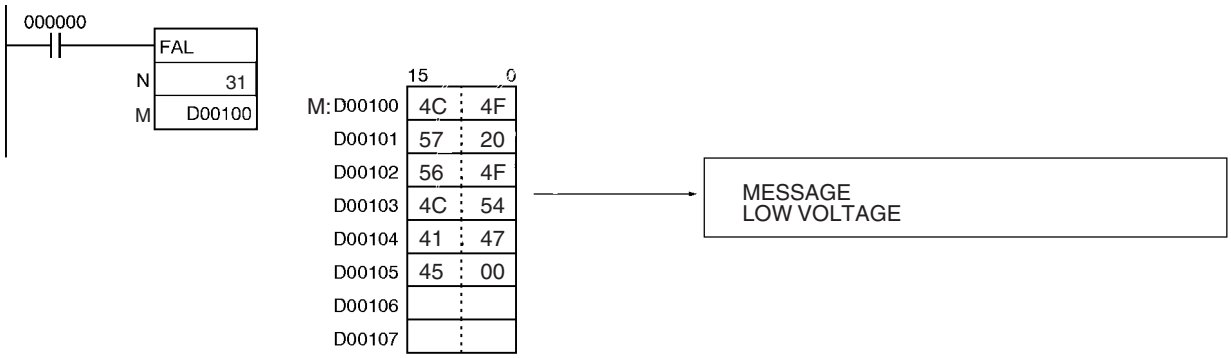
Erzeugen eines nicht schwerwiegenden Fehlers

Wenn CIO 000000 im folgenden Beispiel EIN ist, erzeugt FAL(006) einen nicht schwerwiegenden Fehler mit der FAL Nummer 31 und führt die folgenden Prozesse aus.

1,2,3...

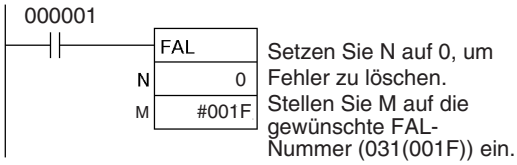
1. Das FAL-Fehler-Flag (A40215) wird auf EIN gesetzt.
2. Das entsprechende Ausgeführte-FAL-Nummer-Flag (A36114) wird auf EIN gesetzt.
3. Der entsprechende Fehlercode (411F) wird in A400 geschrieben.

   Hinweis Wenn zwei oder mehr Fehler gleichzeitig auftreten, wird der Fehlercode des schwersten Fehlers (mit dem höchsten Fehlercode) in A400 gespeichert.
4. Fehlercode und Uhrzeit/Datum des Auftretens des Fehlers werden in den Fehlerprotokollbereich (A100 bis A199) geschrieben.
5. Die ERR-Anzeige auf der CPU-Baugruppe blinkt.
6. Die ASCII-Meldung in D00100 bis D00107 wird auf dem Peripheriegerät angezeigt. (Wenn eine Meldung nicht erforderlich ist, spezifizieren Sie für S eine Konstante.)



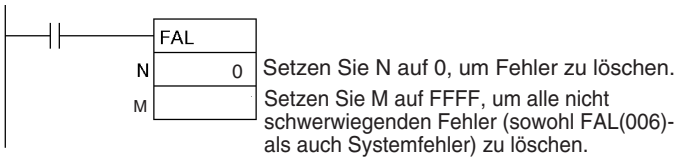
Löschen eines bestimmten, nicht schwerwiegenden Fehlers

Wenn CIO 000001 im folgenden Beispiel EIN ist, löscht FAL(006) den nicht schwerwiegenden Fehler mit der FAL Nummer 31 und setzt das entsprechende Ausgeführte-FAL-Nummer-Flag (A36114) und das FAL-Fehler-Flag (A40215) auf AUS.



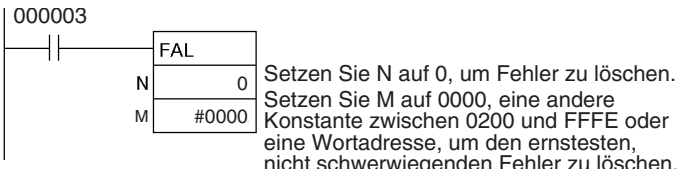
Löschen aller nicht schwerwiegenden Fehler

Wenn CIO 000002 im folgenden Beispiel EIN ist, löscht FAL(006) alle nicht schwerwiegenden Fehler und setzt die entsprechenden Ausgeführte-FAL-Nummer-Flags (A36001 bis A39115) und das FAL-Fehler-Flag (A40215) auf AUS.



Löschen des ernstesten, nicht schwerwiegenden Fehlers

Wenn CIO 000003 im folgenden Beispiel EIN ist, löscht FAL(006) den ernstesten, nicht schwerwiegenden Fehler, der aufgetreten ist, und setzt den Fehlercode in A400 zurück. Wenn der gelöschte Fehler ursprünglich durch FAL(006) erzeugt wurde, werden das entsprechende Ausgeführte-FAL-Nummer-Flag und das FAL-Fehler-Flag (A40215) auf AUS gesetzt.



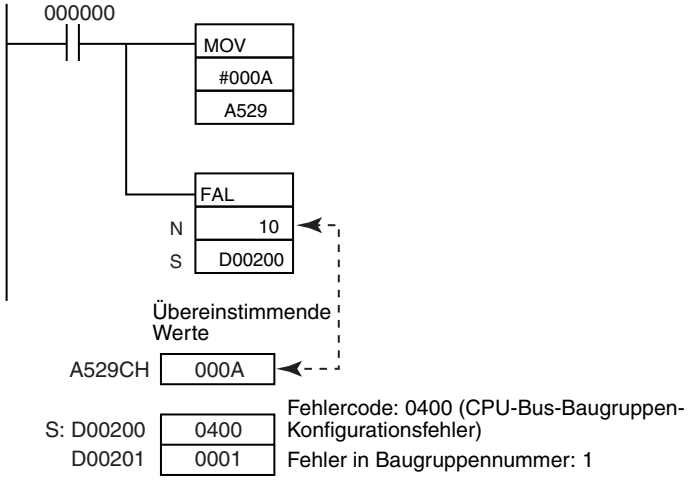
Erzeugen eines nicht schwerwiegenden Systemfehlers (nur CS1-H, CJ1-H, CJ1M und CS1D)

Wenn CIO 000000 im folgenden Beispiel EIN ist, erzeugt FAL(006) einen CPU Bus-Baugruppen-Konfigurationsfehler für Baugruppennummer 1. In diesem Fall wird die Dummy-FAL-Nummer 10 verwendet und der entsprechende Wert (000A hex) wird in A529 gespeichert.

1,2,3...

1. Der spezifizierte Fehlercode (0400) wird in A400 geschrieben, wenn es sich um den ernstesten Fehler handelt.
2. Fehlercode und Uhrzeit/Datum des Auftretens des Fehlers werden in den Fehlerprotokollbereich (A100 bis A199) geschrieben.

3. Das CPU-Bus-Baugruppen-Konfigurationsfehler-Flag (A40203) und das CPU-Bus-Baugruppen-Konfigurationsfehler-Flag für Baugruppennummer 1 (A42701) werden auf EIN gesetzt.
4. Die ERR-Anzeige der CPU-Baugruppe blinkt.
5. Auf der Programmierkonsole wird eine Meldung (CPU BU ST ERR 01) angezeigt, die auf einen Fehler in der CPU-Bus-Baugruppe 1 hinweist.



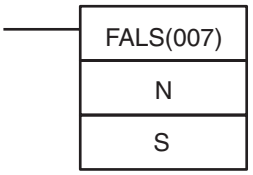
3-30-2 SEVERE FAILURE ALARM: FALS(007)

Funktion

Erzeugt schwerwiegende anwenderdefinierte Fehler. Durch schwerwiegende Fehler wird der SPS-Betrieb angehalten.
Bei CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen kann FAL(007) auch zum Erzeugen von schwerwiegenden Systemfehlern verwendet werden.

Kontaktplansymbol

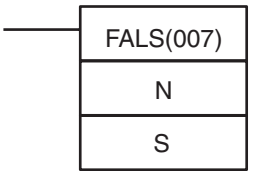
- Erzeugen von anwenderdefinierten schwerwiegenden Fehlern



N: FALS-Nummer

S: Erstes Meldungswort oder Konstante (0000 bis FFFF)

- Erzeugen von schwerwiegenden Systemfehlern (nur CS1-H, CJ1-H, CJ1M und CS1D)



N: FALS-Nummer (Wert in A529)

S: Erstes der Worte, die Fehlercode und Fehlerdetails enthalten

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FALS(007)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Erzeugen von anwenderdefinierten schwerwiegenden Fehlern

In der folgenden Tabelle sind die Funktionen der Operanden aufgeführt.

Hinweis Der Wert von Operand N muss **verschieden** sein vom Inhalt von A529 (die vom System generierte FAL/FALS-Nummer).

Operand	Funktion
N	1 bis 511 (Diese FALS-Nummern werden mit FAL-Nummern gemeinsam genutzt.)
S	Spezifiziert das erste von acht Worten, die eine ASCII-Meldung enthalten, die auf dem Programmiergerät angezeigt wird. Spezifizieren Sie eine Konstante (0000 bis FFFF), wenn keine Meldung erforderlich ist.

Erzeugen von schwerwiegenden Systemfehlern (nur CS1-H, CJ1-H, CJ1M und CS1D)

In der folgenden Tabelle sind die Funktionen der Operanden aufgeführt.

Hinweis Der Wert von Operand N muss **identisch mit** dem Inhalt von A529 (die vom System generierte FAL/FALS-Nummer) sein.

Operand	Funktion
N	1 bis 511 (Diese FALS-Nummern werden mit FAL-Nummern gemeinsam genutzt.)
S	Zu generierender Fehlercode. (Siehe *Beschreibung* weiter unten.)
S+1	Zu generierender Fehlerdetailcode. (Siehe *Beschreibung* weiter unten.)

Operanden-Spezifikationen

Bereich	N	S
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A959
Zeitgeberbereich	---	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	Nur spezifizierte Werte	#0000 bis #FFFF (binär)
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR+(++)0 bis IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

FALS(007) erzeugt einen schwerwiegenden Fehler. Bei CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen kann FAL(007) auch zum Erzeugen von schwerwiegenden Systemfehlern sowie von schwerwiegenden, anwenderdefinierten Fehlern verwendet werden. (Wenn der Wert von N gleich dem Inhalt von A529 ist, wird ein Systemfehler generiert.)

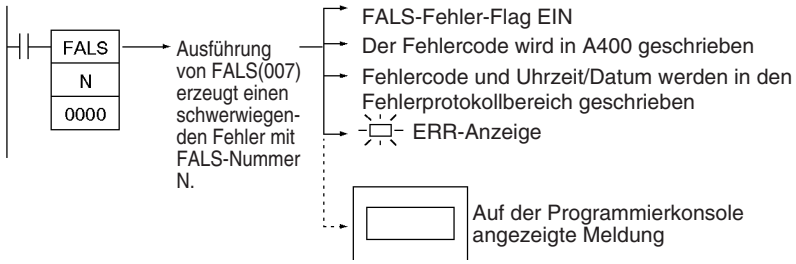
Erzeugen von schwerwiegenden, anwenderdefinierten Fehlern

Wenn FALS(007) ausgeführt wird und N auf eine FALS-Nummer (1 bis 511) gesetzt ist, die nicht gleich dem Inhalt von A529 (die vom System generierte FAL/FALS-Nummer) ist, wird ein schwerwiegender Fehler mit dieser FALS-Nummer generiert und die folgende Verarbeitung erfolgt:

1,2,3...

1. Das FALS-Fehler-Flag (A40106) wird auf EIN gesetzt. (Der Betrieb der SPS wird gestoppt.)
2. Der Fehlercode wird in A400 geschrieben. Die Fehlercodes C101 bis C2FF entsprechen den FALS-Nummern 0001 bis 01FF (1 bis 511).

   Hinweis Wenn ein ernsterer Fehler als der durch den FALS(007)-Befehl erzeugte (einer mit einem höheren Fehlercode) aufgetreten ist, enthält A400 den Fehlercode des ersteren Fehlers.
3. Fehlercode und Uhrzeit/Datum des Auftretens des Fehlers werden in den Fehlerprotokollbereich (A100 bis A199) geschrieben.
4. Die ERR-Anzeige auf der CPU-Baugruppe leuchtet.
5. Wenn in S eine Wortadresse spezifiziert wurde, wird die bei S beginnende ASCII-Meldung registriert (auf dem Peripheriegerät angezeigt).



In der folgenden Tabelle finden Sie die Fehlercodes für FALS(007).

FALS-Nummer	FALS-Fehlercodes
1 bis 511	C101 bis C2FF

Hinweis Die Methode zur Eingabe der FALS-Nummer N ist bei CX-Programmer und einer Programmierkonsole unterschiedlich. Geben Sie bei CX-Programmer #1 bis #511 ein und bei einer Programmierkonsole 001 bis 511.

Anzeigen von Meldungen bei schwerwiegenden, anwenderdefinierten Fehlern

Wenn S eine Wortadresse ist, wird die mit S beginnende ASCII-Meldung bei Ausführung von FALS(007) auf dem Programmiergerät angezeigt. (Wenn eine Meldung nicht erforderlich ist, setzen Sie S auf eine Konstante.)

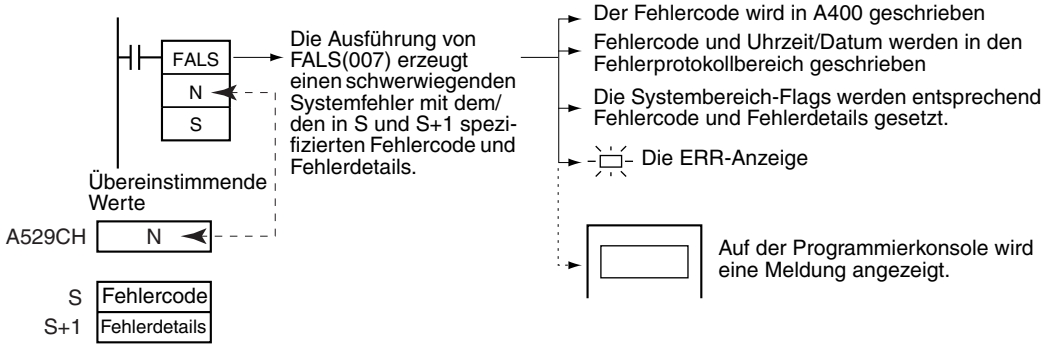
Die mit S beginnende Meldung wird bei Ausführung von FALS(007) registriert. Nach Registrierung der Meldung wird diese bei Anschluss einer Programmierkonsole auf dieser angezeigt.

In S bis S+7 kann eine ASCII-Meldung mit einer Länge von bis zu 16 Zeichen gespeichert werden. Das höchstwertige Byte jedes Worts wird zuerst angezeigt.

Der Endcode für die Meldung ist das Nullzeichen (00 hexadezimal). Beim Auslassen des Nullzeichens werden alle 16 Zeichen in den Worten S bis S+7 angezeigt.

Wenn der Inhalt der Worte, die die Meldung enthalten, nach Ausführung von FALS(007) geändert wird, wird die Meldung entsprechend geändert.

Erzeugen von schwerwiegenden Systemfehlern (nur CS1-H, CJ1-H, CJ1M und CS1D)



Wenn FALS(007) ausgeführt wird und N auf eine FAL-Nummer (1 bis 511) gesetzt ist, die nicht gleich dem Inhalt von A529 (die vom System generierte FAL/FALS-Nummer) ist, wird ein schwerwiegender Fehler mit dem in S und S+1 spezifizierten Fehlercode und Fehlerdetailcode generiert. Gleichzeitig erfolgt die folgende Verarbeitung:

1,2,3...

1. Der spezifizierte Fehlercode wird in A400 geschrieben.
2. Der Fehlercode und Uhrzeit/Datum des Auftretens des Fehlers werden in den Fehlerprotokollbereich (A100 bis A199) geschrieben.
3. Die entsprechenden Systembereich-Flags werden entsprechend Fehlercode und Fehlerdetails gesetzt.
4. Die ERR-Anzeige auf der CPU-Baugruppe leuchtet auf und der SPS-Betrieb wird gestoppt.
5. Die Meldung eines schwerwiegenden Systemfehlers für den spezifizierten Systemfehler wird auf der Programmierkonsole angezeigt.

Hinweis

1. Der Wert von A529 (die vom System generierte FAL/FALS-Nummer) ist eine Dummy-FAL-Nummer (FAL-, FALS- und FPD-Nummern werden gemeinsam genutzt), die bei der absichtlichen Erzeugung eines schwerwiegenden Fehlers durch das System verwendet wird. Dies ist eine Dummy-FAL-Nummer. Sie wird also nicht im Fehlercode wiedergegeben.
   Wenn die Erzeugung von zwei oder mehr Systemfehlern erforderlich ist, können durch mehrmalige Ausführung der FAL/FALS/FPD-Befehle mit den selben Werten in A529 und N, aber unterschiedlichen Werten in S und S+1, verschiedene Fehler generiert werden.
2. Wenn ein ernsterer Fehler (einschließlich eines vom System generierten schwerwiegenden Fehlers oder FALS(007)-Fehlers) gleichzeitig mit der Ausführung des FALS(007)-Befehls auftritt, wird der Fehlercode des ernsteren Fehlers in A400 geschrieben.
3. Um einen durch FALS(007) erzeugten Systemfehler zu löschen, muss die SPS aus- und wieder eingeschaltet werden. Die SPS kann eingeschaltet bleiben, aber zum Löschen des Fehlers ist der gleiche Vorgang erforderlich, als wenn der spezifizierte Fehler tatsächlich aufgetreten wäre. Ausführliche Informationen hierzu finden Sie im *CS-Serie oder CJ-Serie Bedienerhandbuch* im Abschnitt über Fehlerbehebung.
4. In der folgenden Tabelle ist beschrieben, wie das E/A-Speicher-Haltebit den Status des E/A-Speichers sowie den Status von Ausgängen an Ausgangsbaugruppen nach dem Erzeugen eines schwerwiegenden Fehlers mit FALS(007) beeinflusst.

E/A-Speicher-Haltebit (A50012)	Status des E/A-Speichers	Status von Ausgängen an Ausgangsbaugruppen
EIN	Beibehalten	AUS
AUS	Gelöscht	AUS

Hinweis Im Gegensatz zu anwenderdefinierten schwerwiegenden Fehlern löschen die durch FALS(007) erzeugten Systemfehler den E/A-Speicher, wenn das E/A-Speicher-Haltebit auf AUS gesetzt ist. Folgende Bereiche werden gelöscht: CIO-Bereich, Arbeitsbereich nicht remanent, Zeitgeber-Flags und -Istwerte, Indexregister und Datenregister.

In der folgenden Tabelle ist dargestellt, wie Fehlercodes und Fehlerdetails in S und S+1 spezifiziert werden.

Fehler-bezeichnung	S	S+1
	Fehlercode	**Fehlerdetails**
Speicherfehler	80F1 hex	• Bits 00 bis 09: Speicherfehlerposition Bit 00: Anwenderprogramm Bit 04: SPS-Setup Bit 05: Registrierte E/A-Tabelle Bit 07: Routing-Tabelle Bit 08: CPU-Bus-Baugruppen-Konfiguration Bit 09: Fehler bei der Speicherkartenübertragung • Bits 10 bis 15: Ungültig
E/A-Bus-Fehler	80C0 hex	• Bits 00 bis 07: Steckplatz, an dem der E/A-Busfehler aufgetreten ist Steckplatz 0 bis 9: 00 bis 09 hex Steckplatz unbekannt: 0F hex • Bits 08 bis 15: Nummer des Baugruppenträgers, in dem der E/A-Busfehler aufgetreten ist Baugruppenträger 0 bis 7: 00 bis 07 hex Baugruppenträger unbekannt: 0F hex
Baugruppen-nummer-Duplizierungs-fehler	80E9 hex	Doppelt vergebene Baugruppennummer für CPU-Bus-Baugruppe 0000 bis 000F hex
		Doppelt vergebene Baugruppennummer für Spezial-E/A-Baugruppe 8000 bis 805F hex
Baugruppen-trägernummer-Duplizierungs-fehler	80EA hex	Doppelt vergebene Baugruppenträgernummer (überlappende Wortzuordnungen) 0000 bis 0006 hex
Schwerwie-gender Fehler in internem Sondermodul	82F0 hex	Fehlerursache Bits 00 bis 03: Durch internes Sondermodul definierter Fehler Bits 04 bis 15: Ungültig

Fehler-bezeichnung	S	S+1
	Fehlercode	**Fehlerdetails**
Zu-viele-E/A-Punkte-Fehler	80E1 hex	Bits 13 bis 15: Fehlerursache Bits 00 bis 12: Details • Gesamtzahl der E/A-Punkte ist zu hoch. Bits 13 bis 15: 000 Bits 00 bis 12: Anzahl der E/A-Punkte (binär) • Anzahl der Interrupt-Eingänge ist zu hoch. Bits 13 bis 15: 001 Bits 00 bis 12: Anzahl der Interrupt-Eingänge (binär) Bits 00 bis 12: Alle auf null • Eine Baugruppennummer für Slave-Baugruppen ist doppelt vergeben oder eine C500 Slave-Baugruppe hat mehr als 320 E/A-Punkte. Bits 13 bis 15: 010 Bits 00 bis 12: Baugruppennummer der Slave-Baugruppe (binär) • Die Baugruppennummer einer E/A-Schnittstelle (mit Ausnahme von solchen in Remote-Slave-Baugruppenträgern) ist doppelt vergeben. Bits 13 bis 15: 011 Bits 00 bis 12: Baugruppennummer (binär) • Eine Baugruppennummer für Master-Baugruppen ist doppelt vergeben oder liegt außerhalb des zulässigen Einstellbereichs. Bits 13 bis 15: 100 Bits 00 bis 12: Baugruppennummer der Master-Baugruppe (binär) • Die Anzahl der Erweiterungs-Baugruppenträger ist zu hoch. Bits 13 bis 15: 101 Bits 00 bis 12: Anzahl der Erweiterungs-Baugruppenträger (binär) • C200H Spezial-E/A-Baugruppe oder dezentrale E/A wurde nicht erkannt. Bits 13 bis 15: 110
E/A-Tabellen-Einstellungs-fehler	80E0 hex	--- (nicht festgelegt)
Programmfehler	80F0 hex	• Bits 08 bis 15: Fehlerursache Bit 15: Anwenderprogrammspeicher-Überlauffehler Bit 14: Ungültiger Befehl Bit 13: Flankenauswertungs-Überlauffehler Bit 12: Task-Fehler Bit 11: END-Befehl fehlt Bit 10: Zugriffsfehler Bit 09: BCD-Fehler bei indirekter DM/EM-Adressierung Bit 08: Befehlsfehler • Bits 00 bis 07: Ungültig
Zykluszeit-Überlauffehler	809F hex	--- (nicht festgelegt)

Löschen von schwerwiegenden FALS(007)-Systemfehlern (nur CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen)

Es gibt zwei Wege, schwerwiegende Systemfehler zu löschen, die mit FALS(007) erzeugt wurden.

1. Schalten Sie die Versorgungsspannung der SPS aus und wieder ein.
2. Soll die SPS eingeschaltet bleiben, muss der Systemfehler gelöscht werden, als ob der spezifizierte Fehler tatsächlich aufgetreten wäre.

Löschen von anwenderdefinierten schwerwiegenden (FALS(007)-Fehlern

Um durch FALS(007) erzeuge Fehler zu löschen, muss zunächst die Fehlerursache behoben werden. Dann muss der Fehler mit einem Programmiergerät gelöscht oder die SPS muss aus- und wieder eingeschaltet werden.

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn N nicht innerhalb des spezifizierten Bereichs von 0001 bis 01FF hex (1 bis 511 dezimal) liegt. EIN, wenn ein schwerwiegender Systemfehler erzeugt wird (nur CS1-H/CJ1-H/CJ1M/CS1D), aber der spezifizierte Fehlercode oder Fehlerdetailcode falsch ist. In allen anderen Fällen AUS.

In den nachfolgende Tabellen sind die relevanten Worte und Flags im Systembereich ausgeführt.

- Systembereichs-Worte/-Flags nur für anwenderdefinierte Fehler

Name	Adresse	Funktion
FALS-Fehler-Flag	A40106	EIN, wenn ein Fehler mit FALS(007) erzeugt wurde.

- Systembereichs-Worte/-Flags nur für Systemfehler (nur CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen)

Name	Adresse	Funktion
Vom System erzeugte FAL/FALS-Nummer	A529	Wenn ein Systemfehler mit FALS(007) erzeugt wird, wird eine Dummy-FAL/FALS-Nummer verwendet. Dieselbe Dummy-FAL/FALS-Nummer (0001 bis 01FF hex, 1 bis 511 dezimal) wird in dieses Wort gesetzt.

- Systembereichs-Worte/-Flags für anwenderdefinierte Fehler und Systemfehler

Name	Adresse	Funktion
Fehlerprotokoll-bereich	A100 bis A199	Der Fehlerprotokollbereich enthält Fehlercodes und Uhrzeit/Datum der 20 zuletzt aufgetretenen Fehler, einschließlich durch FALS(007) erzeugter Fehler.
Fehlercode	A400	Beim Auftreten eines Fehlers wird der entsprechende Fehlercode in A400 gespeichert. Die Fehlercodes für die FALS-Nummern 0001 bis 01FF (1 bis 511 dezimal) lauten C101 bis C2FF. Wenn zwei oder mehr Fehler gleichzeitig auftreten, wird der Fehlercode des schwersten Fehlers in A400 gespeichert.

Vorsichtsmaßnahmen

Der Endcode für die Meldung ist das Nullzeichen (00 hexadezimal). Beim Auslassen des Nullzeichens werden alle 16 Zeichen in den Worten S bis S+7 angezeigt.

N muss einen Wert zwischen 0001 und 01FF haben. Wenn der Wert für N außerhalb des spezifizierten Bereichs liegt, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Beispiele

Erzeugen eines anwenderdefinierten Fehlers

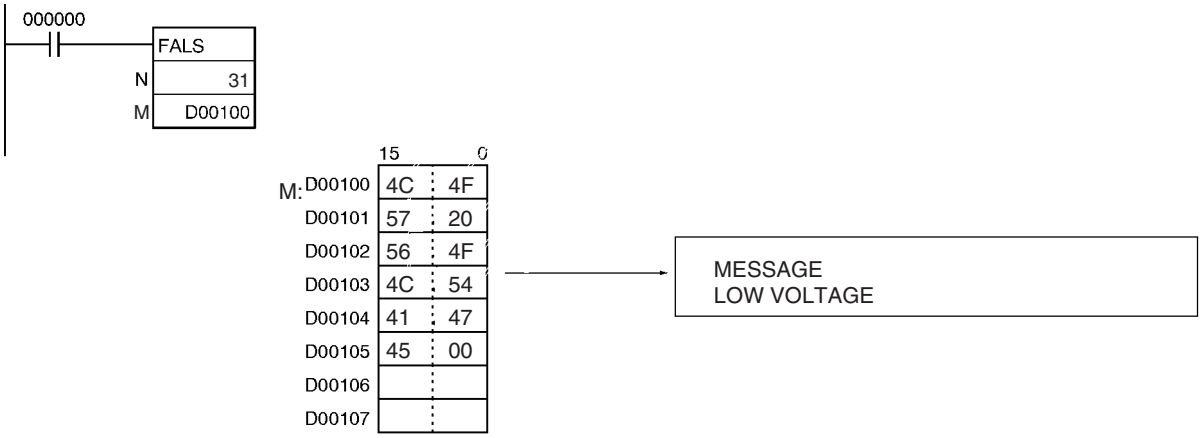
Wenn CIO 000000 im folgenden Beispiel EIN ist, erzeugt FALS(007) einen schwerwiegenden Fehler mit FAL Nummer 31 und führt die folgenden Prozesse aus.

1,2,3...

1. Das FALS-Fehler-Flag (A40106) wird auf EIN gesetzt.
2. Der entsprechende Fehlercode (C11F) wird in A400 geschrieben.

   Hinweis A400 enthält den Fehlercode des ernstesten aller aufgetretenen Fehler, einschließlich nicht schwerwiegender und schwerwiegender Systemfehler, sowie durch FAL(006) und FAL(007) erzeugter Fehler.

3. Fehlercode und Uhrzeit/Datum des Auftretens des Fehlers werden in den Fehlerprotokollbereich (A100 bis A199) geschrieben.
4. Die ERR-Anzeige auf der CPU-Baugruppe leuchtet.
5. Die ASCII-Meldung in D00100 bis D00107 wird auf dem Peripheriegerät angezeigt. (Wenn eine Meldung nicht erforderlich ist, spezifizieren Sie für S eine Konstante.)

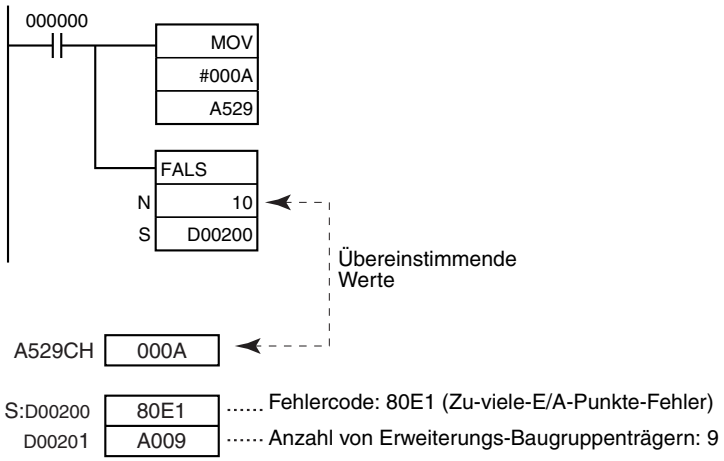


Erzeugen eines schwerwiegenden Systemfehlers (nur CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen)

Wenn CIO 000000 im folgenden Beispiel EIN ist, erzeugt FALS(007) einen Zu-viele-E/A-Punkte-Fehler (zu viele Erweiterungs-Baugruppenträger angeschlossen, in diesem Fall 9 Baugruppenträger). In diesem Fall wird die Dummy-FAL-Nummer 10 verwendet und der entsprechende Wert (000A hex) wird in A529 gespeichert.

1,2,3...

1. Der spezifizierte Fehlercode (80E1) wird in A400 geschrieben, wenn es sich um den ernstesten Fehler handelt.
2. Fehlercode und Uhrzeit/Datum des Auftretens des Fehlers werden in den Fehlerprotokollbereich (A100 bis A199) geschrieben.
3. Das Zu-viele-E/A-Punkte-Flag (A40111) wird auf EIN gesetzt.
4. Die Fehleranzeige (ERR) der CPU-Baugruppe leuchtet auf und der SPS-Betrieb wird gestoppt.
5. Eine Meldung (TOO MANY I/O PNT), die auf einen Zu-viele-E/A-Punkte-Fehler hinweist, wird auf der Programmierkonsole angezeigt.

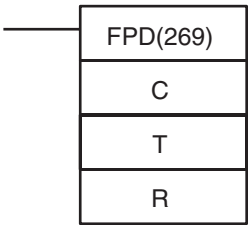


3-30-3 FAILURE POINT DETECTION: FPD(269)

Funktion

Ermittelt einen Fehler in einem Befehlsblock durch Überwachung der Zeit zwischen der Ausführung von FPD(269) und der Ausführung eines Diagnose-Ausgangs und Ermittlung, welcher Eingang verhindert, dass der Ausgang auf EIN wechselt.

Kontaktplansymbol



C: Steuerwort

T: Überwachungszeit

R: Erstes Registerwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FPD(269)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

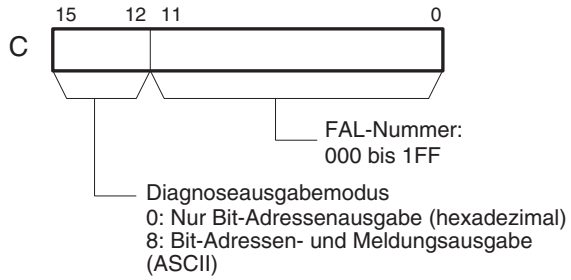
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
Nicht zulässig	OK	OK	Nicht zulässig

Operanden

C: Steuerwort

C muss eine Konstante zwischen 0000 und 01FF oder zwischen 8000 und 81FF sein.

In der folgenden Abbildung sind die Funktionen der einzelnen Stellen im Steuerwort beschrieben.



T: Überwachungszeit

T muss zwischen 0 und 9.999 dezimal (zwischen 0000 und 270F hex) liegen. Ein Wert von 0 deaktiviert die Zeitüberwachung und Werte im Bereich von 1 bis 270F stellen die Überwachungszeit auf Werte zwischen 0,1 und 999,9 Sekunden ein.

R: Erstes Registerwort

Die Funktionen von Registerworten sind auf Seite 1098 beschrieben.

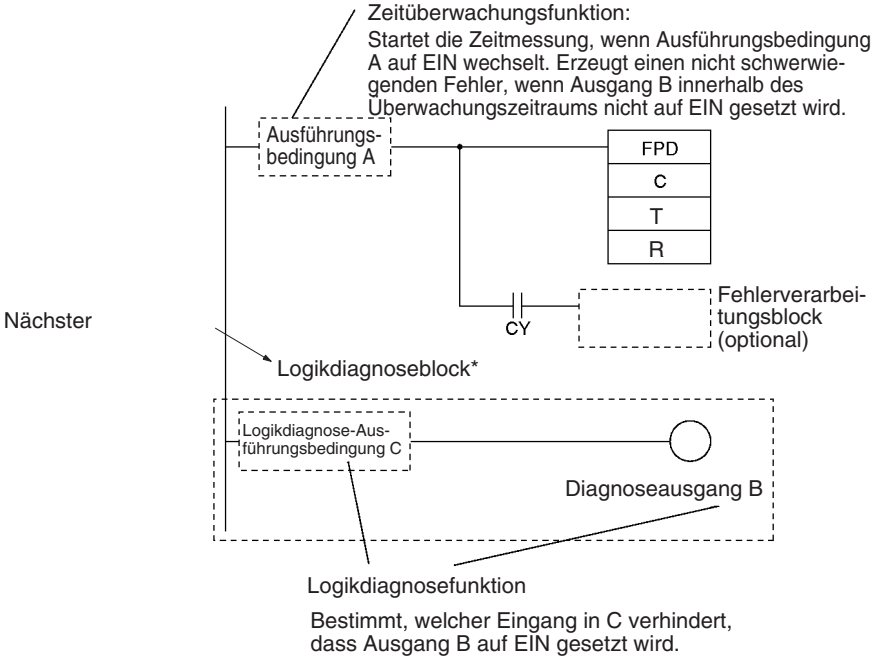
Operanden-Spezifikationen

Bereich	C	T	R
CIO-Bereich	---	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	---	W000 bis W511	
Arbeitsbereich remanent	---	H000 bis H511	
Systembereich	---	A000 bis A447 A448 bis A959	A448 bis A959
Zeitgeberbereich	---	T0000 bis T4095	

Bereich	C	T	R
Zählerbereich	---	C0000 bis C4095	
DM-Bereich	---	D00000 bis D32767	
EM-Bereich ohne Bank	---	E00000 bis E32767	
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	Nur spezifizierte Werte	#0000 bis #270F (binär)	---
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	

Beschreibung

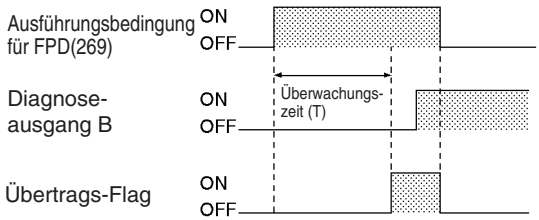
Mit FPD(269) können Sie Zeitüberwachung und Logikdiagnose durchführen. Die Zeitüberwachungsfunktion löst einen nicht schwerwiegenden Fehler mit der spezifizierten FAL-Nummer aus, wenn der Diagnoseausgang innerhalb der angegebenen Überwachungszeit nicht auf EIN gesetzt wird. Die Logikdiagnosefunktion dient zur Ermittlung, welcher Eingang verhindert, dass der Ausgang auf EIN gesetzt wird.



Hinweis *Der Logikdiagnoseblock beginnt mit dem ersten LD- (nicht LD TR) oder LD NOT-Befehl nach FPD(269) und endet mit dem ersten OUT- (nicht OUT TR) Befehl oder einem anderen, auf der rechten Seite programmierten Befehl.

Zeitüberwachungsfunktion

FPD(269) startet die Zeitmessung mit der Ausführung (wenn die Ausführungsbedingung A auf EIN gesetzt wird). Der Befehl erzeugt einen nicht schwerwiegenden Fehler und setzt das Übertrags-Flag auf EIN, wenn der Diagnoseausgang nicht innerhalb der festgelegten Überwachungszeit auf EIN gesetzt wird.



Nicht schwerwiegender Fehler erzeugt.

Hinweis Der Diagnoseausgang muss innerhalb der Überwachungszeit auf EIN gesetzt werden. Zur automatischen Einstellung der Überwachungszeit kann die Teach-In-Funktion verwendet werden.

Der folgende Vorgang wird ausgeführt, wenn das Übertrags-Flag auf EIN gesetzt wird. (Der Vorgang wird nicht ausgeführt, wenn die FAL-Nummer in C auf 000 gesetzt ist.)

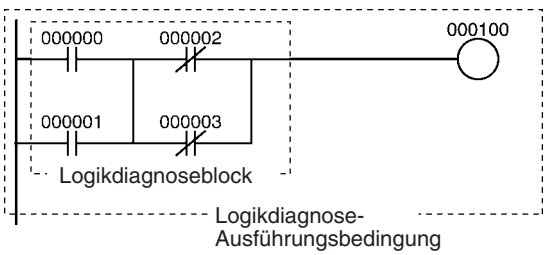
1,2,3...

1. Das FAL-Fehler-Flag (A40215) wird auf EIN gesetzt. (Der Betrieb der SPS wird fortgesetzt.)
2. Das Ausgeführte-FAL-Nummer-Flag für die spezifizierte FAL-Nummer wird auf EIN gesetzt. (Die Flags A36001 bis A39115 entsprechen den FAL-Nummern 001 bis 1FF.)
3. Der entsprechende Fehlercode wird in A400 geschrieben. Die Fehlercodes 4101 bis 42FF entsprechen den FAL-Nummern 001 bis 1FF. (Wenn gleichzeitig ein ernsterer Fehler aufgetreten ist (einer mit einem höheren Fehlercode), wird der Fehlercode des ernsteren Fehlers in A400 gespeichert.)
4. Fehlercode und Uhrzeit/Datum des Auftretens des Fehlers werden in den Fehlerprotokollbereich (A100 bis A199) geschrieben.
5. Die ERR-Anzeige auf der CPU-Baugruppe blinkt.
6. Wenn der Ausgabemodus auf Bit-Adresse und Meldungsausgabe (höchstwertige Stelle von C auf 8 eingestellt) eingestellt wurde, wird die in R+2 bis R+10 gespeicherte ASCII-Meldung als Fehlermeldung für einen nicht schwerwiegenden Fehler angezeigt.

Logikdiagnosefunktion

In jedem Zyklus, in dem die Ausführungsbedingung für FPD(269) auf EIN gesetzt ist, ermittelt FPD(269), welches Eingangsbit dazu führt, dass der Diagnoseausgang AUS ist, und schreibt die Adresse dieses Bits in den mit R beginnenden Registerbereich.

Wenn die Eingangsbits CIO 000000 bis CIO 000003 im folgenden Beispiel alle EIN sind, ermittelt FPD(269), dass die Öffnerbedingung CIO 000002 dazu führt, dass der Ausgang CIO 000100 AUS bleibt. FPD(269) setzt das Bit-Adresse-gefunden-Bit (Bit 15 von R) auf EIN und schreibt die Bit-Adresse in die Registerworte R+2 bis R+4.



Die Logikdiagnosefunktion wird so lange in jedem Zyklus ausgeführt, wie die Ausführungsbedingung für FPD(269) auf EIN gesetzt ist. Die Logikdiagnosefunktion ist unabhängig von der Zeitüberwachungsfunktion.

Wenn zwei oder mehr Eingangsbits das Einschalten des Diagnoseausgangs verhindern, wird die Adresse des ersten Eingangsbits in der Ausführungsbedingung (in der höchsten Befehlszeile und am nächsten an der linken Stromschiene) in R+2 bis R+4 ausgegeben.

Durch die Logikdiagnosefunktion werden Eingangsbits in LD-, LD NOT-, AND-, AND NOT-, OR- und OR NOT-Befehlen (einschließlich deren Varianten mit Flankenauswertung und direkter Aktualisierung) geprüft. Eingangsbits in anderen Befehlen und Operanden, die indirekt über Indexregister adressiert werden, werden nicht geprüft.

Der Logikdiagnoseblock beginnt mit dem ersten LD- (nicht LD TR) oder LD NOT-Befehl nach FPD(269) und endet mit dem ersten OUT- (nicht OUT TR) Befehl oder einem anderen, auf der rechten Seite zu programmierenden Befehl.

Es gibt zwei Diagnoseausgangsmodi, die durch die höchstwertige Stelle von C eingestellt werden.

1,2,3...

1. Bitadressen-Ausgangsmodus (höchstwertige Stelle von C = 0)

   Bit 15 von R (das Bit-Adresse-gefunden-Bit) wird auf EIN gesetzt, wenn eine Eingangsbitadresse gefunden wurde. Bit 14 von R zeigt an, ob es sich bei dem Eingang um einen Öffner oder Schließer handelt.
   Die 8-stellige hexadezimale SPS-Speicheradresse des Eingangsbits wird in R+3 und R+2 ausgegeben.

2. Bitadressen- und Meldungsausgabemodus (höchstwertige Stelle von C = 8)

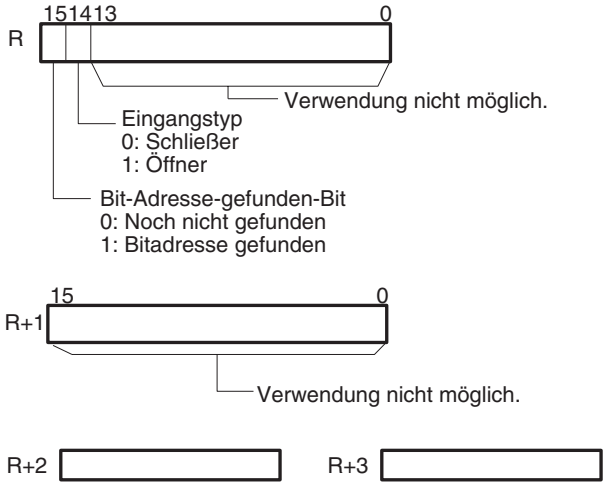
   Bit 15 von R (das Bit-Adresse-gefunden-Bit) wird auf EIN gesetzt, wenn eine Eingangsbitadresse gefunden wurde. Bit 14 von R zeigt an, ob es sich bei dem Eingang um einen Öffner oder Schließer handelt.
   Die Adresse des Eingangsbits wird in Form von 6 ASCII-Zeichen in R+2 bis R+4 ausgegeben.

Registerwortfunktionen

Die Registerworte enthalten die Ergebnisse der Diagnosefunktion und können auch eine ASCII-Fehlermeldung enthalten, die angezeigt wird, wenn von der Zeitüberwachungsfunktion ein Fehler erzeugt wird. Die Funktion der Registerworte hängt vom Diagnoseausgangsmodus ab, der mit der höchstwertigen Stelle von C gewählt wird.

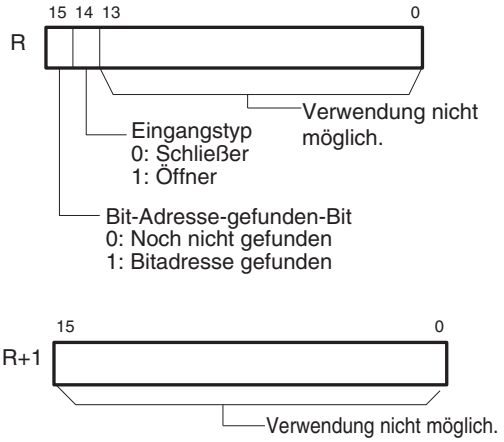
Bitadressenausgabe (C = 0□□□)

Wenn die höchstwertige Stelle von C auf 0 gesetzt ist, wird die SPS-Speicheradresse des Eingangsbits als 8-stellige Hexadezimalzahl in R+2 und R+3 ausgegeben. R enthält zwei Flags, die angeben, ob ein Eingangsbit gefunden wurde und ob es in einer Öffner- oder Schließer-Eingangsbedingung verwendet wird.



Bitadressen- und Meldungsausgabe (C = 8□□□)

Wenn die höchstwertige Stelle von C auf 8 gesetzt wird, wird die ASCII-Adresse des Eingangsbits in R+2 bis R+4 ausgegeben. R enthält zwei Flags, die angeben, ob ein Eingangsbit gefunden wurde und ob es in einer Schließer- oder Öffner-Eingangsbedingung verwendet wird.



Die Registerworte R+2 bis R+4 enthalten die Adresse des Eingangs, der ein Einschalten des Diagnoseausgangs verhinderte. Die Bitadresse wird im ASCII-Format in diese Worte ausgegeben. Der folgenden Tabelle sind die ASCII-Entsprechungen für die einzelnen Bereiche zu entnehmen.

Bereich	ASCII-Text	Hinweise
Systembereich	A00000 bis A95915	---
Arbeitsbereich remanent	H00000 bis H51115	---
Arbeitsbereich nicht remanent	W00000 bis W51115	---
CIO-Bereich	000000 bis 665515	---
Task-Flags	TK0000 bis TK0031	---
Zeitgeberbereich	_T0000 bis _T4095	Die Unterstriche „_“ stellen ein ASCII-Leerzeichen dar. (Zeichencode 20)
Zählerbereich	_C0000 bis _C4095	

	15	
R+2	W	5
R+3	1	1
R+4	1	5

In ASCII geschriebene Bitadresse

Die Registerworte R+2 bis R+5 würden für W51115 die folgenden Werte annehmen:

Wort	Bits 8 bis 15	Bits 0 bis 7
R+2	W	5
R+3	1	1
R+4	1	5
R+5	2D (hexadezimal)	Eingangsart (hexadezimal) 30: Schließer 31: Öffner

Der Anwender kann eine ASCII-Meldung in den Registerworten R+6 bis R+10 speichern. Diese Meldung wird auf dem Programmiergerät angezeigt, wenn von der Überwachungsfunktion ein nicht schwerwiegender Fehler erzeugt wird. Kennzeichnen Sie das Ende der Meldung durch das Nullzeichen (00 hexadezimal).

	15 8	7 0
R+6		
R+7		
R+8		
R+9		
R+10		

Deaktivierung von Fehlerprotokolleinträgen bei nicht schwerwiegenden FPD(269)-Fehlern (nur CS1-H, CJ1-H, CJ1M oder CS1D)

Wenn die Zeitüberwachungsfunktion FPD(269) einen nicht schwerwiegenden Fehler erzeugt, werden Fehlercode und Uhrzeit/Datum seines Auftretens normalerweise in den Fehlerprotokollbereich (A100 bis A199) geschrieben. Bei CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen kann das SPS-Setup so eingestellt werden, dass die von FAL(006) erzeugten, nicht schwerwiegenden Fehler nicht im Fehlerprotokoll gespeichert werden.

Auch wenn der Fehler nicht im Fehlerprotokoll gespeichert wird, wird das FAL-Fehler-Flag (40215) auf EIN gesetzt, das entsprechende Flag in den Ausgeführte-FAL-Nummer-Flags (A36001 bis A39115) auf EIN gesetzt und der Fehlercode in A400 geschrieben.

Deaktivieren Sie Fehlerprotokolleinträge für FPD(269)-Zeitüberwachungsfehler, wenn nur die vom System erzeugten Fehler gespeichert werden sollen. Diese Funktion ist z.B. während der Fehlersuche hilfreich, wenn FPD(269) und FAL(006)-Befehle in verschiedenen Anwendungen verwendet werden und sich das Fehlerprotokoll mit diesen Fehlern füllt. In der folgenden Tabelle ist die Einstellung des SPS-Setups dargestellt:

Parameter	Einstellung	
Einstellungsadresse der Programmierkonsole	Wort	129
	Bit	15
Name	FAL-Fehlerprotokoll-Registrierung	
Einstellungen	0: FAL-Fehler im Fehlerprotokoll aufzeichnen. 1: FAL-Fehler nicht im Fehlerprotokoll aufzeichnen.	
Standardeinstellung	0: FAL-Fehler im Fehlerprotokoll aufzeichnen.	
Zeitpunkt, an dem die SPS-Setup-Einstellung gelesen wird	Jeden Zyklus (bei Auftreten eines FAL-Fehlers)	

Auch wenn das SPS-Setup-Wort 129 Bit 15 auf 1 gesetzt ist (FAL-Fehler nicht im Fehlerprotokoll aufzeichnen), werden die folgenden Fehler aufgezeichnet:

- Durch FALS(007) erzeugte schwerwiegende Fehler
- Nicht schwerwiegende Systemfehler

- Schwerwiegende Systemfehler
- Nicht schwerwiegende Systemfehler, die absichtlich mit FAL(006) oder FPD(269) erzeugt wurden.
- Schwerwiegende Systemfehler, die absichtlich durch FALS(007) erzeugt wurden.

Einstellung der Überwachungszeit mit der Teach-In-Funktion

Wenn für T eine Wortadresse spezifiziert wird, kann die Überwachungszeit automatisch mit der Teach-In-Funktion eingestellt werden. Führen Sie folgenden Vorgang aus, wenn für T eine Wortadresse eingestellt wurde.

1,2,3...

1. Setzen Sie das FPD-Teach-In-Bit (A59800) auf EIN.
2. FPD(269) misst die Zeit ab dem Zeitpunkt, an dem die Ausführungsbedingung für FPD(269) auf EIN gesetzt wird bis zu dem Zeitpunkt, an dem der Diagnoseausgang auf EIN gesetzt wird.
3. Wenn die gemessene Zeit die eingestellte Überwachungszeit überschreitet, wird die 1,5-fache Zeit der gemessenen Zeit in T gespeichert.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn C nicht im spezifizierten Bereich von 0000 bis 01FF oder 8000 bis 81FF liegt. EIN, wenn T nicht im spezifizierten Bereich von 0000 bis 270F liegt. In allen anderen Fällen AUS.
Übertrags-Flag	CY	EIN, wenn der Diagnoseausgang nach Ablauf der Überwachungszeit immer noch auf AUS ist. In allen anderen Fällen AUS.

In der nachfolgenden Tabelle sind die relevanten Worte und Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
FAL-Fehler-Flag	A40215	EIN, wenn bei der Zeitüberwachung ein nicht schwerwiegender (FAL-) Fehler erkannt wird.
Ausgeführte-FAL-Nummer-Flags	A36001 bis A39115	Wenn bei der Zeitüberwachung ein nicht schwerwiegender (FAL-) Fehler erkannt wird, wird das entsprechende Flag auf EIN gesetzt. Die Flags A36001 bis A39115 entsprechen den FAL-Nummern 0001 bis 01FF.
Fehlerprotokollbereich	A100 bis A199	Der Fehlerprotokollbereich enthält Fehlercodes und Uhrzeit/Datum der 20 zuletzt aufgetretenen Fehler, einschließlich durch FAL(269) erzeugter Fehler.
Fehlercode	A400	Beim Auftreten eines Fehlers wird der entsprechende Fehlercode in A400 gespeichert. Die Fehlercodes für die FAL-Nummern 0001 bis 01FF lauten 4101 bis 42FF. Wenn zwei oder mehr Fehler gleichzeitig auftreten, wird der Fehlercode des schwersten Fehlers in A400 gespeichert.
FPD-Teach-In-Bit	A59800	Setzen Sie dieses Bit auf EIN, wenn die Überwachungszeit bei der Ausführung von FPD(269) automatisch eingestellt werden soll (Teach-In-Funktion).

Vorsichtsmaßnahmen

Wenn die Zeitüberwachungsfunktion verwendet wird, muss die Ausführungsbedingung für FPD(269) über die gesamte, in T festgelegte Überwachungszeit auf EIN bleiben.

Die Ausführungsbedingung für FPD(269) muss aus einer Kombination von Schließer- und Öffnereingängen bestehen.

Der Fehlerverarbeitungsblock ist optional. Wenn ein Fehlerverarbeitungsblock einbezogen wird, müssen Ausgänge oder andere Befehle, die auf der rechten Seite programmiert werden, verwendet werden. LD und LD NOT können an dieser Stelle nicht verwendet werden.

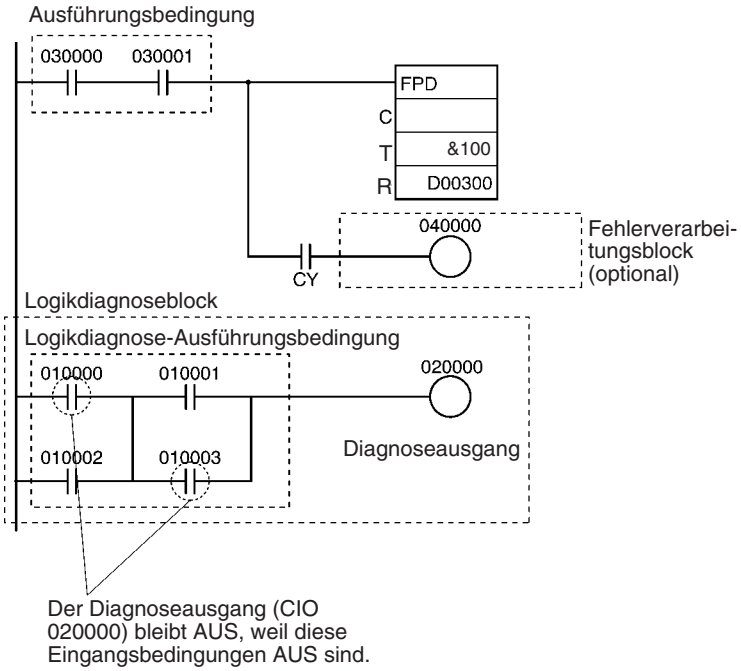
FPD(269) kann im Programm mehr als einmal verwendet werden, aber jeder Befehl muss eine eigene Registereinstellung (R) aufweisen.

Die Überwachungszeit wird nur bei Ausführung von FPD(269) aktualisiert. Wenn die Zykluszeit länger als 100 ms ist, wird die Überwachungszeit nicht

wie üblich aktualisiert und FPD(269) arbeitet nicht korrekt, da die Überwachungszeit in Einheiten von 100 ms aktualisiert wird.

Beispiele

Das folgende Programmbeispiel wird dazu verwendet, den Betrieb der Zeitüberwachungsfunktion sowie der Logikdiagnosefunktion zu demonstrieren. In diesem Beispiel wird der Diagnoseausgang (CIO 020000) nicht auf EIN gesetzt, da CIO 010000 und CIO 010003 in der Logikdiagnose-Ausführungsbedingung auf AUS gesetzt bleiben.



Zeitüberwachungsfunktion

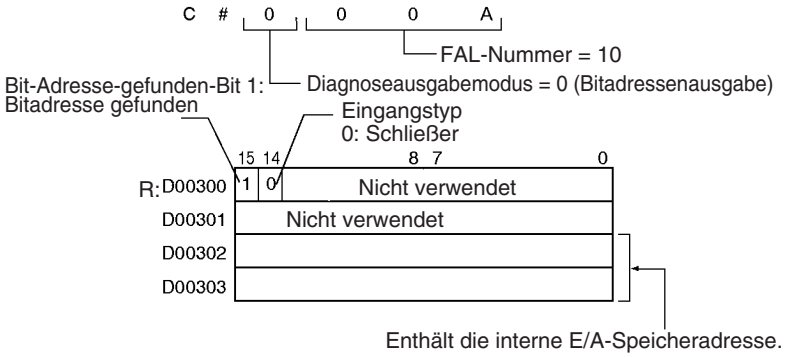
Wenn der Diagnoseausgang (CIO 020000) nicht innerhalb von 10 Sekunden auf EIN gesetzt wird, nachdem CIO 030000 und CIO 030001 beide auf EIN gesetzt wurden, wird ein nicht schwerwiegender Fehler erzeugt und die folgende Verarbeitung ausgeführt.

1,2,3...

1. Das Übertrags-Flag wird auf EIN gesetzt.
2. Wenn die niederwertigen 3 Stellen von C eine FAL-Nummer von 00A hex (10 dezimal) spezifizieren, wird das entsprechende Ausgeführte-FAL-Nummer-Flag (A36010) auf EIN gesetzt, der entsprechende Fehlercode (410A) wird in A400 geschrieben und das FAL-Fehler-Flag (A40215) wird auf EIN gesetzt.

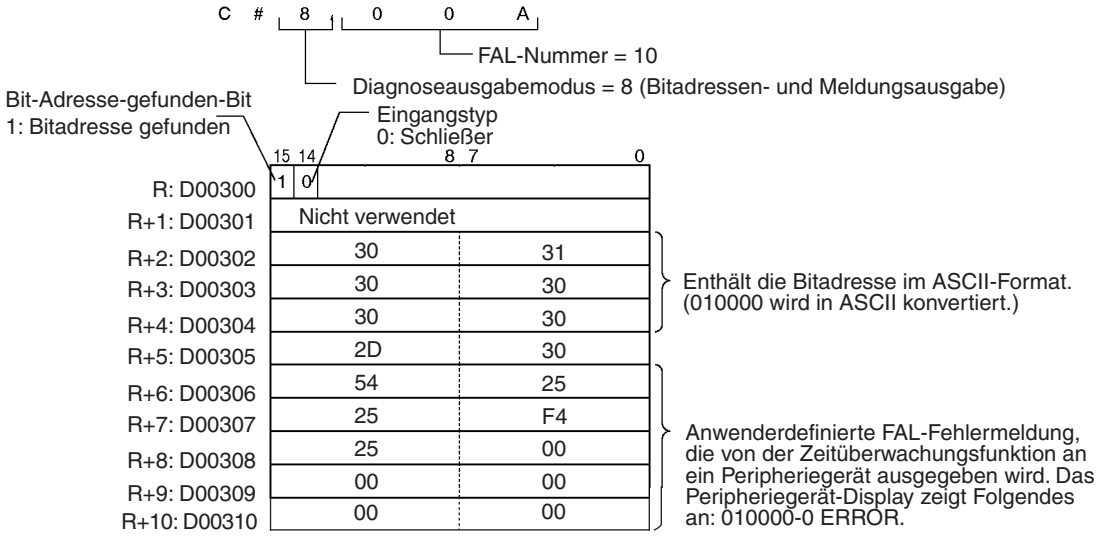
Logikdiagnosefunktion (C = 000A)

Da die höchstwertige Stelle von C gleich 0 (Bitadressen-Ausgabemodus) ist, wird die SPS-Speicheradresse von CIO 010000 in D00303 und D00302 ausgegeben. (CIO 010000 befindet sich in einer höheren Befehlszeile als CIO 010003.)



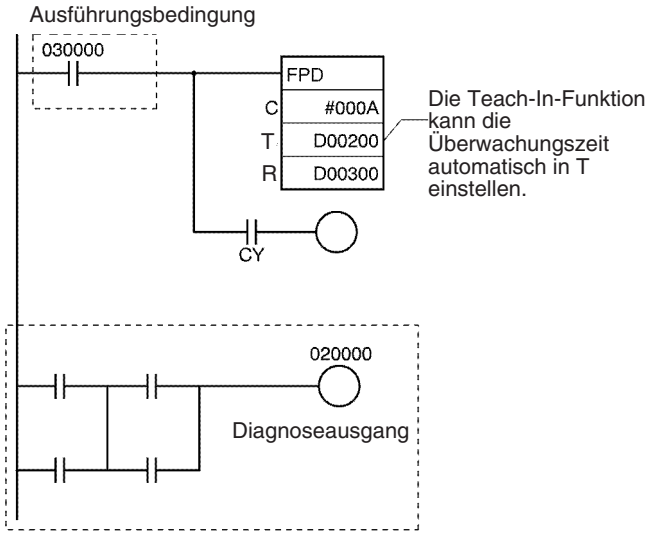
Logikdiagnosefunktion (C = 800A)

Da die höchstwertige Stelle von C gleich 8 (Bitadressen- und Meldungsausgabemodus) ist, wird die Adresse von CIO 010000 (010000) in D00302 bis D00304 im ASCII-Format ausgegeben.

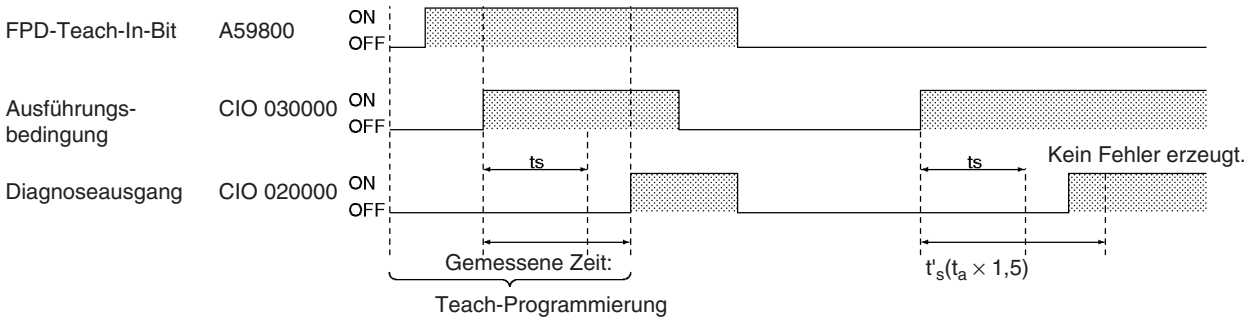


Einstellen der Überwachungszeit mit der Teach-In-Funktion

Wenn für T eine Wortadresse spezifiziert wurde, kann die Überwachungszeit mit der Teach-In-Funktion automatisch eingestellt werden.



Zum Starten der Teach-In-Funktion wird A59800 (FPD-Teach-In-Bit) auf EIN gesetzt. Während A59800 auf EIN gesetzt ist, misst FPD(269), wie lange es dauert, bis der Diagnoseausgang (CIO 020000) auf EIN gesetzt wird, nachdem die Ausführungsbedingung (CIO 030000) auf EIN gesetzt wurde. Überschreitet die gemessene Zeit die Überwachungszeit in T, wird die gemessene Zeit mit 1,5 multipliziert und dieser Wert als neue Überwachungszeit in T gespeichert.



t_s: Anfangseinstellung in T
t_a: Gemessene Zeit
t'_s: Neue Einstellung in T nach Teach-In
(Wenn $t_a > t_s$, $t'_s = t_a \times 1{,}5$)

3-31 Sonstige Befehle

In diesem Abschnitt werden Befehle für die Beeinflussung des Übertrags-Flags, zur Auswahl der EM-Bank und zur Verlängerung der maximalen Zykluszeit beschrieben.

Befehl	AWL	Funktionscode	Seite
SET CARRY	STC	040	1104
CLEAR CARRY	CLC	041	1105
SELECT EM BANK	EMBC	281	1106
EXTEND MAXIMUM CYCLE TIME	WDT	094	1108
SAVE CONDITION FLAGS	CCS	282	1110
LOAD CONDITION FLAGS	CCL	283	1112
CONVERT ADDRESS FROM CV	FRMCV	284	1113
CONVERT ADDRESS TO CV	TOCV	285	1117
DISABLE PERIPHERAL SERVICING	IOSP	287	1121
ENABLE PERIPHERAL SERVICING	IORS	288	1123

3-31-1 SET CARRY: STC(040)

Setzt das Übertrags-Flag (CY).

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	STC(040)
	Ausführung einmal bei steigender Flanke	@STC(040)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Beschreibung

Wenn die Ausführungsbedingung EIN ist, setzt STC(040) das Übertrags-Flag (CY) auf EIN. Obwohl STC(040) das Übertrags-Flag auf EIN setzt, wird das Flag durch die anschließende Ausführung von das Übertrags-Flag betreffenden Befehlen auf EIN/AUS gesetzt.

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Übertrags-Flag	CY	EIN
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Vorsichtsmaßnahmen ROL(027), ROLL(572), ROR(028) und RORL(573) machen bei ihrer Rotationsverschiebung vom Übertrags-Flag Gebrauch. Bei Verwendung eines dieser Befehle verwenden Sie STC(040) und CLC(041) zum Setzen und Löschen des Übertrags-Flags.

3-31-2 CLEAR CARRY: CLC(041)

Funktion Setzt das Übertrags-Flag (CY) auf AUS.

Kontaktplansymbol



Variationen

Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	CLC(041)
	Ausführung einmal bei steigender Flanke	@CLC(041)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Beschreibung Wenn die Ausführungsbedingung EIN ist, setzt CLC(040) das Übertrags-Flag (CY) auf AUS. Obwohl STC(040) das Übertrags-Flag auf EIN setzt, wird das Flag durch die anschließende Ausführung von das Übertrags-Flag nutzenden Befehlen auf EIN/AUS gesetzt.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	AUS oder unverändert (siehe Hinweis)
Gleich-Flag	=	AUS oder unverändert (siehe Hinweis)
Übertrags-Flag	CY	AUS
Negativ-Flag	N	AUS oder unverändert (siehe Hinweis)

Hinweis In CS1 und CJ1 CPU-Baugruppen werden diese Flags auf AUS gesetzt. In CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen bleiben diese Flags unverändert.

Vorsichtsmaßnahmen +C(402), +CL(403), +BC(406) und +BCL(407) verwenden das Übertrags-Flag bei ihren Additionsfunktionen. Führen Sie vor Verwendung eines dieser Befehle den Befehl CLC(041) aus, um jeglichen Einfluss von vorherigen Befehlen zu vermeiden.

+C(412), +CL(413), +BC(416) und +BCL(417) verwenden das Übertrags-Flag bei ihren Subtraktionsfunktionen. Führen Sie vor Verwendung eines dieser Befehle den Befehl CLC(041) aus, um jeglichen Einfluss von vorherigen Befehlen zu vermeiden.

ROL(027), ROLL(572), ROR(028) und RORL(573) machen bei ihrer Rotationsverschiebung vom Übertrags-Flag Gebrauch. Bei Verwendung eines dieser Befehle verwenden Sie STC(040) und CLC(041) zum Setzen und Löschen des Übertrags-Flags.

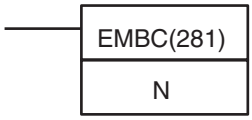
Hinweis Die Befehle +(400), +L(401), +B(404), +BL(405), –(410), –L(411), –B(414) und –BL(415) nutzen das Übertrags-Flag nicht in ihren Additions- und Subtraktionsfunktionen. Im Allgemeinen sind diese Befehle für die Durchführung von Additionen oder Subtraktionen zu verwenden.

3-31-3 SELECT EM BANK: EMBC(281)

Funktion

Wechselt die aktuelle EM-Bank.

Kontaktplansymbol



N: Nummer der EM-Bank

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	EMBC(281)
	Ausführung einmal bei steigender Flanke	@EMBC(281)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: EM-Banknummer
Spezifiziert die neue EM-Banknummer in hexadezimal (0000 bis 000C).

Operanden-Spezifikationen

Bereich	N
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A000 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767
Konstanten	#0000 bis #000C (binär)
Datenregister	DR0 bis DR15

Bereich	N
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

EMBC(281) ändert die aktuelle EM (erweiterter Datenspeicher) -Bank in die durch die EM-Banknummer (N) angegebene EM-Bank. Gleichzeitig wird die neue EM-Banknummer in A301 ausgegeben.

Im EM-Bereich sind bis zu 13 Bänke (0 bis C) verfügbar und jede Bank umfasst 32.768 Worte (E00000 bis E32767). EM-Adressen können auf die zwei folgenden Arten spezifiziert werden. Bei Anwendung der ersten Methode muss EMBC(281) zur Änderung der aktuellen EM-Bank verwendet werden.

1,2,3...

1. EM-Adressen können ohne Banknummer spezifiziert werden, also als E00000 bis E32767, um Adressen in der aktuellen EM-Bank anzugeben.
2. EM-Adressen können mit Banknummer spezifiziert werden, also als En_00000 bis En_32767 (n = 0 bis C), um Adressen in einer bestimmten EM-Bank anzugeben.

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn N nicht einen Wert innerhalb des Bereichs von 0000 bis 000C hat. EIN, wenn N eine nicht existente EM-Banknummer spezifiziert. (Dieser Fehler tritt auf, wenn die spezifizierte EM-Bank im SPS-Setup als Dateispeicher registriert wurde.) In allen anderen Fällen AUS.

In der folgenden Tabelle sind die relevanten Flags im Systembereich aufgeführt.

Name	Adresse	Funktion
Aktuelle EM-Bank	A301	Enthält die aktuelle EM-Banknummer in hexadezimal (0000 bis 000C).

Vorsichtsmaßnahmen

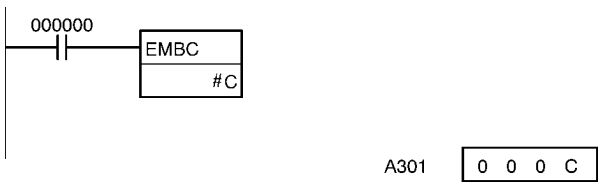
Die aktuelle, in einem zyklischen Task geänderte EM-Banknummer wird beibehalten, wenn der Betrieb zwischen den Tasks wechselt. Wenn z.B. EMBC(281) in Task 1 zum Ändern der aktuellen EM-Bank von Bank B zu Bank C verwendet wird, bleibt Bank C in allen zyklischen Tasks die aktuelle EM-Bank, auch wenn der Betrieb zu Task 2 wechselt.

Die aktuelle EM-Banknummer, die in einer Interrupt-Task geändert wurde, ist nur während der Ausführung der Interrupt-Tasks gültig, in der sie geändert wurde. Wenn die Ausführung des Interrupt-Tasks beendet ist, wird die vorherige EM-Banknummer wieder zur aktuellen EM-Banknummer.

Ein Fehler tritt auf, wenn die spezifizierte EM-Bank im SPS-Setup als Dateispeicher registriert wurde.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, wird die aktuelle EM-Banknummer in Bank C geändert und die neue Banknummer (000C hex) wird in A301 ausgegeben.

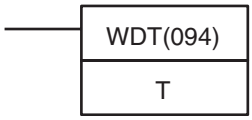


3-31-4 EXTEND MAXIMUM CYCLE TIME: WDT(094)

Funktion

Verlängert die maximale Zykluszeit, allerdings nur für den Zyklus, in dem der Befehl ausgeführt wird. WDT(094) kann zur Vermeidung von Fehlern bei langen Zykluszeiten verwendet werden, wenn aufgrund spezieller Verarbeitungen vorübergehend eine längere Zykluszeit benötigt wird.

Kontaktplansymbol



T: Zeitgebereinstellung

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	WDT(094)
	Ausführung einmal bei steigender Flanke	@WDT(094)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

T: Zeitgebereinstellung
Spezifiziert die Einstellung der Laufzeitüberwachung zwischen 0000 und 0F9F hexadezimal bzw. zwischen &0000 und &3999 dezimal.

Operanden-Spezifikationen

Bereich	T
CIO-Bereich	---
Arbeitsbereich nicht remanent	---
Arbeitsbereich remanent	---
Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	0000 bis 0F9F (binär)
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	---

Beschreibung

WDT(094) verlängert die maximale Zykluszeit für den Zyklus, in dem dieser Befehl ausgeführt wird. Die Einstellung der Laufzeitüberwachung im SPS-Setup wird um eine Zeitspanne von T × 10 ms (0 bis 39.990 ms) verlängert.

Der folgenden Tabelle sind die Einstellungen für die Laufzeitüberwachung im SPS-Setup zu entnehmen. Der Standardwert für die maximale Zykluszeit beträgt 1.000 ms, obwohl sie beliebig in 10-ms Schritten zwischen 1 und 40.000 ms eingestellt werden kann.

Name	Funktion	Einstellungen
Zykluszeit-überwachung	Ein Zykluszeit-zu-lang-Fehler (schwerwiegender Fehler) tritt auf, wenn die Zykluszeit die maximale Einstellung überschreitet.	0: Standardeinstellung (1.000 ms) 1: Anwender-Zeiteinstellung
	Stellt die maximale Zykluszeit ein. (Diese Einstellung ist nur dann gültig, wenn die erste Einstellung auf 1 gesetzt wurde.)	0001 bis 0FA0 (1 bis 40.000 ms, 10-ms-Schritte)

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn die Einstellung der Laufzeitüberwachung 40 Sekunden überschreitet. In allen anderen Fällen AUS.

In der nachfolgenden Tabelle sind die relevanten Flags und Worte im Systembereich aufgeführt.

Name	Adresse	Funktion
Zykluszeit-zu-lang-Flag	A40108	EIN, wenn die Ist-Zykluszeit die im SPS-Setup eingestellte maximale Zykluszeit (Watchdog-Zykluszeit) überschreitet. Dies ist ein schwerwiegender Fehler, der die Programmausführung abbricht.
Maximale Zykluszeit	A262 und A263	Diese Worte enthalten die maximale Zykluszeit als 32-Bit-Binärwert. Dieser Wert wird jeden Zyklus aktualisiert.
Ist-Zykluszeit	A264 und A265	Diese Worte enthalten die Ist-Zykluszeit als 32-Bit-Binärwert. Dieser Wert wird jeden Zyklus aktualisiert.

Vorsichtsmaßnahmen

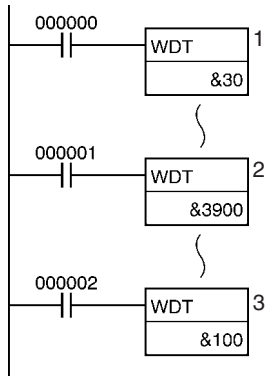
WDT(094) kann mehr als einmal pro Zyklus verwendet werden. Wenn WDT(094) mehr als einmal ausgeführt wird, werden die Zykluszeitverlängerungen addiert, obwohl die Gesamtzeit von 40.000 ms nicht überschreiten darf. WDT(094) kann nicht mehr ausgeführt werden, wenn der Zyklus bereits auf 40.000 ms verlängert wurde.

Beispiele

In diesem Beispiel wird der Standardwert für die maximale Zykluszeit (1.000 ms) verwendet.

1,2,3...

1. Wenn CIO 000000 auf EIN gesetzt wird, verlängert der erste WDT(094)-Befehl die maximale Zykluszeit um 300 ms (30 × 10 ms). Dementsprechend beträgt die maximale Zykluszeit an diesem Punkt 1.300 ms.
2. Wenn CIO 000001 auf EIN gesetzt wird, versucht der zweite WDT(094)-Befehl, die maximale Zykluszeit um weitere 39.000 ms zu verlängern. Da die neue maximale Zykluszeit (40.300 ms) die Obergrenze von 40.000 ms überschreitet, werden die übermäßigen 300 ms ignoriert. Dadurch verlängert der zweite WDT(094)-Befehl die maximale Zykluszeit tatsächlich um 38.700 ms.
3. Wenn CIO 000002 auf EIN gesetzt wird, versucht der dritte WDT(094)-Befehl die maximale Zykluszeit um weitere 1.000 ms zu verlängern. Da die neue maximale Zykluszeit die Obergrenze von 40.000 ms bereits erreicht hat, wird der dritte WDT(094)-Befehl ignoriert.

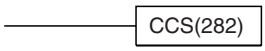


3-31-5 SAVE CONDITION FLAGS: CCS(282)

Speichert den aktuellen Status der Bedingungs-Flags in einem separaten Bereich innerhalb der CPU-Baugruppe. Der aktuelle Status der Flags wird beibehalten, so dass er mit Hilfe von CCL(283) an einer anderen Stelle des Programms, in einem anderen Task oder sogar in einem späteren Zyklus gelesen (wiederhergestellt) werden kann.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CCS(282)
	Ausführung einmal bei steigender Flanke	@CCS(282)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

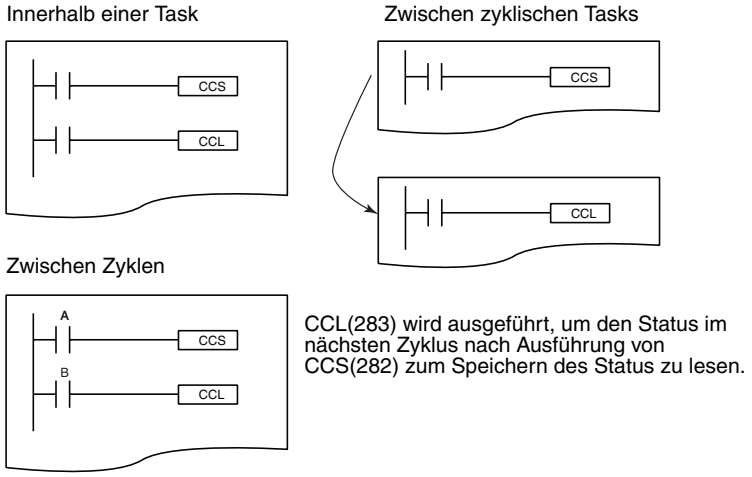
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Beschreibung

Wenn die Ausführungsbedingung EIN ist, speichert CCS(282) den aktuellen Status der Bedingungs-Flags (außer den der Immer-EIN- und Immer-AUS-Flags) in einem getrennten Bereich der CPU-Baugruppe. Der Status der folgenden Bedingungs-Flags wird erhalten: ER, CY, >, =, <, N, OF, UF, >=, <> und <=.

Der erhaltene Status der Bedingungs-Flags kann später nur mit Hilfe des Befehls LOAD CONDITION FLAGS ((CCL(283)) gelesen (wiederhergestellt) werden. Der Status kann bei einer der folgenden Gelegenheiten gelesen werden:

- Innerhalb einer Task
- Zwischen unterschiedlichen zyklischen Tasks
- Zwischen Zyklen



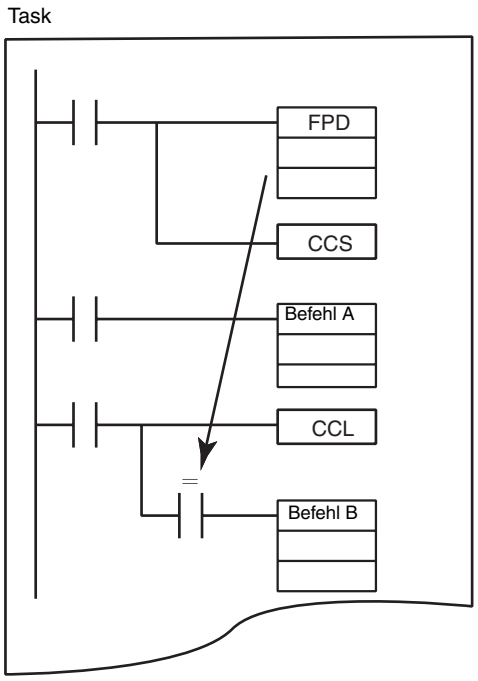
CCL(283) wird ausgeführt, um den Status im nächsten Zyklus nach Ausführung von CCS(282) zum Speichern des Status zu lesen.

Hinweis

1. Der Status des Bedingungs-Flags kann nicht zwischen einer zyklischen Task und einer Interrupt-Task gespeichert/geladen werden.
2. Bei Ausführung von CCS(282) werden die zuvor gespeicherten Bedingungs-Flag-Daten überschrieben.

Alle Bedingungs-Flags werden gelöscht, wenn der Betrieb von einer Task zur einer anderen wechselt. Verwenden Sie die Befehle CCS(282) und CCL(283) zum Speichern und Laden des Bedingungs-Flag-Status zwischen Tasks oder Zyklen.

Zum Beispiel ermöglichen die Befehle CCS(282) und CCL(283) die Verwendung des durch die Ausführung von FPD(269) gesetzten CY-Flag-Status (Zeitüberwachungs-Diagnosefehler) zu einem späteren Zeitpunkt im Programm, nicht unmittelbar nach Ausführung des Befehls.



Die Ergebnisse des Vergleichs werden in den Bedingungs-Flags gespeichert.
(In diesem Fall können die Ergebnisse des COMPARE-Befehls in Befehl B verwendet werden, auch wenn diese Ergebnisse von der Ausführung von Befehl A beeinflusst werden.)

Erhält den aktuellen Status der Bedingungs-Flags an einer separaten Stelle in der CPU-Baugruppe.

Stellt den Status der Bedingungs-Flags wieder her.

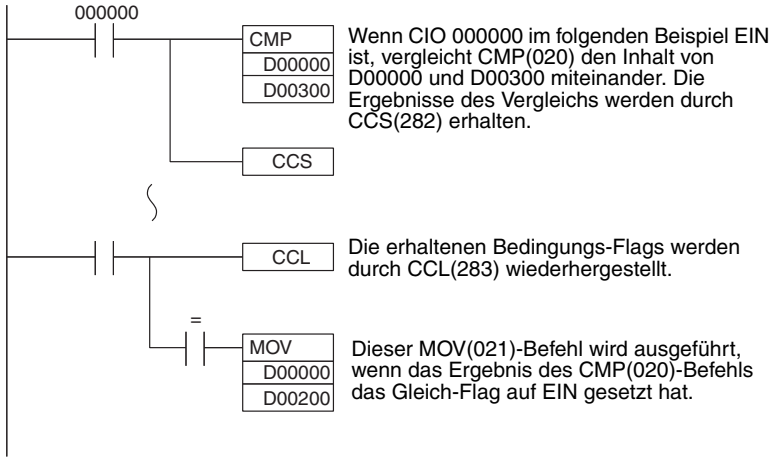
Das Gleich-Flag spiegelt das Ergebnis des COMPARE-Befehls wider, nicht das Ergebnis von Befehl A.

Flags

Diese Befehle haben keinen Einfluss auf Flags.

Beispiele

Im folgenden Beispiel erhält CCS(282) die Ergebnisse eines Vergleichs, sodass dieses Ergebnis später im Programm als Ausführungsbedingung verwendet werden kann.

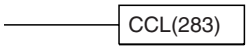


3-31-6 LOAD CONDITION FLAGS: CCL(283)

Stellt den Status der Bedingungs-Flags wieder her, die durch CCS(282) in einem separaten Bereich innerhalb der CPU-Baugruppe gespeichert wurden. CCL(283) kann auch unabhängig zum Löschen der Bedingungs-Flags verwendet werden.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	CCL(283)
	Ausführung einmal bei steigender Flanke	@CCL(283)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

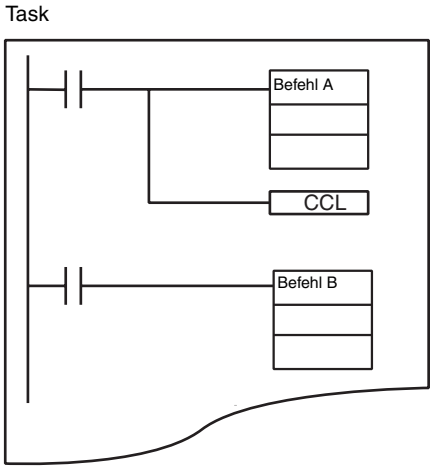
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Beschreibung

Wenn die Ausführungsbedingung EIN ist, stellt (liest) CCL(283) den Status der Bedingungs-Flags (außer den der Immer-EIN- und Immer-AUS-Flags) wieder her. Der Status der folgenden Bedingungs-Flags wird wieder hergestellt (gelesen): ER, CY, >, =, <, N, OF, UF, >=, <> und <=.

Bedingungs-Flags werden von allen Befehlen verwendet, so dass der Status dieser Flags bei der Ausführung jedes Befehls im Laufe des SPS-Zyklus oft wechseln kann. Früher war es notwendig, Bedingungen mit Hilfe der Bedingungs-Flags sofort hinter dem steuernden Befehl zu platzieren, sodass der Status der Bedingungs-Flags durch eingreifende Befehle nicht beeinträchtigt wird. Die Befehle CCS(282) und CCL(283) erlauben es, dass der steuernde Befehl von den auf dem Ergebnis basierenden Ausführungsbedingungen getrennt wird.

Zum Beispiel kann CCS(282) den Status des Gleich-Flags nach Ausführung eines Vergleichsbefehls speichern und das Ergebnis kann später wiederhergestellt werden. Das Ergebnis muss nicht sofort nach Ausführung des Befehls verwendet werden.



CCL(283) wird alleine zum Löschen der Bedingungs-Flags nach Ausführung von Befehl A verwendet, so dass diese Ergebnisse den Befehl B und spätere Befehle nicht beeinflussen.

Weitere Beispiele zur Verwendung von CCS(282) und CCL(283) finden Sie unter *3-31-5 SAVE CONDITION FLAGS: CCS(282)*.

Flags

Diese Befehle haben keinen Einfluss auf Flags.

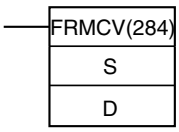
3-31-7 CONVERT ADDRESS FROM CV: FRMCV(284)

Funktion

Konvertiert eine SPS-Speicheradresse der CV-Serie in die entsprechende SPS-Speicheradresse der CS/CJ-Serie. Der Befehl FRMCV(284) kann beim Konvertieren von Programmen der CV-Serie hilfreich sein, die SPS-Speicheradressen verwenden, so dass sie mit SPS der CS/CJ-Serie kompatibel sind.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Wort, das die SPS-Speicheradresse der CV-Serie enthält

D: Ziel-Indexregister

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FRMCV(284)
	Ausführung einmal bei steigender Flanke	@FRMCV(284)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

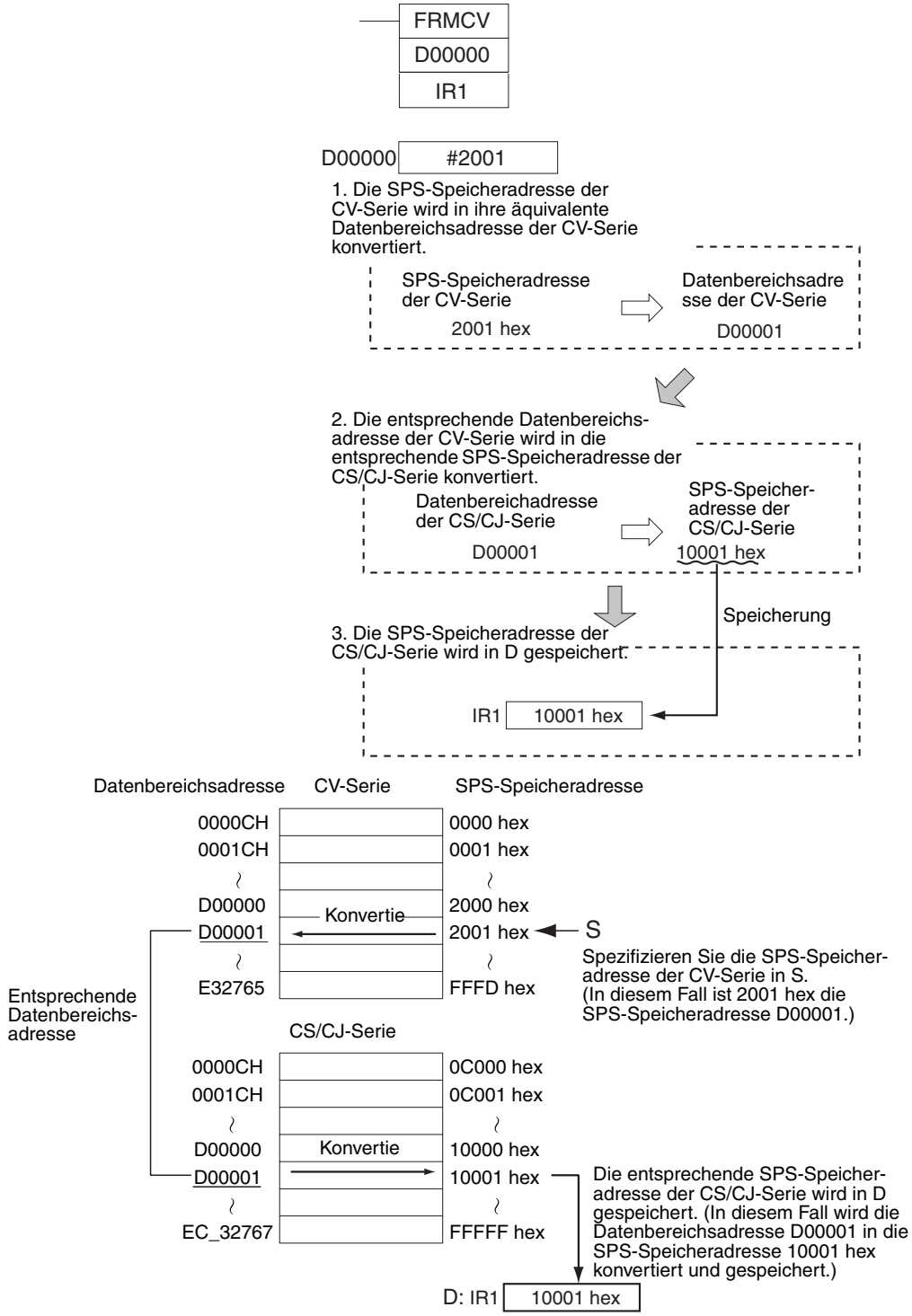
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Beschreibung

Der Befehl FRMCV(284) führt die folgenden Operationen aus, wenn die Ausführungsbedingung EIN ist.

1. Die in S spezifizierte SPS-Speicheradresse der CV-Serie wird in ihre äquivalente Datenbereichsadresse der CV-Serie konvertiert.
2. FRMCV(284) ermittelt die SPS-Speicheradresse der CS/CJ-Serie, die dieser Datenbereichsadresse der CV-Serie entspricht.
3. Die SPS-Speicheradresse der CS/CJ-Serie wird in D ausgegeben. (Für D muss ein Indexregister (IR0 bis IR15) spezifiziert werden.)

Das folgende Beispiel zeigt, wie der Befehl FRMCV(284) zur Konvertierung der SPS-Speicheradresse D00001 einer SPS der CV-Serie verwendet wird.



Hinweis Wenn bei der CS/CJ-Serie kein Äquivalent zu der spezifizierten SPS-Speicheradresse der CV-Serie gibt, tritt ein Fehler auf, das Fehler-Flag wird auf EIN gesetzt und die Adresse wird nicht konvertiert.

Wird als Operand ein Indexregister mit dem Präfix „IR“ verwendet, wird nicht das Indexregister selbst, sondern das durch die im Indexregister enthaltene SPS-Speicheradresse bezeichnete Wort als Operand verwendet. Wenn die gewünschte SPS-Speicheradresse in einem Indexregister gespeichert ist, kann das Indexregister selbst als Operand für einen Befehl verwendet werden.

Der Befehl FRMCV(284) kann zur Konvertierung eines Programms der CV-Serie mit den folgenden beiden Arten von Programmierung verwendet werden, damit dieses Programm bei einer SPS der CS/CJ-Serie eingesetzt werden kann. Siehe *Beispiele* im weiteren Verlauf dieses Abschnitts.

1. Wenn die indirekte DM-Adressierung im Binärmodus (*DM) verwendet wird
   (zur indirekten Spezifizierung einer Datenbereichsadresse bei einer SPS-Speicheradresse im DM-Bereich)
2. Wenn SPS-Speicheradressen der CV-Serie direkt als Wert verwendet werden
   (zum Speichern von SPS-Speicheradressen in Indexregistern mit direkter Adressierung unter Verwendung eines Befehls wie z.B. MOV(021))

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	---
Arbeitsbereich nicht remanent	W000 bis W511	---
Arbeitsbereich remanent	H000 bis H511	---
Systembereich	A448 bis A959	---
Zeitgeberbereich	T0000 bis T4095	---
Zählerbereich	C0000 bis C4095	---
DM-Bereich	D00000 bis D32767	---
EM-Bereich ohne Bank	E00000 bis E32767	---
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	---
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	---
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	---
Konstanten	Beliebige Konstanten, außer 09FF hex, 0A00 bis 0AFF hex und 0D00 bis 0E3F hex	---
Datenregister	DR0 bis DR15	---
Indexregister	---	IR0 bis IR15
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15	---

Flags

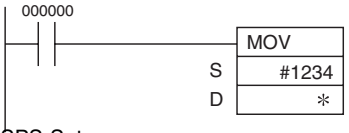
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn S eine der folgenden SPS-Speicheradressen spezifiziert, die bei der CS/CJ-Serie nicht vorhanden sind: Temporärer Relaisbereich (TR-Bereich) (09FF hex) CPU-Bus-Link-Bereich (G-Bereich) (0A00 bis 0AFF hex) SFC-Bereiche (0D00 bis 0E3F hex) In allen anderen Fällen AUS.

Beispiele

Beispiel 1: Konvertieren eines Programms der CV-Serie mit indirekter *DM-Adressierung im Binärmodus

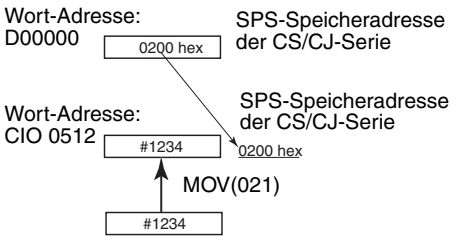
In diesem Beispiel für den Befehl FRMCV(284) ist ein DM-Wort in S spezifiziert, die SPS-Speicheradresse ist in einem Indexregister gespeichert und das Indexregister wird zur indirekten Adressierung verwendet.

- Programm der CV-Serie
  (das Programm verwendet die indirekte DM-Adressierung im Binärmodus)



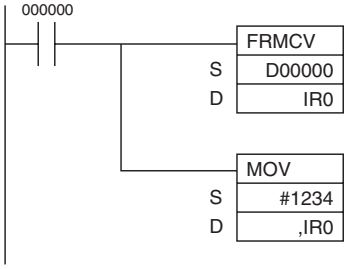
SPS-Setup
Indirekte DM-Daten:
Wenn indirekte DM-Adressen im Binärformat verwendet werden, wird der Inhalt des DM-Worts als eine SPS-Speicheradresse behandelt, die die entsprechende Adresse im E/A-Speicher spezifiziert.

Der Wert in D00000 ist in diesem Fall 0200 hex. Die entsprechende Datenbereichsadresse lautet CIO 0512. Somit wird #1234 nach CIO 0512 übertragen.

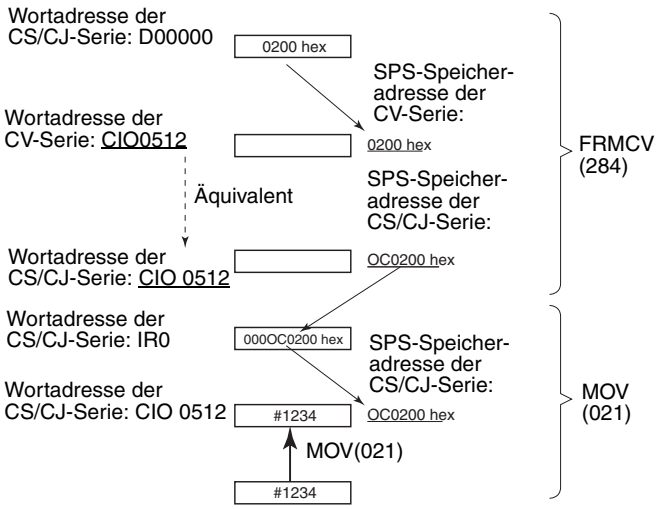


Äquivalentes Programm

- Programm der CS/CJ-Serie



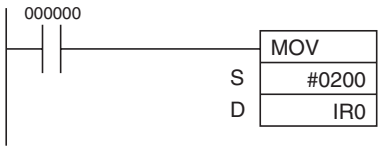
Der Wert in D00000 ist in diesem Fall 0200 hex. Die entsprechende Datenbereichsadresse der CV-Serie lautet CIO 0512. Die SPS-Speicheradresse der CS/CJ-Serie für CIO 0512 lautet 0000C200 hex, somit wird dieser Wert in IR0 gespeichert. Der Zieloperand in MOV(021) adressiert den Inhalt von IR0 indirekt, sodass #1234 nach CIO 0512 übertragen wird.



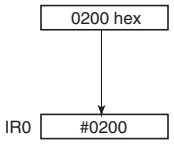
Beispiel 2: Konvertieren eines Programms der CV-Serie, wobei die SPS-Speicheradresse direkt im Indexregister gespeichert ist.

In diesem Beispiel für den Befehl FRMCV(284) wird die SPS-Speicheradresse der CV-Serie direkt in S spezifiziert.

- Programm der CV-Serie
  (Das Programm verwendet SPS-Speicheradressen, die direkt in IR gespeichert sind)

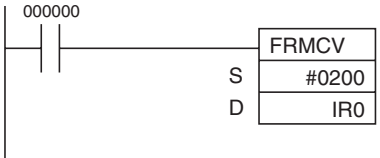


In diesem Fall ist die SPS-Speicheradresse 0200 hex im Indexregister IR0 gespeichert.

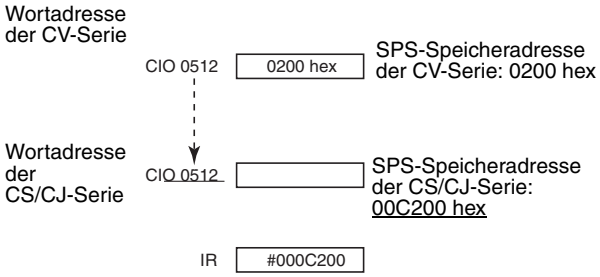


Äquivalentes Programm

- Programm der CS/CJ-Serie



In diesem Fall entspricht die SPS-Speicheradresse 0200 hex der CV-Serie der Wortadresse CIO 0512. Die SPS-Speicheradresse der CS/CJ-Serie für CIO 0512 lautet 0000C200 hex. Somit wird dieser Wert in IR0 gespeichert.



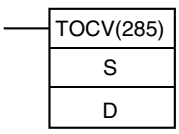
3-31-8 CONVERT ADDRESS TO CV: TOCV(285)

Funktion

Konvertiert die SPS-Speicheradresse der CS/CJ-Serie zur entsprechenden SPS-Speicheradresse der CV-Serie. TOCV(285) kann beim Konvertieren von Programmen der CS/CJ-Serie, die SPS-Speicheradressen verwenden, hilfreich sein, damit diese Programme mit SPS der CV-Serie kompatibel sind.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H, CJ1M und CS1D unterstützt.

Kontaktplansymbol



S: Indexregister, das die SPS-Speicheradresse der CS/CJ-Serie enthält

D: Zielwort

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TOCV(285)
	Ausführung einmal bei steigender Flanke	@TOCV(285)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

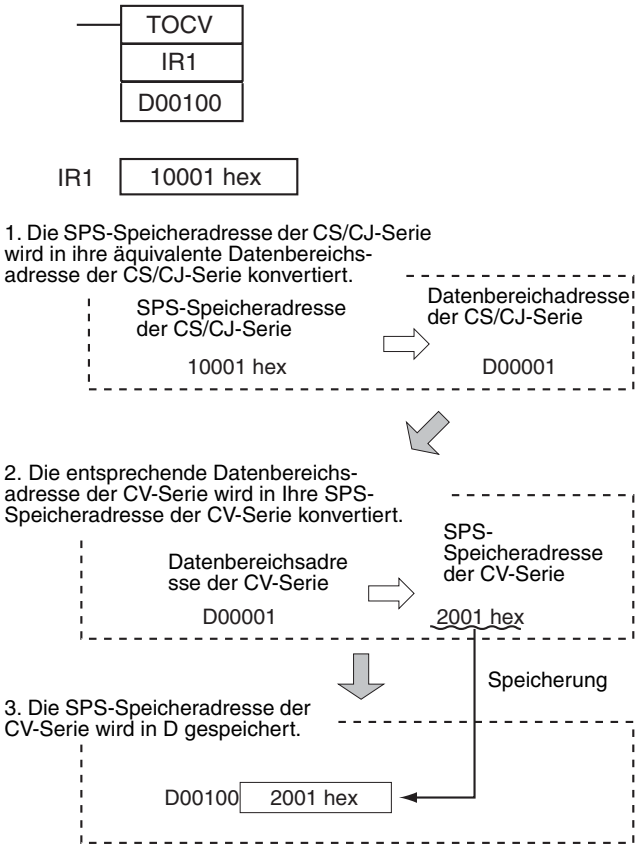
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

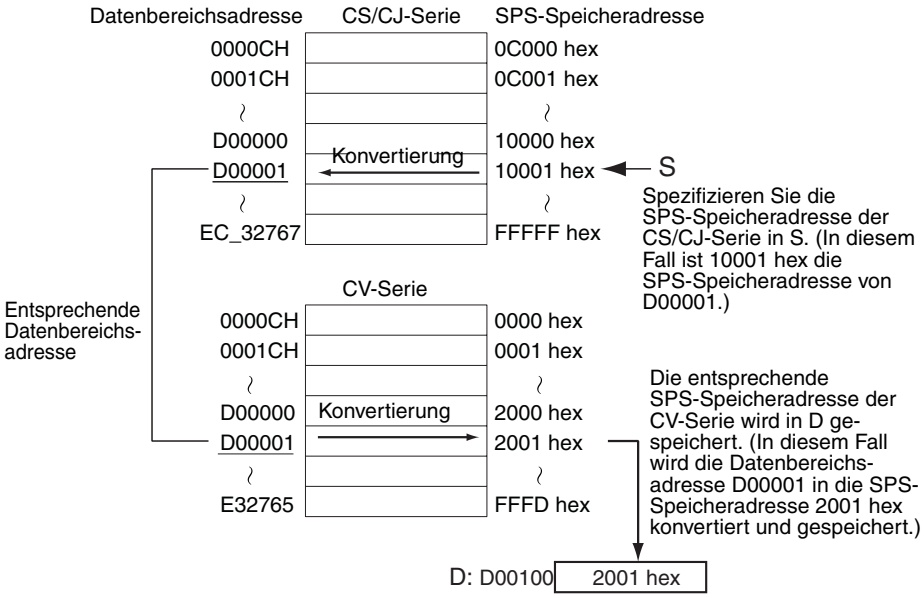
Beschreibung

Der Befehl TOCV(285) führt die folgenden Operationen aus, wenn die Ausführungsbedingung EIN ist.

1. Die in S spezifizierte SPS-Speicheradresse der CS/CJ-Serie wird in ihre äquivalente Datenbereichsadresse der CS/CJ-Serie konvertiert. (Für S muss ein Indexregister (IR0 bis IR15) spezifiziert sein.)
2. TOCV(284) ermittelt die SPS-Speicheradresse der CV-Serie, die der gleichen Datenbereichsadresse der CS/CJ-Serie entspricht.
3. Die SPS-Speicheradresse der CV-Serie wird in D ausgegeben.

Das folgende Beispiel zeigt die Verwendung von TOCV(285) zur Konvertierung der SPS-Speicheradresse D00001 der CS/CJ-Serie.





Hinweis

1. Wenn bei der CV-Serie kein Äquivalent zur spezifizierten SPS-Speicheradresse der CS/CJ-Serie gibt, tritt ein Fehler auf, das Fehler-Flag wird auf EIN gesetzt und die Adresse wird nicht konvertiert.
2. Die durch TOCV(285) gespeicherte SPS-Speicheradressendaten der CV-Serie kann mit Hilfe von CX-Programmer an eine SPS der CV-Serie übertragen werden.
3. Dieselbe Datenbereichsadresse, die im Programm der CS/CJ-Serie verwendet wurde, kann im Programm der CV-Serie spezifiziert werden, indem die indirekte Adressierung mittels Indexregister (Präfix „IR“) oder die indirekte DM-Adressierung im Binärmodus (*DM) verwendet wird.

Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A448 bis A959
Zeitgeberbereich	---	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	Siehe Hinweis 1	---
Datenregister	---	DR0 bis DR15

Bereich	S	D
Indexregister	IR0 bis IR15	---
Indirektes Adressieren mittels Indexregistern	---	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15

Hinweis

1. Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn S eine der folgenden SPS-Speicheradressen spezifiziert, die bei der CV-Serie nicht vorhanden sind.

Bereich oder Adresse	SPS-Speicheradresse
Task-Flag-Bereich	0000 B800 bis 0000 B801 hex
A512 bis A959	0000 BA40 bis 0000 BBFF hex
CIO 2556 bis CIO 6143	0000 C9FC bis 0000 D7FF hex
T1024 bis T4095	0000 BE40 bis 0000 BEFF hex und 0000 E400 bis 0000 EFFF hex
C1024 bis C4095	0000 BF40 bis 0000 BFFF hex und 0000 F400 bis 0000 FFFF hex
HR-Bereich	0000 D800 bis 0000 D9FF hex
WR-Bereich	0000 DE00 bis 0000 DFFF hex
D24576 bis D32767	0001 6000 bis 0001 7FFF hex
EM-Bank-Spezifikation	0001 8000 bis 000F 7FFF hex
E32766 bis D32767	000F FFFE bis 000F FFFF hex

2. Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn für S ein anderer Bereich als der des Indexregisters spezifiziert wird.

Flags

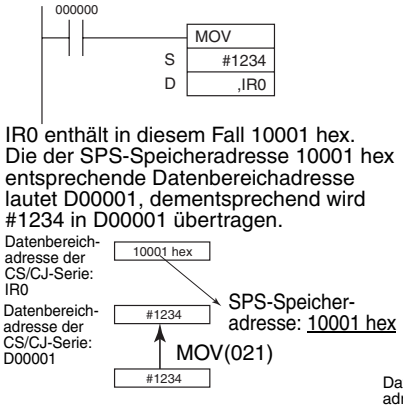
Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn S eine SPS-Speicheradresse spezifiziert, die in den SPS der CV-Serie nicht vorhanden ist. EIN, wenn S keine Konstante oder Indexregister ist. In allen anderen Fällen AUS.

Beispiel

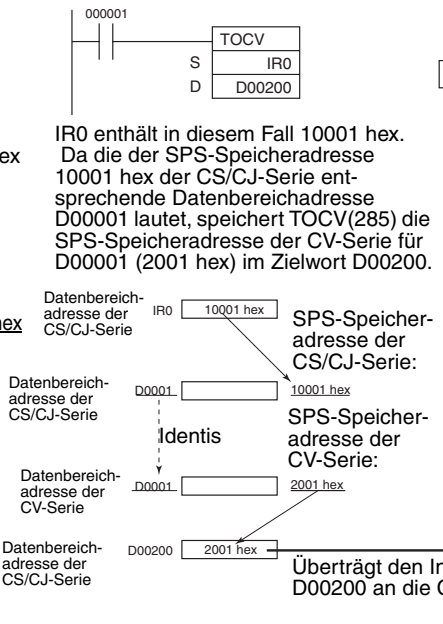
Konvertieren eines Programms der CS/CJ-Serie mit indirekter Indexregister-Adressierung

1. In diesem Beispiel für die Verwendung des Befehls TOCV(285) wird in S ein Indexregister spezifiziert. Die SPS-Speicheradresse der CS/CJ-Serie in diesem Indexregister wird in ihr Äquivalent der CV-Serie konvertiert.
2. Die SPS-Speicheradresse der CV-Serie wird in die spezifizierte Datenbereichadresse übertragen.
3. Verwenden Sie die SPS-Speicheradresse der CV-Serie im Programm der CV-Serie.

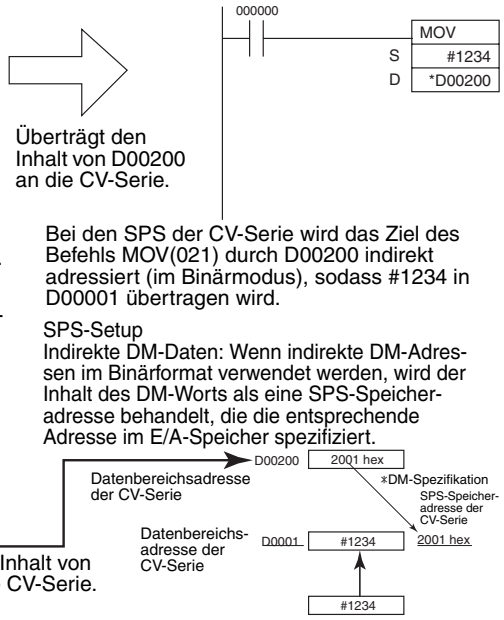
- Programm der CS/CJ-Serie (das Programm verwendet indirekte Indexregister-Adressierung)



- Programm der CS/CJ-Serie



- Programm der CV-Serie



3-31-9 DISABLE PERIPHERAL SERVICING: IOSP(287) (nur CS1-H/CJ1-H/CJ1M)

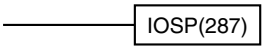
Funktion

Deaktiviert die Bedienung von Peripheriegeräten während der Programmausführung im Parallelverarbeitungs- oder im Peripherie-Prioritätsmodus.

Weitere Informationen über den Parallelverarbeitungs- und Peripherie-Prioritätsmodus sind in *Kapitel 6, Erweiterte Funktionen*, im *CS/CJ-Serie Speicherprogrammierbare Steuerungen Programmierhandbuch* zu finden.

Hinweis Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M unterstützt. Er kann nicht mit den CPU-Baugruppen der SPS-Systeme CS1, CJ1 oder CS1D verwendet werden.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	IOSP(287)
	Ausführung einmal bei steigender Flanke	@IOSP(287)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

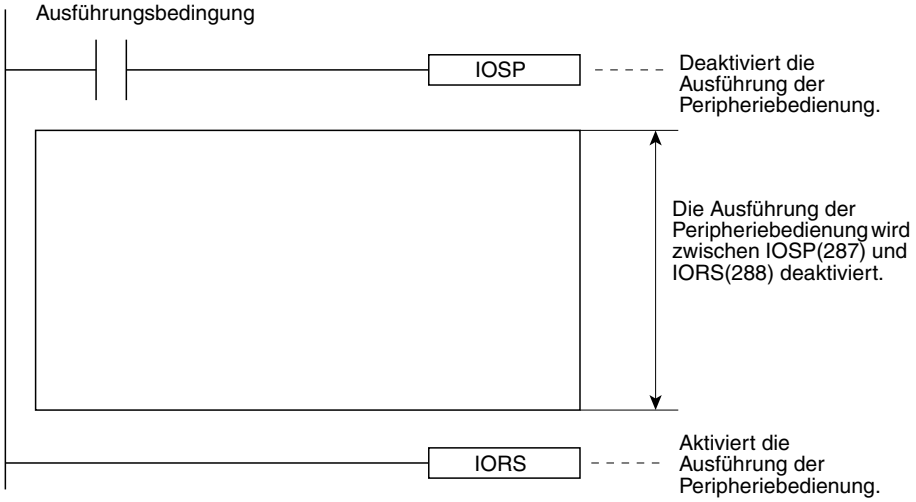
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	Nicht zulässig

Beschreibung

Verwenden Sie den Befehl IOSP(287) in einer zyklischen Task im Parallelverarbeitungsmodus (mit synchronem oder asynchronem Speicherzugriff), um folgenden Arten der Bedienung von Peripheriegeräten zu deaktivieren. Die Bedienung von Peripheriegeräten wird wieder aktiviert, wenn der Befehl ENABLE PERIPHERAL SERVICING (IORS(288)) ausgeführt wird.

- Ereignisbedienung bei Spezial-E/A-Baugruppen
- Ereignisbedienung bei CPU-Bus-Baugruppen

- Bedienung der Peripherieschnittstelle
- Bedienung der RS-232C-Schnittstelle
- Ereignisbedienung für interne Sondermodule (nur CS-Serie)
- Ereignisbedienung (einschl. Hintergrund-Befehlsverarbeitung), die eine Kommunikationsschnittstellennummer verwendet, d. h., eine interne logische Schnittstelle.



Wenn die Peripheriebedienung durch IOSP(287) deaktiviert wurde, bleibt sie deaktiviert, bis IORS(288) oder END(001) ausgeführt wird oder der SPS-Betrieb gestoppt wird.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn IOSP(287) in einer Interrupt-Task ausgeführt wird. In allen anderen Fällen AUS.

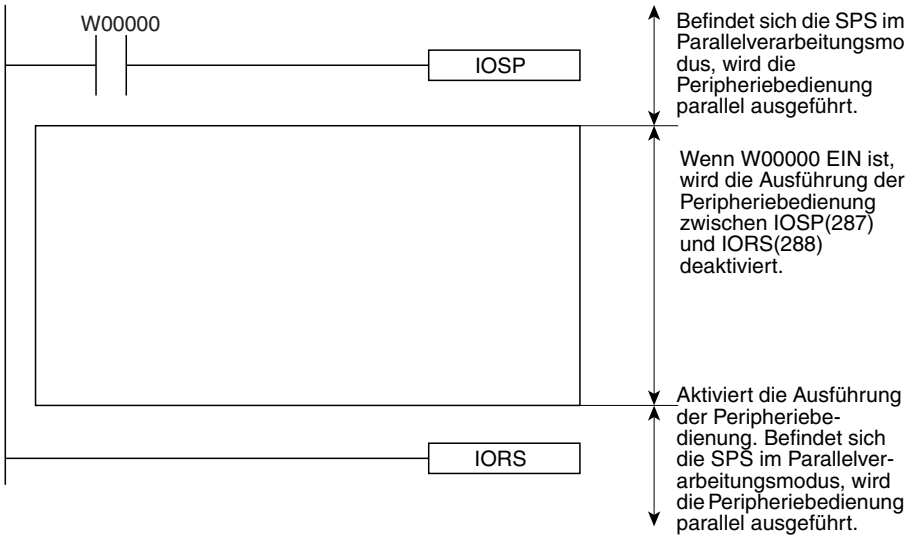
Vorsichtsmaßnahmen

IOSP(287) kann nicht in einer Interrupt-Task ausgeführt werden. Wenn IOSP(287) in einer Interrupt-Task ausgeführt wird, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

IOSP(287) kann die Peripheriebedienung in nicht mehr als einer Task deaktivieren. Wenn die Peripheriebedienung in mehr als einer Task deaktiviert werden soll, muss IOSP(287) separat in jeder Task programmiert werden.

Beispiel

Das folgende Beispiel zeigt, wie die Befehle verwendet werden IOSP(287) und IORS(288), um die Peripheriebedienung in einem Programmabschnitt zu deaktivieren.



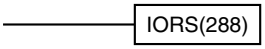
3-31-10 ENABLE PERIPHERAL SERVICING: IOSP(288) (nur CS1-H/CJ1-H/CJ1M)

Funktion

Aktiviert die Peripheriebedienung während der Programmausführung im Parallelverarbeitungsmodus, die durch den Befehl DISABLE PERIPHERAL SERVICING (IOSP(287)) deaktiviert wurde.

Dieser Befehl wird nur durch die CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M unterstützt.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	IORS(288)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	Nicht zulässig

Beschreibung

Verwenden Sie IORS(288) in einer zyklischen Task, um die Deaktivierung der Peripheriebedienung durch den Befehl DISABLE PERIPHERAL SERVICING (IOSP(287)) aufzuheben.

Es ist nicht erforderlich, IORS(288) mit einer Ausführungsbedingung zu programmieren.

IORS(288) kann nicht in einer Interrupt-Task ausgeführt werden. Wenn IORS(288) in einer Interrupt-Task ausgeführt wird, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn IOSP(288) in einer Interrupt-Task ausgeführt wird. In allen anderen Fällen AUS.

3-32 Blockprogrammierbefehle

In diesem Abschnitt werden Blockprogramme und die Blockprogrammierbefehle beschrieben.

Befehl	AWL	Funktionscode	Seite
BLOCK PROGRAM BEGIN	BPRG	096	1128
BLOCK PROGRAM END	BEND	801	1128
BLOCK PROGRAM PAUSE	BPPS	811	1131
BLOCK PROGRAM RESTART	BPRS	812	1131
CONDITIONAL BLOCK EXIT (NOT)	EXIT (NOT)	806	1137
IF (NOT)	IF (NOT)	802	1133
ELSE	ELSE	803	1133
IF END	IEND	804	1133
ONE CYCLE AND WAIT (NOT)	WAIT (NOT)	805	1140
TIMER WAIT	TIMW (BCD)	813	1144
	TIMWX (binär)	816	
COUNTER WAIT	CNTW (BCD)	814	1147
	CNTWX (binär)	818	
HIGH-SPEED TIMER WAIT	TMHW (BCD)	817	1150
	TMHWX (binär)	815	
LOOP	LOOP	809	1153
LOOP END (NOT)	LEND (NOT)	810	1153

3-32-1 Einführung

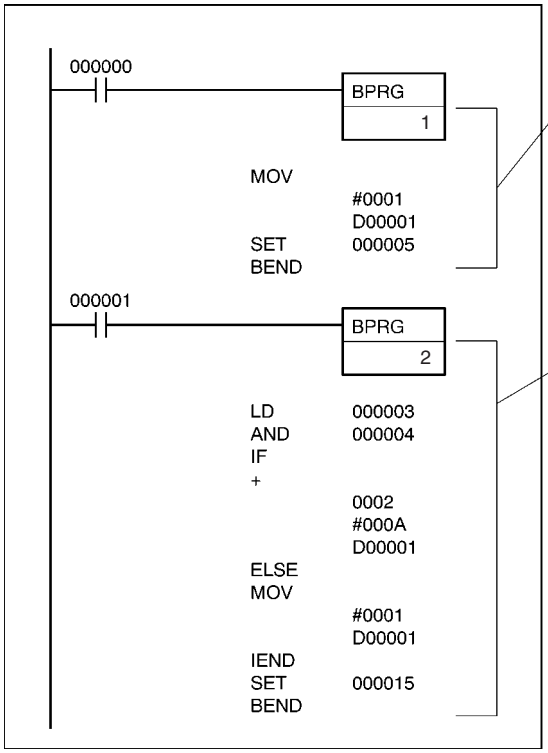
Blockprogramme

Bei der CS/CJ-Serie können bis zu 128 Blockprogramme innerhalb des gesamten Anwenderprogramms (alle Tasks) verwendet werden. Die Ausführung jedes Blockprogramms wird durch eine einzelne Ausführungsbedingung gesteuert. Alle Befehle zwischen BPRG(096) und BEND(801) werden bedingungslos ausgeführt, wenn die Ausführungsbedingung für BPRG(096) auf EIN gesetzt wird. Die Ausführung aller Blockprogrammierbefehle außer BPRG(096) wird nicht vom Status der Ausführungsbedingung beeinflusst. Dadurch kann die Programmierung, die durch eine einzelne Ausführungsbedingung ausgeführt werden soll, in einem Blockprogramm zusammengefasst werden.

Jeder Block wird durch eine Ausführungsbedingung im Kontaktplan gestartet, und alle Befehle innerhalb des Blocks werden in AWL-Form geschrieben. Das Blockprogramm ist somit eine Kombination von Kontaktplan- und AWL-Befehlen.

Blockprogramme aktivieren Programmoperationen, die mit Kontaktplänen schwierig zu programmieren sein können, wie z.B. bedingte Verzweigungen und Schrittprogressionen.

Im folgenden Beispiel sind zwei Blockprogramme dargestellt.



Blockprogrammbereich Nr. 1

Wenn CIO 000000 EIN ist, wird der Inhalt von Blockprogramm 1 ausgeführt. Die Befehle MOV(021) und SET werden bedingungslos ausgeführt und das Blockprogramm wird beendet.

Blockprogrammbereich Nr. 2

Wenn CIO 000001 EIN ist, wird der Inhalt von Blockprogramm 2 ausgeführt. Wenn CIO 000003 und CIO 000004 beide EIN sind, wird die binäre Addition ausgeführt (CIO 0002 + #000A→D00001).
Wenn eines oder beide dieser Bits AUS sind, wird #0001 in D00001 übertragen. CIO 000015 wird dann bedingungslos gesetzt und das Blockprogramm wird beendet.

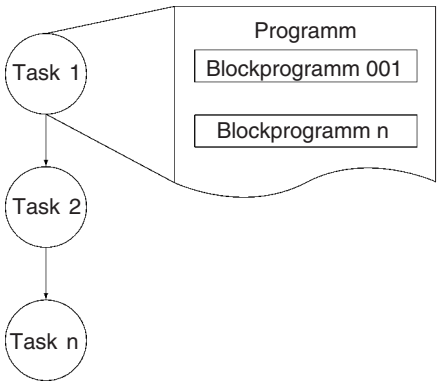
Tasks und Blockprogramme

Blockprogramme können innerhalb von Tasks platziert werden. Während Tasks zum Teilen von großen Programmiereinheiten verwendet werden, können Blockprogramme innerhalb von Tasks zur weiteren Aufteilung von Programmiereinheiten in kleinere Einheiten benutzt werden, die mit einer einzelnen Kontaktplan-Ausführungsbedingung gesteuert werden.

Ähnlich wie bei Tasks benötigen nicht ausgeführte Blockprogramme (also solche mit einer auf AUS gesetzten Ausführungsbedingung) keine Ausführungszeit und können deshalb (ähnlich wie Sprungbefehle) zur Reduzierung der Zykluszeit verwendet werden. Genau wie dies bei Tasks der Falls ist, können auch andere Blöcke aus einem Blockprogramm heraus auf Pause gesetzt oder neu gestartet werden.

Dennoch gibt es Unterschiede zwischen Tasks und Blockprogrammen. Ein Unterschied ist, dass Blockprogramme keine Eingangsbedingungen verwenden, es sei denn, sie wurden absichtlich mit IF(802), WAIT(805), EXIT(806), IEND(810) oder anderen Befehlen programmiert. Es gibt außerdem einige Befehle, die nicht in Blockprogrammen verwendet werden können, wie z.B. solche, die zur Erkennung von steigenden oder fallenden Flanken verwendet werden.

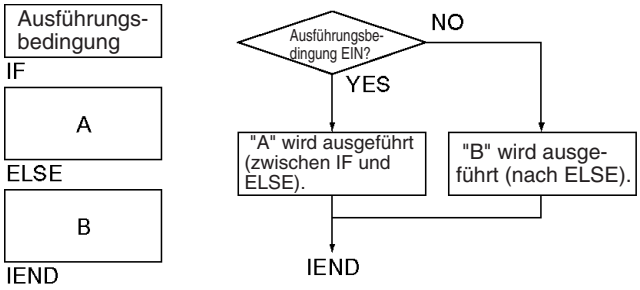
Blockprogramme können entweder in zyklischen Tasks oder Interrupt-Tasks verwendet werden. Jede Blockprogrammnummer von 0 bis 127 kann nur einmal verwendet werden. Eine erneute Verwendung darf selbst in einer anderen Task nicht stattfinden.



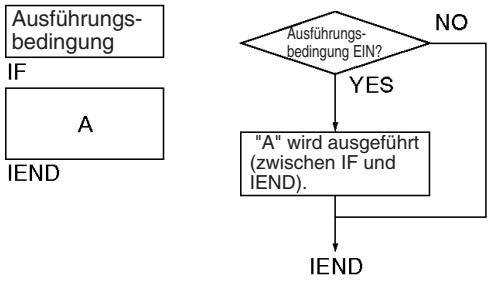
Verwendung von Blockprogrammierbefehlen

IF(802), ELSE(803) und IEND(810) werden im Wesentlichen für Ausführungsbedingungen (zusammen mit Bits) in Blockprogrammen verwendet.

Wenn „A“ oder „B“ ausgeführt werden soll, dann werden IF A ELSE B IEND wie nachstehend gezeigt verwendet.



Wenn „A“ oder nichts ausgeführt werden soll, dann werden IF A IEND wie nachstehend gezeigt verwendet.



Wenn die Ausführung warten soll, bis eine Ausführungsbedingung oder ein Bit auf EIN gesetzt ist (z.B. für Schrittprogressionen), wird WAIT(805) verwendet.

Wenn die Ausführung für eine spezifizierte Zeitspanne warten soll (z.B. für zeitgesteuerte Schrittprogressionen), wird TIMW(813), TIMX(816), TMHW(815) oder TMHWX(817) verwendet.

Wenn die Ausführung warten soll, bis ein bestimmter Zählwert erreicht ist (z.B. für Schrittprogressionen mit Zählern), wird CNTW(814)/CNTWX(818) verwendet.

Wenn die Ausführung innerhalb eines Teils eines Blockprogramms wiederholt werden soll, bis eine Bedingung erfüllt ist, werden LOOP(809) und LEND(810) verwendet.

Wenn die Ausführung des Blockprogramms in der Mitte auf Grundlage einer Ausführungsbedingung beendet werden soll, wird EXIT(806) verwendet.

Wenn ein anderes, ausgeführtes Blockprogramm aus einem Blockprogramm heraus auf Pause gesetzt oder neu gestartet werden soll, werden BPPS(811) und BPRS(812) verwendet.

Befehle innerhalb von Blockprogrammen, die Ausführungsbedingungen verwenden

Die folgenden Befehle können innerhalb eines Blockprogramms Ausführungsbedingungen verwenden.

Befehlstyp	Befehlsname	AWL
Blockprogrammierbefehle	IF (NOT)	IF(802) (NOT)
	ONE CYCLE AND WAIT (NOT)	WAIT(805) (NOT)
	EXIT	EXIT(806) NOT
	LOOP END	LEND(810) NOT
Kontaktplanbefehle	CONDITIONAL JUMP	CJP(510)
	CONDITIONAL JUMP NOT	CJPN(511)

Befehle mit Anwendungsbeschränkungen innerhalb von Blockprogrammen

Die in der folgenden Tabelle aufgelisteten Befehle können nur zur Erstellung von Ausführungsbedingungen für IF(802), WAIT(805), EXIT(806), LEND(810), CJP(510 oder CJPN(511) verwendet werden, die nicht von ihnen selbst genutzt werden können. Die Ausführung dieser Befehle kann unvorhersehbar sein, wenn sie von ihnen selbst oder in Kombination mit anderen Befehlen verwendet werden.

AWL	Name
LD/LD NOT	LOAD/LOAD NOT
AND/AND NOT	AND/AND NOT
OR/OR NOT	OR/OR NOT
UP/DOWN	CONDITION ON/CONDITION OFF
>, <,=, >=, <=, <> (S) (L)	Symbolvergleichsbefehl (nicht auf der rechten Seite zu programmierende Befehle)
LD TST/TST NOT	LOAD-Bit-Test-Befehle
AND TST/TST NOT	AND-Bit-Test-Befehle
OR TST/TST NOT	OR-Bit-Test-Befehle
>$, <$,=$, >=$, <=$, <>$	Textzeichenketten-Vergleichsbefehle

Gutes Beispiel

```
LD    000000
AND   000100
TST   D00000   #0010
IF
```

Als Ausführungsbedingung für IF verwendet.

Schlechtes Beispiel

```
LD    000000
AND   000100
TST   D00000   #0010
MOV   #0000    0010
```

Kann nicht als Ausführungsbedingung für MOV(021) verwendet werden.

Befehle, die nicht in Blockprogrammen verwendet werden können

Die in der folgenden Tabelle aufgezählten Befehle können nicht in Blockprogrammen verwendet werden.

Befehlstyp	AWL	Name	Alternative
Sequenzausgabebefehle	OUT	OUTPUT	Verwendung von SET und RSET
	OUT NOT	OUTPUT NOT	
	DIFU(013)	DIFFERENTIATE UP	Keine
	DIFD(014)	DIFFERENTIATE DOWN	Keine
	KEEP(011)	KEEP	Keine

Befehlstyp	AWL	Name	Alternative
Sequenz-Steuer-befehle	FOR(512) und NEXT(513)	FOR-NEXT LOOPS	Verwendung von LOOP(809) und LEND(810) (NOT)
	BREAK(514)	BREAK LOOP	
	IL(002) und ILC(003)	INTERLOCK und INTERLOCK CLEAR	Aufteilung des Blockprogramms in kleinere Blöcke
	JMP(004)0 und JME(005) 0	MULTIPLE JUMP and MULTIPLE JUMP END	Verwendung von JMP(004) und JME(005) (jedoch erfolgt der Sprung bedingungslos)
	END(001)	END	Verwendung von BEND(801)
Zeitgeber- und Zähler-befehle	TIM	TIMER	Verwendung von TIMW(813), TIMWX(816), TMHW(815), TMHWX(817), CNTW(814) und CNTWX(818). Andere Befehle im Blockprogramm werden so lange nicht ausgeführt, bis Zeitgeber oder Zähler abgelaufen sind.
	TIMH(015)	HIGH-SPEED TIMER	
	TMHH(540)	ONE-MS TIMER	
	TTIM(087)	ACCUMULATIVE TIMER	
	TIML(542)	LONG TIMER	
	MTIM(543)	MULTI-OUTPUT TIMER	
	CNT	COUNTER	
	CNTR(012)	REVERSIBLE COUNTER	
Unterpro-gramm-befehle	SBN(092) und RET(093)	SUBROUTINE ENTRY und SUBROUTINE RETURN	Keine
Verschie-bungs-befehle	SFT(010)	SHIFT REGISTER	Verwendung anderer Verschiebungsbefehle
Schritt-Befehle	STEP(008) und SNXT(009)	STEP und STEP NEXT	Verwendung von WAIT(805)
Daten-steuerungs-befehle	PID(190)	PID CONTROL	Keine
Diagnose-befehle	FPD(269)	FAILURE POINT DETECTION	Keine
Befehle mit Über-wachung auf steigende und fallende Flanke	AWL mit @	Befehle mit Überwachung auf steigende Flanke	Keine
	AWL mit %	Befehle mit Überwachung auf fallende Flanke	Keine

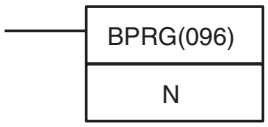
3-32-2 BLOCK PROGRAM BEGIN/END: BPRG(096)/BEND(801)

Funktion

Definiert einen Blockprogrammierbereich. Für jeden BPRG(096)-Befehl muss es einen entsprechenden BEND(801)-Befehl geben.

Kontaktplansymbole

BLOCK PROGRAM BEGIN



N: Blockprogramm-Nummer

BLOCK PROGRAM END

BEND(801)

Variationen

BPRG(096)

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	BPRG(096)
	Ausführung einmal bei steigender Flanke	Nicht unterstützt
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

BEND(801)

Variationen	**Wird im Blockprogramm stets ausgeführt**

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
(Siehe Hinweis.)	OK	OK	OK

Hinweis BPRG(096) ist nur einmal zu Beginn jedes Blockprogramms zulässig.

Operanden

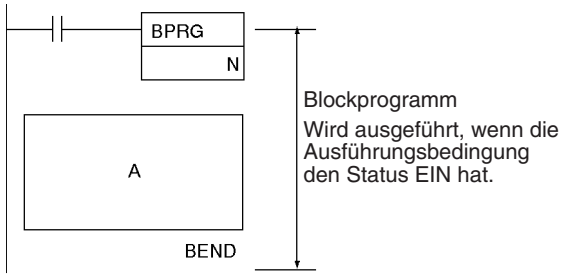
N: Blockprogramm-Nummer
Die Blockprogramm-Nummer muss zwischen 0 und 127 (dezimal) liegen.

Operanden-Spezifikationen (BPRG(096))

Bereich	N
CIO-Bereich	---
Arbeitsbereich nicht remanent	---
Arbeitsbereich remanent	---
Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	0 bis 127 (dezimal)
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	---

Beschreibung

BPRG(096) führt das Blockprogramm mit der in N spezifizierten Blocknummer aus, d.h. das unmittelbar auf den Befehl folgende und mit BEND(801) endende Blockprogramm. Alle Befehle zwischen BPRG(096) und BEND(801) werden mit EIN-Ausführungsbedingungen ausgeführt (also bedingungslos).



Wenn die Ausführungsbedingung für BPRG(096) AUS ist, wird das Blockprogramm nicht ausgeführt und für die Befehle im Blockprogramm ist keine Ausführungszeit erforderlich.

Die Ausführung des Blockprogramms kann mit BPPS(811) aus einem anderen Blockprogramm heraus gestoppt werden, auch wenn die Ausführungsbedingung für BPRG(096) EIN ist.

Flags

BPRG(096)

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn BPRG(096) bereits ausgeführt wird. EIN, wenn N nicht zwischen 0 und 127 liegt. EIN, wenn die selbe Blockprogramm-Nummer mehr als einmal verwendet wird. In allen anderen Fällen AUS.

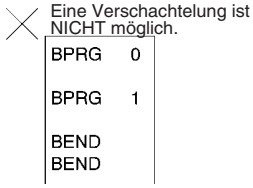
BEND(801)

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn ein Blockprogramm nicht ausgeführt wird. In allen anderen Fällen AUS.

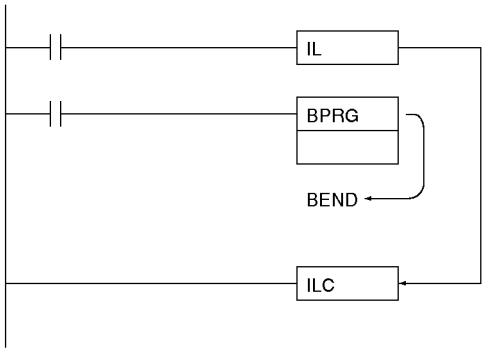
Vorsichtsmaßnahmen

Jede Blockprogramm-Nummer kann innerhalb des gesamten Anwenderprogramms nur einmal verwendet werden.

Blockprogramme können nicht verschachtelt werden.



Wenn das Blockprogramm ein verriegelter Programmbereich ist und die Ausführungsbedingung für IL(002) auf AUS gesetzt ist, wird das Blockprogramm nicht ausgeführt.

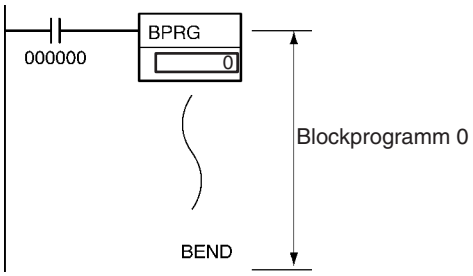


Der Befehl BPRG(096) und der zugehörige Befehl BEND(801) müssen in der gleichen Task sein.

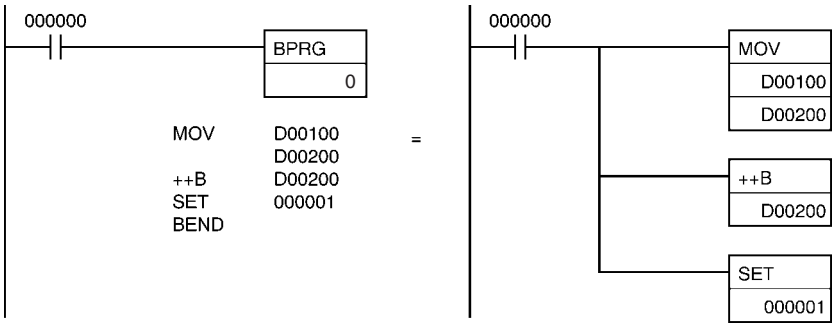
Ein Fehler tritt auf und der das Fehler-Flag wird auf EIN gesetzt, wenn sich BPRG(096) in der Mitte eines Blockprogramms befindet, wenn sich BEND(801) nicht in einem Blockprogramm befindet, wenn N nicht zwischen #0000 und #007F (binär) liegt, wenn kein Blockprogramm vorhanden ist oder wenn dieselbe Blockprogramm-Nummer mehr als einmal verwendet wird.

Beispiele

Wenn CIO 000000 im folgenden Beispiel auf EIN gesetzt wird, wird Blockprogramm 0 ausgeführt. Wenn CIO 000000 AUS ist, wird das Blockprogramm nicht ausgeführt.



Die zwei nachfolgend gezeigten Programmabschnitte führen beide MOV(021), ++B(594) sowie SET bei der selben Ausführungsbedingung aus (wenn CIO 000000 auf EIN gesetzt wird).



3-32-3 BLOCK PROGRAM PAUSE/RESTART: BPPS(811)/BPRS(812)

Funktion

Anhalten des spezifizierten Blockprogramms und Neustart des Blockprogramms aus einem anderen Blockprogramm heraus.

Kontaktplansymbol

BPPS(811)	N	**N**: Blockprogramm-Nummer
BPRS(812)	N	

Variationen

Variationen	Wird im Blockprogramm stets ausgeführt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Hinweis BPRG(096) und BPRS(812) müssen in Blockprogrammierbereichen verwendet werden, auch innerhalb von Unterprogrammen und Interrupt-Tasks.

Operanden

N: Blockprogramm-Nummer
Die Blockprogramm-Nummer muss zwischen 0 und 127 (dezimal) liegen.

Operanden-Spezifikationen

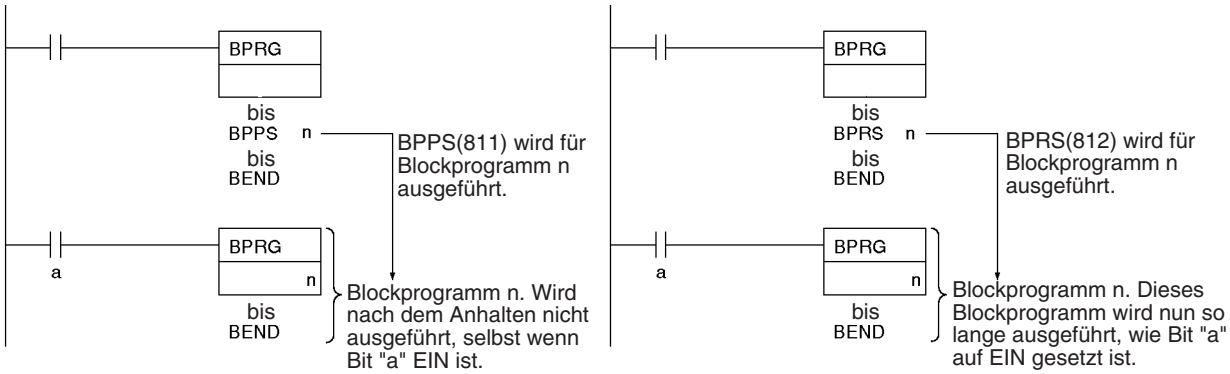
Bereich	N
CIO-Bereich	---
Arbeitsbereich nicht remanent	---
Arbeitsbereich remanent	---
Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---

Bereich	N
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	0 bis 127 (dezimal)
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	---

Beschreibung

BPPS(811) wird innerhalb eines Blockprogramms verwendet, um die Ausführung eines anderen, durch N (die Blockprogramm-Nummer) spezifizierten Blockprogramms anzuhalten. Das Blockprogramm wird mit BPPS(811) angehalten, auch wenn der BPRG(096)-Befehl für das Blockprogramm eine EIN-Ausführungsbedingung hat. Das Blockprogramm wird nicht neu gestartet, bis BPRS(812) für das Blockprogramm ausgeführt wird.

BPRS(812) startet das durch N (die Blockprogramm-Nummer) spezifizierte Blockprogramm neu. Nach dem Neustart wird das Blockprogramm ausgeführt, solange BPRG(096) eine EIN-Ausführungsbedingung hat.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn sich BPPS(811) bzw. BPRS(812) nicht in einem Blockprogramm befindet. EIN, wenn N nicht zwischen 0 und 127 liegt. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

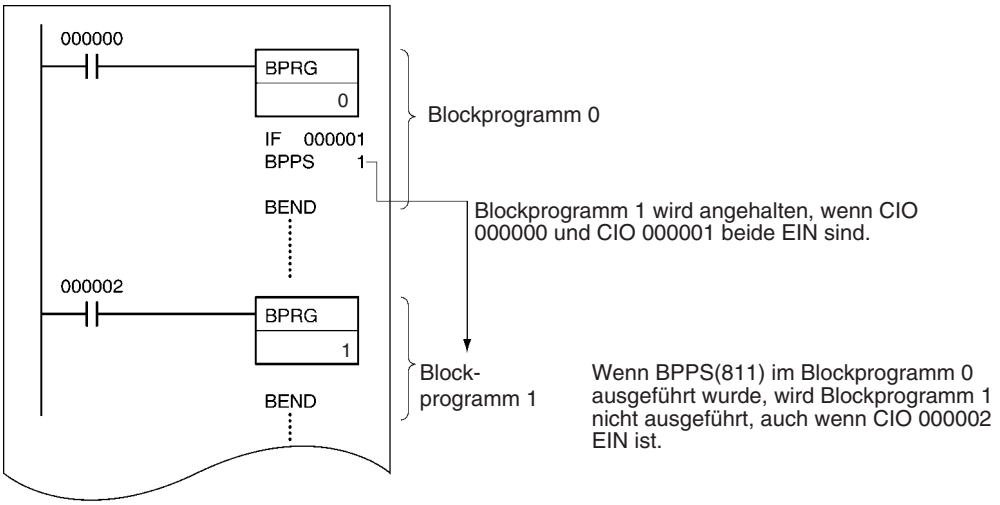
Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn sich BPPS(811) bzw. BPRS(812) nicht in einem Blockprogramm befindet oder N nicht zwischen #0000 und #007F (binär) liegt.

BPPS(811) kann zum Unterbrechen des Blockprogramms verwendet werden, in dem der Befehl enthalten ist. Wird das Blockprogramm dann mit Hilfe von BPRS(812) aus einem anderen Blockprogramm heraus neu gestartet, wird das angehaltene Blockprogramm durch den nächsten Befehl nach BPPS(811) wieder gestartet.

Wenn ein auf Pause gesetztes Blockprogramm die Befehle TIMW(813), TIMWX(816), TMHW(815) oder TMHWX(817) enthält, läuft der Zeit-Istwert weiter ab, auch wenn das Blockprogramm angehalten wurde.

Beispiele

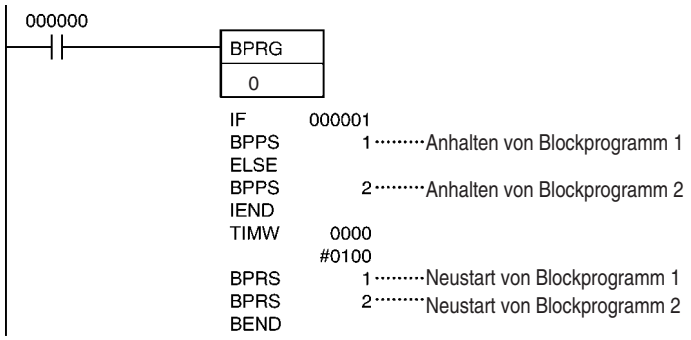
Die folgende Abbildung zeigt ein Beispiel für das Anhalten eines Blockprogramms.



Hinweis

Wenn das angehaltene Blockprogramm nach BPPS(811) platziert ist, wird es nicht ausgeführt. Wenn das Blockprogramm vor BPPS(811) platziert ist, wird an dem nächsten Zyklus auf Pause gesetzt.

Wenn CIO 000000 EIN ist, hält das folgende Programm die Ausführung von Blockprogramm 1 oder Blockprogramm 2 an, abhängig vom Status von CIO 000001. Das angehaltene Blockprogramm wird dann nach 10 Sekunden neu gestartet.



Adresse	Befehl	Operanden
000000	LD	000000
000001	BPRG(096)	00
000002	IF(802)	000001
000003	BPPS(811)	01
000004	ELSE(803)	
000005	BPPS(811)	02
000006	IEND(804)	
000007	TIMW(803)	0000
		# 0100
000008	BPRS(812)	1
000009	BPRS(812)	2
000010	BEND(801)	

3-32-4 Verzweigung: IF(802), ELSE(803) und IEND(804)

Funktion

Verzweigt das Blockprogramm auf Basis einer Ausführungsbedingung oder des Status eines Operand-Bits.

Kontaktplansymbol

IF(802) B

IF(802)

IF(802) NOT B

ELSE(803)

IEND(804)

B: Bit-Operand

Variationen

Variationen	Wird im Blockprogramm stets ausgeführt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Hinweis IF (802), ELSE(803) und IEND(804) müssen in Blockprogrammierbereichen verwendet werden, selbst innerhalb von Unterprogrammen und Interrupt-Tasks.

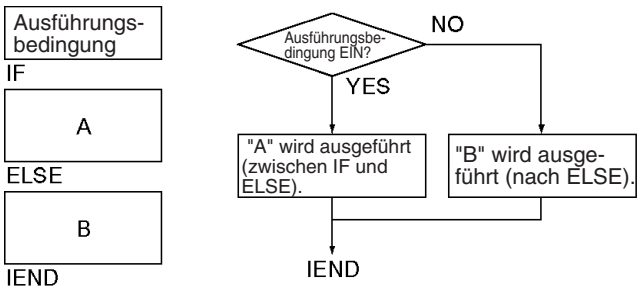
Operanden-Spezifikationen

Bereich	B
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A44715 A44800 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Task-Flags	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

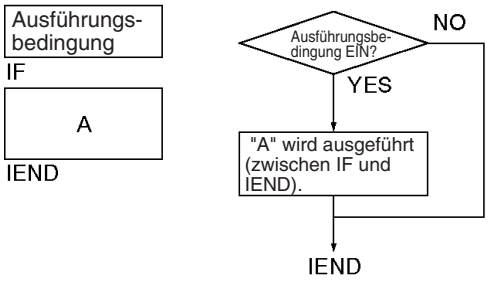
Beschreibung

Funktionsweise ohne Operand für IF(802)

Wenn kein Operand-Bit spezifiziert ist, muss vor dem Start von IF(802) durch LD eine Ausführungsbedingung erzeugt werden. Wenn die Ausführungsbedingung EIN ist, werden die Befehle zwischen IF(802) und ELSE(803) ausgeführt. Wenn die Ausführungsbedingung AUS ist, werden die Befehle zwischen ELSE(803) und IEND(804) ausgeführt.

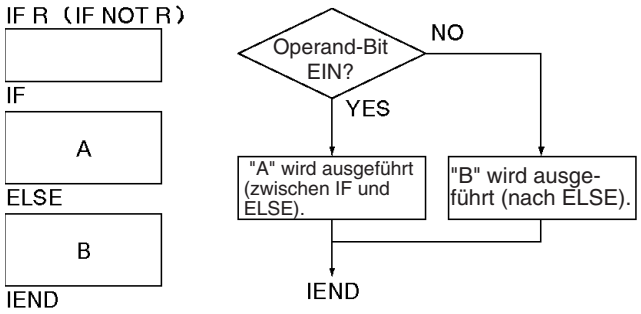


Wenn der Befehl ELSE(803) ausgelassen und die Ausführungsbedingung EIN wird, werden die Befehle zwischen IF(802) und IEND(804) ausgeführt. Wenn die Ausführungsbedingung AUS ist, werden nur die Befehle nach IEND(804) ausgeführt.

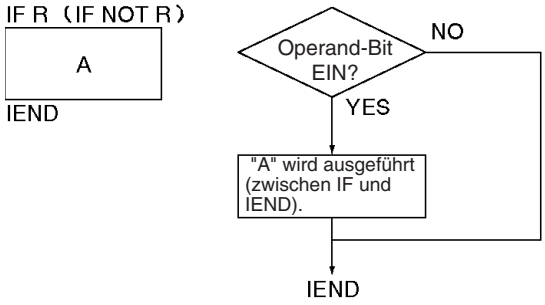


Operation mit einem Operanden für IF(802) oder IF NOT(802)

Ein Operand-Bit (B) kann für IF(802) oder IF NOT(802) spezifiziert werden. Wenn das Operand-Bit EIN ist, werden die Befehle zwischen IF(802) und ELSE(803) ausgeführt. Wenn das Operand-Bit AUS ist, werden die Befehle zwischen ELSE(803) und IEND(804) ausgeführt. Bei IF NOT(802) werden die Befehle zwischen IF(802) und ELSE(803) ausgeführt, wenn das Operand-Bit EIN ist und die Befehle ELSE(803) und IEND(804) werden ausgeführt, wenn das Operand-Bit AUS ist.



Wenn der Befehl ELSE(803) ausgelassen und das Operand-Bit auf EIN gesetzt wird, werden die Befehle zwischen IF(802) und IEND(804) ausgeführt. Wenn das Operand-Bit AUS ist, werden nur die Befehle nach IEND(804) ausgeführt. Das gleiche passiert bei invertiertem Status der Operand-Bits, wenn IF NOT(802) verwendet wird.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn sich die Verzweigungsbefehle nicht in einem Blockprogramm befinden. EIN, wenn mehr als 254 Verzweigungen verschachtelt sind. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Befehle in Blockprogrammen werden gewöhnlich bedingungslos ausgeführt. Eine Verzweigung kann dennoch zur Erzeugung einer bedingten Ausführung auf Basis von Ausführungsbedingungen oder Operanden-Bits verwendet werden.

Verwenden Sie IF A ELSE B IEND für eine Verzweigung zwischen A und B.

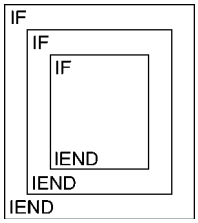
Verwenden Sie IF A IEND für eine Verzweigung zwischen A und keinem Vorgang.

Verzweigungen können in bis zu 253 Ebenen verschachtelt werden.

Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn sich die Verzweigungsbefehle nicht in einem Blockprogramm befinden oder wenn mehr als 254 Verzweigungen verschachtelt sind.

Verschachtelung von Verzweigungen

Bis zu 253 Verzweigungen können innerhalb der obersten Verzweigung verschachtelt werden.

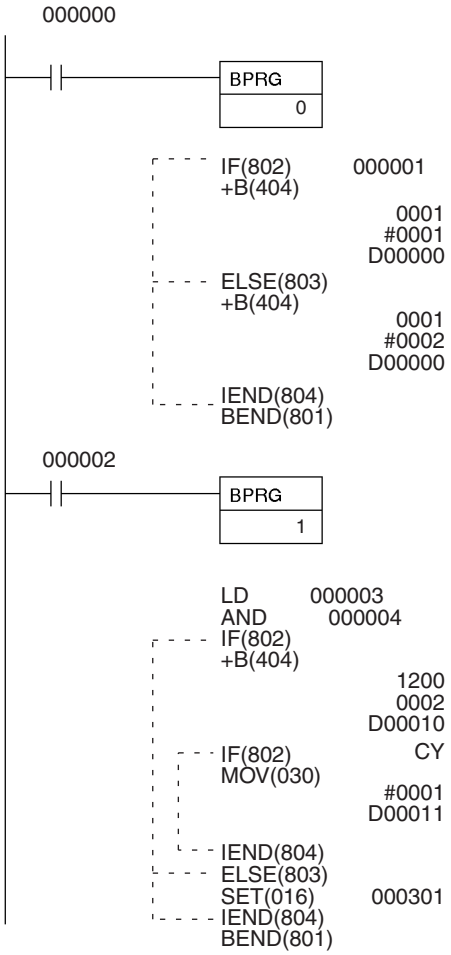


Beispiele

Das folgende Beispiel zeigt zwei unterschiedliche Blockdiagramme, die von CIO 000000 und CIO 000002 gesteuert werden.

Der erste Block führt eine der zwei Additionen in Abhängigkeit von Status von CIO 000001 aus. Dieser Block wird ausgeführt, wenn CIO 000000 EIN ist. Wenn CIO 000001 EIN ist, wird 0001 zum Inhalt von CIO 0001 addiert. Wenn CIO 000001 AUS ist, wird 0002 zum Inhalt von CIO 0001 addiert. In beiden Fällen wird das Ergebnis in D00000 ausgegeben.

Der zweite Block wird ausgeführt, wenn CIO 000002 EIN ist. Dieser Block ist in zwei Ebenen verschachtelt. Wenn CIO 000003 und CIO 000004 beide EIN sind, werden die Inhalte von CIO 1200 und CIO 0002 addiert und das Ergebnis in D00010 ausgegeben. Anschließend wird 0001, je nach Status von CY, in D00011 übertragen. Wenn entweder CIO 000003 oder CIO 000004 AUS ist, wird die gesamte Addition übersprungen und CIO 000301 wird auf EIN gesetzt.



Adresse	Befehl	Operanden
000000	LD	000000
000001	BPRG(096)	00
000002	IF(802)	000001
000003	+B(404)	
		0001
		#0001
		D00000
000004	ELSE(803)	
000005	+B(404)	
		0001
		#0002
		D00000
000006	IEND(804)	
000007	BEND(801)	
000008	LD	000002
000009	BPRG(096)	1
000010	LD	000003
000011	AND	000004
000012	IF(802)	
000013	+B(404)	
		1200
		0002
		D00010
000014	IF(802)	A50004
000015	MOV(030)	
		#0001
		D00011
000016	IEND(804)	
000017	ELSE(803)	
000018	SET(016)	000301
000019	IEND(804)	
000020	BEND(801)	

3-32-5 CONDITIONAL BLOCK EXIT (NOT): EXIT (NOT)(806)

Funktion

Verlässt das Blockprogramm (im Blockprogramm wird also kein anderer Befehl bis BEND(801) ausgeführt), abhängig vom Status des Operand-Bits oder der Ausführungsbedingung. EXIT(806) ohne ein Operand-Bit verlässt das Programm, wenn die Ausführungsbedingung den Zustand EIN hat. EXIT(806) mit einem Operand-Bit verlässt das Programm, wenn das Bit EIN ist. EXIT NOT(806) muss ein Operand-Bit haben und verlässt das Programm, wenn das Bit EIN ist.

Kontaktplansymbol

EXIT(806)

EXIT(806) B

EXIT NOT(806) B

B: Bit-Operand

Variationen

Variationen	Wird im Blockprogramm stets ausgeführt	EXIT(806)
		EXIT(806) B
		EXITNOT(806) B

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Hinweis EXIT(806) und EXIT NOT(806) müssen in Blockprogrammierbereichen verwendet werden, selbst innerhalb von Unterprogrammen und Interrupt-Tasks.

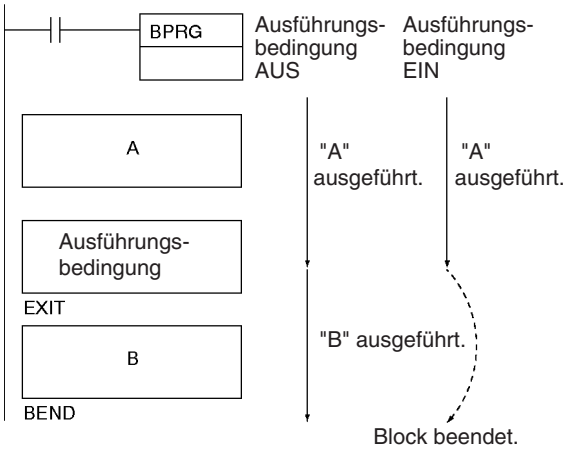
Operanden-Spezifikationen

Bereich	B
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A44715 A44800 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Task-Flags	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

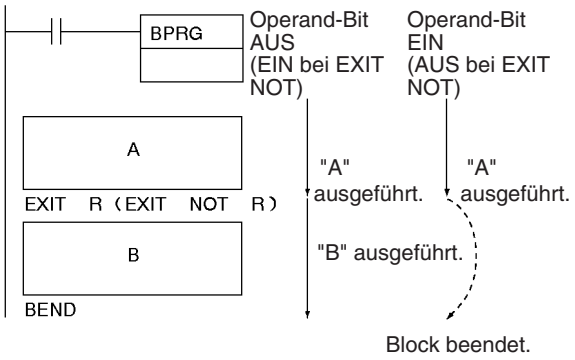
Funktion ohne Operand

EXIT(806) kann ohne Operand ausgeführt werden. In diesem Fall muss dafür eine mit LD beginnende Ausführungsbedingung erzeugt werden. Wenn die Ausführungsbedingung AUS ist, wird der Rest des Blockprogramms normal ausgeführt. Ist die Ausführungsbedingung EIN, wird der Rest der Befehle im Blockprogramm bis BEND(801) nicht ausgeführt.



Funktion mit Operand

Wenn das Operand-Bit B für EXIT(806) AUS ist, wird der Rest des Blockprogramms normal ausgeführt. Ist der Operand für EXIT(806) EIN, wird der Rest der Befehle im Blockprogramm bis BEND(801) nicht ausgeführt. Bei EXIT NOT(806) wird der Rest des Blockprogramms ausgeführt, wenn das Operand-Bit EIN ist und übersprungen, wenn das Operand-Bit AUS ist.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn sich EXIT(806) bzw. EXIT NOT(806) nicht in einem Blockprogramm befindet. In allen anderen Fällen AUS.

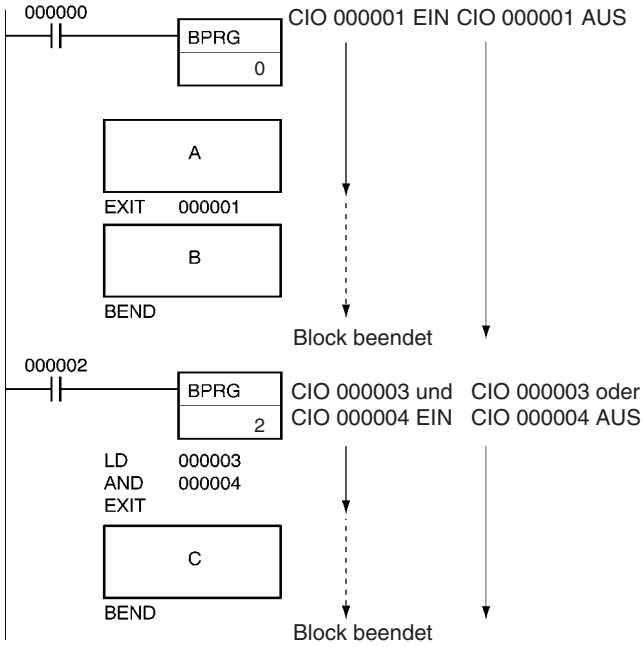
Vorsichtsmaßnahmen

Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn sich EXIT(806) bzw. EXIT NOT(806) nicht in einem Blockprogramm befindet.

Beispiele

Wenn CIO 000000 AUS ist, wird das Blockprogramm ausgeführt. Wenn CIO 000001 EIN ist, wird A ausgeführt. B wird übersprungen und die Programmsteuerung springt zu BEND(801). Abschnitt B des Programms wird so lange weiterhin übersprungen, bis CIO 000001 wieder auf AUS gesetzt wird.

Obwohl sich EXIT (NOT)(806) und die IF-IEND-Programmierung ähnlich sind, ist die Ausführungszeit für EXIT (NOT)(806) gewöhnlich kürzer, da die Befehle von EXIT (NOT)(806) bis zum Ende des Blockprogramms überhaupt nicht ausgeführt werden.



3-32-6 ONE CYCLE AND WAIT (NOT): WAIT(805)/WAIT(805) NOT

Funktion

Stoppt die Ausführung des restlichen Blockprogramms, bis eine Ausführungsbedingung auf EIN oder ein Operand-Bit auf EIN oder AUS gesetzt wird.

Kontaktplansymbol

WAIT(805)		
WAIT(805)	B	**B**: Bit-Operand
WAIT(805) NOT	B	

Variationen

Variationen	Wird im Blockprogramm stets ausgeführt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Hinweis WAIT(805)/WAIT(805) NOT muss in Blockprogrammierbereichen verwendet werden, selbst innerhalb von Unterprogrammen und Interrupt-Tasks.

Operanden-Spezifikationen

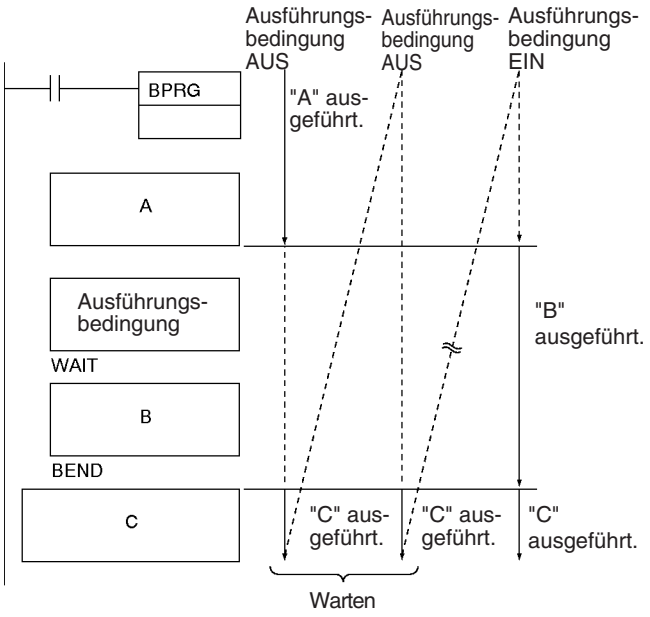
Bereich	B
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A44715 A44800 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Task-Flags	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, >, =, <, N, OF, UF, >=, <>, <=ON, OFF, AER

Bereich	B
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

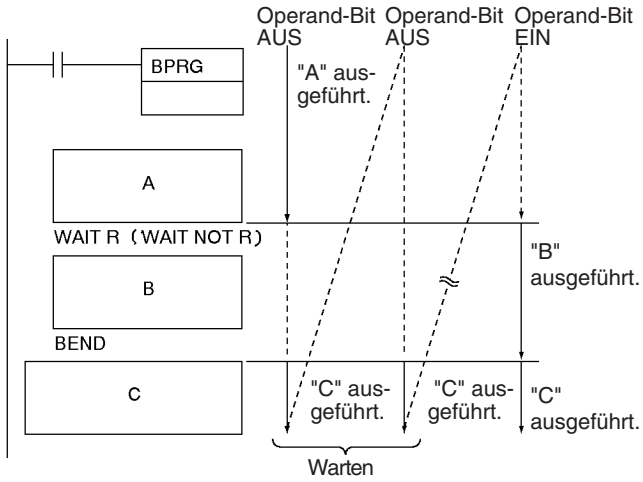
Funktion ohne Operand

Wenn kein Operand-Bit spezifiziert ist, muss vor WAIT(805)/WAIT(805 NOT eine Ausführungsbedingung erzeugt werden, die mit LD beginnt. Wenn die Ausführungsbedingung für WAIT(805) EIN ist, wird der Rest des Blockprogramms übersprungen. Im nächsten Zyklus wird kein Teil der Blockprogramms ausgeführt, außer der Teil für die Ausführungsbedingung für WAIT(805). Wenn die Ausführungsbedingung auf EIN gesetzt wird, erfolgt die Ausführung der Befehle von WAIT(805) bis zum Ende des Programms.



Funktion mit Operand

Für WAIT(805) oder WAIT NOT(805) kann ein Operand-Bit (B) spezifiziert werden. Wenn das Operand-Bit den Zustand AUS hat (EIN bei WAIT NOT(805)), werden die restlichen Befehle im Blockprogramm übersprungen. Im nächsten Zyklus wird kein Teil der Blockprogramms ausgeführt, außer der Teil für die Ausführungsbedingung für WAIT(805) bzw. WAIT NOT(805). Wenn die Ausführungsbedingung auf EIN wechselt (AUS bei WAIT NOT(805)), werden die Befehle von WAIT(805) bzw. WAIT NOT(805) bis zum Ende des Programms ausgeführt.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn sich WAIT(805) bzw. WAIT(805) NOT nicht in einem Blockprogramm befindet. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

WAIT(805) und WAIT(805) NOT können für die Schrittprogressionen in Blockprogrammen verwendet werden.

Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn sich WAIT(805) bzw. WAIT(805) NOT nicht in einem Blockprogramm befindet.

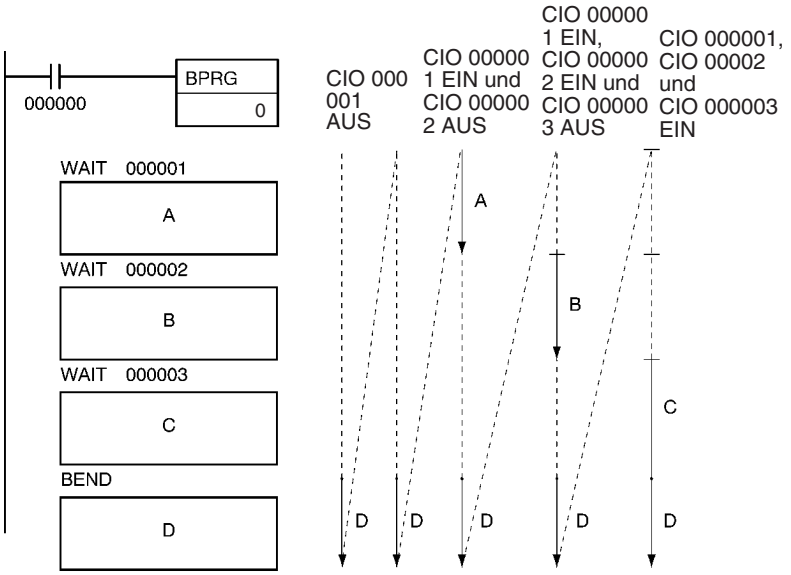
Hinweis Die Programmadresse von WAIT-Befehlen mit spezifizierten Operanden und die Programmadressen des ersten Befehls, die die Ausführungsbedingungen für WAIT-Befehle ohne Operanden erzeugen, werden gespeichert, um die weitere Ausführung auf Basis der Ausführungsbedingung/Bit-Operand zu ermöglichen. Wenn von einem Peripheriegerät aus eine Online-Bearbeitung durchgeführt wird, kann der WAIT-Status allerdings auch gelöscht werden und das Blockprogramm wird wieder von Anfang an ausgeführt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel EIN ist, wird Blockprogramm 00 ausgeführt. Die Ausführung erfolgt wie folgt:

1,2,3...

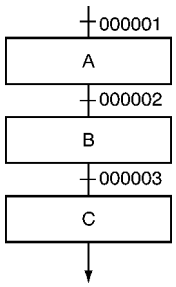
1. Wenn CIO 000001 AUS ist, wird kein Teil des Blockprogramms ausgeführt, bis CIO 000001 auf EIN gesetzt wird. Wenn CIO 000001 auf EIN gesetzt wird, erfolgt die Ausführung von „A“.
2. Wenn CIO 000002 nach Ausführung von „A“ AUS ist, wird der Rest des Blockprogramms nicht ausgeführt, bis CIO 000002 auf EIN gesetzt wird. Wenn CIO 000002 auf EIN gesetzt wird, erfolgt die Ausführung von „B“.
3. Wenn CIO 000003 nach Ausführung von „B“ AUS ist, wird der Rest des Blockprogramms nicht ausgeführt, bis CIO 000003 auf EIN gesetzt wird. Wenn CIO 000003 auf EIN gesetzt ist, wird „C“ ausgeführt und der Ausführungsprozess wird wiederholt.



Die folgende Tabelle zeigt die Beziehung zwischen Operand-Bits und Blockprogrammausführung.

Operand-Bits			Programmausführung		
CIO 000001	**CIO 000002**	**CIO 000003**	**Erster Zyklus, CIO 000000 ist EIN**	**Nächster Zyklus**	**Folgende Zyklen**
AUS	Beliebiger Status	Beliebiger Status	Keine Ausführung	Keine Ausführung, warten auf CIO 000001.	Wenn CIO 000001 auf EIN gesetzt ist, wird „A“ ausgeführt und der Status von CIO 000002 wird geprüft.
EIN	AUS	Beliebiger Status	„A“ ausgeführt	Warten auf CIO 000002.	Wenn CIO 000002 auf EIN gesetzt ist, wird „B“ ausgeführt und der Status von CIO 000003 geprüft.
EIN	EIN	AUS	„A“ und „B“ ausgeführt	Warten auf CIO 000003.	Wenn CIO 000003 auf EIN gesetzt ist, wird „C“ ausgeführt
EIN	EIN	EIN	„A“, „B“ und „C“ ausgeführt	„A“, „B“ und „C“ ausgeführt	

Wie in diesem Beispiel ersichtlich, kann WAIT(805) und WAIT(805) NOT zur progressiven Ausführung von Schritten in einem Blockprogramm verwendet werden.



3-32-7 TIMER WAIT: TIMW(813) und TIMWX(816)

Funktion

Verzögert die Ausführung des restlichen Blockprogramms, bis die spezifizierte Zeitspanne verstrichen ist. Die Ausführung fährt mit dem nächsten Befehl nach TIMW(813)/TMWX(816) fort, wenn der Zeitgeber abgelaufen ist.

Kontaktplansymbol

Istwert-Aktualisierungsmethode: BCD

TIMW(813)	N	**N**: Zeitgebernummer
	SV	**SV**: Sollwert

Istwert-Aktualisierungsmethode: Binär

TIMWX(816)	N	**N**: Zeitgebernummer
	SV	**SV**: Sollwert

Variationen

Variationen	Wird im Blockprogramm stets ausgeführt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	Nicht zulässig

Hinweis TIMW(813)/TIMWX(816) muss in Blockprogrammierbereichen verwendet werden, selbst wenn sich die in Unterprogrammen befinden.

Operanden

N: Zeitgebernummer
BCD: 0 bis 4095 (dezimal)
Binär: 0 bis 4095 (dezimal)

S: Sollwert
BCD: #0000 bis #9999 (BCD)
Binär: &0 bis &65535 (dezimal)
#0000 bis #FFFF (hex)

Operanden-Spezifikationen

Bereich	N	Sollwert
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A447 A448 bis A959
Zeitgeberbereich	0000 bis 4095	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)

Bereich	N	Sollwert
Konstanten	---	BCD: #0000 bis 9999 (BCD) "&" kann nicht verwendet werden. Binär: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)
Datenregister	---	DR0 bis DR15
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

TIMW(813)/TIMWX(816) erzeugt einen Einschaltverzögerungszeitgeber (in SV eingestellter 100-ms-Zeitgeber) zwischen der Ausführung des vorherigen und des folgenden Blockprogrammbefehls. TIMW(813) kann eine Zeitspanne von 0 bis 999,9 s mit einer Zeitgebergenauigkeit von 0 bis 0,01 s messen. TIMWX(816) kann eine Zeit von 0 bis 6.553,5 s mit einer Zeitgebergenauigkeit von 0 bis 0,01 s messen.

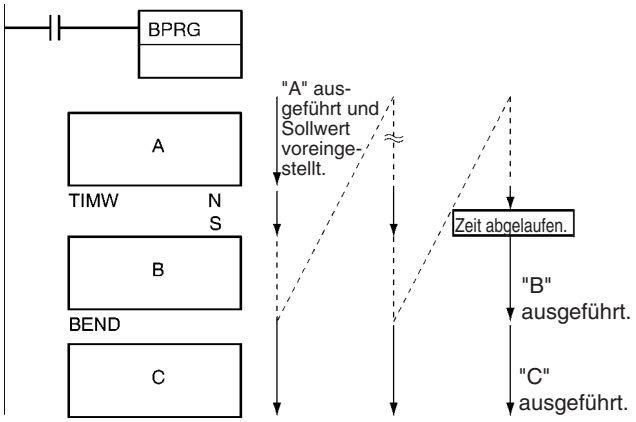
Hinweis

Die Zeitgebergenauigkeit bei CS1D CPU-Baugruppen beträgt 10 ms + die Zykluszeit.

Der erste Teil des Blockprogramms wird beim ersten Zugriff auf das Blockprogramm ausgeführt. Wenn TIMW(813)/TIMWX(816) erreicht wird, wird das Fertig-Bit auf AUS zurückgesetzt, der Zeitgeber auf den Sollwert eingestellt und die Ausführung des restlichen Blockprogramms wartet, bis der Sollwert verstrichen ist.

Während der Zeitgeber die Zeit herunterzählt, wird nur TIMW(813)/TIMWX(816) zur Aktualisierung des Zeitgebers ausgeführt. Wenn der Zeitgeber abgelaufen ist, wird das Fertig-Bit auf EIN gesetzt und der Rest des Blockprogramms wird ausgeführt. Nach Ausführung des gesamten Blockprogramms wird der Prozess wiederholt.

TIMW(813)/TIMWX(816) kann als WAIT-Befehl mit einem Zeitgeber für die Ausführungsbedingung bezeichnet werden, der daher für die zeitgesteuerte Schrittprogressionen verwendet werden kann.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn sich TIMW(813)/TIMWX(816) nicht in einem Blockprogramm befindet. EIN, wenn für N eine indirekte IR-Adressierung im BCD-Modus verwendet wird und die Adresse nicht für einen Zeitgeber-Istwert ist. EIN, wenn der BCD-Modus eingestellt ist und der Sollwert nicht in BCD ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Rest des Blockprogramms nach dem Zeitgeber wird ausgeführt, wenn das Fertig-Bit für den Zeitgeber zwangsweise gesetzt wird.

Wenn das Fertig-Bit für den Zeitgeber zwangsweise zurückgesetzt wird, wird im Blockprogramm nur TIMW(813/TIMWX(816)) ausgeführt, bis die Status der zwangsweisen Rücksetzung gelöscht wird.

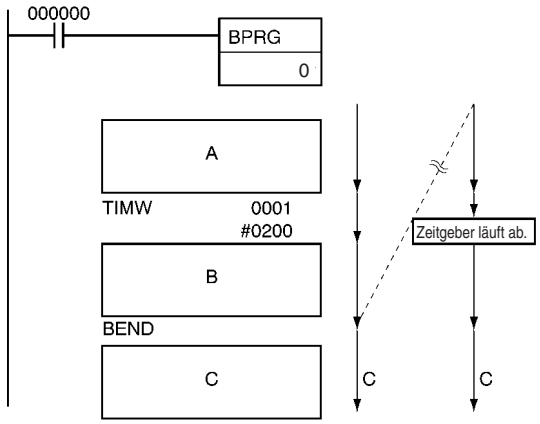
Der Istwert von Zeitgebern, die mit Zeitgebernummern von 0000 bis 2047 programmiert wurden, wird auch aktualisiert, wenn der Zeitgeber auf Standby gesetzt ist. Der Istwert von Zeitgebern, die mit Zeitgebernummern von 2048 bis 4095 programmiert wurden, wird gehalten, wenn der Zeitgeber auf Standby gesetzt ist.

Die Zeitgebernummern werden auch von den anderen Zeitgeberbefehlen verwendet. Das Verhalten ist nicht vorhersehbar, wenn die selbe Zeitgebernummer für mehr als einen Zeitgeberbefehl verwendet wird. Jede Zeitgebernummer darf nur einmal verwendet werden. Der einzige Art, auf die dieselbe Zeitgebernummer mehrfach verwendet werden kann, besteht darin, dass nie zwei Zeitgeber gleichzeitig arbeiten. Wenn die selbe Zeitgebernummer in mehr als einem Zeitgeberbefehl verwendet wird, tritt bei der Programmprüfung ein Fehler auf.

Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn für N eine indirekte IR-Adressierung im BCD-Modus verwendet wird und die Adresse nicht für einen Zeitgeber-Istwert gilt oder der Sollwert kein BCD-Wert ist.

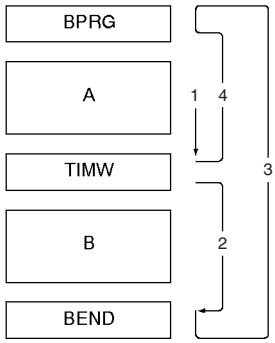
Beispiele

Wenn CIO 000000 EIN ist, wird „B“ im folgenden Beispiel 20 Sekunden nach „A“ ausgeführt.



Adresse	Befehl	Operand
000200	LD	000000
000201	BPRG	0
. .	A	. .
000210	TIMW	0001
		#0200
. .	B	. .
000220	BEND	---

Die Programmausführung bewegt sich während der 20 s vor Ausführung von „B“ von 2 zu 3 zu 4 und zurück zu 2 (siehe folgende Abbildung).



3-32-8 COUNTER WAIT: CNTW(814) und CNTWX(818)

Funktion

Verzögert die Ausführung des restlichen Blockprogramms, bis der spezifizierte Zählwert erreicht ist. Die Ausführung fährt mit dem nächsten Befehl nach CNTW(814)/CNTWX(818) fort, wenn der Zählwert erreicht ist.

Kontaktplansymbol

Istwert-Aktualisierungsmethode: BCD

CNTW(814)	N	**N**: Zählernummer
	SV	**SV**: Sollwert
	I	**I**: Zähleingang

Istwert-Aktualisierungsmethode: Binär

CNTWX(818)	N	**N**: Zählernummer
	SV	**SV**: Sollwert
	I	**I**: Zähleingang

Variationen

Variationen	Wird im Blockprogramm stets ausgeführt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Hinweis CNTW(814)/CNTWX(818) muss in Blockprogrammierbereichen verwendet werden, selbst wenn sich diese innerhalb von Unterprogrammen und Interrupt-Tasks befinden.

Operanden

N: Zählernummer
BCD: 0 bis 4095 (dezimal)
Binär: 0 bis 4095 (dezimal)

S: Sollwert
BCD: #0000 bis #9999 (BCD)
Binär: &0 bis &65535 (dezimal)
#0000 bis #FFFF (hex)

Operanden-Spezifikationen

Bereich	N	SV	I
CIO-Bereich	---	CIO 0000 bis CIO 6143	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	---	W000 bis W511	W00000 bis W51115
Arbeitsbereich remanent	---	H000 bis H511	H00000 bis H51115
Systembereich	---	A000 bis A447 A448 bis A959	A00000 bis A44715 A44800 bis A95915
Zeitgeberbereich	---	T0000 bis T4095	T0000 bis T4095
Zählerbereich	C0000 bis C4095	C0000 bis C4095	C0000 bis C4095
Task-Flags	---		TK0000 bis TK0031
Bedingungs-Flags	---		ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON,OFF, AER
Taktimpulse	---		0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM-Bereich	---	D00000 bis D32767	---
EM-Bereich ohne Bank	---	E00000 bis E32767	---
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)	---
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	---
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	---
Konstanten	---	BCD: #0000 bis 9999 (BCD) „&“ kann nicht verwendet werden. Binär: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)	---
Datenregister	---	DR0 bis DR15	---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

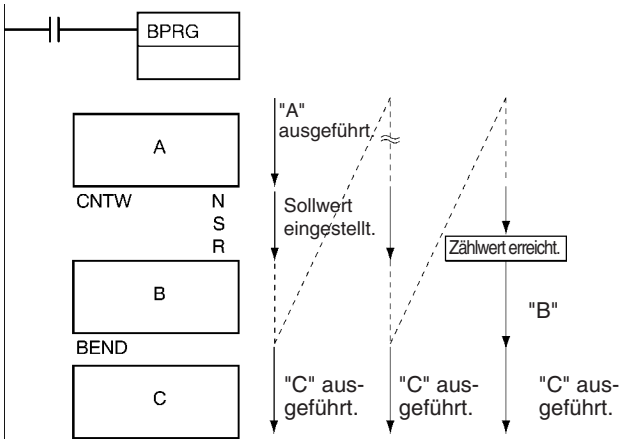
Beschreibung

CNTW(814)/CNTWX(818) erzeugt einen dekrementierenden Zähler, der die Ausführung der nachfolgenden Befehle im Blockprogramm verzögert, bis der Zähler heruntergezählt hat. Der Sollwert für CNTW(814) wird als BCD-Wert zwischen 0000 und 9999 spezifiziert. Der Sollwert für CNTWX(818) wird als Binärwert zwischen 0000 und FFFF hex spezifiziert.

Der erste Teil des Blockprogramms wird beim ersten Zugriff auf das Blockprogramm ausgeführt. Wenn der Befehl CNTW(814)/CNTWX(818) erreicht wird, wird das Fertig-Bit auf 0 zurückgesetzt, der Zähler auf den Sollwert voreingestellt und die Ausführung des restlichen Blockprogramms wartet, bis der Zähler heruntergezählt hat. Der Zähler zählt die Impulse (steigende Flanke) von I (Zählereingang).

Während der Zähler herunterzählt, wird nur CNTW(814)/CNTWX(818) zur Aktualisierung des Zählers ausgeführt. Wenn der Zähler heruntergezählt hat, wird das Fertig-Bit auf EIN gesetzt und der Rest des Blockprogramms wird ausgeführt. Nach Ausführung des gesamten Blockprogramms wird der Prozess wiederholt.

TIMW(814)/TIMWX(818) kann als WAIT-Befehl mit einem Zähler für die Ausführungsbedingung betrachtet werden, der dementsprechend für zeitgesteuerte Schrittprogressionen verwendet werden kann.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn sich CNTW(814)/CNTWX(818) nicht in einem Blockprogramm befindet. EIN, wenn für N eine indirekte IR-Adressierung im BCD-Modus verwendet wird und die Adresse nicht für einen Zähler-Istwert gilt. EIN, wenn der Sollwert beim Einstellung des BCD-Modus kein BCD-Wert ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Rest des Blockprogramms nach CNTW(814)/CNTWX(818) wird ausgeführt, wenn das Fertig-Bit für den Zähler zwangsweise gesetzt wird.

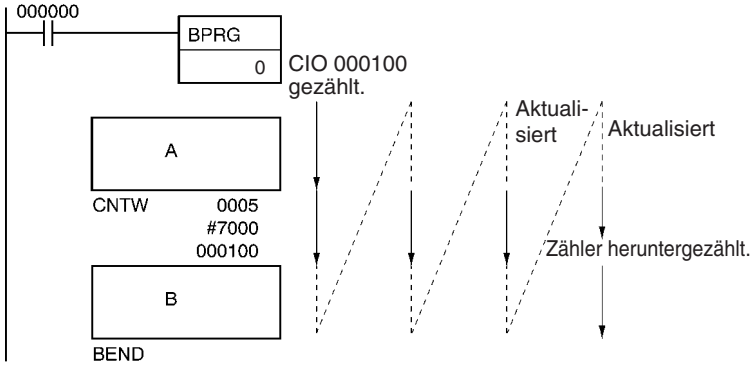
Wenn das Fertig-Bit für den Zähler zwangsweise zurückgesetzt wird, wird im Blockprogramm nur CNTW(814)/CNTWX(818) ausgeführt, bis die Status der zwangsweisen Rücksetzung aufgehoben wird.

Die Zählernummern werden auch von den anderen Zählerbefehlen verwendet. Die Funktion ist nicht vorhersehbar, wenn dieselbe Zählernummer für mehr als einen Zählerbefehl verwendet wird. Jede Zählernummer darf nur einmal verwendet werden. Der einzige Weg, dieselbe Zählernummer mehrfach zu nutzen, besteht darin, dass nie zwei Zähler gleichzeitig arbeiten. Wenn dieselbe Zählernummer in mehr als einem Zählerbefehl verwendet wird, tritt bei der Programmprüfung ein Fehler auf.

Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn für N eine indirekte IR-Adressierung im BCD-Modus verwendet wird und die Adresse nicht für einen Zähler-Istwert gilt oder der Sollwert bei Einstellung des BCD-Modus kein BCD-Wert ist.

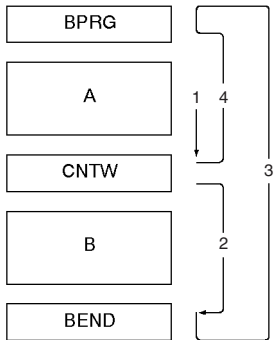
Beispiele

Wenn CIO 000000 im folgenden Beispiel EIN ist, wird „A“ ausgeführt und dann wartet der Rest des Blockprogramms „B“ auf seine Ausführung, bis 7.000 Zählimpulse von CIO 000100 erreicht sind.



Adresse	Befehl	Operand
000200	LD	000000
000201	BPRG	0
. .	A	. .
000210	CNTW	0005
		#7000
		000100
. .	B	. .
000220	BEND	---

Die Programmausführung geht vor Ausführung von „B“ von 2 zu 3 zu 4 und zurück zu 2, bis der Zählwert 7.000 erreicht hat (siehe folgende Abbildung).



3-32-9 HIGH-SPEED TIMER WAIT: TMHW(815) and TMHWX(817)

Funktion

Verzögert die Ausführung des restlichen Blockprogramms, bis die spezifizierte Zeitspanne verstrichen ist. Die Ausführung fährt mit dem nächsten Befehl nach TMHW(815)/TMHWX(817) fort, wenn der Zeitgeber abgelaufen ist.

Kontaktplansymbol

Istwert-Aktualisierungsmethode: BCD

TMHW(815)	N	**N**: Zeitgebernummer
	SV	**SV**: Sollwert

Istwert-Aktualisierungsmethode: Binär

TMHWX(817)	N	**N**: Zeitgebernummer
	SV	**SV**: Sollwert

Variationen

Variationen	Wird im Blockprogramm stets ausgeführt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	Nicht zulässig

Hinweis TMHW(815)/TMHWX(817) muss in Blockprogrammierbereichen verwendet werden, selbst wenn sich diese in Unterprogrammen befinden.

Operanden

N: Zeitgebernummer
BCD: 0 bis 4095 (dezimal)
Binär: 0 bis 4095 (dezimal)

S: Sollwert
BCD: #0000 bis #9999 (BCD)
Binär: &0 bis &65535 (dezimal)
#0000 bis #FFFF (hex)

Operanden-Spezifikationen

Bereich	N	SV
CIO-Bereich	---	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	---	W000 bis W511
Arbeitsbereich remanent	---	H000 bis H511
Systembereich	---	A000 bis A447 A448 bis A959
Zeitgeberbereich	0000 bis 4095	T0000 bis T4095
Zählerbereich	---	C0000 bis C4095
DM-Bereich	---	D00000 bis D32767
EM-Bereich ohne Bank	---	E00000 bis E32767
EM-Bereich mit Bank	---	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	---	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	---	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---	BCD: #0000 bis 9999 (BCD) „&“ kann nicht verwendet werden. Binär: &0 bis &65535 (dezimal) #0000 bis #FFFF (hex)
Datenregister	---	DR0 bis DR15
Indexregister	---	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

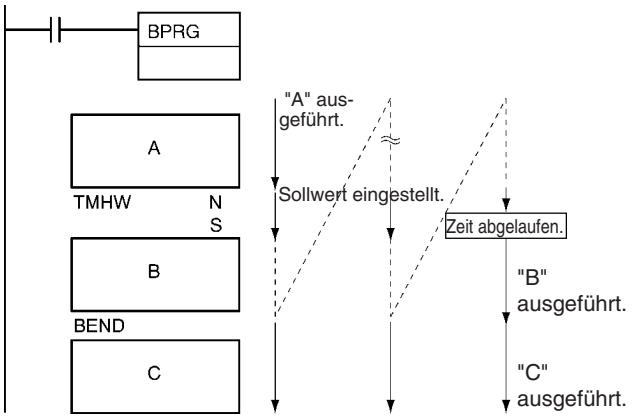
TMHW(815)/TMHWX(817) erzeugt einen Einschaltverzögerungszeitgeber (in SV eingestellter 10-ms-Zeitgeber) zwischen der Ausführung des vorherigen und der folgenden Blockprogrammbefehle. TMHW(815) kann die Zeit von 0 bis 99,99 s mit einer Zeitgebergenauigkeit von 0 bis 0,01 s messen. TMHWX(817) kann die Zeit von 0 bis 655,35 s mit einer Zeitgebergenauigkeit von 0 bis 0,01 s messen.

Hinweis Die Zeitgebergenauigkeit bei CS1D CPU-Baugruppen beträgt 10 ms + die Zykluszeit.

Der erste Teil des Blockprogramms wird beim ersten Zugriff auf das Blockprogramm ausgeführt. Wenn TMHW(815)/TMHWX(817) erreicht wird, wird das Fertig-Bit auf AUS zurückgesetzt, der Zeitgeber auf den Sollwert voreingestellt und die Ausführung des restlichen Blockprogramms wartet, bis der Sollwert abgelaufen ist.

Während der Zeitgeber abläuft, wird nur TMHW(815)/TMHWX(817) zur Aktualisierung des Zeitgebers ausgeführt. Wenn die Zeitgeberzeit abgelaufen ist, wird das Fertig-Bit auf EIN gesetzt und der Rest des Blockprogramms wird ausgeführt. Nach Ausführung des gesamten Blockprogramms wird der Prozess wiederholt.

TMHW(815)/TMHWX(817) kann als WAIT-Befehl mit einem Zeitgeber für die Ausführungsbedingung betrachtet werden, der dementsprechend für zeitgesteuerte Schrittprogressionen verwendet werden kann.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn sich TMHW(815)/TMHWX(817) nicht in einem Blockprogramm befindet. EIN, wenn für N eine indirekte IR-Adressierung im BCD-Modus verwendet wird und die Adresse nicht für einen Zeitgeber-Istwert ist. EIN, wenn der BCD-Modus eingestellt ist und der Sollwert nicht in BCD ist. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Rest des Blockprogramms nach TMHW(815)/TMHWX(817) wird ausgeführt, wenn das Fertig-Bit für den Zeitgeber zwangsweise gesetzt wird.

Wenn das Fertig-Bit für den Zeitgeber zwangsweise zurückgesetzt wird, wird im Blockprogramm nur TMHW(815)/TMHWX(817) ausgeführt, bis die Status der zwangsweisen Rücksetzung aufgehoben wird.

Der Istwert von Zeitgebern, die mit Zeitgebernummern von 0000 bis 2047 programmiert wurden, wird auch aktualisiert, wenn der Zeitgeber auf Standby gesetzt ist. Der Istwert von Zeitgebern, die mit Zeitgebernummern von 2048 bis 4095 programmiert wurden, wird gehalten, wenn der Zeitgeber auf Standby gesetzt ist.

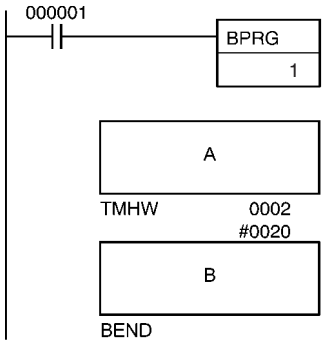
Die Zeitgebernummern werden auch von den anderen Zeitgeberbefehlen verwendet. Das Verhalten ist nicht vorhersehbar, wenn die selbe Zeitgebernummer für mehr als einen Zeitgeberbefehl verwendet wird. Jede Zeitgeber-

nummer darf nur einmal verwendet werden. Der einzige Art, auf die dieselbe Zeitgebernummer mehrfach verwendet werden kann, besteht darin, dass nie zwei Zeitgeber gleichzeitig arbeiten. Wenn die selbe Zeitgebernummer in mehr als einem Zeitgeberbefehl verwendet wird, tritt bei der Programmprüfung ein Fehler auf.

Ein Fehler tritt auf und das Fehler-Flag wird auf EIN gesetzt, wenn für N eine indirekte IR-Adressierung im BCD-Modus verwendet wird und die Adresse nicht für einen Zeitgeber-Istwert gilt oder der Sollwert kein BCD-Wert ist.

Beispiele

Wenn CIO 000000 EIN ist, wird „B“ im folgenden Beispiel 20 Sekunden nach „A“ ausgeführt.



Adresse	Befehl	Operand
000221	LD	000001
000222	BPRG	1
. .	A	. .
000250	TMHW	0002
		#0020
. .	B	. .
000281	BEND	---

3-32-10 Schleifensteuerung: LOOP(809)/LEND(810)/LEND(810) NOT

Funktion

Erstellt eine Schleife, die wiederholt ausgeführt wird, bis eine Ausführungsbedingung auf EIN oder AUS gesetzt wird oder bis eine Ausführungsbedingung auf EIN gesetzt wird.

Kontaktplansymbol

LOOP(809)

LEND(810)

LEND(810)	B	**B**: Bit-Operand
LEND(810) NOT	B	

Variationen

Variationen	Wird im Blockprogramm stets ausgeführt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Hinweis LOOP(809), LEND(810) und LEND(810) NOT müssen in Blockprogrammierbereichen verwendet werden, selbst wenn sich diese in Unterprogrammen und Interrupt-Tasks befinden.

Operanden-Spezifikationen

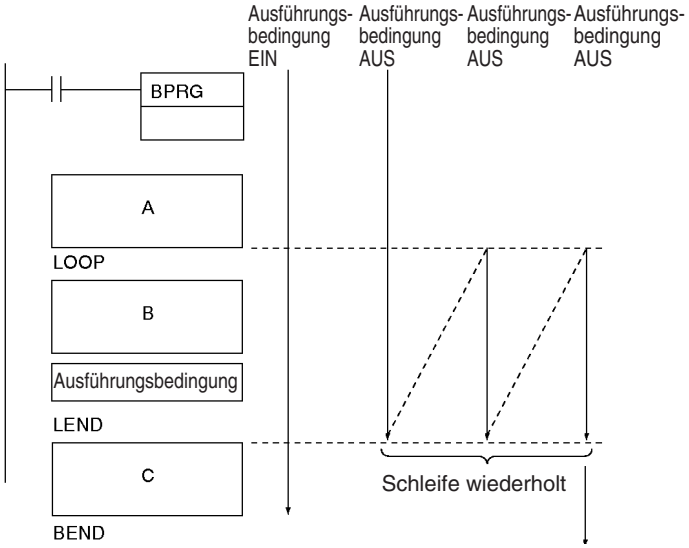
Bereich	B
CIO-Bereich	CIO 000000 bis CIO 614315
Arbeitsbereich nicht remanent	W00000 bis W51115
Arbeitsbereich remanent	H00000 bis H51115
Systembereich	A00000 bis A44715 A44800 bis A95915
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
Task-Flags	TK0000 bis TK0031
Bedingungs-Flags	ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON,OFF, AER
Taktimpulse	0,02 s, 0,1 s, 0,2 s, 1 s, 1 min
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

LOOP(809) spezifiziert den Anfang einer Programmschleife. LEND(810) bzw. LEND(810) NOT spezifizieren das Ende der Schleife. Wenn der Befehl LEND(810) bzw. LEND(810) NOT erreicht wird, geht die Programmausführung zurück zum vorhergehenden LOOP(809)-Befehl, bis das Operand-Bit für LEND(810) bzw. LEND(810) NOT auf EIN bzw. AUS wechselt oder bis die Ausführungsbedingung für LEND(810) auf EIN wechselt.

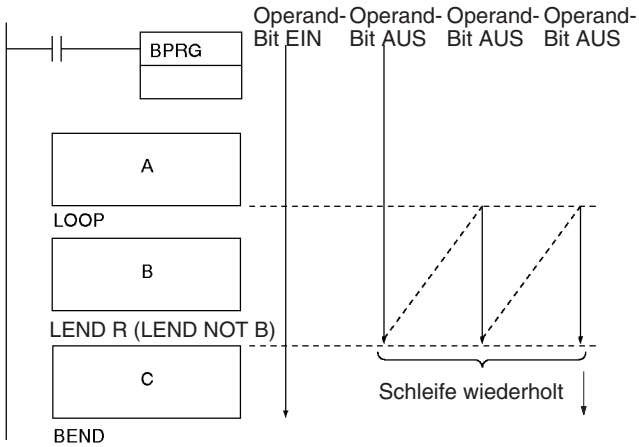
Verwendung einer Ausführungsbedingung für LEND(810)

LEND(810) kann mit oder ohne Operand-Bit programmiert werden. Wenn kein Operand-Bit spezifiziert ist, muss vor LEND(810) eine mit LD beginnende Ausführungsbedingung erzeugt werden. Wenn die Ausführungsbedingung AUS ist, wird die Ausführung der Schleife wiederholt, wobei mit dem nächsten Befehl nach LOOP(809) begonnen wird. Wenn die Ausführungsbedingung EIN ist, wird die Schleife beendet und die Ausführung mit dem nächsten Befehl nach LEND(810) fortgesetzt.



Verwendung eines Bit-Operanden für LEND(810) oder LEND(810) NOT

LEND(810) und LEND(810) NOT können mit einem Operand-Bit programmiert werden. Wenn das Operand-Bit für LEND(810) AUS ist (bzw. auf EIN bei LEND(810) NOT) wird die Ausführung der Schleife beginnend mit dem nächsten Befehl nach LOOP(809) wiederholt. Wenn das Operand-Bit für LEND(810) auf EIN gesetzt ist (bzw. auf AUS bei LEND(810) NOT) wird die Schleife beendet, und die Ausführung wird mit dem nächsten Befehl nach LEND(810) bzw. LEND(810) NOT fortgeführt.



Hinweis Der Status des Operand-Bits ist im Falle von LEND(810) NOT umgekehrt.

Hinweis

1. Eine Ausführung innerhalb einer Schleife aktualisiert die E/A-Daten nicht. Verwenden Sie IORF(184), wenn E/A-Daten aktualisiert werden müssen.
2. Wenn Schleifen zu lange wiederholt werden, kann die maximale Zykluszeit überschritten werden. Das Programm muss so ausgelegt sein, dass die maximale Zykluszeit nicht überschritten wird.

Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn sich ein Schleifensteuerbefehl nicht in einem Blockprogramm befindet. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Schleifen können nicht innerhalb von Schleifen verschachtelt werden.

Falsch:

```
LOOP(809)
LOOP(809)
LEND(810)
LEND(810)
```

Vertauschen Sie die Reihenfolge von LOOP und LEND nicht.

Falsch:

```
LEND(810)
   :
   :
LOOP(809)
```

Eine bedingte Blockverzweigung kann innerhalb einer Schleife verwendet werden, aber die gesamte Verzweigungsoperation muss innerhalb der Schleife ablaufen.

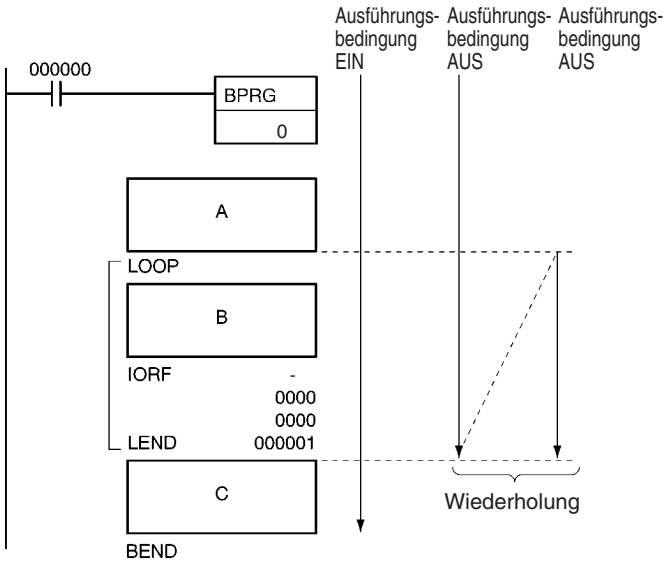
Richtig:	**Falsch:**
LOOP(809)	LOOP(809)
IF(802)	IF(802)
IF(802)	IF(802)
IEND(804)	IEND(804)
IEND(804)	LEND(810)
LEND(810)	IEND(804)

Wenn LOOP(809) nicht ausgeführt wird, erfolgt eine NOP-Verarbeitung.

Wenn sich ein Schleifensteuerungsbefehl nicht in einem Blockprogramm befindet, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Beispiele

Wenn CIO 000000 im folgenden Beispiel EIN ist, wird das Blockprogramm ausgeführt. Nach Ausführung von „A“ werden „B“ und der folgende Befehl IORF(184) wiederholt ausgeführt, bis CIO 000001 EIN ist. An diesem Zeitpunkt wird C ausgeführt und das Blockprogramm beendet.



Adresse	Befehl	Operand
000220	LD	000000
000201	BPRG	0
. .	A	. .
000210	LOOP	---
. .	B	. .
000220	IORF	. .
		0000
		0000
000221	LEND	000001
. .	C	. .
000220	BEND	---

3-33 Textverarbeitungsbefehle

Dieser Abschnitt enthält eine Beschreibung der zur Bearbeitung von Textzeichenketten verwendeten Befehle.

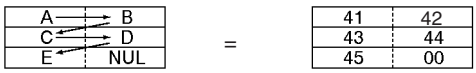
Befehl	AWL	Funktionscode	Seite
MOV STRING	MOV$	664	1159
CONCATENATE STRING	+$	656	1161
GET STRING LEFT	LEFT$	652	1164
GET STRING RIGHT	RGHT$	653	1166
GET STRING MIDDLE	MID$	654	1168
FIND IN STRING	FIND$	660	1171
STRING LENGTH	LEN$	650	1173
REPLACE IN STRING	RPLC$	661	1175
DELETE STRING	DEL$	658	1178
EXCHANGE STRING	XCHG$	665	1180
CLEAR STRING	CLR$	666	1182
INSERT INTO STRING	INS$	657	1184
Zeichenkettenvergleichsbefehle	=$, <>$, <$, <=$, >$, >=$	670 bis 675	1187

3-33-1 Übersicht über Textzeichenkettenverarbeitung

Daten vom Anfang bis zum NULL-Code (00 hex) werden als in ASCII ausgedrückte Textzeichenkettendaten (mit Ausnahme von 1-Byte-Sonderzeichen) behandelt. Sie werden vom höchstwertigen Byte zum niederwertigsten Byte und vom niederwertigen Wort zum höchstwertigsten Wort gespeichert.

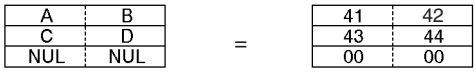
Bei einer ungeraden Zeichenzahl wird im freien Platz im niederwertigsten Byte des letzten Worts 00 hex (NULL-Code) gespeichert.

Beispiel: Textzeichenkette ABCDE



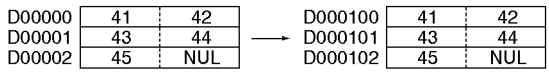
Bei einer geraden Zeichenzahl werden in den höchstwertigen und niederwertigsten Bytes des auf das Endwort folgenden Worts 0000 hex (zwei NULL-Codes) gespeichert.

Beispiel: Textzeichenkette ABCD

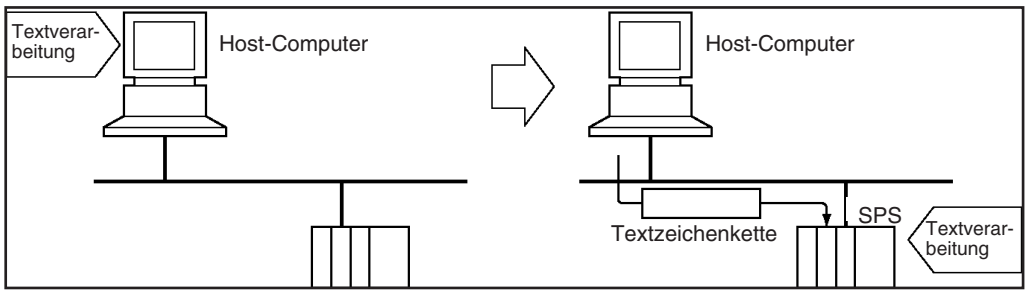


Wie in der folgenden Abbildung dargestellt kann eine Textzeichenkette einfach durch Angabe des Anfangsworts dieser Textzeichenkette spezifiziert werden. Die Textzeichenkettendaten bis zum nächsten NULL-Code (00 hex) werden dann als einziger Block von ASCII-Daten verarbeitet.

Beispiel: MOV$ D00000 D00100



Mit Textverarbeitungsbefehlen können von einer SPS verschiedene Arten der Textzeichenketten-Verarbeitung (Produktdaten usw.) ausgeführt werden, die früher im Host-Computer ausgeführt wurden.



Beispielsweise können Produktionsplandaten wie Produktnamen vom Host-Computer an die SPS übertragen werden. Verschiedene Vorgänge wie das Einfügen und Neuanordnen von Textzeichenketten können anschließend von der SPS durchgeführt werden, wodurch die Datenverarbeitungslast des Host-Computers verringert wird.

ASCII-Zeichen

Die ASCII-Zeichen, die von Textverarbeitungsbefehlen verarbeitet werden können, sind in der folgenden Tabelle aufgeführt.

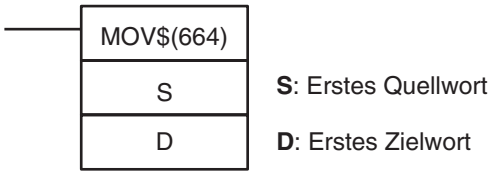
Die vier niederwertigsten Bits \ Die vier höchstwertigen Bits	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	‘	p				ー	タ	ミ		
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			”	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			$	4	D	T	d	t			、	エ	ト	ヤ		
5			%	5	E	U	e	u			・	オ	ナ	ユ		
6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7			’	7	G	W	g	w			ァ	キ	ヌ	ラ		
8			(	8	H	X	h	x			ィ	ク	ネ	リ		
9			)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ェ	コ	ハ	レ		
B			+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C			,	<	L	¥	l	|			ャ	シ	フ	ワ		
D			−	=	M	]	m	}			ュ	ス	ヘ	ン		
E			.	>	N	^	n	~			ョ	セ	ホ	゛		
F			/	?	O	_	o				ッ	ソ	マ	°		

3-33-2 MOV STRING: MOV$(664)

Funktion

Überträgt eine Textzeichenkette.

Kontaktplansymbol



S: Erstes Quellwort

D: Erstes Zielwort

Variationen

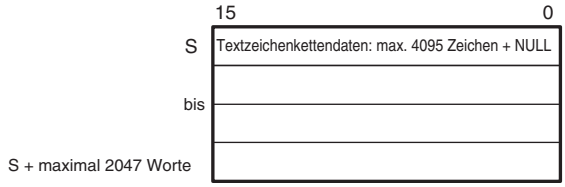
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MOV$(664)
	Ausführung einmal bei steigender Flanke	@MOV$(664)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

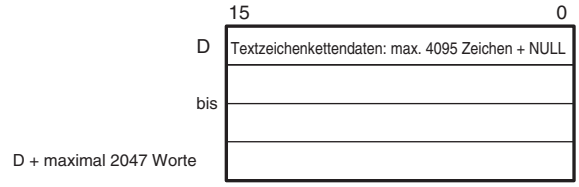
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Erstes Quellwort



D: Erstes Zielwort



Hinweis

1. Die Daten von S bis S + maximal 2047 Worte und von D bis D + maximal 2047 Worte müssen sich im gleichen Bereich befinden.
2. Die Daten von S bis S + maximal 2047 Worte und von D bis D + maximal 2047 Worte können sich überlappen.

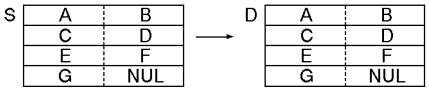
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A447 A448 bis A959	A448 bis A959
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	

Bereich	S	D
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Mit MOV$(664) werden die durch S Spezifizierten Textzeichenkettendaten in ihrem aktuellen Zustand als Textzeichenkettendaten (einschließlich abschließendem NULL-Code) in D übertragen. Die maximale Anzahl an Zeichen, die durch S spezifiziert werden kann, beträgt 4095 (0FFF hex).



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn durch S mehr als 4095 Zeichen spezifiziert sind. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn 0000 (hex) in D übertragen wird. In allen anderen Fällen AUS.

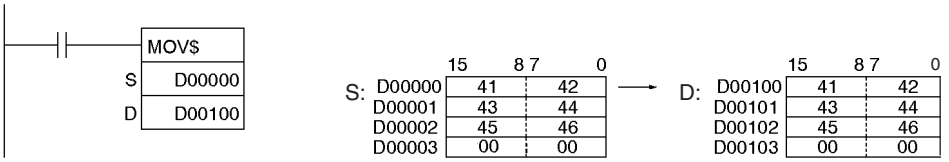
Vorsichtsmaßnahmen

Wenn von S mehr als 4095 Zeichen spezifiziert werden, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Wenn 0000 (hex) in D übertragen wird, wird das Gleich-Flag auf EIN gesetzt.

Beispiel

In diesem Beispiel wird MOV$(664) zur Übertragung der Textzeichenkette ABCDEF verwendet.

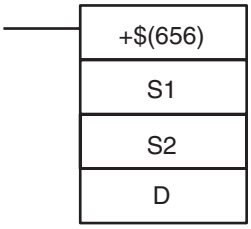


3-33-3 CONCATENATE STRING: +$(656)

Funktion

Verknüpft eine Textzeichenkette mit einer anderen Textzeichenkette.

Kontaktplansymbol



S1: Textzeichenkette 1

S2: Textzeichenkette 2

D: Erstes Zielwort

Variationen

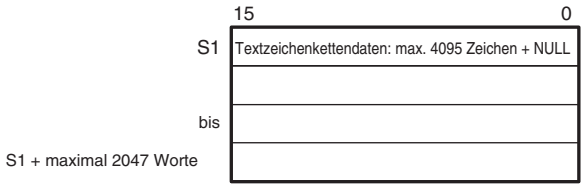
Variationen	Ausführung in jedem Zyklus bei EIN-Bedingung	+$(656)
	Ausführung einmal bei steigender Flanke	@+$(656)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

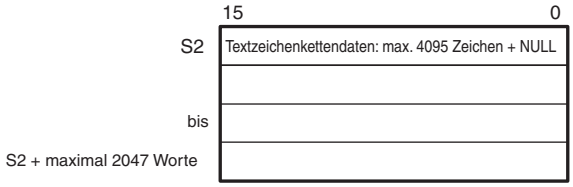
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

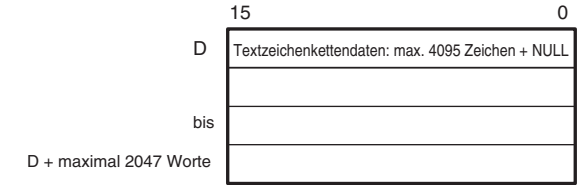
S1: Textzeichenkette 1



S2: Textzeichenkette 2



D: Erstes Zielwort



Hinweis

1. Die Daten von S1 bis S1 + maximal 2047 Worte, von S2 bis S2 + maximal 2047 Worte und von D bis D + maximal 2047 Worte müssen sich im gleichen Bereich befinden.
2. Die Daten von S2 bis S2 + maximal 2047 Worte und von D bis D + maximal 2047 Worte dürfen sich nicht überlappen.

Operanden-Spezifikationen

Bereich	S1	S2	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A447 A448 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis 32767 (n = 0 bis C)		

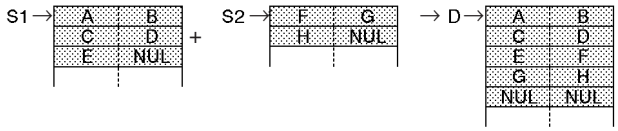
Bereich	S1	S2	D
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0V bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Mit +$(664) werden die durch S1 spezifizierten Textzeichenkettendaten mit den durch S2 spezifizierten Textzeichenkettendaten verbunden und das Ergebnisse als Textzeichenkettendaten (einschließlich abschließendem NULL-Code) in D ausgegeben.

Die maximale Anzahl an Zeichen, die durch S1 und S2 spezifiziert werden können, beträgt 4095 (0FFF hex). Wenn auch nach 4096 Zeichen kein NULL-Code auftritt, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt. Darüber hinaus kann das Ergebnis der Verknüpfung nicht mehr als 4095 Zeichen (0FFF hex) aufweisen. Wenn das Verknüpfungsergebnis zu mehr Zeichen führt, werden nur die ersten 4096 Zeichen (wobei NULL als 4096. Zeichen hinzugefügt wird) in D ausgegeben.

Wenn sowohl für S1 als auch für S2 eine NULL vorliegt, werden die beiden NULL-Zeichen (0000 hex) in D ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn durch S1 und S2 mehr als 4095 Zeichen spezifiziert werden. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn 0000 (hex) in D übertragen wird. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

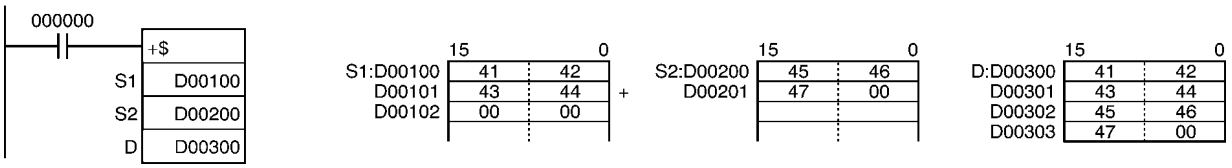
Wenn durch S1 und S2 mehr als 4095 Zeichen spezifiziert werden, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Wenn 0000 (hex) in D übertragen wird, wird das Gleich-Flag auf EIN gesetzt.

Vermeiden Sie eine Überlappung des durch D spezifizierten Anfangsworts mit dem Zeichendatenbereich für S2. Wenn diese überlappen, kann der Befehl nicht ordnungsgemäß ausgeführt werden.

Beispiel

In diesem Beispiel wird +$(656) verwendet, um die Textzeichenketten ABCD und EFG zu verknüpfen und das Ergebnis in D auszugeben.

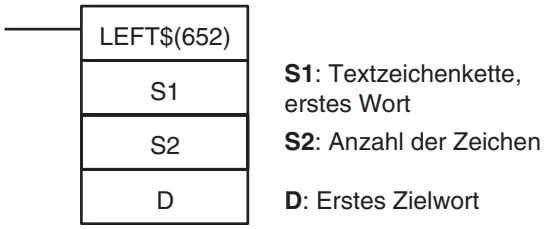


3-33-4 GET STRING LEFT: LEFT$(652)

Funktion

Liest eine spezifizierte Anzahl von Zeichen von der linken Seite (dem Anfang) einer Textzeichenkette.

Kontaktplansymbol



S1: Textzeichenkette, erstes Wort

S2: Anzahl der Zeichen

D: Erstes Zielwort

Variationen

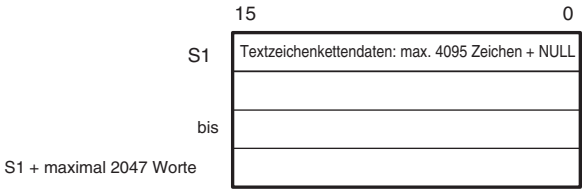
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	LEFT$(652)
	Ausführung einmal bei steigender Flanke	@LEFT$(652)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

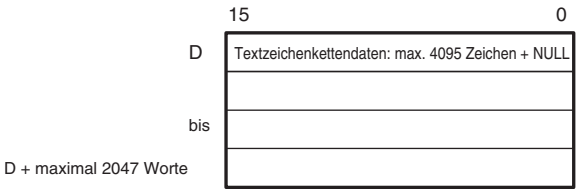
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S1: Textzeichenkette



S2: Anzahl der Zeichen (0000 bis 0FFF hex bzw. &0 bis &4095)



Hinweis

1. Die Daten von S1 bis S1 + maximal 2047 Worte und von D bis D + maximal 2047 Worte müssen sich im gleichen Bereich befinden.
2. Die Daten von S1 bis S1 + maximal 2047 Worte und von D bis D + maximal 2047 Worte dürfen sich überlappen.

Operanden-Spezifikationen

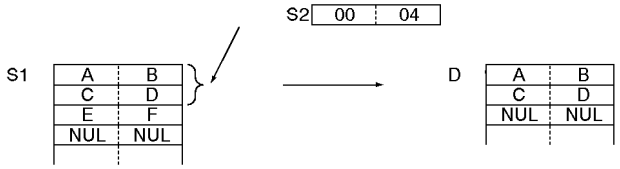
Bereich	S1	S2	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A447 A448 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0000 bis #0FFF (binär) oder &0 bis &4095	---
Datenregister	---	DR0 bis DR15	---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl LEFT$(652) liest die durch S2 spezifizierte Zeichenanzahl von links (Anfang) des ersten Worts der durch S1 spezifizierten Textzeichenkette bis zum NULL-Code (00 hex) und gibt das Ergebnis in D aus (wobei am Ende NULL hinzugefügt wird).

Wenn die Anzahl der abgerufenen Zeichen die durch S1 spezifizierte Zeichenzahl übersteigt, wird die gesamte S1-Textzeichenkette ausgegeben.

Wenn 0 (0000 hex) als Anzahl der zu lesenden Zeichen spezifiziert wird, werden zwei NULL-Zeichen (0000 hex) in D ausgegeben.



Flags

Name	Bezeich-nung	Funktion
Fehler-Flag	ER	EIN, wenn durch S1 mehr als 4095 Zeichen spezifiziert werden. EIN, wenn durch S2 mehr als 4095 Zeichen (0FFF hex) spezifiziert werden. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn 0000 (hex) in D ausgegeben wird. In allen anderen Fällen AUS.

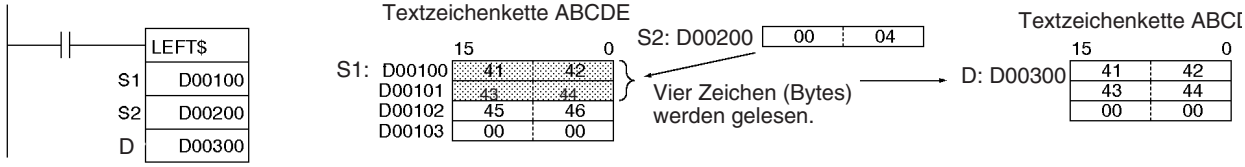
Vorsichtsmaßnahmen

Die maximale Anzahl an zu lesenden Zeichen, die durch S2 spezifiziert werden kann, beträgt 4095 (0FFF hex). Wenn eine höhere Anzahl spezifiziert wird, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Wenn 0000 (hex) in D ausgegeben wird, wird das Gleich-Flag auf EIN gesetzt.

Beispiel

In diesem Beispiel wird der Befehl LEFT$(652) zum Lesen von vier Zeichen verwendet.

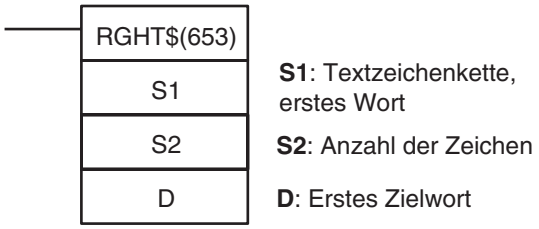


3-33-5 GET STRING RIGHT: RGHT$(653)

Funktion

Liest eine spezifizierte Anzahl von Zeichen von der rechten Seite (dem Ende) einer Textzeichenkette.

Kontaktplansymbol



Variationen

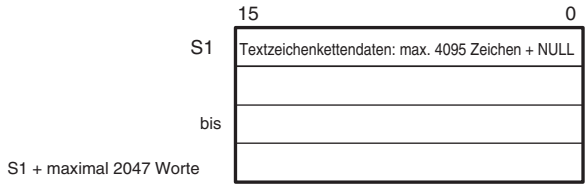
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	RGHT$(653)
	Ausführung einmal bei steigender Flanke	@RGHT$(653)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

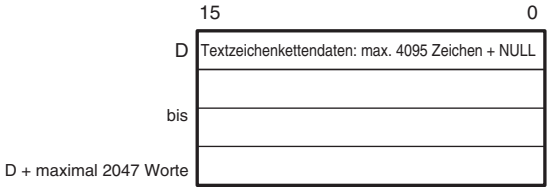
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S1: Textzeichenkette



S2: Anzahl der Zeichen (0000 bis 0FFF hex bzw. &0 bis &4095)



Hinweis

1. Die Daten von S1 bis S1 + maximal 2047 Worte und von D bis D + maximal 2047 Worte müssen sich im gleichen Bereich befinden.
2. Die Daten von S1 bis S1 + maximal 2047 Worte und von D bis D + maximal 2047 Worte dürfen sich überlappen.

Operanden-Spezifikationen

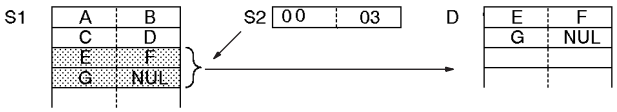
Bereich	S1	S2	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A447 A448 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0000 bis #0FFF (binär) oder &0 bis &4095	---
Datenregister	---	DR0 bis DR15	---
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl RGHT$(653) liest die durch S2 spezifizierte Zeichenanzahl von links (vom Anfang) des ersten Worts der durch S1 spezifizierten Textzeichenkette bis zum NULL-Code (00 hex) und gibt das Ergebnis in D aus (wobei am Ende NULL hinzugefügt wird).

Wenn die Anzahl der zu lesenden Zeichen die durch S1 spezifizierte Zeichenanzahl übersteigt, wird die gesamte S1-Textzeichenkette ausgegeben.

Wenn 0 (0000 hex) als Anzahl der zu lesenden Zeichen spezifiziert wird, werden zwei NULL-Zeichen (0000 hex) in D ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn durch S1 mehr als 4095 Zeichen spezifiziert werden. EIN, wenn durch S2 mehr als 4095 Zeichen (0FFF hex) spezifiziert werden. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn 0000 (hex) in D ausgegeben wird. In allen anderen Fällen AUS.

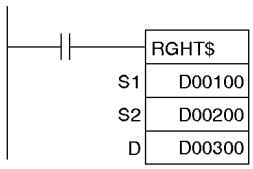
Vorsichtsmaßnahmen

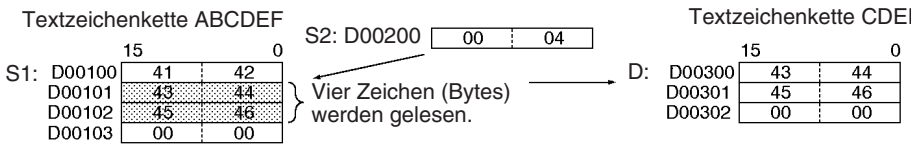
Die maximale Anzahl an zu lesenden Zeichen, die durch S2 spezifiziert werden kann, beträgt 4095 (0FFF hex). Wenn eine höhere Anzahl spezifiziert wird, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Wenn 0000 (hex) in D ausgegeben wird, wird das Gleich-Flag auf EIN gesetzt.

Beispiel

In diesem Beispiel wird der Befehl RGHT$(653) zum Lesen von vier Zeichen verwendet.



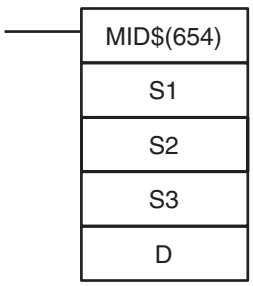


3-33-6 GET STRING MIDDLE: MID$(654)

Funktion

Liest eine spezifizierte Anzahl von Zeichen ab einer beliebigen Stelle innerhalb einer Textzeichenkette.

Kontaktplansymbol



S1: Textzeichenkette, erstes Wort

S2: Anzahl der Zeichen

S3: Anfangsposition

D: Erstes Zielwort

Variationen

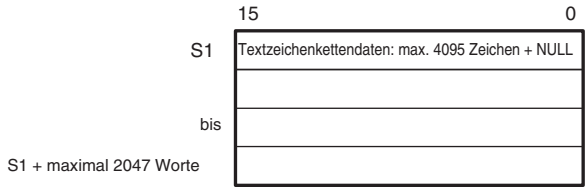
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	MID$(654)
	Ausführung einmal bei steigender Flanke	@MID$(654)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

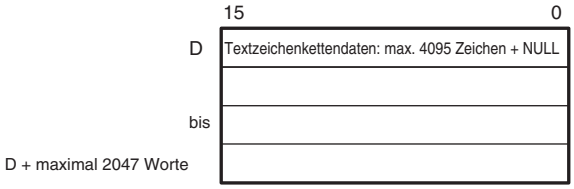
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S1: Textzeichenkette



S2: Anzahl der Zeichen (0000 bis 0FFF hex bzw. &0 bis &4095)
S3: Anfangsposition (0001 bis 0FFF hex bzw. &1 bis &4095)



Hinweis

1. Die Daten von S1 bis S1 + maximal 2047 Worte und von D bis D + maximal 2047 Worte müssen sich im gleichen Bereich befinden.
2. Die Daten von S1 bis S1 + maximal 2047 Worte und von D bis D + maximal 2047 Worte dürfen sich überlappen.

Operanden-Spezifikationen

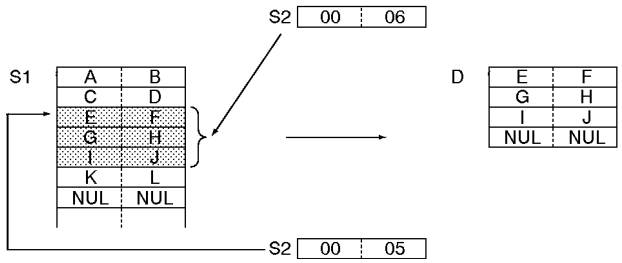
Bereich	S1	S2	S3	D
CIO-Bereich	CIO 0000 bis CIO 6143			
Arbeitsbereich nicht remanent	W000 bis W511			
Arbeitsbereich remanent	H000 bis H511			
Systembereich	A000 bis A447 A448 bis A959			A448 bis A959
Zeitgeberbereich	T0000 bis T4095			
Zählerbereich	C0000 bis C4095			
DM-Bereich	D00000 bis 32767			
EM-Bereich ohne Bank	E00000 bis E32767			
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)			
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)			

Bereich	S1	S2	S3	D
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)			
Konstanten	---	#0000 bis #0FFF (binär) oder &0 bis &4095	#0001 bis #0FFF (binär) oder &1 bis &4095	---
Datenregister	---	DR0 bis DR15		
Indexregister	---			
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15			

Beschreibung

Innerhalb der Textzeichenkette, die anhand des durch S1 spezifizierten ersten Worts bis zum NULL-Code (00 hex) identifiziert wird, liest der Befehl MID$(654) die durch S2 spezifizierte Zeichenanzahl ab dem durch S3 spezifizierten Anfangswort und gibt das Ergebnis in D aus (wobei am Ende NULL hinzugefügt wird).

Wenn sich die Anzahl der zu lesenden Zeichen über das Ende der durch S1 spezifizierten Textzeichenkette hinaus erstreckt, wird die Textzeichenkette bis zum Ende ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn durch S1 mehr als 4095 Zeichen spezifiziert werden. EIN, wenn durch S2 mehr als 4095 Zeichen (0FFF hex) spezifiziert werden. EIN, wenn die Daten für S3 nicht innerhalb des Bereichs von 1 bis 4.095 (0001 bis 0FFF hex) liegen. EIN, wenn S3 größer als S1 ist. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn 0000 (hex) in D ausgegeben wird. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Der Bereich für die durch S3 spezifizierte Anfangsposition umfasst das 1. bis das 4095. Zeichen (0001 to 0FFF hex). Wenn die Einstellung außerhalb dieses Bereichs liegt, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

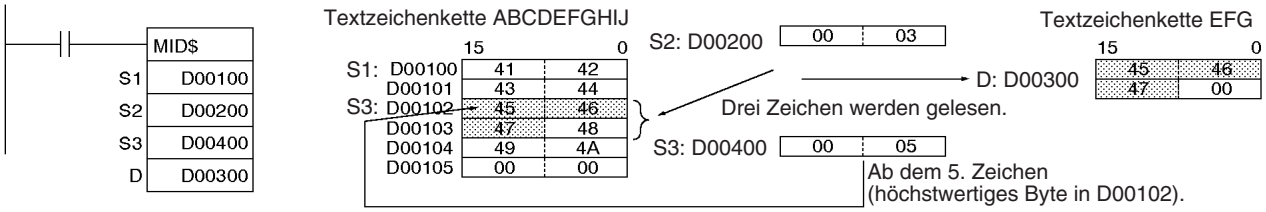
Die maximale Anzahl an zu lesenden Zeichen, die durch S2 spezifiziert werden kann, beträgt 4095 (0FFF hex). Wenn eine höhere Anzahl spezifiziert wird, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Wenn 0 (0000 hex) als Anzahl der zu lesenden Zeichen spezifiziert wird, werden zwei NULL-Zeichen (0000 hex) in D ausgegeben.

Wenn 0000 (hex) in D ausgegeben wird, wird das Gleich-Flag auf EIN gesetzt.

Beispiel

In diesem Beispiel wird der Befehl MID$(654) zum Lesen von drei Zeichen verwendet.

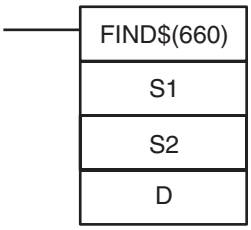


3-33-7 FIND IN STRING: FIND$(660)

Funktion

Sucht eine bestimmte Textzeichenkette innerhalb einer Textzeichenkette.

Kontaktplansymbol



S1: Quell-Textzeichenkette, erstes Wort
S2: Zu suchende Textzeichenkette, erstes Wort
D: Erstes Zielwort

Variationen

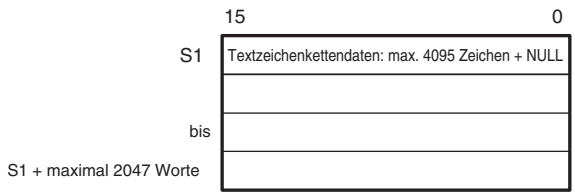
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	FIND$(660)
	Ausführung einmal bei steigender Flanke	@FIND$(660)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

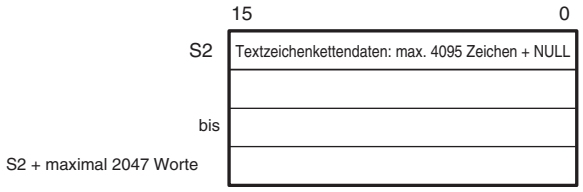
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S1: Quell-Textzeichenkette



S2: Zu suchende Textzeichenkette



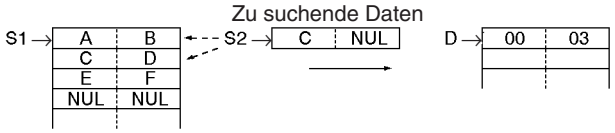
Hinweis Die Daten von S1 bis S1 + maximal 2047 Worte und von S2 bis S2 + maximal 2047 Worte müssen sich im gleichen Bereich befinden.

Operanden-Spezifikationen

Bereich	S1	S2	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A447 A448 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	---		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl FIND$(660) sucht die durch S2 spezifizierte Textzeichenkette in der durch S1 spezifizierten Textzeichenkette und gibt das Ergebnis (eine gegebene Anzahl von Zeichen ab dem Anfang von S1) als Binärdaten in D aus. Wenn keine übereinstimmende Textzeichenkette vorhanden ist, wird 0000 hex in D ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn von S1 oder S2 mehr als 4095 Zeichen spezifiziert werden. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn 0000 (hex) in D ausgegeben wird. In allen anderen Fällen AUS.

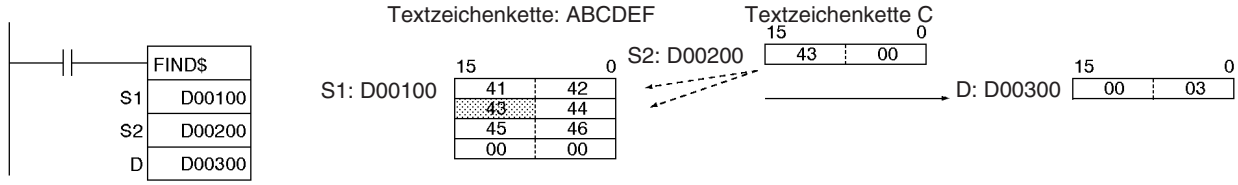
Vorsichtsmaßnahmen

Die maximale Anzahl an zu lesenden Zeichen, die durch S1 oder S2 spezifiziert werden kann, beträgt 4095 (0FFF hex). Wenn eine höhere Anzahl spezifiziert wird, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Wenn 0000 (hex) in D ausgegeben wird, wird das Gleich-Flag auf EIN gesetzt.

Beispiel

In diesem Beispiel wird der Befehl FIND$(660) verwendet, um ein Zeichen in einer Textzeichenkette zu suchen.

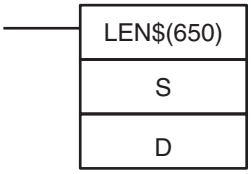


3-33-8 STRING LENGTH: LEN$(650)

Funktion

Berechnet die Länge einer Textzeichenkette.

Kontaktplansymbol



S: Textzeichenkette, erstes Wort

D: Erstes Zielwort

Variationen

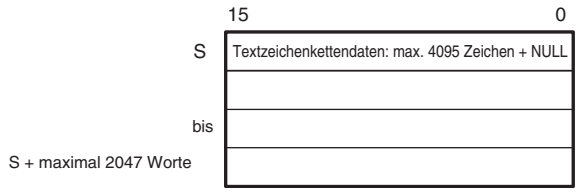
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	LEN$(650)
	Ausführung einmal bei steigender Flanke	@LEN$(650)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Textzeichenkette



Hinweis Die Daten von S bis S + maximal 2047 Worte müssen sich im gleichen Bereich befinden.

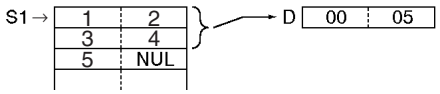
Operanden-Spezifikationen

Bereich	S	D
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A447 A448 bis A959	A448 bis A959

Bereich	S	D
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	DR0 bis DR15
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Der Befehl LENS$(650) berechnet die Anzahl der Zeichen vom durch S spezifizierten ersten Wort der Textzeichenkette bis zum NULL-Code (00 hex), einschließlich des NULL-Codes selbst, und gibt das Ergebnis als Binärdaten in D aus. Wenn sich am Anfang der Textzeichenkette ein NULL-Code befindet, ist das berechnete Ergebnis 0000 hex.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn das berechnete Ergebnis die Anzahl von 4095 Zeichen übersteigt. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das berechnete Ergebnis 0 ist. In allen anderen Fällen AUS.

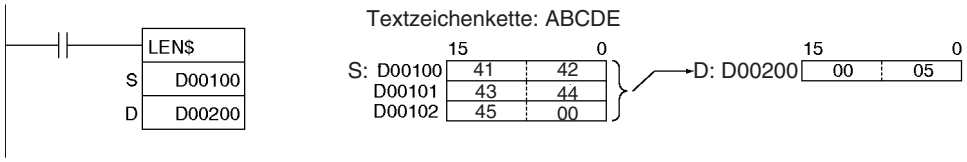
Vorsichtsmaßnahmen

Die maximale Anzahl an Zeichen beträgt 4095 (0FFF hex). Wenn mehr als diese Anzahl vorhanden sind (d. h., wenn sich vor dem 4096. Zeichen kein NULL-Code befindet), tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Wenn 0000 (hex) in D ausgegeben wird, wird das Gleich-Flag auf EIN gesetzt.

Beispiel

In diesem Beispiel wird mit LENS$(650) die Anzahl der Zeichen berechnet und das Ergebnis ausgegeben.

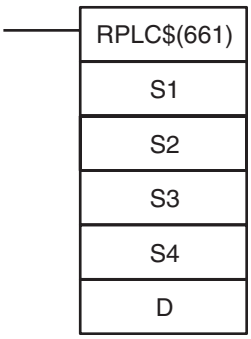


3-33-9 REPLACE IN STRING: RPLC$(661)

Funktion

Ersetzt eine Textzeichenkette durch eine spezifizierte Textzeichenkette ab einer bestimmten Position.

Kontaktplansymbol



S1: Textzeichenkette, erstes Wort

S2: Ersatztextzeichenkette, erstes Wort

S3: Anzahl der Zeichen

S4: Anfangsposition

D: Erstes Zielwort

Variationen

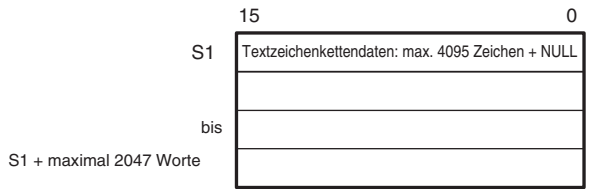
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	RPLC$(661)
	Ausführung einmal bei steigender Flanke	@RPLC$(661)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

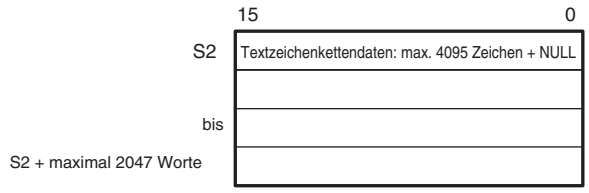
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

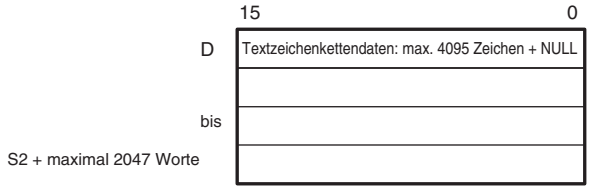
S1: Textzeichenkette



S2: Ersatztextzeichenkette



S3: Anzahl der Zeichen (0000 bis 0FFF hex bzw. &0 bis &4095)
S4: Anfangsposition (0001 bis 0FFF hex bzw. &0 bis &4095)



Hinweis

1. Die Daten von S1 bis S1 + maximal 2047 Worte, von S2 bis S2 + maximal 2047 Worte und von D bis D + maximal 2047 Worte müssen sich im gleichen Bereich befinden.
2. Die Daten von D bis D + maximal 2047 Worte und entweder von S1 bis S1 + maximal 2047 Worte oder von S2 bis S2 + maximal 2047 Worte dürfen sich überlappen.

Operanden-Spezifikationen

Bereich	S1	S2	S3	S4	D
CIO-Bereich	CIO 0000 bis CIO 6143				
Arbeitsbereich nicht remanent	W000 bis W511				
Arbeitsbereich remanent	H000 bis H511				
Systembereich	A000 bis A447 A448 bis A959				A448 bis A959
Zeitgeberbereich	T0000 bis T4095				
Zählerbereich	C0000 bis C4095				
DM-Bereich	D00000 bis D32767				
EM-Bereich ohne Bank	E00000 bis E32767				
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)				
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)				
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)				
Konstanten	---		#0000 bis #0FFF (binär) oder &0 bis &4095	#0001 bis #0FFF (binär) oder &1 bis &4095	---
Datenregister	---	DR0 bis DR15			---
Indexregister	---				
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15				

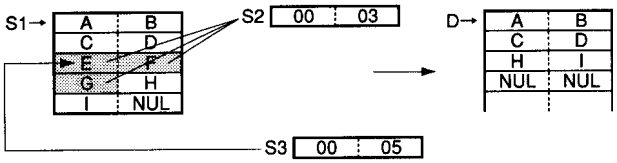
Beschreibung

Der Befehl RPLC$(661) ersetzt in der durch S1 spezifizierten Textzeichenkette den bei der durch S4 spezifizierten Anfangsposition beginnenden Teil durch die durch S2 spezifizierte Textzeichenkette und gibt das Ergebnis als Textzeichenkettendaten in D aus (wobei am Ende NULL hinzugefügt wird). Die Anzahl der zu ersetzenden Zeichen wird durch S3 spezifiziert.

Die maximale Anzahl an Zeichen im Ergebnis beträgt 4095 (0FFF hex). Wenn die Anzahl größer ist, werden nur 4095 Zeichen ausgegeben (wobei NULL als 4096. Zeichen hinzugefügt wird).

Es können 0 bis 4095 Zeichen (0000 bis 0FFF hex) ersetzt werden. Wenn die Anzahl 0 beträgt, wird die durch S1 spezifizierte Textzeichenkette ohne Änderung in D ausgegeben. Wenn die S2-Textzeichenkette NULL ist, entspricht die Funktion im Ergebnis dem Löschen des spezifizierten Textbereichs in S1.

Wenn die S1-Textzeichenkette von Anfang bis Ende durch NULL ersetzt wird, werden in D zwei NULL-Zeichen (0000 hex) ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn von S1 oder S2 mehr als 4095 Zeichen spezifiziert werden. EIN, wenn durch S3 mehr als 4095 Zeichen (0FFF hex) spezifiziert werden. EIN, wenn die S4-Daten nicht innerhalb des Bereichs von 1 bis 4095 (0001 bis 0FFF hex) liegen. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn 0000 (hex) in D ausgegeben wird. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die maximale Anzahl an Zeichen für S1 oder S2 beträgt 4095 (0FFF hex). Wenn mehr als diese Anzahl vorhanden sind (d. h., wenn sich vor dem 4096. Zeichen kein NULL-Code befindet), tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Der Bereich für die durch S4 spezifizierte Anfangsposition umfasst das 1. bis zum 4095. Zeichen (0001 to 0FFF hex). Wenn die Einstellung außerhalb dieses Bereichs liegt, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

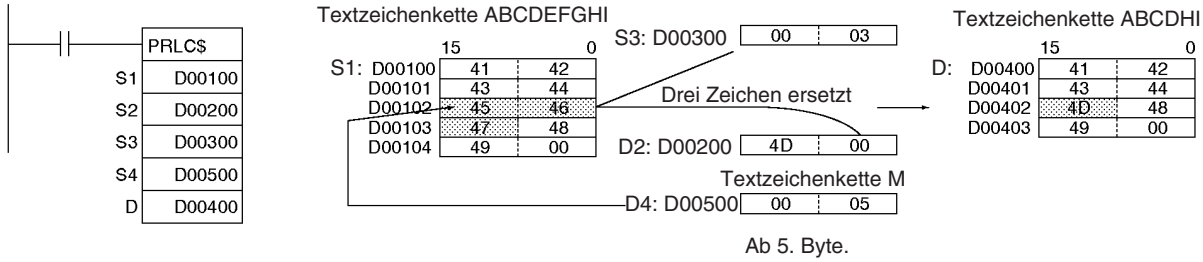
Wenn die durch S4 spezifizierte Anfangsposition über das Ende der durch S1 spezifizierten Textzeichenkette hinausgeht, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Wenn 0000 (hex) in D ausgegeben wird, wird das Gleich-Flag auf EIN gesetzt.

Legen Sie das erste Zielwort D so fest, dass es sich nicht mit den durch das erste Wort S2 der Ersatztextzeichenkette festgelegten Bereichen überlappt. RPLC$(654) funktioniert nicht ordnungsgemäß, wenn sich diese Bereiche überlappen.

Beispiel

In diesem Beispiel wird der Befehl RPLC$(654) zum Ersetzen von drei Zeichen verwendet.



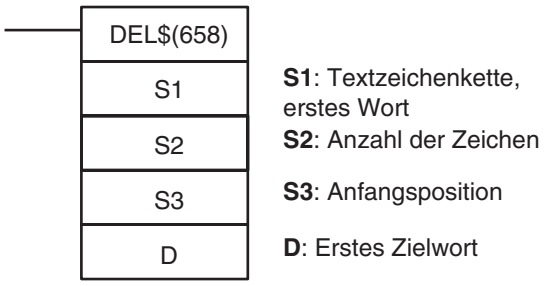
Ab 5. Byte.

3-33-10 DELETE STRING: DEL$(658)

Funktion

Löscht eine spezifizierte Textzeichenkette aus der Mitte einer Textzeichenkette.

Kontaktplansymbol



S1: Textzeichenkette, erstes Wort
S2: Anzahl der Zeichen
S3: Anfangsposition
D: Erstes Zielwort

Variationen

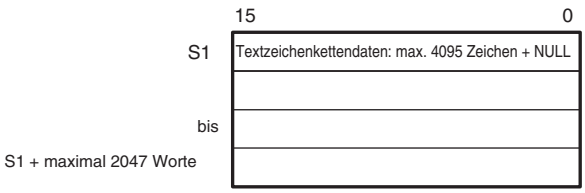
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	DEL$(658)
	Ausführung einmal bei steigender Flanke	@DEL$(658)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

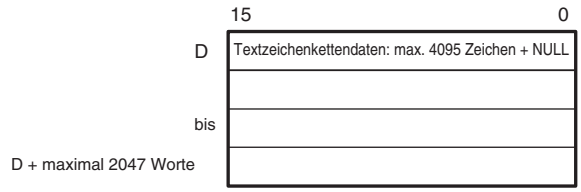
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S1: Textzeichenkette



S2: Anzahl der Zeichen (0000 bis 0FFF hex bzw. &0 bis &4095)
S3: Anfangsposition (0001 bis 0FFF hex bzw. &1 bis &4095)



Hinweis

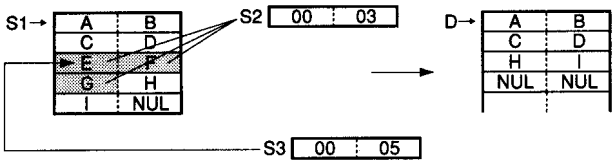
1. Die Daten von S1 bis S1 + maximal 2047 Worte, von S2 bis S2 + maximal 2047 Worte und von D bis D + maximal 2047 Worte müssen sich im gleichen Bereich befinden.
2. Die Daten von S1 bis S1 + maximal 2047 Worte und von D bis D + maximal 2047 Worte dürfen sich überlappen.

Operanden-Spezifikationen

Bereich	S1	S2	S3	D
CIO-Bereich	CIO 0000 bis CIO 6143			
Arbeitsbereich nicht remanent	W000 bis W511			
Arbeitsbereich remanent	H000 bis H511			
Systembereich	A000 bis A447 A448 bis A959			A448 bis A959
Zeitgeberbereich	T0000 bis T4095			
Zählerbereich	C0000 bis C4095			
DM-Bereich	D00000 bis D32767			
EM-Bereich ohne Bank	E00000 bis E32767			
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)			
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)			
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)			
Konstanten	---	#0000 bis #0FFF (binär) oder &0 bis &4095	#0001 bis #0FFF (binär) oder &1 bis &4095	---
Datenregister	---	DR0 bis DR15		---
Indexregister	---			
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15			

Beschreibung

Durch den Befehl DEL$(658) wird innerhalb der durch S1 spezifizierten Textzeichenkette die durch S2 spezifizierte Anzahl von Zeichen ab dem durch S3 spezifizierten Anfangswort gelöscht und das Ergebnis als Textzeichenkettendaten in D ausgegeben (wobei am Ende NULL hinzugefügt wird).



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn durch S1 mehr als 4095 Zeichen spezifiziert werden. EIN, wenn durch S2 mehr als 4095 Zeichen (0FFF hex) spezifiziert werden. EIN, wenn die Daten für S3 nicht innerhalb des Bereichs von 1 bis 4.095 (0001 bis 0FFF hex) liegen. EIN, wenn S3 größer als S1 ist. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn 0000 hex in D ausgegeben wird. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die maximale Anzahl an Zeichen für S1 beträgt 4095 (0FFF hex). Wenn mehr als diese Anzahl vorhanden sind (d. h., wenn sich vor dem 4096. Zeichen kein NULL-Code befindet), tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

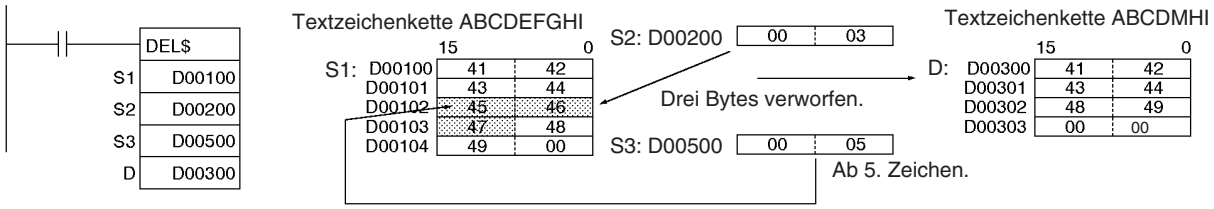
Der Bereich für die durch S3 spezifizierte Anfangsposition umfasst das 1. bis zum 4095. Zeichen (0001 to 0FFF hex). Wenn die Einstellung außerhalb dieses Bereichs liegt, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn die für S1 spezifizierte Wortanzahl die Länge der Textzeichenkette überschreitet, wird das Fehler-Flag auf EIN gesetzt.

Wenn sich die Anzahl der zu löschenden Zeichen über das Ende der S1-Textzeichenkette hinaus erstreckt, werden alle Zeichen bis zum Ende gelöscht. Wenn alle Zeichen vom Anfang bis zum Ende von S1 als zu löschen spezifiziert sind, wird 000 hex in D ausgegeben.

Beispiel

In diesem Beispiel wird der Befehl DEL$(658) zum Löschen von drei Zeichen verwendet.

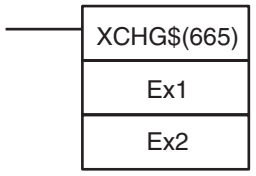


3-33-11 EXCHANGE STRING: XCHG$(665)

Funktion

Ersetzt eine spezifizierte Textzeichenkette durch eine andere Textzeichenkette.

Kontaktplansymbol



Ex1: Erstes Austauschwort 1

Ex2: Erstes Austauschwort 2

Variationen

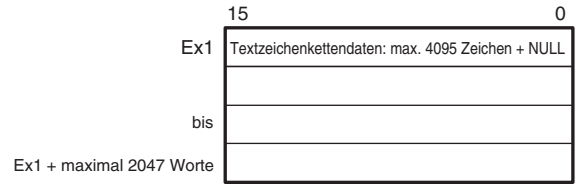
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	XCHG$(665)
	Ausführung einmal bei steigender Flanke	@XCHG$(665)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

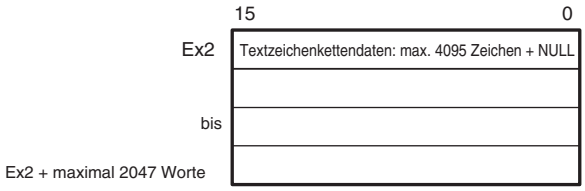
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Ex1: Erstes Austauschwort 1



Ex2: Erstes Austauschwort 2



Hinweis

1. Die Daten von Ex1 bis Ex1 + maximal 2047 Worte und von Ex2 bis Ex2 + maximal 2047 Worte müssen sich im gleichen Bereich befinden.
2. Die Daten von Ex1 bis Ex1 + maximal 2047 Worte und von Ex2 bis Ex2 + maximal 2047 Worte dürfen sich nicht überlappen.

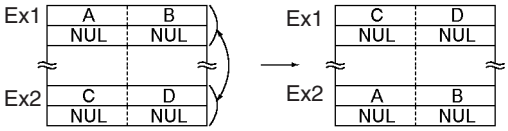
Operanden-Spezifikationen

Bereich	Ex1	Ex2
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A448 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	

Bereich	Ex1	Ex2
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

XCHG$(665) ersetzt die durch Ex1 spezifizierte Textzeichenkette durch die durch Ex2 spezifizierte Textzeichenkette. Wenn entweder Ex1 oder Ex2 NULL ist, werden in das jeweils andere zwei NULL-Zeichen (0000 hex) ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn durch Ex1 oder Ex2 mehr als 4095 Zeichen spezifiziert werden. EIN, wenn sich die Ex1- und Ex2-Daten überlappen. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

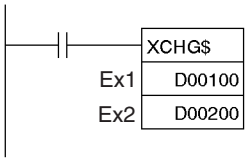
Vorsichtsmaßnahmen

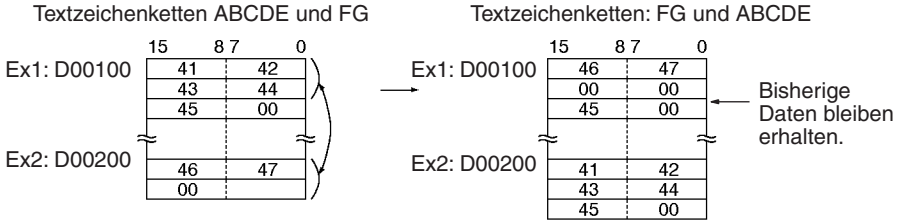
Die maximale Anzahl an Zeichen, die durch Ex1 oder Ex2 spezifiziert werden können, beträgt 4095 (0FFF hex). Wenn eine höhere Anzahl spezifiziert wird, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Wenn sich die durch Ex1 und Ex2 spezifizierten Textzeichenkettendaten überlappen, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Beispiel

In diesem Beispiel wird XCHG$(665) zum Austauschen von zwei Textzeichenketten verwendet.



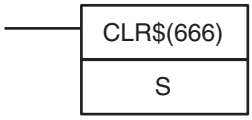


3-33-12 CLEAR STRING: CLR$(666)

Funktion

Löscht eine komplette Textzeichenkette zu NULL (00 hex).

Kontaktplansymbol



S: Textzeichenkette, erstes Wort

Variationen

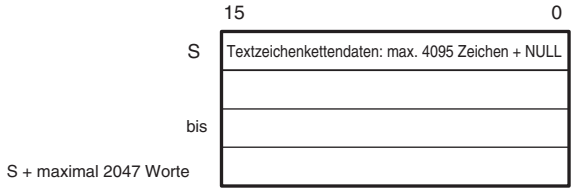
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	CLR$(666)
	Ausführung einmal bei steigender Flanke	@CLR$(666)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Textzeichenkette, erstes Wort



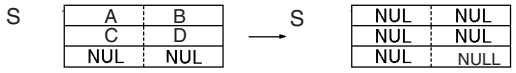
Hinweis Die Daten von S bis S + maximal 2047 Worte müssen sich im gleichen Bereich befinden.

Operanden-Spezifikationen

Bereich	S
CIO-Bereich	CIO 0000 bis CIO 6143
Arbeitsbereich nicht remanent	W000 bis W511
Arbeitsbereich remanent	H000 bis H511
Systembereich	A448 bis A959
Zeitgeberbereich	T0000 bis T4095
Zählerbereich	C0000 bis C4095
DM-Bereich	D00000 bis D32767
EM-Bereich ohne Bank	E00000 bis E32767
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)
Konstanten	---
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15

Beschreibung

Der Befehl CLR$(666) löscht die gesamte Textzeichenkette vom ersten durch S spezifizierten Wort bis zum NULL-Code (00 hex) auf NULL (00 hex). Die maximale Anzahl von Zeichen, die gelöscht werden können, beträgt 4096. Wenn vor dem 4096. Zeichen kein NULL-Code auftritt, werden nur 4096 Zeichen gelöscht.

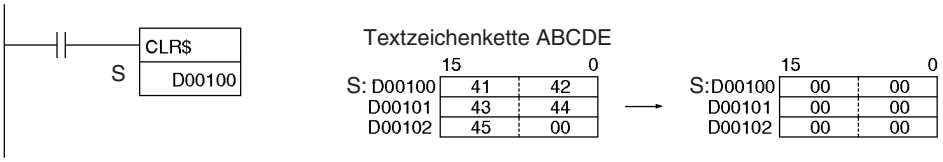


Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.

Beispiel

In diesem Beispiel wird CLR$(666) zum Löschen der Textzeichenkette ABCDE verwendet.

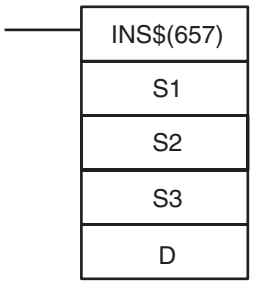


3-33-13 INSERT INTO STRING: INS$(657)

Funktion

Fügt eine spezifizierte Textzeichenkette in die Mitte einer Textzeichenkette ein.

Kontaktplansymbol



S1: Basis-Textzeichenkette, erstes Wort
S2: Einzufügende Textzeichenkette, erstes Wort
S3: Anfangsposition
D: Erstes Zielwort

Variationen

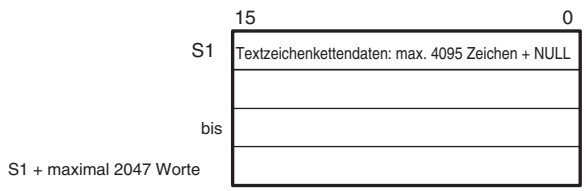
Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	INS$(657)
	Ausführung einmal bei steigender Flanke	@INS$(657)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

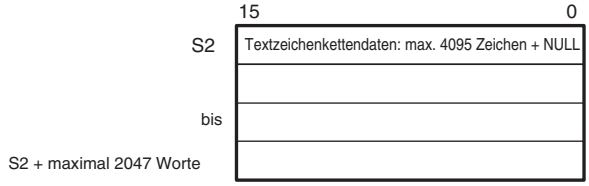
Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

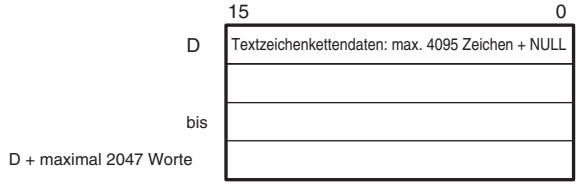
S1: Basis-Textzeichenkette



S2: Einzufügende Textzeichenkette



S3: Anfangsposition (0000 bis 0FFF hex bzw. &0 bis &4095)



Hinweis

1. Die Daten von S1 bis S1 + maximal 2047 Worte, von S2 bis S2 + maximal 2047 Worte und von D bis D + maximal 2047 Worte müssen sich im gleichen Bereich befinden.
2. Die Daten von S2 bis S2 + maximal 2047 Worte und von D bis D + maximal 2047 Worte dürfen sich nicht überlappen. Die Daten von S1 bis S1 + maximal 2047 Worte und von D bis D + maximal 2047 Worte dürfen sich überlappen. Die Daten von S1 bis S1 + maximal 2047 Worte und von S2 bis S2 + maximal 2047 Worte dürfen sich auch überlappen.

Operanden-Spezifikationen

Bereich	S1	S2	S3	D
CIO-Bereich	CIO 0000 bis CIO 6143			
Arbeitsbereich nicht remanent	W000 bis W511			
Arbeitsbereich remanent	H000 bis H511			
Systembereich	A000 bis A447 A448 bis A959			A448 bis A959
Zeitgeberbereich	T0000 bis T4095			
Zählerbereich	C0000 bis C4095			
DM-Bereich	D00000 bis D32767			
EM-Bereich ohne Bank	E00000 bis E32767			
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)			
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)			
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)			
Konstanten	---		#0000 bis #0FFF (binär) oder &0 bis &4095	---
Datenregister	---		DR0 bis DR15	---

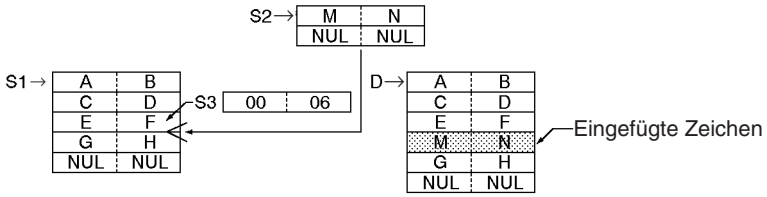
Bereich	S1	S2	S3	D
Indexregister	---			
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15			

Beschreibung

Der Befehl INS$(657) fügt die durch S2 spezifizierte Textzeichenkette hinter dem durch S3 spezifizierten Anfangswort in die durch S1 spezifizierte Textzeichenkette ein und gibt das Ergebnis als Textzeichenkettendaten in D aus (wobei am Ende NULL hinzugefügt wird).

Die maximale Anzahl an Zeichen, die eingefügt werden können, beträgt 4095 (0FFF hex). Wenn mehr Zeichen vorhanden sind, werden nur 4095 Zeichen in D ausgegeben (wobei NULL als 4096. Zeichen hinzugefügt wird).

Wenn entweder S1 oder S2 NULL ist, wird die von der jeweils anderen Variablen spezifizierte Textzeichenkette unverändert in D ausgegeben. Wenn S1 und S2 beide NULL sind, werden zwei NULL-Zeichen (0000 hex) in D ausgegeben.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn von S1 oder S2 mehr als 4095 Zeichen spezifiziert werden. EIN, wenn S3 den Wert 4095 (0FFF hex) übersteigt. EIN, wenn bei Spezifikation der Hintergrundausführung das Kommunikationsschnittstelle-Frei-Flag der als *Komm.-Port-Nummer* bei *Hintergrundausführung* spezifizierten Kommunikationsschnittstellen-Nummer auf AUS gesetzt ist. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn 0000 (hex) in D ausgegeben wird. In allen anderen Fällen AUS.

Vorsichtsmaßnahmen

Die maximale Anzahl an Zeichen für S1 und S2 beträgt 4095 (0FFF hex). Wenn mehr als diese Anzahl vorhanden sind (d. h., wenn sich vor dem 4096. Zeichen kein NULL-Code befindet), tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

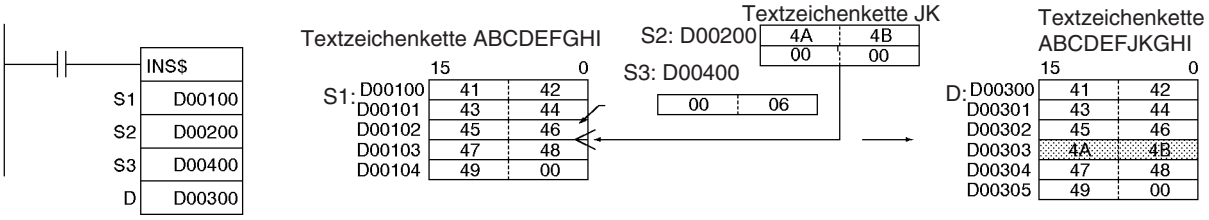
Der Bereich für die durch S3 spezifizierte Anfangsposition ist 0 bis 4095. Wenn die Einstellung außerhalb dieses Bereichs liegt, führt dies zu einem Fehler und das Fehler-Flag wird auf EIN gesetzt.

Wenn 0000 (hex) in D ausgegeben wird, wird das Gleich-Flag auf EIN gesetzt.

Die durch D spezifizierten Zielworte dürfen nicht mit den durch S2 spezifizierten Textzeichenkettendaten überlappen. Andernfalls wird die Funktion nicht ordnungsgemäß ausgeführt.

Beispiel

In diesem Beispiel wird der Befehl INS$(657) zum Einfügen von zwei Zeichen verwendet.



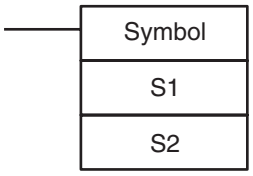
3-33-14 Zeichenketten-Vergleichsbefehle (670 bis 675)

Funktion

Zeichenketten-Vergleichsbefehle (=$, <>$, <$, <=$, >$, >=$) vergleichen zwei Textzeichenketten, an deren Anfang beginnend. Dabei werden die Werte der ASCII-Codes verglichen. Wenn das Ergebnis des Vergleichs wahr ist, wird eine EIN-Ausführungsbedingung für LOAD, AND oder OR erzeugt.

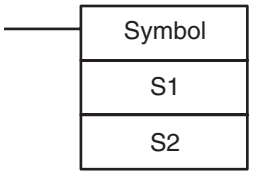
Kontaktplansymbol

LD



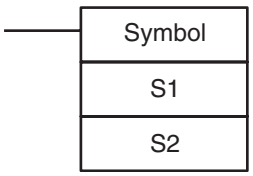
S1:
Textzeichenkette 1
S2:
Textzeichenkette 2

AND (Reihenschaltung)



S1:
Textzeichenkette 1
S2:
Textzeichenkette 2

OR (Parallelschaltung)



S1:
Textzeichenkette 1
S2:
Textzeichenkette 2

Variationen

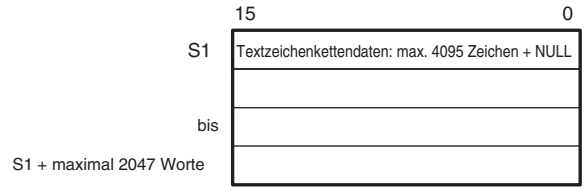
Variationen	Erzeugt in jedem Zyklus ein EIN, in dem die Vergleichsbedingung wahr ist	Zeichenketten-Vergleichsbefehle
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

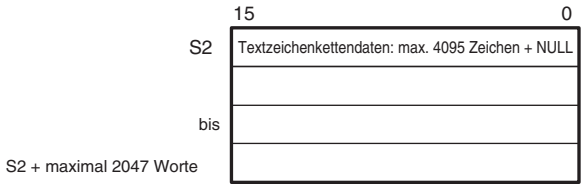
Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S1: Textzeichenkette 1



S2: Textzeichenkette 2



Hinweis

1. Die Daten von S1 bis S1 + maximal 2047 Worte und von S2 bis S2 + maximal 2047 Worte müssen sich im selben Bereich befinden.
2. Die Daten von S1 bis S1 + maximal 2047 Worte und von S2 bis S2 + maximal 2047 Worte dürfen sich nicht überlappen.

Operanden-Spezifikationen

Bereich	S1	S2
CIO-Bereich	CIO 0000 bis CIO 6143	
Arbeitsbereich nicht remanent	W000 bis W511	
Arbeitsbereich remanent	H000 bis H511	
Systembereich	A000 bis A447 A448 bis A959	
Zeitgeberbereich	T0000 bis T4095	
Zählerbereich	C0000 bis C4095	
DM-Bereich	D00000 bis D32767	
EM-Bereich ohne Bank	E00000 bis E32767	
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)	
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)	
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)	
Konstanten	---	
Datenregister	---	
Indexregister	---	
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15	

Beschreibung

Mit Zeichenketten-Vergleichsbefehlen werden die durch S1 und S2 spezifizierten Textzeichenketten miteinander verglichen. Wenn das Ergebnis des Vergleichs wahr ist, wird im Kontaktplan eine EIN-Ausführungsbedingung erstellt. Die maximale Anzahl an Zeichen für S1 bzw. S2 beträgt jeweils 4095 (0FFF hex).

Zeichenketten-Vergleichsbefehle werden durch die nachfolgend aufgeführten 18 verschiedenen AWL-Kürzel ausgedrückt. (LD, AND und OR werden im Kontaktplan nicht angegeben.)

LD=$, AND=$, OR=$
LD<>$, AND<>$, OR<>$
LD<$, AND<$, OR<$
LD<=$, AND<=$, OR<=$
LD>$, AND>$, OR>$
LD>=$, AND>=$, OR>=$

Die folgende Tabelle enthält Details zu diesen Befehlen.

AWL-Kürzel (einschließlich Funktionscode)	Name	Funktion
LD=$(670)	LOAD STRING EQUALS	Wahr, wenn die S1-Textzeichenkette gleich der S2-Textzeichenkette ist.
AND=$(670)	AND STRING EQUALS	
OR=$(670)	OR STRING EQUALS	
LD<>$(671)	LOAD STRING NOT EQUAL	Wahr, wenn die S1-Textzeichenkette nicht gleich der S2-Textzeichenkette ist.
AND<>$(671)	AND STRING NOT EQUAL	
OR<>$(671)	OR STRING NOT EQUAL	
LD<$(672)	LOAD STRING LESS THAN	Wahr, wenn die S1-Textzeichenkette kleiner als die S2-Textzeichenkette ist.
AND<$(672)	AND STRING LESS THAN	
OR<$(672)	OR STRING LESS THAN	
LD<=$(673)	LOAD STRING LESS THAN OR EQUALS	Wahr, wenn die S1-Textzeichenkette kleiner oder gleich der S2-Textzeichenkette ist.
AND<=$(673)	AND STRING LESS THAN OR EQUALS	
OR<=$(673)	OR STRING LESS THAN OR EQUALS	
LD>$(674)	LOAD STRING GREATER THAN	Wahr, wenn die S1-Textzeichenkette größer als die S2-Textzeichenkette ist.
AND>$(674)	AND STRING GREATER THAN	
OR>$(674)	OR STRING GREATER THAN	
LD>=$(675)	LOAD STRING GREATER THAN OR EQUALS	Wahr, wenn die S1-Textzeichenkette größer oder gleich der S2-Textzeichenkette ist.
AND>=$(675)	AND STRING GREATER THAN OR EQUALS	
OR>=$(675)	OR STRING GREATER THAN OR EQUALS	

Vergleichsverfahren

Es gibt folgende Vergleichsverfahren:

Das erste Zeichen (Byte) jeder Textzeichenkette wird mit seinem Gegenstück aus der anderen Zeichenkette als ASCII-Code verglichen. Wenn die beiden ASCII-Codes nicht gleich sind, wird das Größer-als-/Kleiner-als-Verhältnis zum Größer-als-/Kleiner-als-Verhältnis der zwei Textzeichenketten. Wenn die beiden ASCII-Codes gleich sind, werden die nächsten Zeichen miteinander verglichen. Wenn diese beiden ASCII-Codes nicht gleich sind, wird dieses Größer-als-/Kleiner-als-Verhältnis zum Größer-als-/Kleiner-als-Verhältnis der zwei Textzeichenketten selber.

Auf diese Weise werden die beiden Textzeichenketten der Reihenfolge nach Zeichen für Zeichen miteinander verglichen. Wenn alle Zeichen, einschließlich des NULL-Codes, gleich sind, haben die beiden Textzeichenketten ein Gleich-Verhältnis.

Wenn die beiden Textzeichenketten unterschiedlich lang sind, dann werden NULL-Zeichen (00 hex) zur kürzeren der beiden Textzeichenketten hinzugefügt, um die Differenz auszugleichen, und die Vergleiche finden auf dieser Basis statt.

Vergleichsbeispiele

AD (414400 hex) und BC (424300 hex):
AD < BC, da am Anfang der Textzeichenketten 41 (hex) kleiner ist als 42 (hex).

ADC (41444300 hex) und B (4200 hex):
ADC < B, da am Anfang der Textzeichenketten 41 (hex) kleiner ist als 42 (hex).

ABC (41424300 hex) und ABD (41424400 hex):
ABC < ABD, da die Werte 41 und 42 (hex) am Anfang der Textzeichenketten übereinstimmen und somit das Ergebnis durch die 43 bestimmt wird, die kleiner als 44 ist.

ABC (41424300 hex) und AB (414200 hex):
ABC > AB, da die Werte 41 und 42 (hex) am Anfang der Textzeichenketten übereinstimmen und somit das Ergebnis durch die 43 bestimmt wird, die größer als 00 ist.

AB (414200 hex) und AB (414200 hex):
AB = AB, da alle Zeichen (41, 42 und 00) übereinstimmen.

Fahren Sie mit der Programmierung der Befehle nacheinander fort, wobei LD, AND und OR auf die selbe Weise behandelt werden. LD- und OR-Befehle können direkt mit der Stromschiene verbunden werden, was nicht für AND-Befehle gilt.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn von S1 oder S2 mehr als 4095 Zeichen spezifiziert werden. In allen anderen Fällen AUS.
Größer-als-Flag	>	EIN, wenn die Vergleichsergebnisse in S1 größer als S2 sind. In allen anderen Fällen AUS.
Größer-oder-gleich-Flag	>=	EIN, wenn die Vergleichsergebnisse in S1 größer oder gleich S2 sind. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn die Vergleichsergebnisse in S1 gleich S2 sind. In allen anderen Fällen AUS.
Ungleich-Flag	<>	EIN, wenn die Vergleichsergebnisse in S1 nicht gleich S2 sind. In allen anderen Fällen AUS.
Kleiner-als-Flag	<	EIN, wenn die Vergleichsergebnisse in S1 kleiner als S2 sind. In allen anderen Fällen AUS.
Kleiner-oder-gleich-Flag	<=	EIN, wenn die Vergleichsergebnisse in S1 kleiner oder gleich S2 sind. In allen anderen Fällen AUS.

Hinweis Zeichenketten-Vergleichsbefehle werden zur Neuanordnung der Reihenfolge von Textzeichenketten gemäß der Reihenfolge der ASCII-Zeichen verwendet. Die aufsteigende Reihenfolge der ASCII-Zeichen entspricht der Reihenfolge des Alphabets von A bis Z, sodass Textzeichenketten in alphabetischer Reihenfolge angeordnet werden können.

Vorsichtsmaßnahmen

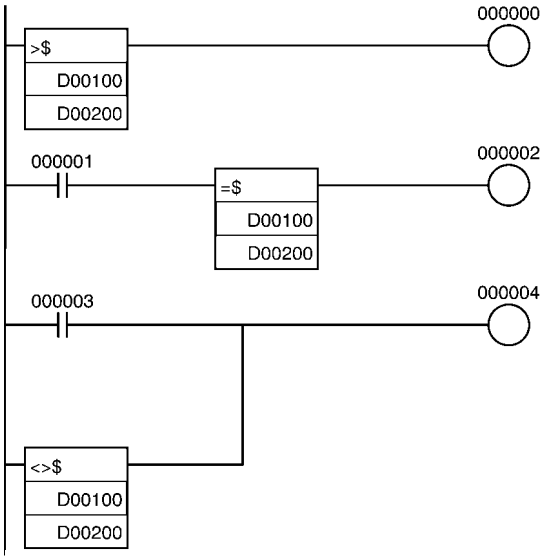
Platzieren Sie hinter diesen Befehlen einen auf der rechten Seite zu programmierenden Befehl. Zeichenketten-Vergleichsbefehle dürfen nicht auf der rechten Seite des Kontaktplans verwendet werden.

Diese Befehle können nicht im letzten Netzwerk eines logischen Blocks verwendet werden.

Die maximale Anzahl an Zeichen, die verglichen werden können, beträgt 4095 (0FFF hex). Wenn diese Anzahl überschritten wird (d. h., wenn sich vor dem 4096. Zeichen kein NULL-Code befindet), tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt. Wenn das geschieht, wird an den nächsten Befehl eine AUS-Ausführungsbedingung ausgegeben.

Beispiel

In diesem Beispiel werden Zeichenketten-Vergleichsbefehle zum Datenvergleich verwendet.



Adresse	AWL	Operand
000000	LD > $	--- D00100 D00200
000001	OUT	000000
000002	LD	000001
000003	AND=$	--- D00100 D00200
000004	OUT	000002
000005	LD	000003
000006	OR <> $	--- D00100 D00200
000007	OUT	000004

> $	= $	<>$
D00100	D00100	D00100
D00200	D00200	D00200
ON	OFF	ON

Textzeichenkette ABCD

D00100	41	42
D00101	44	43
D00102	00	00

Textzeichenkette ABC

D00200	41	42
D00201	43	00

Textzeichenkette ABC

D00100	41	42
D00101	43	00

Textzeichenkette ABC

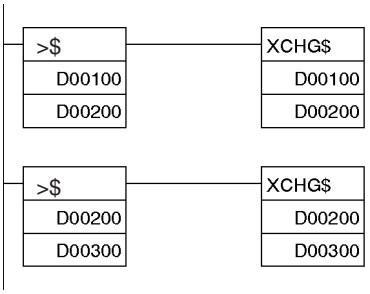
D00200	41	42
D00201	43	00

OFF	ON	OFF

In diesem Beispiel werden drei Textzeichenketten in alphabetischer Reihenfolge angeordnet. Die ursprüngliche Reihenfolge ist wie folgt:

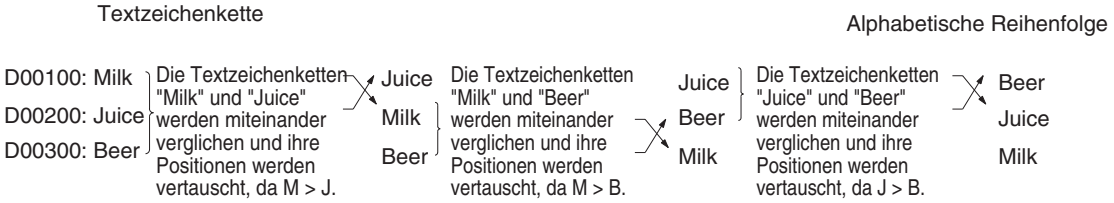
D00100: Milk
D00200: Juice
D00300: Beer

Nach der alphabetischen Neuordnung ändert sich die Reihenfolge wie folgt: Beer, Juice, Milk.



Zwei mit D00100 und D00200 beginnende Textzeichenketten werden hinsichtlich der aufsteigenden ASCII-Reihenfolge miteinander verglichen. Wenn die mit D00100 beginnende Textzeichenkette in der ASCII-Reihenfolge höher als die mit D00200 beginnende Textzeichenkette ist, dann wird die Position der zwei Textzeichenketten vertauscht.

Zwei mit D00200 und D00300 beginnende Textzeichenketten werden hinsichtlich der aufsteigenden ASCII-Reihenfolge miteinander verglichen. Wenn die mit D00200 beginnende Textzeichenkette in der ASCII-Reihenfolge höher als die mit D00300 beginnende Textzeichenkette ist, dann wird die Position der zwei Textzeichenketten vertauscht.



Auf diese Weise können drei Textzeichenketten in alphabetischer Reihenfolge angeordnet werden.

3-34 Task-Steuerbefehle

In diesem Abschnitt werden die zum Steuern von Tasks verwendeten Befehle beschrieben.

Befehl	AWL	Funktionscode	Seite
TASK ON	TKON	820	1192
TASK OFF	TKOF	821	1196

3-34-1 TASK ON: TKON(820)

Funktion

Macht die spezifizierte Task ausführbar. Zudem erhält eine Interrupt-Task die Funktion einer zyklischen Zusatztask. (Zyklische Zusatztasks werden nur von den CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M unterstützt.)

Kontaktplansymbol



N: Task-Nummer

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TKON(820)
	Ausführung einmal bei steigender Flanke	@TKON(820)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	Nicht zulässig

Operanden

N: Task-Nummer
Der gültige Bereich für N hängt von der Art der spezifizierten Task ab.

- Zyklische Tasks:
  N muss eine Konstante zwischen 0 und 31 (dezimal) sein. (Die Werte 0 bis 31 spezifizieren die zyklischen Tasks 0 bis 31.)
- Zyklische Zusatztasks (nur CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen):
  N muss eine Konstante zwischen 8000 und 8255 (dezimal) sein. (Die Werte 8000 bis 8255 spezifizieren die zyklischen Zusatztasks 0 bis 255.)

Operanden-Spezifikationen

Bereich	N
CIO-Bereich	---
Arbeitsbereich nicht remanent	---
Arbeitsbereich remanent	---
Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	00 bis 31 oder 8000 bis 8255 (dezimal)
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	---

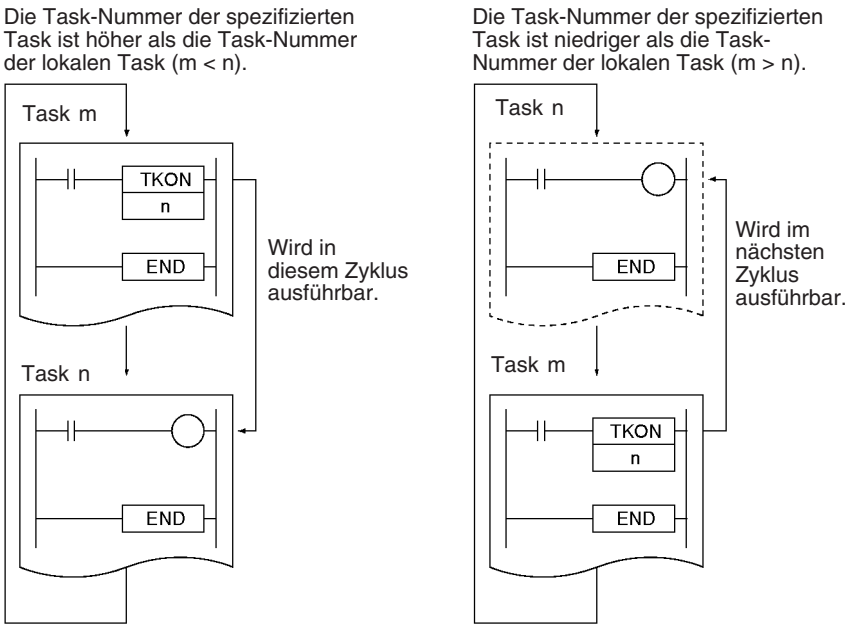
Beschreibung

Der Befehl TKON(820) versetzt spezifizierte zyklische Task oder zyklische Zusatztask in den ausführbaren Status. Wenn N den Wert 0 bis 31 (Spezifikation einer zyklischen Task) hat, wird das entsprechende Task-Flag (TK00 bis TK31) gleichzeitig auf EIN gesetzt.

Dieser Befehl kann nur in einer normalen zyklischen Task oder einer zyklischen Zusatztask ausgeführt werden. Ein Fehler tritt auf, wenn der Versuch zur Ausführung dieses Befehls in einer Interrupt-Task erfolgt.

Die in TKON(820) spezifizierte zyklische Task oder zyklische Zusatztask ist auch in folgenden Zyklen ausführbar, so lange die Task nicht durch TKOF(821) in den Standby-Status versetzt wird.

Jede Task kann aus einer beliebige zyklische Task heraus ausführbar gemacht werden, obwohl die spezifizierte Task bis zum nächsten Zyklus nicht ausgeführt wird, wenn deren Task-Nummer niedriger als die der lokalen Task ist. Die Task wird im selben Zyklus ausgeführt, wenn ihre Task-Nummer höher als die der lokalen Task ist.



TKON(820) wird als NOP(000) verarbeitet, wenn die spezifizierte Task bereits ausführbar oder die lokale Task spezifiziert ist.

Eine Task im ausführbaren Status kann mit Hilfe von TKOF(821), mit CX-Programmer oder einem FINS-Befehl in den Standby-Status versetzt werden.

Die Begriffe „ausführbar“ und „ausgeführt“ sind nicht austauschbar. Ausführbare Tasks werden während der zyklischen Programmausführung in der Reihenfolge ihrer Task-Nummern ausgeführt. Eine ausführbare Task wird nicht ausgeführt, wenn sie in den Standby-Status versetzt wird, bevor die Programmausführung ihre Task-Nummer erreicht hat.

Hinweis

1. Die Registerkarte *General Properties* (Allgemeine Eigenschaften) in CX-Programmer hat für jede Task eine Einstellung (das Kontrollkästchen *Operation start* (Betriebsstart)), die festlegt, ob die zyklische Task beim Systemstart ausführbar ist. Wenn das Kontrollkästchen *Operation start* (Betriebsstart) aktiviert wurde, wird die entsprechende zyklische Task bei Betriebsstart der SPS automatisch in den ausführbaren Status versetzt. Alle anderen zyklischen Tasks befinden sich im nicht ausführbaren Status. (Wenn die Funktion zum Löschen des gesamten Speichers über die Programmierkonsole ausgeführt wird, wird die zyklische Task 0 dennoch automatisch ausführbar gemacht.)
2. Wenn sich eine Task im nicht ausführbaren Status befindet, kann TKON(820) ausgeführt werden, um diese Task in den ausführbaren Status zu versetzen. Ebenso kann eine zyklische Task im ausführbaren Zustand mit dem Befehl TKOF(821) in den nicht ausführbaren Zustand versetzt werden.
3. Zyklische Tasks und zyklische Zusatztasks, die ausführbar gemacht wurden, werden in diesem Zyklus in der Reihenfolge der Task-Nummern in den ausführbaren Status versetzt. Demnach wird eine Task nicht ausgeführt, wenn sie in einen Standby-Status versetzt wird, bevor die Programmverarbeitung im Zyklus diese Task erreicht, da alle Tasks in der Reihenfolge der Task-Nummern ausgeführt werden.

Flags

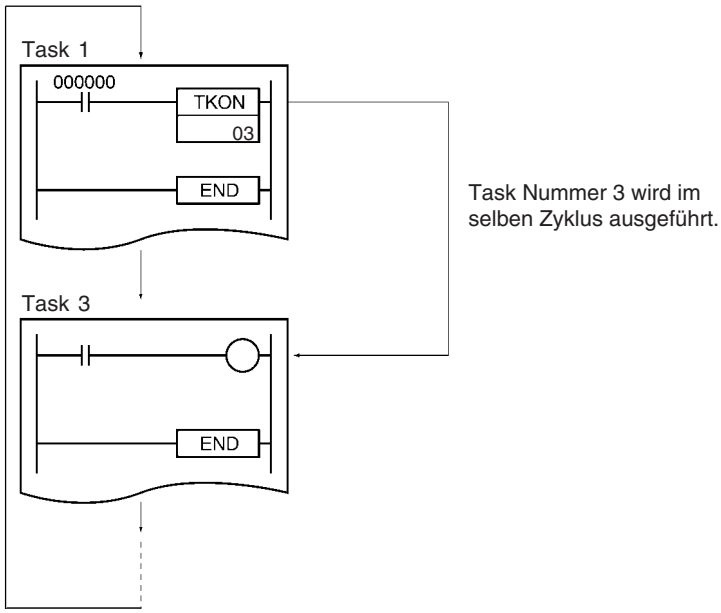
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N keine Konstante zwischen 00 und 31 bzw. zwischen 8000 und 8255 ist (nur CS1-H, CJ1-H und CJ1M CPU-Baugruppen). EIN, wenn die mit N spezifizierte Task nicht vorhanden ist. EIN, wenn TKON(820) in einer Interrupt-Task ausgeführt wird. In allen anderen Fällen AUS.

Name	Adressen	Funktion
Task-Flags	TK00 bis TK31	Diese Flags werden auf EIN gesetzt, wenn die zugehörige zyklische Task ausführbar ist. Die Flags sind auf AUS gesetzt, wenn die Task nicht ausführbar ist oder sich im Standby-Status befindet. TK00 bis TK31 entsprechen den zyklischen Tasks mit den Nummern 00 bis 31.

Beispiele

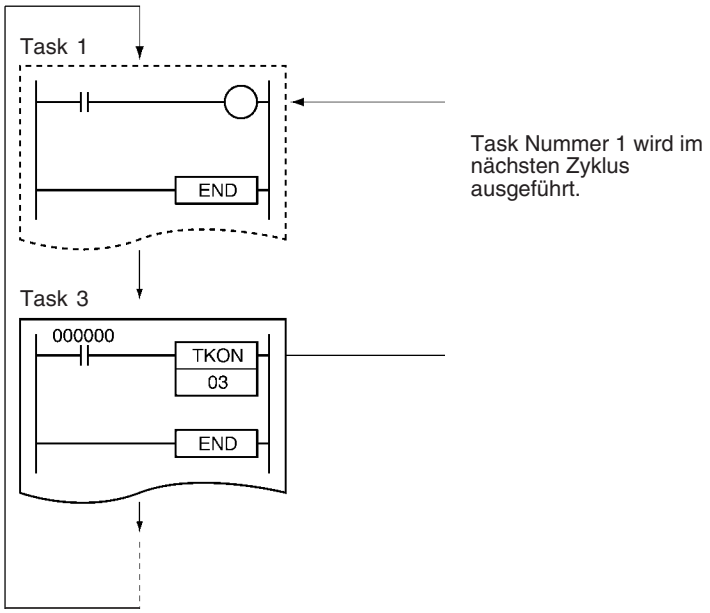
Spezifikation einer späteren Task

Wenn CIO 000000 im folgenden Beispiel EIN ist, wird Task Nummer 3 in Task Nummer 1 ausführbar gemacht. Task Nummer 3 wird im selben Zyklus ausgeführt, wenn die Programmausführung die Task Nummer 3 erreicht.



Spezifikation einer früheren Task

Wenn CIO 000000 im folgenden Beispiel EIN ist, wird Task Nummer 1 in Task Nummer 3 ausführbar gemacht. Task Nummer 1 wird im nächsten Zyklus ausgeführt, wenn die Programmausführung die Task Nummer 1 erreicht.

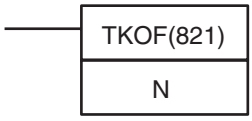


3-34-2 TASK OFF: TKOF(821)

Funktion

Versetzt die spezifizierte zyklische Task oder zyklische Zusatztask in den Standby-Status, d.h., die Ausführung der Tasks wird deaktiviert. (Zyklische Zusatztasks werden nur von den CPU-Baugruppen der SPS-Systeme CS1-H, CJ1-H und CJ1M unterstützt.)

Kontaktplansymbol



N: Task-Nummer

Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	TKOF(821)
	Ausführung einmal bei steigender Flanke	@TKOF(821)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	Nicht zulässig

Operanden

N: Task-Nummer
Der gültige Bereich für N hängt von der Art der spezifizierten Task ab.

- Zyklische Tasks:
  N muss eine Konstante zwischen 0 und 31 (dezimal) sein. (Die Werte 0 bis 31 spezifizieren die zyklischen Tasks 0 bis 31.)
- Zyklische Zusatztasks (nur CS1-H, CJ1-H, CJ1M und CS1D CPU-Baugruppen):
  N muss eine Konstante zwischen 8000 und 8255 (dezimal) sein. (Die Werte 8000 bis 8255 spezifizieren die zyklischen Zusatztasks 0 bis 255.)

Operanden-Spezifikationen

Bereich	N
CIO-Bereich	---
Arbeitsbereich nicht remanent	---
Arbeitsbereich remanent	---
Systembereich	---
Zeitgeberbereich	---
Zählerbereich	---
DM-Bereich	---
EM-Bereich ohne Bank	---
EM-Bereich mit Bank	---
Indirekte DM/EM-Binäradressen	---
Indirekte DM/EM-Adressen in BCD	---
Konstanten	00 bis 31 oder 8000 bis 8255 (dezimal)
Datenregister	---
Indexregister	---
Indirektes Adressieren mittels Indexregistern	---

Beschreibung

TKOF(821) versetzt die spezifizierte zyklische Task oder zyklische Zusatztask in den Standby-Status und setzt das entsprechenden Task-Flag (TK00 bis TK31) auf AUS.

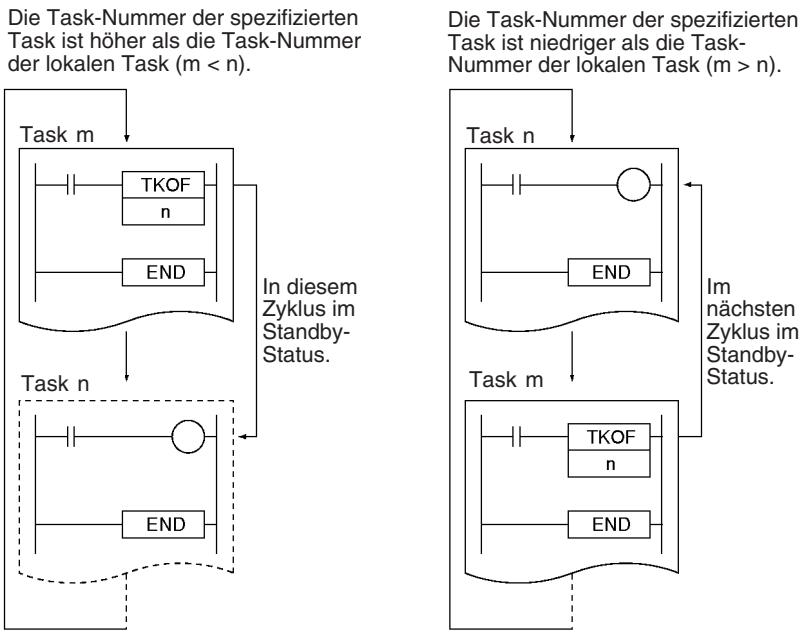
Die in TKOF(821) spezifizierte Task hat in den folgenden Zyklen ebenfalls den Standby-Status, so lange sie nicht durch TKON(820), ein Peripheriegerät mit CX-Programmer oder einen FINS-Befehl in den ausführbaren Status versetzt wird.

Eine Task kann aus jeder regulären Task heraus in den Standby-Status versetzt werden. Allerdings wird die spezifizierte Task bis zum nächsten Zyklus nicht in den Standby-Status versetzt, wenn ihre Task-Nummer niedriger als die Task-Nummer der lokalen Task (die bereits ausgeführt worden wäre) ist. Die Task wird im selben Zyklus in den Standby-Status versetzt, wenn ihre Task-Nummer höher als die der lokalen Task ist.

Wenn die lokale Task in TKOF(821) spezifiziert wird, wird die Task sofort in den Standby-Status versetzt und die nachfolgenden Befehle in der Task werden nicht ausgeführt.

Hinweis

1. Die Registerkarte *General Properties* (Allgemeine Eigenschaften) in CX-Programmer hat für jede Task eine Einstellung (das Kontrollkästchen *Operation start* (Betriebsstart)), die festlegt, ob die zyklische Task beim Systemstart ausführbar ist. Wenn das Kontrollkästchen *Operation start* (Betriebsstart) aktiviert wurde, wird die entsprechende zyklische Task bei Betriebsstart der SPS automatisch in den ausführbaren Status versetzt. Alle anderen zyklischen Tasks befinden sich im nicht ausführbaren Status. (Wenn die Funktion zum Löschen des gesamten Speichers über die Programmierkonsole ausgeführt wird, wird die zyklische Task 0 dennoch automatisch ausführbar gemacht.)
2. Wenn sich eine Task im nicht ausführbaren Status befindet, kann TKON(820) ausgeführt werden, um diese Task in den ausführbaren Status zu versetzen. Ebenso kann eine zyklische Task im ausführbaren Zustand mit dem Befehl TKOF(821) in den nicht ausführbaren Zustand versetzt werden.
3. Zyklische Tasks oder zyklische Zusatztasks im ausführbaren Zustand können durch den Befehl TKOF(821) in den Standby-Status versetzt werden.



Eine auf Ausführbarkeit beim Systemstart eingestellte reguläre Task wird bei Betriebsbeginn der SPS automatisch in den ausführbaren Zustand versetzt. Alle anderen regulären Tasks befinden sich im nicht ausführbaren Status.

Eine im ausführbaren Status befindliche Task kann durch den Befehl TKOF(821), ein Peripheriegerät mit CX-Programmer oder durch einen FINS-Befehl in den Standby-Status versetzt werden.

Die Begriffe „ausführbar“ und „ausgeführt“ sind nicht austauschbar. Ausführbare Tasks werden während der zyklischen Programmausführung in der Reihenfolge ihrer Task-Nummern ausgeführt. Eine ausführbare Task wird nicht ausgeführt, wenn sie in den Standby-Status versetzt wird, bevor die Programmausführung ihre Task-Nummer erreicht hat.

Im Unterschied zu TKON(820) kann dieser Befehl sowohl in Interrupt-Tasks als auch in zyklischen Tasks platziert werden.

Flags

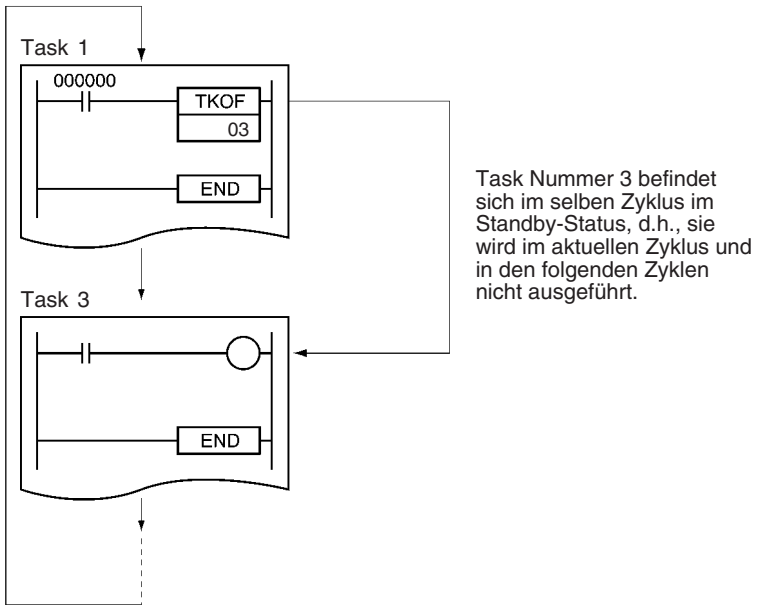
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn N keine Konstante zwischen 00 und 31 bzw. zwischen 8000 und 8255 ist (nur CS1-H, CJ1-H und CJ1M CPU-Baugruppen). EIN, wenn die mit N spezifizierte Task nicht vorhanden ist. EIN, wenn TKOF(821) in einer Interrupt-Task ausgeführt wird. In allen anderen Fällen AUS.

Name	Adressen	Funktion
Task-Flags	TK00 bis TK31	Diese Flags werden auf EIN gesetzt, wenn die zugehörige zyklische Task ausführbar ist. Die Flags sind auf AUS gesetzt, wenn die Task ist nicht ausführbar ist oder sich im Standby-Status befindet. TK00 bis TK31 entsprechen den zyklischen Tasks mit den Nummern 00 bis 31.

Beispiele

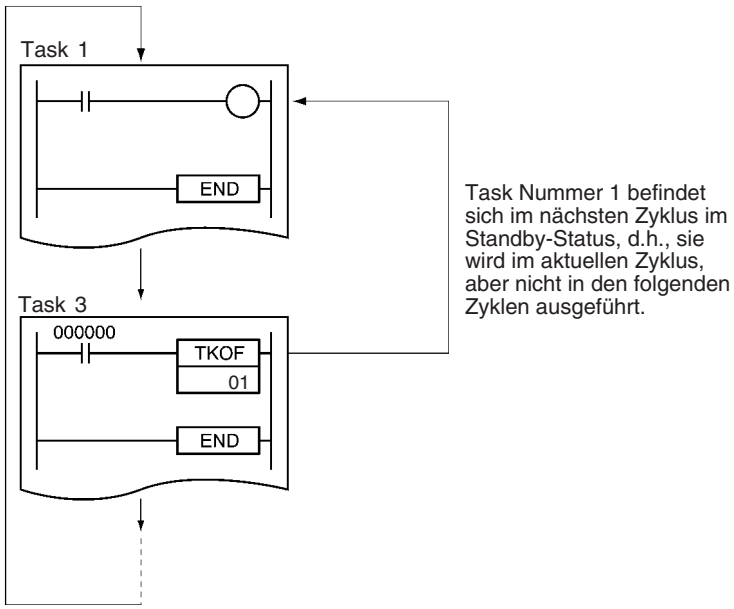
Spezifikation einer späteren Task

Wenn CIO 000000 im folgenden Beispiel EIN ist, wird Task Nummer 3 in Task Nummer 1 in den Standby-Status versetzt. Task Nummer 3 wird in diesem Zyklus nicht ausgeführt, wenn die Programmausführung Task Nummer 3 erreicht.



Spezifikation einer früheren Task

Wenn CIO 000000 im folgenden Beispiel EIN ist, wird Task Nummer 1 in Task Nummer 3 in den Standby-Status versetzt. Task Nummer 1 wird im nächsten Zyklus nicht ausgeführt, wenn die Programmausführung Task Nummer 1 erreicht.



3-35 Befehle zur Modellkonvertierung (nur CPU-Baugruppen ab Ver. 3.0)

In diesem Abschnitt werden Befehle beschrieben, die beim Wechsel von SPS-Modellen verwendet werden.

Befehl	AWL	Funktionscode	Seite
BLOCK TRANSFER	XFERC	565	1201
SINGLE WORD DISTRIBUTE	DISTC	566	1203
DATA COLLECT	COLLC	567	1206
MOVE BIT	MOVBC	568	1211
BIT COUNTER	BCNTC	621	1212

Modell-Konvertierungsbefehle bieten dieselbe Funktionalität wie andere Befehle, verwenden aber, wie auch die Befehle für die C-Serie, BCD-Daten für die Operanden. (Die CJ/CS-Serie verwendet Binärdaten für die Operanden.) Wie in der obigen Tabelle ersichtlich, gibt es fünf Modell-Konvertierungsbefehle, von denen alle am Ende des AWL-Kürzels der äquivalenten Funktion für binäre Operandendaten ein C angefügt tragen.

Die Modell-Konvertierungsbefehle erlauben eine Konvertierung von Programmen der C-Serie in solche für die CS/CJ-Serie ohne Änderung der Operandendaten für diese Befehle.

Beim Konvertieren von Programmen der C-Serie in solche für die CS/CJ-Serie mit CX-Programmer Version 5.0 oder höher (siehe Hinweis) werden diese Befehle bei der Konvertierung automatisch verwendet (z.B. wird XFER in XFERC konvertiert), wodurch eine manuelle Korrektur der Operandendaten entfällt.

Beim Konvertieren von Programmen der C-Serie in solche für die CS/CJ-Serie mit CX-Programmer Version 4.0 oder niedriger (siehe Hinweis) wird jeder Operand, für den eine Konstante spezifiziert ist, von BCD in binär konvertiert, jedoch müssen alle Operandendaten, für die eine Wortadresse spezifiziert ist, manuell korrigiert werden.

Hinweis Die Konvertierung wird durch Spezifizieren der CS/CJ-Serie als „Gerätetyp“ im Dialogfeld "SPS ändern" erreicht.

Unterschiede zu Befehlen der C-Serie

Die „C-Serie“ beinhaltet die SPS-Systeme C200H, C1000H, C2000H, C200HS, C2000HX/HG/HE(-Z), CQM1, CQM1H, CPM1/CPM1A, CPM2C und SRM1.

Name	Modell-Konvertierungsbefehle (ab Baugruppenversion 3.0)	Entsprechender Befehl der C-Serie	Unterschiede zu Befehlen der C-Serie		Bei der Konvertierung des Gerätetyps zu CS/CJ mit CX-Programmer Ver. 4.0 oder früher	Bei der Konvertierung des Gerätetyps zu CS/CJ mit CX-Programmer ab Ver. 5.0
	AWL (Funktionscode)	AWL (Funktionscode)	C200H, C1000H oder C2000H	C200HS, C2000HX/HG/HE(-Z), CQM1, CQM1H, CPM1/CPM1A, CPM2C oder SRM1		
BLOCK TRANSFER	XFERC(565)	XFER(70)	Identisch	Identisch	Konvertiert in XFER. Wenn eine Wortadresse für den ersten Operanden spezifiziert ist (Anzahl der zu übertragenden Worte), muss diese im Programm manuell in Binärdaten korrigiert werden.	XFER wird in XFERC konvertiert. Operanden müssen nicht korrigiert werden.
SINGLE WORD DISTRIBUTE	DISTC(566)	DIST(80)	Bietet neben der Datenverteilungsfunktion eine zuvor nicht unterstützte Funktion zum Schreiben von Daten in einen Stapel.	Identisch(Verteilungsfunktion und Funktion zum Schreiben von Daten in einen Stapel)	Konvertierung in DIST. Wenn eine Wortadresse für den dritten Operanden spezifiziert ist (Offset-Daten), muss diese im Programm manuell in Binärdaten korrigiert werden.	DIST wird in DISTC konvertiert. Operanden müssen nicht korrigiert werden.
DATA COLLECT	COLLC(567)	COLL(81)	Bietet neben der Datenerfassungsfunktion eine zuvor nicht unterstützte Stapel-Lesefunktion.	Identisch (Datenerfassungsfunktion und Stapel-Lesefunktion)	Konvertiert in COLL. Wenn eine Wortadresse für den zweiten Operanden spezifiziert ist (Offset-Daten), muss diese im Programm manuell in Binärdaten korrigiert werden.	COLL wird in COLLC konvertiert. Operanden müssen nicht korrigiert werden.
MOVE BIT	MOVBC(568)	MOVB(82)	Identisch	Identisch	Konvertiert in MOVB. Wenn eine Wortadresse für den zweiten Operanden (Steuerdaten) spezifiziert ist, muss diese im Programm manuell in Binärdaten korrigiert werden.	MOVB wird in MOVBC konvertiert. Operanden müssen nicht korrigiert werden.
BIT COUNTER	BCNTC(621)	BCNT(67)	Identisch	Identisch	Konvertiert in BCNT. Wenn eine Wortadresse für den ersten Operanden (Anzahl der zu zählenden Worte) spezifiziert wird, muss diese im Programm manuell in Binärdaten korrigiert werden.	BCNT wird in BCNTC konvertiert. Operanden müssen nicht korrigiert werden.

Hinweis Die Funktion der Bedingungs-Flags unterscheidet sich wie folgt. Detaillierte Informationen hierzu finden Sie in der Beschreibung der Bedingungs-Flags für die einzelnen Befehle.

- Die Funktion der Bedingungs-Flags unterscheidet sich bei allen Befehle, wenn der Inhalt der zur indirekten Adressierung verwendeten Worte im DM-Bereich nicht in BCD (*BCD) vorliegt oder der DM-Adressenbereich überschritten wird.
- Bei DISTC(566) unterscheidet sich die Funktion der Bedingungs-Flags im Vergleich mit der Funktion bei den SPS C200H, C1000H und C2000H für die Stapel-Schreibfunktion.
- Bei COLLC(567) unterscheidet sich die Funktion der Bedingungs-Flags im Vergleich mit der Funktion bei den SPS C200H, C1000H und C2000H für die Stapel-Lesefunktion.

Unterschiede zu bisherigen Befehlen der CS/CJ-Serie

Name	Modell-Konvertierungsbefehle (ab Baugruppenversion 3.0)	Entsprechender Befehl der C-Serie	Unterschiede zu bisherigen Befehlen der CS/CJ-Serie
	AWL (Funktionscode)	AWL(Funktionscode)	
BLOCK TRANSFER	XFERC(565)	XFER(70)	Der Datentyp für den ersten Operand (Anzahl der zu übertragenden Worte) ist BCD (0000 bis 9999) anstelle von binär (0000 bis FFFF hex).
SINGLE WORD DISTRIBUTE	DISTC(566)	DIST(80)	Zusätzlich zu der Datenverteilungsfunktion wird eine Stapel-Schreibfunktion unterstützt. Der Datentyp für den dritten Operand (Offset-Daten) ist BCD (Datenverteilung: 0000 bis 7999, Schreiben in Stapel: 0000 bis 9999) anstelle von binär (0000 bis FFFF hex).

Name	Modell-Konvertierungsbefehle (ab Baugruppenversion 3.0)	Entsprechender Befehl der C-Serie	Unterschiede zu bisherigen Befehlen der CS/CJ-Serie
	AWL (Funktionscode)	AWL(Funktionscode)	
DATA COLLECT	COLLC(567)	COLL(81)	Zusätzlich zu der Datenverteilungsfunktion wird eine Stapel-Lesefunktion unterstützt. Der Datentyp für den zweiten Operand (Offset-Daten) ist BCD (Datenverteilung: 0000 bis 7999, Stapel-Lesefunktion bei FIFO: 9000 bis 9999, Stapel-Lesefunktion bei LIFO: 8000 bis 8999) anstelle von binär (0000 bis FFFF hex).
MOVE BIT	MOVBC(568)	MOVB(82)	Der Datentyp für die Quell- und Zielbit-Spezifikationen im zweiten Operanden (Steuerdaten) ist BCD (00 bis 15) anstelle von binär (00 bis 0F hex).
BIT COUNTER	BCNTC(621)	BCNT(67)	Der Datentyp für den ersten Operanden (Anzahl der zu zählenden Worte) ist BCD (0000 bis 9999) anstelle von binär (0000 bis FFFF hex). Der für den dritten Operanden (Zählergebnisse) gespeicherte Datentyp ist BCD (0000 bis 9999) anstelle von binär (0000 bis FFFF hex).

Hinweis Die Funktion der Bedingungs-Flags unterscheidet sich wie folgt. Detaillierte Informationen hierzu finden Sie in der Beschreibung der Bedingungs-Flags für die einzelnen Befehle.

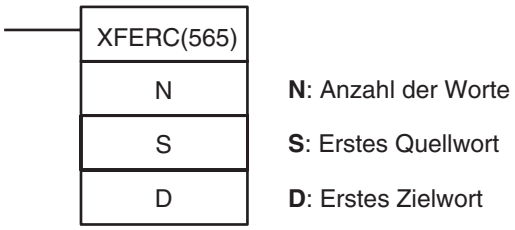
- Das Fehler-Flag wird auf EIN gesetzt, wenn die Daten für die oben aufgeführten Operanden nicht in BCD vorliegen.
- Bei DISTC(566) wurde die Funktion der Bedingungs-Flags für die Stapel-Schreibfunktion hinzugefügt.
- Bei COLLC(567) wurde die Funktion der Bedingungs-Flags für die Stapel-Lesefunktion hinzugefügt.

3-35-1 BLOCK TRANSFER: XFERC(565)

Funktion

Überträgt die spezifizierte Anzahl von aufeinander folgenden Worten.

Kontaktplansymbol



Variationen

Variationen		
	Ausführung in jedem Zyklus bei EIN-Bedingung	XFERC(565)
	Ausführung einmal bei steigender Flanke	@XFERC(565)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogrammbereiche	Schrittprogrammbereiche	Unterprogramme	Interrupt-Tasks
OK	OK	OK	OK

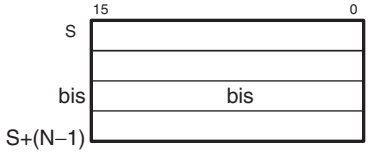
Operanden

N: Anzahl der Worte

Spezifiziert die Anzahl der zu übertragenden Worte. Der mögliche Bereich für N ist 0000 bis 9999 BCD.

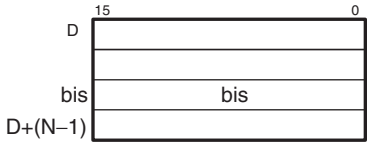
S: Erstes Quellwort

Spezifiziert das erste Quellwort.



D: Erstes Zielwort

Spezifiziert das erste Zielwort.

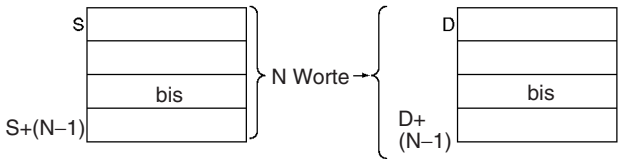


Operanden-Spezifikationen

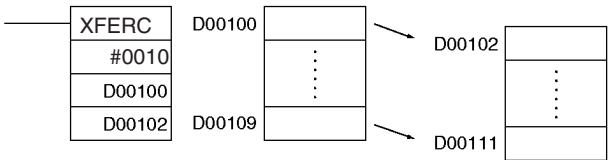
Bereich	N	S	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #9999 (BCD)	---	---
Datenregister	DR0 bis DR15	---	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

XFERC(565) kopiert N Worte mit S (S bis S+(N–1)) beginnend in die N Worte, die mit D (D bis D+(N–1)) beginnen.



Quellworte und Zielworte können sich überlappen, deshalb kann XFERC(565) auch Wortverschiebefunktionen durchführen.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn die Daten in N (die Anzahl Worte) nicht in BCD sind.

Hinweis Bei SPS der C-Serie bewirkt der Befehl BLOCK TRANSFER (XFER), dass das Fehler-Flag auf EIN gesetzt wird, wenn der Inhalt eines indirekt adressierten DM-Worts (*DM) keinen BCD-Wert enthält oder die DM-Bereichsgrenze überschritten wird. XFERC(565) bewirkt in diesen Fällen nicht, dass das Fehler-Flag auf EIN gesetzt wird.

Vorsichtsmaßnahmen

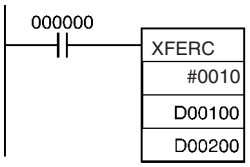
Achten Sie darauf, dass sich die Quellworte (S bis S+N–1) und die Zielworte (D bis D+N–1) nicht über das Ende des Datenbereichs hinaus erstrecken.

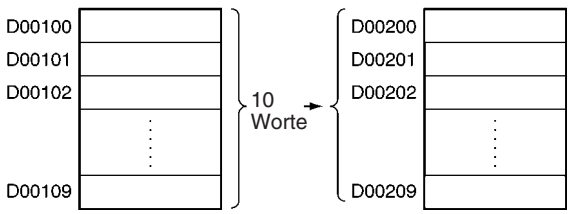
Die Ausführung von XFERC(565) nimmt einige Zeit in Anspruch, wenn eine große Anzahl von Worten übertragen wird. Wenn es während der Ausführung des Befehls zu einer Unterbrechung der Spannungsversorgung kommt, kann die Übertragung von XFERC(565) möglicherweise nicht beendet werden.

Der Inhalt von N muss ein BCD-Wert sein. Wenn N nicht in BCD vorliegt, tritt ein Fehler auf und das Fehler-Flag wird auf EIN gesetzt.

Beispiel

Wenn CIO 000000 im folgenden Beispiel EIN ist, werden die 10 Worte D00100 bis D00109 in D00200 bis D00209 kopiert.



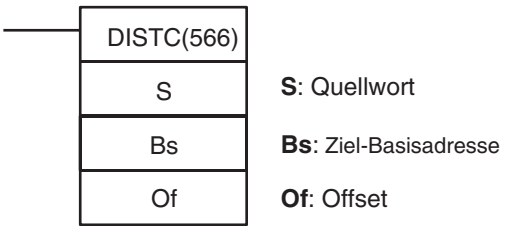


3-35-2 SINGLE WORD DISTRIBUTE: DISTC(566)

Funktion

Überträgt das Quellwort an das durch Addition eines Offsetwerts zur Basisadresse errechnete Zielwort.

Kontaktplansymbol



Variationen

Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	DISTC(566)
	Ausführung einmal bei steigender Flanke	@DISTC(566)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

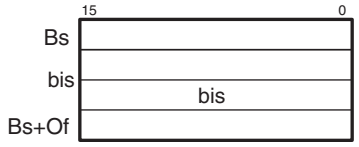
Bs: Ziel-Basisadresse

Spezifiziert die Ziel-Basisadresse. Zu dieser Adresse wird der Offset addiert, um die Adresse des Zielworts zu berechnen.

Of: Offset

- Datenverteilungsfunktion (0000 bis 7999 BCD)

  Dieser Wert wird zu der Basisadresse addiert, um die Adresse des Zielworts zu berechnen. Der Offset kann ein beliebiger Wert von 0000 bis 7999 in BCD sein, allerdings müssen Bs und Bs+Of im selben Datenbereich liegen.



- Stapel-Schreibfunktion (9000 bis 9999 BCD)

  Wenn die höchstwertige Stelle von Of den Wert 9 hat, spezifizieren die niederwertigsten 3 Stellen von Of die Anzahl der Worte im Stapel. Der Offset kann ein beliebiger Wert von 9000 bis 9999 BCD sein.

Operanden-Spezifikationen

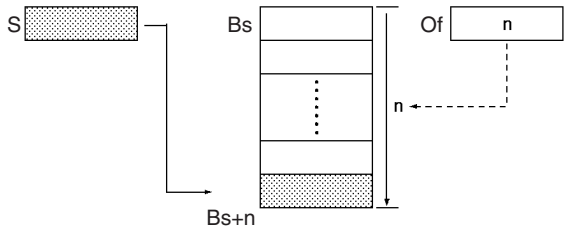
Bereich	S	Bs	Of
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959	A448 bis A959	A000 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0000 bis #FFFF (binär)	---	#0000 bis #7999 bei Verteilung #9000 bis #9999 bei Stapelfunktion

Bereich	S	Bs	Of
Datenregister	DR0 bis DR15	---	DR0 bis DR15
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Datenverteilungsfunktion

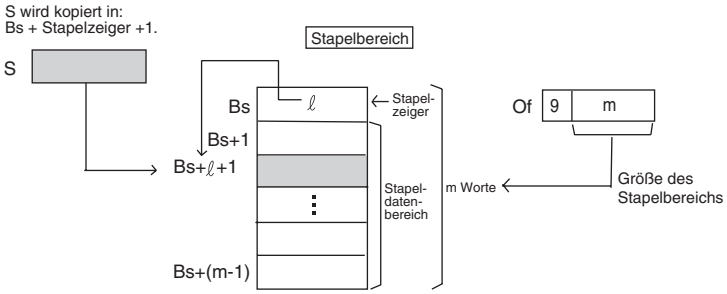
DISTC(566) kopiert S in das Zielwort, dessen Adresse durch Addition von Of zu Bs berechnet wird. Derselbe DISTC(566)-Befehl kann durch Änderung des Werts von Of zur Verteilung des Quellworts an verschiedene Worte im Datenbereich verwendet werden.



Stapel-Schreibfunktion

Wenn die höchstwertige Stelle (Bits 12 bis 15) von Of auf 9 BCD gesetzt ist, steuert der Befehl DISTC(566) einen Stapel von Bs bis Bs+Of-9000. Die Ziel-Basisadresse (Bs) enthält den Stapelzeiger und der Rest der Worte des Stapels enthält die Stapeldaten.

DISTC(566) kopiert S in das Zielwort, dessen Adresse durch Addition des Stapelzeigers (Inhalt von Bs) + 1 zur Adresse Bs berechnet wird. Derselbe DISTC(566)-Befehl kann durch Änderung des Werts von Of zur Verteilung des Quellworts an verschiedene Worte im Datenbereich verwendet werden.



Bei jedem Kopieren des Inhalts von S in ein Wort im Stapelbereich wird der Stapelzeiger in Bs automatisch um 1 inkrementiert.

Hinweis Verwenden Sie COLLC(567) zum Lesen von Stapeldaten aus dem Stapelbereich.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn die Stapel-Schreibfunktion spezifiziert ist, aber der Stapelzeiger in Bs nicht in BCD vorliegt. EIN, wenn die Stapel-Schreibfunktion spezifiziert ist und der Stapelzeiger auf ein Wort weist, das den Stapeldatenbereich überschreitet.
Gleich-Flag	=	EIN, wenn die Quelldaten 0000 sind. In allen anderen Fällen AUS.

Hinweis Bei SPS der C-Serie bewirkt der Befehl SINGLE WORD DISTRIBUTE (DIST), dass das Fehler-Flag auf EIN gesetzt wird, wenn der Inhalt eines indirekt adressierten DM-Worts (*DM) nicht in BCD ist oder die DM-Bereichsgrenze überschritten wird. DISTC(566) bewirkt in diesen Fällen nicht, dass das Fehler-Flag auf EIN gesetzt wird.

Vorsichtsmaßnahmen

Wenn DISTC(566) zur Zuweisung eines Stapelbereichs mit der Stapel-Schreibfunktion ausgeführt wurde, muss in den nachfolgenden DISTC(566)-Befehlen immer dieselbe Stapelbereichslänge spezifiziert werden. Die Funktion ist unzuverlässig, wenn in nachfolgenden DISTC(566)-Befehlen eine andere Stapelbereichgröße spezifiziert wird.

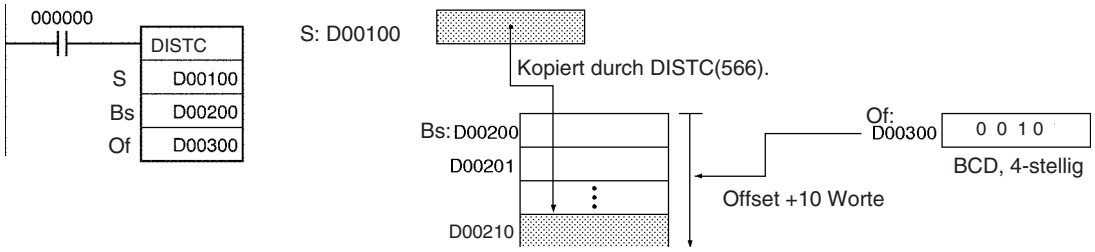
Achten Sie darauf, dass die durch Of spezifizierte Offset- bzw. Stapelgröße bei Addition zu Bs nicht das Ende des Datenbereichs überschreitet.

Beispiele

Datenverteilungsfunktion

Das höchstwertige Byte von D00300 ist 0, sodass DISTC(566) die Datenverteilungsfunktion ausführt.

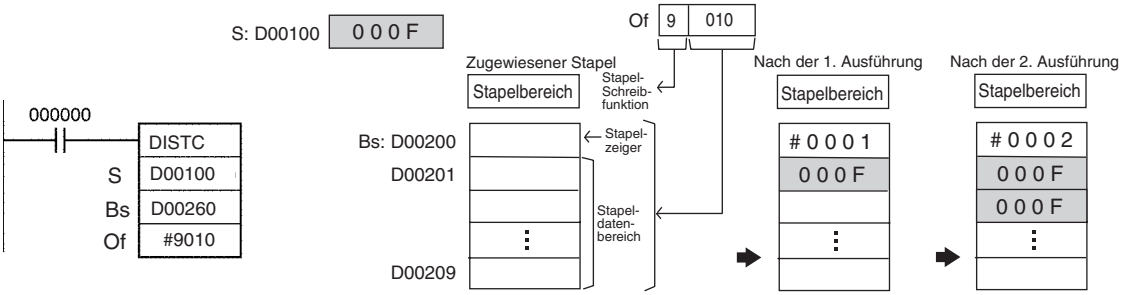
Wenn CIO 000000 im folgenden Beispiel EIN ist, wird der Inhalt von D00100 in D00210 (D00200 + 10) kopiert, falls der Inhalt von D00300 den Wert 0010 BCD hat. Der Inhalt von D00100 kann durch Ändern des Offsets in D00300 in andere Worte kopiert werden.



Stapel-Schreibfunktion

Das höchstwertige Byte von Of ist 9, sodass DISTC(566) die Stapel-Schreibfunktion ausführt.

Wenn CIO 000000 im folgenden Beispiel EIN ist, weist DISTC(566) einen 10-Worte-Stapelbereich (da die niederwertigsten 3 Stellen von Of den Wert #010 haben) zwischen D00200 und D00209 zu. Gleichzeitig wird der Inhalt von D00100 in das durch Addition von D00200 + Stapelzeiger + 1 berechnete Wort kopiert. Abschließend wird der Stapelzeiger um 1 inkrementiert.

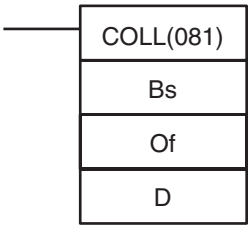


3-35-3 DATA COLLECT: COLLC(567)

Funktion

Überträgt das Quellwort (errechnet durch Addition eines Offsetwerts zur Basisadresse) in das Zielwort.

Kontaktplansymbol



Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	COLLC(567)
	Ausführung einmal bei steigender Flanke	@COLLC(567)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

Bs: Quell-Basisadresse

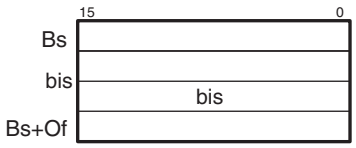
Spezifiziert die Quell-Basisadresse. Zu dieser Adresse wird der Offset addiert, um die Adresse des Quellworts zu berechnen.

Of: Offset

Der Wert von Of bestimmt die Funktion von COLLC(567).

- Datenerfassungsfunktion (Of = 0000 bis 7999 BCD)

  Der Wert von Of wird zur Berechnung der Adresse des Quellworts zur Basisadresse addiert. Der Offset kann ein beliebiger Wert zwischen 0000 und 7999 in BCD sein, allerdings müssen Bs und Bs+Of im selben Datenbereich liegen.



- LIFO-Stapel-Leseoperation (Of = 8000 bis 8999 BCD)
  Wenn die höchstwertige Stelle von Of gleich 8 ist, arbeitet COLLC(567) als LIFO-Stapelbefehl. Der Stapel beginnt bei Bs und hat eine durch die 3 niederwertigsten Stellen von Of definierte Größe.
- FIFO-Stapel-Leseoperation (Of = 9000 bis 9999 BCD)
  Wenn die niederwertige Stelle von Of gleich 9 ist, arbeitet COLLC(567) als FIFO-Stapelbefehl. Der Stapel beginnt bei Bs und hat eine durch die 3 niederwertigsten Stellen von Of definierte Größe.

Operanden-Spezifikationen

Bereich	Bs	Of	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		

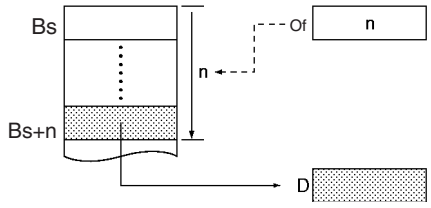
Bereich	Bs	Of	D
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---	#0000 bis #7999 bei Datenerfassung #8000 bis #8999 bei LIFO-Stapel-Lesefunktion #9000 bis #9999 bei FIFO-Stapel-Lesefunktion	---
Datenregister	---	DR0 bis DR15	
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Abhängig vom Wert von Of funktioniert COLLC(567) als Datenerfassungsbefehl, FIFO-Stapelbefehl oder LIFO-Stapelbefehl.

Datenerfassungsfunktion (Of = 0000 bis 7999 BCD)

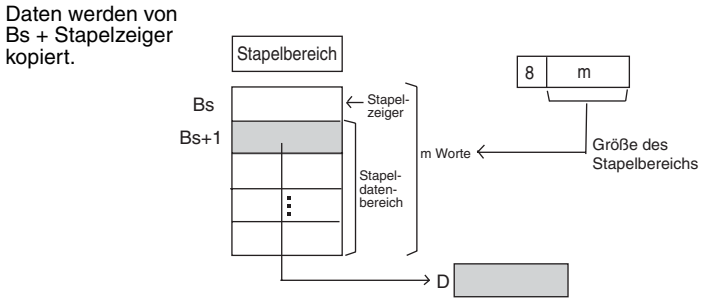
COLLC(567) kopiert das Quellwort (dessen Adresse durch Addition von Of zu Bs berechnet wird) in das Zielwort. Derselbe COLLC(567)-Befehl kann durch Änderung des Werts von Of zur Erfassung von Daten aus verschiedenen Quellworten im Datenbereich verwendet werden.



LIFO-Stapel-Lesefunktion (Of = 8000 bis 8999 BCD)

Wenn die höchstwertige Stelle von Of gleich 8 ist, arbeitet COLLC(567) als LIFO-Stapelbefehl (LIFO steht für Last-In-First-Out, neueste Daten werden zuerst gelesen). In diesem Fall spezifizieren die niederwertigsten 3 Stellen von Of die Größe des Stapels.

COLLC(567) kopiert die zuletzt im Stapel gespeicherten Daten in D. Die Adresse des Quellwort ist Bs + Stapelzeiger (Inhalt von Bs). Nachdem dem Kopieren der Daten wird der Stapelzeiger um 1 dekrementiert.

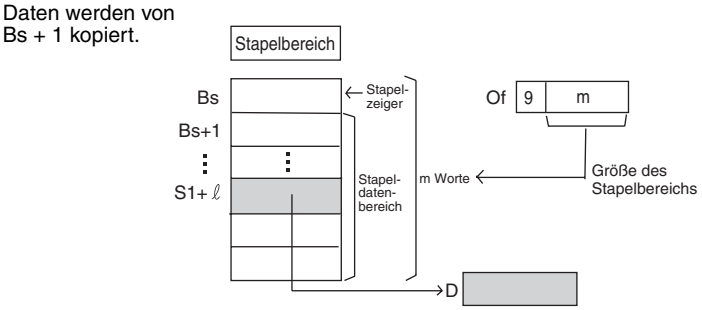


Hinweis Verwenden Sie DISTC(566) zum Schreiben von Stapeldaten in den Stapelbereich.

FIFO-Stapel-Lesefunktion (Of = 9000 bis 9999 BCD)

Wenn die höchstwertige Stelle von Of gleich 9 ist, arbeitet COLLC(567) als FIFO-Stapelbefehl (FIFO steht für First-In-First-Out, älteste Daten werden zuerst gelesen). In diesem Fall spezifizieren die niederwertigsten 3 Stellen von Of die Größe des Stapels.

COLLC(567) kopiert die Daten aus dem ältesten im Stapel gespeicherten Wort in D. Das Quellwort ist Bs + 1. Nachdem die Daten kopiert wurden, wird der Stapelzeiger um 1 dekrementiert.



Hinweis Verwenden Sie DISTC(566) zum Schreiben von Stapeldaten in den Stapelbereich.

Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn die Offset-Daten in Of nicht in BCD sind. EIN, wenn die LIFO- oder FIFO-Stapelfunktion spezifiziert wurde, aber der Stapelzeiger in Bs nicht in BCD ist. EIN, wenn die LIFO- oder FIFO-Stapelfunktion spezifiziert wurde und der Stapelzeiger auf ein Wort weist, das den Stapeldatenbereich überschreitet. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn die Quelldaten 0000 sind. In allen anderen Fällen AUS.

Hinweis Bei SPS der C-Serie bewirkt der Befehl DATA COLLECT (COLL), dass das Fehler-Flag auf EIN gesetzt wird, wenn der Inhalt eines indirekt adressierten DM-Worts (*DM) nicht in BCD vorliegt oder die DM-Bereichsgrenze überschritten wird. COLLC(567) bewirkt in diesen Fällen nicht, dass das Fehler-Flag auf EIN gesetzt wird.

Vorsichtsmaßnahmen Wenn DISTC(566) zur Zuweisung eines Stapelbereichs mit der Stapel-Schreibfunktion ausgeführt wurde, muss in den nachfolgenden COLLC(567)-Befehlen immer dieselbe Stapelbereichsgröße spezifiziert werden. Die Funktion ist unzuverlässig, wenn in COLLC(567)-Befehlen eine andere Stapelbereichgröße spezifiziert wird.

Achten Sie darauf, dass die durch Of spezifizierte Offset- bzw. Stapelgröße bei Addition zu Bs nicht das Ende des Datenbereichs überschreitet.

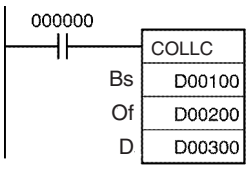
Die Offset-Daten in Of müssen BCD-Werte sein.

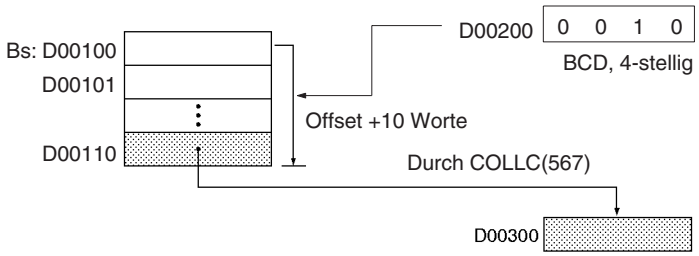
Beispiele

Datenerfassungsfunktion

Das höchstwertige Byte von D00200 ist 0, sodass COLLC(567) die Datenerfassungsfunktion ausführt.

Wenn CIO 000000 im folgenden Beispiel EIN ist, wird der Inhalt von D00110 (D00100 + 10) in D00300 kopiert, falls der Inhalt von D00200 den Wert 10 (0010 BCD) hat. Der Inhalt anderer Worte kann in D00300 kopiert werden, indem der Offset in D00200 geändert wird.

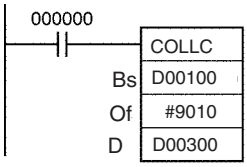


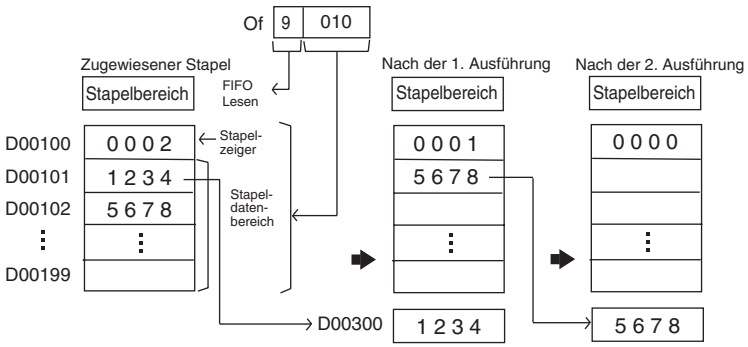


FIFO-Stapelfunktion

Das höchstwertige Byte von Of ist 9, sodass COLLC(567) die FIFO-Stapelfunktion ausführt.

Wenn CIO 000000 im folgenden Beispiel EIN ist, weist COLLC(567) ein 10-Worte-Stapelbereich (da die niederwertigsten 3 Stellen von Of den Wert #010 haben) zwischen D00100 und D00109 zu. Gleichzeitig wird der Inhalt von D00101 (Bs +1) in D00300 kopiert. Abschließend wird der Stapelzeiger um 1 dekrementiert.

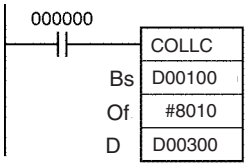


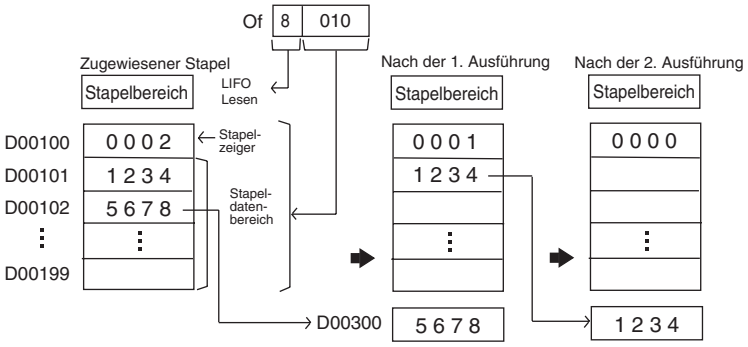


LIFO-Stapelfunktion

Das höchstwertige Byte von Of ist 8, sodass COLLC(567) die LIFO-Stapelfunktion ausführt.

Wenn CIO 000000 im folgenden Beispiel EIN ist, weist COLLC(567) ein 10-Worte-Stapelbereich (da die niederwertigsten 3 Stellen von Of den Wert #010 haben) zwischen D00100 und D00109 zu. Gleichzeitig wird der Inhalt des Quellworts (D00101 + Stapelzeiger) in D00300 kopiert. Abschließend wird der Stapelzeiger um 1 dekrementiert.



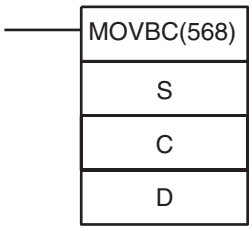


3-35-4 MOVE BIT: MOVBC(568)

Funktion

Überträgt das spezifizierte Bit.

Kontaktplansymbol



S: Quellwort oder -daten

C: Steuerwort

D: Zielwort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	MOVBC(568)
	Ausführung einmal bei steigender Flanke	@MOVBC(568)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

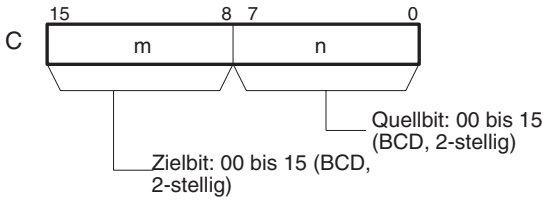
Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

C: Steuerwort

Die niederwertigen zwei Stellen von C spezifizieren, welches Bit von S das Quellbit ist, und die höherwertigen zwei Stellen von C spezifizieren, welches Bit von D das Zielbit ist.



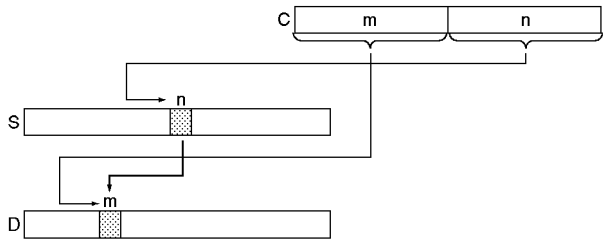
Operanden-Spezifikationen

Bereich	S	C	D
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		

Bereich	S	C	D
Konstanten	#0000 bis #FFFF (binär)	Nur spezifizierte Werte	---
Datenregister	DR0 bis DR15		
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

MOVBC(568) kopiert das spezifizierte Bit (n) von S in das spezifizierte Bit (m) in D. Die anderen Bits im Zielwort bleiben unverändert.



Hinweis Für S und D kann das gleiche Wort spezifiziert werden, um ein Bit innerhalb eines Worts zu kopieren.

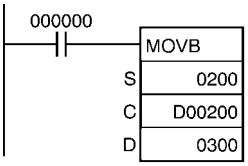
Flags

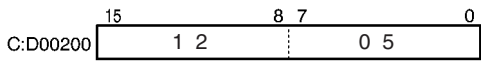
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn die niederwertigsten und höchstwertigen zwei Stellen von C keine BCD-Werte sind oder außerhalb des spezifizierten Bereichs von 00 bis 15 liegen. In allen anderen Fällen AUS.

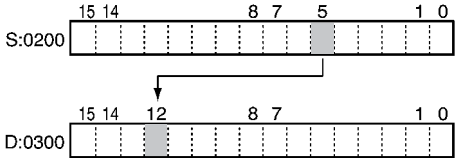
Hinweis Bei SPS der C-Serie bewirkt der Befehl MOVE BIT (MOVB), dass das Fehler-Flag auf EIN gesetzt wird, wenn der Inhalt eines indirekt adressierten DM-Worts (*DM) nicht in BCD vorliegt oder die DM-Bereichsgrenze überschritten wird. MOVBC(568) bewirkt in diesen Fällen nicht, dass das Fehler-Flag auf EIN gesetzt wird.

Beispiele

Wenn CIO 000000 im folgenden Beispiel EIN ist, wird das 5. Bit des Quellworts (CIO 0200) entsprechend des Steuerwort-Werts von 1205 in das 12. Bit des Zielworts (CIO 0300) kopiert.





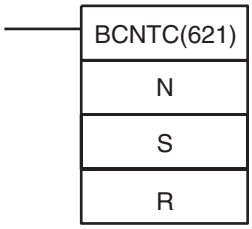


3-35-5 BIT COUNTER: BCNTC(621)

Funktion

Zählt die Gesamtanzahl der auf EIN gesetzten Bits in dem/den spezifizierten Wort(en).

Kontaktplansymbol



N: Anzahl der Worte

S: Erstes Quellwort

R: Ergebniswort

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	BCNTC(621)
	Ausführung einmal bei steigender Flanke	@BCNTC(621)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

N: Anzahl der Worte

Die Anzahl der Worte muss im Bereich zwischen 0001 und 9999 (BCD) liegen.

S: Erstes Quellwort

S und S+(N–1) müssen im gleichen Datenbereich liegen.

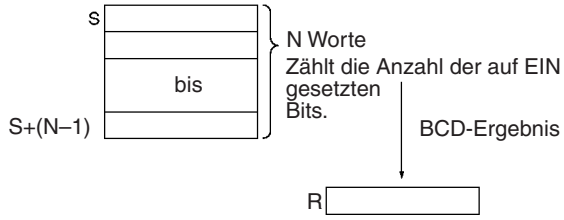
Operanden-Spezifikationen

Bereich	N	S	R
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		A448 bis A959
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	#0001 to #9999 (BCD)	---	
Datenregister	DR0 bis DR15	---	DR0 bis DR15

Bereich	N	S	R
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

Der Befehl BCNTC(621) zählt die Gesamtzahl der Bits mit EIN-Zustand in allen Worten zwischen S und S+(N–1) und gibt das BCD-Ergebnis in R aus.



Flags

Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn sich N nicht innerhalb des Bereichs von 0000 bis 9999 (BCD) befindet. EIN, wenn das Ergebnis 9999 (BCD) übersteigt. In allen anderen Fällen AUS.
Gleich-Flag	=	EIN, wenn das Ergebnis 0000 ist. In allen anderen Fällen AUS.

Hinweis

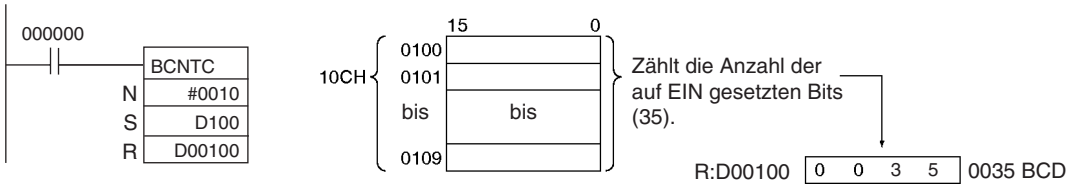
Bei SPS der C-Serie bewirkt der Befehl BIT COUNTER (BCNT) dass das Fehler-Flag auf EIN gesetzt wird, wenn der Inhalt eines indirekt adressierten DM-Worts (*DM) nicht in BCD vorliegt oder die DM-Bereichsgrenze überschritten wird. BCNTC(621) bewirkt in diesen Fällen nicht, dass das Fehler-Flag auf EIN gesetzt wird.

Vorsichtsmaßnahmen

Wenn N kein BCD-Wert zwischen 0001 und 9999 ist oder das Ergebnis 9.999 überschreitet, tritt ein Fehler auf.

Beispiel

Wenn CIO 000000 in dem folgenden Beispiel EIN ist, zählt der Befehl BCNTC(621) die Gesamtzahl der Bits mit EIN-Status in den 10 Worten von CIO 0100 bis CIO 0109 und schreibt das Ergebnis in D00100.

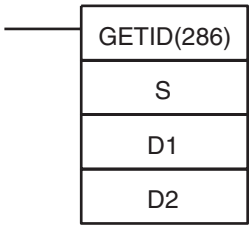


3-35-6 GET VARIABLE ID: GETID(286)

Funktion

Gibt den FINS-Befehlsvariablen-Typcode (Datenbereich) und die Wortadresse der spezifizierten Variable oder Adresse aus. Dieser Befehl wird im Allgemeinen verwendet, um die einer Variable in einem Funktionsblock zugeordnete Adresse zu ermitteln.

Kontaktplansymbol



S: Quelldaten

D1:

D2: Wortadresse

Variationen

Variationen		
Variationen	**Ausführung in jedem Zyklus bei EIN-Bedingung**	GETID(286)
	Ausführung einmal bei steigender Flanke	@GETID(286)
	Ausführung einmal bei fallender Flanke	Nicht unterstützt
Spezifikation der direkten Aktualisierung		Nicht unterstützt

Anwendbarkeit in Programmbereichen

Blockprogramm-bereiche	Schrittprogramm-bereiche	Unterpro-gramme	Interrupt-Tasks
OK	OK	OK	OK

Operanden

S: Quelldaten
Spezifiziert die Variable oder Adresse, für die der Variablentyp oder die Wortadresse abgerufen wird.

D1: Variablencode
Enthält den FINS-Variablen-Typcode (Datenbereichscode) der Quelldaten.

D2: Wortadresse
Enthält die Wortadresse der Quelldaten als 4-stellige Hexadezimalzahl.

Operanden-Spezifikationen

Bereich	S	D1	D2
CIO-Bereich	CIO 0000 bis CIO 6143		
Arbeitsbereich nicht remanent	W000 bis W511		
Arbeitsbereich remanent	H000 bis H511		
Systembereich	A000 bis A959		
Zeitgeberbereich	T0000 bis T4095		
Zählerbereich	C0000 bis C4095		
DM-Bereich	D00000 bis D32767		
EM-Bereich ohne Bank	E00000 bis E32767		
EM-Bereich mit Bank	En_00000 bis En_32767 (n = 0 bis C)		
Indirekte DM/EM-Binäradressen	@ D00000 bis @ D32767 @ E00000 bis @ E32767 @ En_00000 bis @ En_32767 (n = 0 bis C)		
Indirekte DM/EM-Adressen in BCD	*D00000 bis *D32767 *E00000 bis *E32767 *En_00000 bis *En_32767 (n = 0 bis C)		
Konstanten	---		
Datenregister	DR0 bis DR15		

Bereich	S	D1	D2
Indexregister	---		
Indirektes Adressieren mittels Indexregistern	,IR0 bis ,IR15 –2048 bis +2047 ,IR0 bis –2048 bis +2047 ,IR15 DR0 bis DR15, IR0 bis IR15 ,IR0+(++) bis ,IR15+(++) ,–(– –)IR0 bis ,–(– –)IR15		

Beschreibung

GETID(286) ruft die Datenbereichsadresse der spezifizierten Quellvariable oder –adresse ab, gibt den Datenbereichscode als 4-stellige Hexadezimalzahl in D1 und die Wortadressennummer als 4-stellige Hexadezimalzahl in D2 aus.

In der folgenden Tabelle sind die Variablen-Typcodes (Datenbereich-Codes) und die entsprechenden Adressenbereiche der SPS-Datenbereiche aufgeführt.

Datenbereich		Datengröße	Datenbereichscode (Ausgabe in D1).	Adresse (Ausgabe in D2).
CIO-Bereich	CIO	Wort	00B0 hex	0000 bis 17FF hex (0000 bis 6143)
Arbeitsbereich nicht remanent	W		00B1 hex	0000 bis 01FF hex (000 bis 511)
Arbeitsbereich remanent	H		00B2 hex	0000 bis 01FF hex (000 bis 511)
Systembereich			00B3 hex	0000 bis 03BF hex (000 bis 959)
DM-Bereich			0082 hex	0000 bis 7FFF hex (00000 bis 32767)
EM-Bereich (bestimmte Bank)	En_ (n = 0 bis C)		00A0 bis 00AC hex	0000 bis 7FFF hex (00000 bis 32767)
EM-Bereich (aktuelle Bank)	E		0098 hex	0000 bis 7FFF hex (00000 bis 32767)

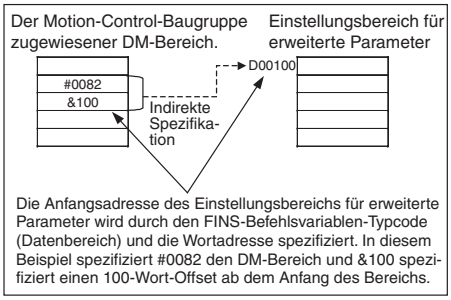
Variablen in Funktionsblöcken sind automatisch durch CX-Programmer ab Ver. 5.0 zugewiesene Adressen, es sei denn, die AT-Spezifikation wird verwendet. Wenn eine indirekte Spezifizierung des Einstellungsbereichs für erweiterte Parameter einer Spezialbaugruppe wie z.B. einer Motion-Control-Baugruppe erforderlich ist und am Anfang des Einstellungsbereichs für erweiterte Parameter eine Variable verwendet wird, muss die Adresse dieser Variablen eingestellt werden. In diesem Fall kann GETID(286) zum Abrufen der Datenbereichsadresse der Variablen verwendet werden.

Flags

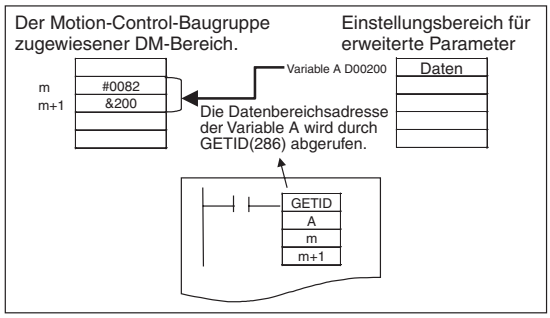
Name	Bezeichnung	Funktion
Fehler-Flag	ER	EIN, wenn S nicht innerhalb des zulässigen Bereichs liegt.

Beispiel

Normaler Betrieb



Verwendung von Funktionsblöcken



KAPITEL 4
Ausführungszeit und Step-Anzahl der Befehle

In diesem Kapitel sind die Befehlsausführungszeiten und die Anzahl der Steps für jeden Befehl der CS/CJ-Serie aufgeführt.

4-1 Ausführungszeit und Step-Anzahl der Befehle für die CS-Serie 1221
4-1-1 Sequenzeingabebefehle. 1222
4-1-2 Sequenzausgabebefehle 1223
4-1-3 Sequenzsteuerungsbefehle 1224
4-1-4 Zeitgeber- und Zählerbefehle 1225
4-1-5 Vergleichsbefehle 1226
4-1-6 Datenübertragungsbefehle 1228
4-1-7 Datenverschiebungsbefehle 1229
4-1-8 Inkrementier-/Dekrementierbefehle 1230
4-1-9 Symbolarithmetische Befehle. 1231
4-1-10 Konvertierungsbefehle 1233
4-1-11 Logikbefehle. 1235
4-1-12 Spezielle arithmetische Befehle 1235
4-1-13 Gleitkomma-Arithmetikbefehle 1236
4-1-14 Gleitkommaarithmetikbefehle doppelter Genauigkeit 1237
4-1-15 Tabellendaten-Verarbeitungsbefehle. 1238
4-1-16 Datensteuerungsbefehle 1240
4-1-17 Unterprogrammbefehle. 1241
4-1-18 Interrupt-Steuerungsbefehle 1241
4-1-19 Schritt-Befehle 1241
4-1-20 Befehle für E/A-Baugruppen 1242
4-1-21 Befehle für serielle Kommunikation. 1243
4-1-22 Netzwerkbefehle. 1243
4-1-23 Dateispeicherbefehle. 1244
4-1-24 Anzeigebefehle 1244
4-1-25 Uhrfunktionsbefehle 1245
4-1-26 Fehlerbehebungsbefehle 1245
4-1-27 Fehlerdiagnosebefehle 1245
4-1-28 Sonstige Befehle. 1246
4-1-29 Blockprogrammierungsbefehle. 1246
4-1-30 Textzeichenketten-Verarbeitungsbefehle 1248
4-1-31 Task-Steuerbefehle 1249
4-1-32 Befehle zur Modellkonvertierung (nur CPU-Baugruppen ab Ver. 3.0) 1249
4-1-33 Spezielle Funktionsblockbefehle (nur CPU-Baugruppen ab Ver. 3.0) . 1250
4-2 Ausführungszeit und Step-Anzahl der Befehle für die CJ-Serie 1250
4-2-1 Sequenzeingabebefehle. 1251
4-2-2 Sequenzausgabebefehle 1252
4-2-3 Sequenzsteuerungsbefehle 1253
4-2-4 Zeitgeber- und Zählerbefehle 1254

4-2-5 Vergleichsbefehle 1255
4-2-6 Datenübertragungsbefehle 1257
4-2-7 Datenverschiebungsbefehle 1258
4-2-8 Inkrementier-/Dekrementierbefehle 1260
4-2-9 Symbolarithmetische Befehle 1260
4-2-10 Konvertierungsbefehle 1262
4-2-11 Logikbefehle 1264
4-2-12 Spezielle arithmetische Befehle 1265
4-2-13 Gleitkomma-Arithmetikbefehle 1265
4-2-14 Gleitkommaarithmetikbefehle doppelter Genauigkeit 1267
4-2-15 Tabellendaten-Verarbeitungsbefehle 1268
4-2-16 Datensteuerungsbefehle 1269
4-2-17 Unterprogrammbefehle 1270
4-2-18 Interrupt-Steuerungsbefehle 1271
4-2-19 Befehle für schnelle Zähler und Impulsausgabe 1271
4-2-20 Schritt-Befehle 1273
4-2-21 Befehle für E/A-Baugruppen 1273
4-2-22 Befehle für serielle Kommunikation 1274
4-2-23 Netzwerkbefehle 1275
4-2-24 Dateispeicherbefehle 1276
4-2-25 Anzeigebefehle 1276
4-2-26 Uhrfunktionsbefehle 1276
4-2-27 Fehlerbehebungsbefehle 1277
4-2-28 Fehlerdiagnosebefehle 1277
4-2-29 Sonstige Befehle 1278
4-2-30 Blockprogrammierungsbefehle 1278
4-2-31 Textzeichenketten-Verarbeitungsbefehle 1280
4-2-32 Task-Steuerbefehle 1281
4-2-33 Befehle zur Modellkonvertierung (nur CPU-Baugruppen ab Ver. 3.0) 1281
4-2-34 Spezielle Funktionsblockbefehle (nur CPU-Baugruppen ab Ver. 3.0) 1282
4-2-35 Anzahl der Programm-Steps von Funktionsblock-Instanzen (CPU-Baugruppen mit Baugruppen-Version 3.0 oder höher) 1282
4-2-36 Bestimmung der Programmkapazität bestehender Programme von älteren OMRON SPS-Systemen 1283
4-2-37 Ausführungszeiten von Funktionsblock-Instanzen (CPU-Baugruppen mit Baugruppen-Version 3.0 oder höher) 1284

4-1 Ausführungszeit und Step-Anzahl der Befehle für die CS-Serie

In den nachstehenden Tabellen finden Sie die Ausführungszeit und Anzahl der Steps aller Befehle für SPS der CS-Serie.

Bei der Berechnung der Zykluszeit wird die Programmausführungszeit als Gesamtausführungsdauer aller Befehle des Anwenderprogramms ermittelt (siehe Hinweis).

Hinweis Programme sind Tasks zugeordnet, die als zyklische Tasks oder als Interrupt-Tasks ausgeführt werden.

Die Ausführungszeiten der meisten Befehle unterscheiden sich in Abhängigkeit davon, welche CPU-Baugruppe verwendet wird (CS1□-CPU6□H, CS1□-CPU6□, CS1□-CPU4□H, CS1□-CPU4□) sowie von den Bedingungen bei Ausführung des Befehls. Wenn in den folgenden Tabellen zu einem Befehl mehrere Zeilen mit Ausführungszeiten gehören, gibt die erste Zeile die minimale Ausführungszeit für diesen Befehl sowie die hierfür angenommenen Bedingungen an, die letzte Zeile entsprechend die maximale Ausführungszeit sowie die hierfür angenommenen Bedingungen.

Nicht erfüllte (AUS) Ausführungsbedingungen tragen ebenfalls zu Abweichungen in der Ausführungszeit bei.

In der Spalte *Länge (Steps)* der folgenden Tabellen wird zudem die Anzahl der Steps der einzelnen Befehle angegeben. Je nach Befehl und Operanden belegt ein Befehl für die CS-Serie zwischen 1 und 7 Steps im Anwenderprogramm. Die Anzahl der Steps in einem Programm entspricht nicht der Anzahl der Befehle.

Hinweis

1. Anders als bei früheren SPS-Systemen von OMRON, bei denen die Programmkapazität in Worten angegeben wurde, erfolgt die Angabe der Programmkapazität bei SPS-Systemen der CS-Serie in Steps. Einfach gesagt heißt das, 1 Step entspricht 1 Wort. Der für jeden Befehl erforderliche Speicherplatz ist bei einigen Befehlen der CS-Serie dennoch abweichend und es treten Fehler auf, wenn die Kapazität eines Anwenderprogramms eines anderen SPS-Systems für eine SPS der CS-Serie auf der Grundlage, dass 1 Wort gleich 1 Step ist, konvertiert wird. Am Ende dieses Kapitels finden Sie im Abschnitt *4-1 Ausführungszeit und Step-Anzahl der Befehle für die CS-Serie* detaillierte Anleitungen zur Konvertierung der Kapazität eines Programms für frühere OMRON SPS-Systeme.

   Die meisten Befehle können mit einer Flankenüberwachung (erkennbar an den Symbolen ↑, ↓, @ oder %) kombiniert werden. Dies verlängert die Ausführungszeit des Befehls um den in der nachstehenden Tabelle aufgeführten Wert.

Symbol	CS1-H CPU-Baugruppen		CS1 CPU-Baugruppen	
	CPU6□H	CPU4□H	CPU6□	CPU4□
↑ oder ↓	+0,24	+0,32	+0,41	+0,45
@ oder %	+0,24	+0,32	+0,29	+0,33

2. Verwenden Sie folgende Zeiten als Richtlinien, wenn Befehle nicht ausgeführt werden.

CS1-H CPU-Baugruppen		CS1 CPU-Baugruppen	
CPU6□H	CPU4□H	CPU6□	CPU4□
ca. 0,1	ca. 0,2	ca. 0,1 bis 0,3	ca. 0,2 bis 0,4

4-1-1 Sequenzeingabebefehle

Befehl	AWL	Code	Länge (Steps)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU6□H	CPU4□H	CPU6□	CPU4□	
LOAD	LD	---	1	0,02	0,04	0,04	0,08	---
	!LD	---	2	+21,14	+21,16	+21,16	+21,16	Erhöhung bei CS-Serie
				+45,1	+45,1	+45,1	+45,1	Erhöhung bei C200H
LOAD NOT	LD NOT	---	1	0,02	0,04	0,04	0,08	---
	!LD NOT	---	2	+21,14	+21,16	+21,16	+21,16	Erhöhung bei CS-Serie
				+45,1	+45,1	+45,1	+45,1	Erhöhung bei C200H
AND	AND	---	1	0,02	0,04	0,04	0,08	---
	!AND	---	2	+21,14	+21,16	+21,16	+21,16	Erhöhung bei CS-Serie
				+45,1	+45,1	+45,1	+45,1	Erhöhung bei C200H
AND NOT	AND NOT	---	1	0,02	0,04	0,04	0,08	---
	!AND NOT	---	2	+21,14	+21,16	+21,16	+21,16	Erhöhung bei CS-Serie
				+45,1	+45,1	+45,1	+45,1	Erhöhung bei C200H
OR	OR	---	1	0,02	0,04	0,04	0,08	---
	!OR	---	2	+21,14	+21,16	+21,16	+21,16	Erhöhung bei CS-Serie
				+45,1	+45,1	+45,1	+45,1	Erhöhung bei C200H
OR NOT	OR NOT	---	1	0,02	0,04	0,04	0,08	---
	!OR NOT	---	2	+21,14	+21,16	+21,16	+21,16	Erhöhung bei CS-Serie
				+45,1	+45,1	+45,1	+45,1	Erhöhung bei C200H
AND LOAD	AND LD	---	1	0,02	0,04	0,04	0,08	---
OR LOAD	OR LD	---	1	0,02	0,04	0,04	0,08	---
NOT	NOT	520	1	0,02	0,04	0,04	0,08	---
CONDITION ON	UP	521	3	0,3	0,42	0,46	0,54	---
CONDITION OFF	DOWN	522	4	0,3	0,42	0,46	0,54	---
LOAD BIT TEST	LD TST	350	4	0,14	0,24	0,25	0,37	---
LOAD BIT TEST NOT	LD TSTN	351	4	0,14	0,24	0,25	0,37	---
AND BIT TEST NOT	AND TSTN	351	4	0,14	0,24	0,25	0,37	---
OR BIT TEST	OR TST	350	4	0,14	0,24	0,25	0,37	---
OR BIT TEST NOT	OR TSTN	351	4	0,14	0,24	0,25	0,37	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle der folgenden Tabelle (Spalte Länge) um 1.

4-1-2 Sequenzausgabebefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
OUTPUT	OUT	---	1	0,02	0,04	0,17	0,21	---
	!OUT	---	2	+21,37	+21,37	+21,37	+21,37	Erhöhung bei CS-Serie
				+49,3	+49,3	+49,3	+49,3	Erhöhung bei C200H
OUTPUT NOT	OUT NOT	---	1	0,02	0,04	0,17	0,21	---
	!OUT NOT	---	2	+21,37	+21,37	+21,37	+21,37	Erhöhung bei CS-Serie
				+49,3	+49,3	+49,3	+49,3	Erhöhung bei C200H
KEEP	KEEP	011	1	0,06	0,08	0,25	0,29	---
DIFFEREN-TIATE UP	DIFU	013	2	0,24	0,40	0,46	0,54	---
DIFFEREN-TIATE DOWN	DIFD	014	2	0,24	0,40	0,46	0,54	---
SET	SET	---	1	0,02	0,06	0,17	0,21	---
	!SET	---	2	+21,37	+21,37	+21,37	+21,37	Erhöhung bei CS-Serie
				+49,3	+49,3	+49,3	+49,3	Erhöhung bei C200H
RESET	RSET	---	1	0,02	0,06	0,17	0,21	Bei Spezifikation des Worts
	!RSET	---	2	+21,37	+21,37	+21,37	+21,37	Erhöhung bei CS-Serie
				+49,3	+49,3	+49,3	+49,3	Erhöhung bei C200H
MULTIPLE BIT SET	SETA	530	4	5,8	6,1	7,8	7,8	Setzen von 1 Bit
				25,7	27,2	38,8	38,8	Setzen von 1.000 Bits
MULTIPLE BIT RESET	RSTA	531	4	5,7	6,1	7,8	7,8	Rücksetzen von 1 Bit
				25,8	27,1	38,8	38,8	Rücksetzen von 1.000 Bits
SINGLE BIT SET	SETB	532	2	0,24	0,34	---	---	---
	!SETB		3	+21,44	+21,54	---	---	---
SINGLE BIT RESET	RSTB	534	2	0,24	0,34	---	---	---
	!RSTB		3	+21,44	+21,54	---	---	---
SINGLE BIT OUTPUT	OUTB	534	2	0,22	0,32	---	---	---
	!OUTB		3	+21,42	+21,52	---	---	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-3 Sequenzsteuerungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
END	END	001	1	5,5	6,0	4,0	4,0	---
NO OPERATION	NOP	000	1	0,02	0,04	0,08	0,12	---
INTERLOCK	IL	002	1	0,06	0,06	0,12	0,12	---
INTERLOCK CLEAR	ILC	003	1	0,06	0,06	0,12	0,12	---
MULTI-INTERLOCK DIFFERENTIATION HOLD (siehe Hinweis 2)	MILH	517	3	6,1	6,5	---	---	Während der Sperre
				7,5	7,9	---	---	Außerhalb der Sperre bei nicht gesetzter Sperre
				8,9	9,7	---	---	Außerhalb der Sperre bei gesetzter Sperre
MULTI-INTERLOCK DIFFERENTIATION RELEASE (siehe Hinweis 2)	MILR	518	3	6,1	6,5	---	---	Während der Sperre
				7,5	7,9	---	---	Außerhalb der Sperre bei nicht gesetzter Sperre
				8,9	9,7	---	---	Außerhalb der Sperre bei gesetzter Sperre
MULTI-INTERLOCK CLEAR (siehe Hinweis 2)	MILC	519	2	5,0	5,6	---	---	Sperre nicht aufgehoben
				5,7	6,2	---	---	Sperre aufgehoben
JUMP	JMP	004	2	0,38	0,48	8,1	8,1	---
JUMP END	JME	005	2	---	---	---	---	---
CONDITIONAL JUMP	CJP	510	2	0,38	0,48	7,4	7,4	Wenn die Sprungbedingung erfüllt ist
CONDITIONAL JUMP NOT	CJPN	511	2	0,38	0,48	8,5	8,5	Wenn die Sprungbedingung erfüllt ist
MULTIPLE JUMP	JMP0	515	1	0,06	0,06	0,12	0,12	---
MULTIPLE JUMP END	JME0	516	1	0,06	0,06	0,12	0,12	---
FOR LOOP	FOR	512	2	0,52	0,54	0,12	0,21	Spezifikation einer Konstanten
BREAK LOOP	BREAK	514	1	0,06	0,06	0,12	0,12	---
NEXT LOOP	NEXT	513	1	0,18	0,16	0,17	0,17	Wenn die Schleife fortgesetzt wird
				0,22	0,40	0,12	0,12	Wenn die Schleife beendet wird

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-1-4 Zeitgeber- und Zählerbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
TIMER	TIM	---	3	0,56	0,88	0,37	0,42	---
	TIMX	550	3	0,56	0,88	---	---	---
COUNTER	CNT	---	3	0,56	0,88	0,37	0,42	---
	CNTX	546	3	0,56	0,88	---	---	---
HIGH-SPEED TIMER	TIMH	015	3	0,88	1,14	0,37	0,42	---
	TIMHX	551	3	0,88	1,14	---	---	---
ONE-MS TIMER	TMHH	540	3	0,86	1,12	0,37	0,42	---
	TMHHX	552	3	0,86	1,12	---	---	---
ACCUMULA-TIVE TIMER	TTIM	087	3	16,1	17,0	21,4	21,4	---
				10,9	11,4	14,8	14,8	Bei Rücksetzung
				8,5	8,7	10,7	10,7	Bei Sperrung
	TTIMX	555	3	16,1	17,0	---	---	---
				10,9	11,4	---	---	Bei Rücksetzung
				8,5	8,7	---	---	Bei Sperrung
LONG TIMER	TIML	542	4	7,6	10,0	12,8	12,8	---
				6,2	6,5	7,8	7,8	Bei Sperrung
	TIMLX	553	4	7,6	10,0	---	---	---
				6,2	6,5	---	---	Bei Sperrung
MULTI-OUT-PUT TIMER	MTIM	543	4	20,9	23,3	26,0	26,0	---
				5,6	5,8	7,8	7,8	Bei Rücksetzung
	MTIMX	554	4	20,9	23,3	---	---	---
				5,6	5,8	---	---	Bei Rücksetzung
REVERSIBLE COUNTER	CNTR	012	3	16,9	19,0	20,9	20,9	---
	CNTRX	548	3	16,9	19,0	---	---	---
RESET TIMER/ COUNTER	CNR	545	3	9,9	10,6	13,9	13,9	Rücksetzen eines Worts
				4,16 ms	4,16 ms	5,42 ms	5,42 ms	Rücksetzen von 1.000 Worten
	CNRX	547	3	9,9	10,6	---	---	Rücksetzen eines Worts
				4,16 ms	4,16 ms	---	---	Rücksetzen von 1.000 Worten

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-5 Vergleichsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
Symbolvergleichsbefehle (vorzeichenlos)	LD, AND, OR +=	300	4	0,10	0,16	0,21	0,37	---
	LD, AND, OR + <>	305						
	LD, AND, OR + <	310						
	LD, AND, OR +<=	315						
	LD, AND, OR + >	320						
	LD, AND, OR +>=	325						
Symbolvergleichsbefehle (Doppelwort, vorzeichenlos)	LD, AND, OR +=+L	301	4	0,10	0,16	0,29	0,54	---
	LD, AND, OR +<>+L	306						
	LD, AND, OR +<+L	311						
	LD, AND, OR +<=+L	316						
	LD, AND, OR +>+L	321						
	LD, AND, OR +>=+L	326						
Symbolvergleichsbefehle (vorzeichenbehaftet)	LD, AND, OR +=+S	302	4	0,10	0,16	6,50	6,50	---
	LD, AND, OR +<>+S	307						
	LD, AND, OR +<+S	312						
	LD, AND, OR +<=	317						
	LD, AND, OR +>+S	322						
	LD, AND, OR +>=+S	327						
Symbolvergleichsbefehle (Doppelwort, vorzeichenbehaftet)	LD, AND, OR +=+SL	303	4	0,10	0,16	6,50	6,50	---
	LD, AND, OR +<>+SL	308						
	LD, AND, OR +<+SL	313						
	LD, AND, OR +<=+SL	318						
	LD, AND, OR +>+SL	323						
	LD, AND, OR +>=+SL	328						

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
Zeitvergleichs-befehle (siehe Hinweis 2)	LD, AND, OR +DT	341	4	25,1	36,4	---	---	Die EIN- und AUS-Ausführungszeiten sind mit den Angaben links identisch.
	LD, AND, OR +<>DT	342	4	25,2	36,4	---	---	
	LD, AND, OR +<DT	343	4	25,2	36,4	---	---	
	LD, AND, OR +<=DT	344	4	25,2	36,4	---	---	
	LD, AND, OR +>DT	345	4	25,1	36,4	---	---	
	LD, AND, OR +>=DT	346	4	25,2	36,4	---	---	
COMPARE	CMP	020	3	0,04	0,04	0,17	0,29	---
	!CMP	020	7	+42,1	+42,1	+42,4	+42,4	Erhöhung bei CS-Serie
				+90,4	+90,4	+90,5	+90,5	Erhöhung bei C200H
DOUBLE COMPARE	CMPL	060	3	0,08	0,08	0,25	0,46	---
SIGNED BINARY COMPARE	CPS	114	3	0,08	0,08	6,50	6,50	---
	!CPS	114	7	+35,9	+35,9	+42,4	+42,4	Erhöhung bei CS-Serie
				+84,1	+84,1	+90,5	+90,5	Erhöhung bei C200H
DOUBLE SIGNED BINARY COMPARE	CPSL	115	3	0,08	0,08	6,50	6,50	---
TABLE COMPARE	TCMP	085	4	14,0	15,2	21,9	21,9	---
MULTIPLE COMPARE	MCMP	019	4	20,5	22,8	31,2	31,2	---
UNSIGNED BLOCK COMPARE	BCMP	068	4	21,5	23,7	32,6	32,6	---
AREA RANGE COMPARE	ZCP	088	3	5,3	5,4	---	---	---
DOUBLE AREA RANGE COMPARE	ZCPL	116	3	5,5	6,7	---	---	---

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-1-6 Datenübertragungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
MOVE	MOV	021	3	0,18	0,20	0,25	0,29	---
	!MOV	021	7	+21,38	+21,40	+42,36	+42,36	Erhöhung bei CS-Serie
				+90,52	+90,52	+90,52	+90,52	Erhöhung bei C200H
DOUBLE MOVE	MOVL	498	3	0,32	0,34	0,42	0,50	---
MOVE NOT	MVN	022	3	0,18	0,20	0,25	0,29	---
DOUBLE MOVE NOT	MVNL	499	3	0,32	0,34	0,42	0,50	---
MOVE BIT	MOVB	082	4	0,24	0,34	7,5	7,5	---
MOVE DIGIT	MOVD	083	4	0,24	0,34	7,3	7,3	---
MULTIPLE BIT TRANSFER	XFRB	062	4	10,1	10,8	13,6	13,6	Übertragung von 1 Bit
				186,4	189,8	269,2	269,2	Übertragung von 255 Bits
BLOCK TRANSFER	XFER	070	4	0,36	0,44	11,2	11,2	Übertragung von 1 Wort
				300,1	380,1	633,5	633,5	Übertragung von 1.000 Worten
BLOCK SET	BSET	071	4	0,26	0,28	8,5	8,5	Setzen von 1 Wort
				200,1	220,1	278,3	278,3	Setzen von 1.000 Worten
DATA EXCHANGE	XCHG	073	3	0,40	0,56	0,5	0,7	---
DOUBLE DATA EXCHANGE	XCGL	562	3	0,76	1,04	0,9	1,3	---
SINGLE WORD DISTRIBUTE	DIST	080	4	5,1	5,4	7,0	7,0	---
DATA COLLECT	COLL	081	4	5,1	5,3	7,1	7,1	---
MOVE TO REGISTER	MOVR	560	3	0,08	0,08	0,42	0,50	---
MOVE TIMER/ COUNTER PV TO REGISTER	MOVRW	561	3	0,42	0,50	0,42	0,50	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-7 Datenverschiebungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
SHIFT REGISTER	SFT	010	3	7,4	10,4	10,4	10,4	Verschieben von 1 Wort
				433,2	488,0	763,1	763,1	Verschieben von 1.000 Worten
REVERSIBLE SHIFT REGISTER	SFTR	084	4	6,9	7,2	9,6	9,6	Verschieben von 1 Wort
				615,3	680,2	859,6	859,6	Verschieben von 1.000 Worten
ASYNCHRO-NOUS SHIFT REGISTER	ASFT	017	4	6,2	6,4	7,7	7,7	Verschieben von 1 Wort
				1,22 ms	1,22 ms	2,01 ms	2,01 ms	Verschieben von 1.000 Worten
WORD SHIFT	WSFT	016	4	4,5	4,7	7,8	7,8	Verschieben von 1 Wort
				171,5	171,7	781,7	781,7	Verschieben von 1.000 Worten
ARITHMETIC SHIFT LEFT	ASL	025	2	0,22	0,32	0,29	0,37	---
DOUBLE SHIFT LEFT	ASLL	570	2	0,40	0,56	0,50	0,67	---
ARITHMETIC SHIFT RIGHT	ASR	026	2	0,22	0,32	0,29	0,37	---
DOUBLE SHIFT RIGHT	ASRL	571	2	0,40	0,56	0,50	0,67	---
ROTATE LEFT	ROL	027	2	0,22	0,32	0,29	0,37	---
DOUBLE ROTATE LEFT	ROLL	572	2	0,40	0,56	0,50	0,67	---
ROTATE LEFT WITHOUT CARRY	RLNC	574	2	0,22	0,32	0,29	0,37	---
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL	576	2	0,40	0,56	0,50	0,67	---
ROTATE RIGHT	ROR	028	2	0,22	0,32	0,29	0,37	---
DOUBLE ROTATE RIGHT	RORL	573	2	0,40	0,56	0,50	0,67	---
ROTATE RIGHT WIT-HOUT CARRY	RRNC	575	2	0,22	0,32	0,29	0,37	---
DOUBLE ROTATE RIGHT WIT-HOUT CARRY	RRNL	577	2	0,40	0,56	0,50	0,67	---
ONE DIGIT SHIFT LEFT	SLD	074	3	5,9	6,1	8,2	8,2	Verschieben von 1 Wort
				561,1	626,3	760,7	760,7	Verschieben von 1.000 Worten
ONE DIGIT SHIFT RIGHT	SRD	075	3	6,9	7,1	8,7	8,7	Verschieben von 1 Wort
				760,5	895,5	1,07 ms	1,07 ms	Verschieben von 1.000 Worten

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
SHIFT N-BIT DATA LEFT	NSFL	578	4	7,5	8,3	10,5	10,5	Verschieben von 1 Bit
				40,3	45,4	55,5	55,5	Verschieben von 1.000 Bits
SHIFT N-BIT DATA RIGHT	NSFR	579	4	7,5	8,3	10,5	10,5	Verschieben von 1 Bit
				50,5	55,3	69,3	69,3	Verschieben von 1.000 Bits
SHIFT N-BITS LEFT	NASL	580	3	0,22	0,32	0,29	0,37	---
DOUBLE SHIFT N-BITS LEFT	NSLL	582	3	0,40	0,56	0,50	0,67	---
SHIFT N-BITS RIGHT	NASR	581	3	0,22	0,32	0,29	0,37	---
DOUBLE SHIFT N-BITS RIGHT	NSRL	583	3	0,40	0,56	0,50	0,67	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-8 Inkrementier-/Dekrementierbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
INCREMENT BINARY	++	590	2	0,22	0,32	0,29	0,37	---
DOUBLE INCREMENT BINARY	++L	591	2	0,40	0,56	0,50	0,67	---
DECREMENT BINARY	– –	592	2	0,22	0,32	0,29	0,37	---
DOUBLE DECREMENT BINARY	– –L	593	2	0,40	0,56	0,50	0,67	---
INCREMENT BCD	++B	594	2	6,4	4,5	7,4	7,4	---
DOUBLE INCREMENT BCD	++BL	595	2	5,6	4,9	6,1	6,1	---
DECREMENT BCD	– –B	596	2	6,3	4,6	7,2	7,2	---
DOUBLE DECREMENT BCD	– –BL	597	2	5,3	4,7	7,1	7,1	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-9 Symbolarithmetische Befehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
SIGNED BINARY ADD WITHOUT CARRY	+	400	4	0,18	0,20	0,25	0,37	---
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	401	4	0,32	0,34	0,42	0,54	---
SIGNED BINARY ADD WITH CARRY	+C	402	4	0,18	0,20	0,25	0,37	---
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	403	4	0,32	0,34	0,42	0,54	---
BCD ADD WITHOUT CARRY	+B	404	4	8,2	8,4	14,0	14,0	---
DOUBLE BCD ADD WITHOUT CARRY	+BL	405	4	13,3	14,5	19,0	19,0	---
BCD ADD WITH CARRY	+BC	406	4	8,9	9,1	14,5	14,5	---
DOUBLE BCD ADD WITH CARRY	+BCL	407	4	13,8	15,0	19,6	19,6	---
SIGNED BINARY SUBTRACT WITHOUT CARRY	–	410	4	0,18	0,20	0,25	0,37	---
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	–L	411	4	0,32	0,34	0,42	0,54	---
SIGNED BINARY SUBTRACT WITH CARRY	–C	412	4	0,18	0,20	0,25	0,37	---
DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	–CL	413	4	0,32	0,34	0,42	0,54	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
BCD SUBTRACT WITHOUT CARRY	–B	414	4	8,0	8,2	13,1	13,1	---
DOUBLE BCD SUBTRACT WITHOUT CARRY	–BL	415	4	12,8	14,0	18,2	18,2	---
BCD SUBTRACT WITH CARRY	–BC	416	4	8,5	8,6	13,8	13,8	---
DOUBLE BCD SUBTRACT WITH CARRY	–BCL	417	4	13,4	14,7	18,8	18,8	---
SIGNED BINARY MULTIPLY	*	420	4	0,38	0,40	0,50	0,58	---
DOUBLE SIGNED BINARY MULTIPLY	*L	421	4	7,23	8,45	11,19	11,19	---
UNSIGNED BINARY MULTIPLY	*U	422	4	0,38	0,40	0,50	0,58	---
DOUBLE UNSIGNED BINARY MULTIPLY	*UL	423	4	7,1	8,3	10,63	10,63	---
BCD MULTIPLY	*B	424	4	9,0	9,2	12,8	12,8	---
DOUBLE BCD MULTIPLY	*BL	425	4	23,0	24,2	35,2	35,2	---
SIGNED BINARY DIVIDE	/	430	4	0,40	0,42	0,75	0,83	---
DOUBLE SIGNED BINARY DIVIDE	/L	431	4	7,2	8,4	9,8	9,8	---
UNSIGNED BINARY DIVIDE	/U	432	4	0,40	0,42	0,75	0,83	---
DOUBLE UNSIGNED BINARY DIVIDE	/UL	433	4	6,9	8,1	9,1	9,1	---
BCD DIVIDE	/B	434	4	8,6	8,8	15,9	15,9	---
DOUBLE BCD DIVIDE	/BL	435	4	17,7	18,9	26,2	26,2	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-10 Konvertierungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
BCD-TO-BINARY	BIN	023	3	0,22	0,24	0,25	0,29	---
DOUBLE BCD-TO-DOUBLE BINARY	BINL	058	3	6,5	6,8	9,1	9,1	---
BINARY-TO-BCD	BCD	024	3	0,24	0,26	8,3	8,3	---
DOUBLE BINARY-TO-DOUBLE BCD	BCDL	059	3	6,7	7,0	9,2	9,2	---
2'S COMPLEMENT	NEG	160	3	0,18	0,20	0,25	0,29	---
DOUBLE 2'S COMPLEMENT	NEGL	161	3	0,32	0,34	0,42	0,5	---
16-BIT TO 32-BIT SIGNED BINARY	SIGN	600	3	0,32	0,34	0,42	0,50	---
DATA DECODER	MLPX	076	4	0,32	0,42	8,8	8,8	Dekodieren einer Stelle (4 zu 16)
				0,98	1,20	12,8	12,8	Dekodieren von vier Stellen (4 zu 16)
				3,30	4,00	20,3	20,3	Dekodieren einer Stelle (8 zu 256)
				6,50	7,90	33,4	33,4	Dekodieren von zwei Stellen (8 zu 256)
DATA ENCODER	DMPX	077	4	7,5	7,9	10,4	10,4	Kodieren einer Stelle (16 zu 4)
				49,6	50,2	59,1	59,1	Kodieren von vier Stellen (16 zu 4)
				18,2	18,6	23,6	23,6	Kodieren einer Stelle (256 zu 8)
				55,1	57,4	92,5	92,5	Kodieren von zwei Stellen (256 zu 8)
ASCII CONVERT	ASC	086	4	6,8	7,1	9,7	9,7	Konvertieren einer Stelle zu ASCII
				11,2	11,7	15,1	15,1	Konvertieren von vier Stellen zu ASCII
ASCII TO HEX	HEX	162	4	7,1	7,4	10,1	10,1	Konvertieren einer Stelle
COLUMN TO LINE	LINE	063	4	19,0	23,1	29,1	29,1	---
LINE TO COLUMN	COLM	064	4	23,2	27,5	37,3	37,3	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
SIGNED BCD-TO-BINARY	BINS	470	4	8,0	8,3	12,1	12,1	Datenformateinstellung 0
				8,0	8,3	12,1	12,1	Datenformateinstellung 1
				8,3	8,6	12,7	12,7	Datenformateinstellung 2
				8,5	8,8	13,0	13,0	Datenformateinstellung 3
DOUBLE SIGNED BCD-TO-BINARY	BISL	472	4	9,2	9,6	13,6	13,6	Datenformateinstellung 0
				9,2	9,6	13,7	13,7	Datenformateinstellung 1
				9,5	9,9	14,2	14,2	Datenformateinstellung 2
				9,6	10,0	14,4	14,4	Datenformateinstellung 3
SIGNED BINARY-TO-BCD	BCDS	471	4	6,6	6,9	10,6	10,6	Datenformateinstellung 0
				6,7	7,0	10,8	10,8	Datenformateinstellung 1
				6,8	7,1	10,9	10,9	Datenformateinstellung 2
				7,2	7,5	11,5	11,5	Datenformateinstellung 3
DOUBLE SIGNED BINARY-TO-BCD	BDSL	473	4	8,1	8,4	11,6	11,6	Datenformateinstellung 0
				8,2	8,6	11,8	11,8	Datenformateinstellung 1
				8,3	8,7	12,0	12,0	Datenformateinstellung 2
				8,8	9,2	12,5	12,5	Datenformateinstellung 3
GRAY CODE CONVERSION (siehe Hinweis 2)	GRY	474	4	46,9	72,1	---	---	8 Bit im Binärformat
				49,6	75,2	---	---	8 Bit im BCD-Format
				57,7	87,7	---	---	8 Bit im Winkelformat
				61,8	96,7	---	---	15 Bit im Binärformat
				64,5	99,6	---	---	15 Bit im BCD-Format
				72,8	112,4	---	---	15 Bit im Winkelformat
				52,3	87,2	---	---	360° Binär-Wert
				55,1	90,4	---	---	360° BCD-Wert
				64,8	98,5	---	---	360° Winkelwert

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-1-11 Logikbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
LOGICAL AND	ANDW	034	4	0,18	0,20	0,25	0,37	---
DOUBLE LOGICAL AND	ANDL	610	4	0,32	0,34	0,42	0,54	---
LOGICAL OR	ORW	035	4	0,22	0,32	0,25	0,37	---
DOUBLE LOGICAL OR	ORWL	611	4	0,32	0,34	0,42	0,54	---
EXCLUSIVE OR	XORW	036	4	0,22	0,32	0,25	0,37	---
DOUBLE EXCLUSIVE OR	XORL	612	4	0,32	0,34	0,42	0,54	---
EXCLUSIVE NOR	XNRW	037	4	0,22	0,32	0,25	0,37	---
DOUBLE EXCLUSIVE NOR	XNRL	613	4	0,32	0,34	0,42	0,54	---
COMPLE-MENT	COM	029	2	0,22	0,32	0,29	0,37	---
DOUBLE COMPLE-MENT	COML	614	2	0,40	0,56	0,50	0,67	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-12 Spezielle arithmetische Befehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
BINARY ROOT	ROTB	620	3	49,6	50,0	530,7	530,7	---
BCD SQUARE ROOT	ROOT	072	3	13,7	13,9	514,5	514,5	---
ARITHMETIC PROCESS	APR	069	4	6,7	6,9	32,3	32,3	Sinus und Cosinus
				17,2	18,4	78,3	78,3	Geraden-segment-Annäherung
FLOATING POINT DIVIDE	FDIV	079	4	116,6	176,6	176,6	176,6	---
BIT COUN-TER	BCNT	067	4	0,3	0,38	22,1	22,1	Zählen von 1 Wort

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-13 Gleitkomma-Arithmetikbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
FLOATING TO 16-BIT	FIX	450	3	10,6	10,8	14,5	14,5	---
FLOATING TO 32-BIT	FIXL	451	3	10,8	11,0	14,6	14,6	---
16-BIT TO FLOATING	FLT	452	3	8,3	8,5	11,1	11,1	---
32-BIT TO FLOATING	FLTL	453	3	8,3	8,5	10,8	10,8	---
FLOATING-POINT ADD	+F	454	4	8,0	9,2	10,2	10,2	---
FLOATING-POINT SUBTRACT	–F	455	4	8,0	9,2	10,3	10,3	---
FLOATING-POINT DIVIDE	/F	457	4	8,7	9,9	12,0	12,0	---
FLOATING-POINT MULTIPLY	*F	456	4	8,0	9,2	10,5	10,5	---
DEGREES TO RADIANS	RAD	458	3	10,1	10,2	14,9	14,9	---
RADIANS TO DEGREES	DEG	459	3	9,9	10,1	14,8	14,8	---
SINE	SIN	460	3	42,0	42,2	61,1	61,1	---
COSINE	COS	461	3	31,5	31,8	44,1	44,1	---
TANGENT	TAN	462	3	16,3	16,6	22,6	22,6	---
ARC SINE	ASIN	463	3	17,6	17,9	24,1	24,1	---
ARC COSINE	ACOS	464	3	20,4	20,7	28,0	28,0	---
ARC TANGENT	ATAN	465	3	16,1	16,4	16,4	16,4	---
SQUARE ROOT	SQRT	466	3	19,0	19,3	28,1	28,1	---
EXPONENT	EXP	467	3	65,9	66,2	96,7	96,7	---
LOGARITHM	LOG	468	3	12,8	13,1	17,4	17,4	---
EXPONENTIAL POWER	PWR	840	4	125,4	126,0	181,7	181,7	---
Symbolvergleichsbefehle mit Gleitkommaoperanden	LD, AND, OR +=F	329	3	6,6	8,3	---	---	---
	LD, AND, OR +<>F	330						
	LD, AND, OR +<F	331						
	LD, AND, OR +<=F	332						
	LD, AND, OR +>F	333						
	LD, AND, OR +>=F	334						

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
FLOATING-POINT TO ASCII	FSTR	448	4	48,5	48,9	---	---	---
ASCII TO FLOATING-POINT	FVAL	449	3	21,1	21,3	---	---	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-14 Gleitkommaarithmetikbefehle doppelter Genauigkeit

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
DOUBLE SYMBOL COMPARISON	LD, AND, OR +=D	335	3	8,5	10,3	---	---	---
	LD, AND, OR +<>D	336						
	LD, AND, OR +<D	337						
	LD, AND, OR +<=D	338						
	LD, AND, OR +>D	339						
	LD, AND, OR +>=D	340						
DOUBLE FLOATING TO 16-BIT BINARY	FIXD	841	3	11,7	12,1	---	---	---
DOUBLE FLOATING TO 32-BIT BINARY	FIXLD	842	3	11,6	12,1	---	---	---
16-BIT BINARY TO DOUBLE FLOATING	DBL	843	3	9,9	10,0	---	---	---
32-BIT BINARY TO DOUBLE FLOATING	DBLL	844	3	9,8	10,0	---	---	---
DOUBLE FLOATING-POINT ADD	+D	845	4	11,2	11,9	---	---	---
DOUBLE FLOATING-POINT SUBTRACT	–D	846	4	11,2	11,9	---	---	---
DOUBLE FLOATING-POINT MULTIPLY	*D	847	4	12,0	12,7	---	---	---
DOUBLE FLOATING-POINT DIVIDE	/D	848	4	23,5	24,2	---	---	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
DOUBLE DEGREES TO RADIANS	RADD	849	3	27,4	27,8	---	---	---
DOUBLE RADIANS TO DEGREES	DEGD	850	3	11,2	11,9	---	---	---
DOUBLE SINE	SIND	851	3	45,4	45,8	---	---	---
DOUBLE COSINE	COSD	852	3	43,0	43,4	---	---	---
DOUBLE TANGENT	TAND	853	3	20,1	20,5	---	---	---
DOUBLE ARC SINE	ASIND	854	3	21,5	21,9	---	---	---
DOUBLE ARC COSINE	ACOSD	855	3	24,7	25,1	---	---	---
DOUBLE ARC TANGENT	ATAND	856	3	19,3	19,7	---	---	---
DOUBLE SQUARE ROOT	SQRTD	857	3	47,4	47,9	---	---	---
DOUBLE EXPONENT	EXPD	858	3	121,0	121,4	---	---	---
DOUBLE LOGARITHM	LOGD	859	3	16,0	16,4	---	---	---
DOUBLE EXPONEN-TIAL POWER	PWRD	860	4	223,9	224,2	---	---	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-15 Tabellendaten-Verarbeitungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
SET STACK	SSET	630	3	8,0	8,3	8,5	8,5	Spezifikation von 5 Worten im Stapelbereich
				231,6	251,8	276,8	276,8	Spezifikation von 1.000 Worten im Stapelbereich
PUSH ONTO STACK	PUSH	632	3	6,5	8,6	9,1	9,1	---
FIRST IN FIRST OUT	FIFO	633	3	6,9	8,9	10,6	10,6	Spezifikation von 5 Worten im Stapelbereich
				352,6	434,3	1,13 ms	1,13 ms	Spezifikation von 1.000 Worten im Stapelbereich
LAST IN FIRST OUT	LIFO	634	3	7,0	9,0	9,9	9,9	---
DIMENSION RECORD TABLE	DIM	631	5	15,2	21,6	142,1	142,1	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
SET RECORD LOCATION	SETR	635	4	5,4	5,9	7,0	7,0	---
GET RECORD NUMBER	GETR	636	4	7,8	8,4	11,0	11,0	---
DATA SEARCH	SRCH	181	4	15,5	19,5	19,5	19,5	Suche nach einem Wort
				2,42 ms	3,34 ms	3,34 ms	3,34 ms	Suche nach 1.000 Worten
SWAP BYTES	SWAP	637	3	12,2	13,6	13,6	13,6	Austausch von 1 Wort
				1,94 ms	2,82 ms	2,82 ms	2,82 ms	Austausch von 1.000 Worten
FIND MAXIMUM	MAX	182	4	19,2	24,9	24,9	24,9	Suche nach einem Wort
				2,39 ms	3,36 ms	3,36 ms	3,36 ms	Suche nach 1.000 Worten
FIND MINIMUM	MIN	183	4	19,2	25,3	25,3	25,3	Suche nach einem Wort
				2,39 ms	3,33 ms	3,33 ms	3,33 ms	Suche nach 1.000 Worten
SUM	SUM	184	4	28,2	38,5	38,5	38,3	Addition von 1 Wort
				1,42 ms	1,95 ms	1,95 ms	1,95 ms	Addition von 1.000 Worten
FRAME CHECKSUM	FCS	180	4	20,0	28,3	28,3	28,3	Tabelle mit 1 Wort
				1,65 ms	2,48 ms	2,48 ms	2,48 ms	Tabelle mit 1.000 Worten
STACK SIZE READ	SNUM	638	3	6,0	6,3	---	---	---
STACK DATA READ	SREAD	639	4	8,0	8,4	---	---	---
STACK DATA OVERWRITE	SWRIT	640	4	7,2	7,6	---	---	---
STACK DATA INSERT	SINS	641	4	7,8	9,9	---	---	---
				354,0	434,8	---	---	Tabelle mit 1.000 Worten
STACK DATA DELETE	SDEL	642	4	8,6	10,6	---	---	---
				354,0	436,0	---	---	Tabelle mit 1.000 Worten

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-16 Datensteuerungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
PID CONTROL	PID	190	4	436,2	678,2	678,2	678,2	Erstmalige Ausführung
				332,3	474,9	474,9	474,9	Abtastung
				97,3	141,3	141,3	141,3	Keine Abtastung
LIMIT CONTROL	LMT	680	4	16,1	22,1	22,1	22,1	---
DEAD BAND CONTROL	BAND	681	4	17,0	22,5	22,5	22,5	---
DEAD ZONE CONTROL	ZONE	682	4	15,4	20,5	20,5	20,5	---
TIME-PROPORTIONAL OUTPUT (siehe Hinweis 2)	TPO	685	4	10,4	14,8	---	---	AUS-Ausführungszeit
				54,5	82,0	---	---	EIN-Aus-führungszeit mit Tastverhältnis-Spezifikation oder angezeigter Ausgabegrenze
				61,0	91,9	---	---	EIN-Aus-führungszeit mit Ausgangsstellwert-Spezifikation und aktivierter Ausgabegrenze
SCALING	SCL	194	4	37,1	53,0	56,8	56,8	---
SCALING 2	SCL2	486	4	28,5	40,2	50,7	50,7	---
SCALING 3	SCL3	487	4	33,4	47,0	57,7	57,7	---
AVERAGE	AVG	195	4	36,3	52,6	53,1	53,1	Mittelwertbildung über einen Zyklus
				291,0	419,9	419,9	419,9	Mittelwertbildung über 64 Zyklen
PID CONTROL WITH AUTOTUNING	PIDAT	191	4	446,3	712,5	---	---	Erstmalige Ausführung
				339,4	533,9	---	---	Abtastung
				100,7	147,1	---	---	Keine Abtastung
				189,2	281,6	---	---	Erstmalige Ausführung des Autotunings
				535,2	709,8	---	---	Autotuning während der Abtastung

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-1-17 Unterprogrammbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
SUBROUTINE CALL	SBS	091	2	1,26	1,96	17,0	17,0	---
SUBROUTINE ENTRY	SBN	092	2	---	---	---	---	---
SUBROUTINE RETURN	RET	093	1	0,86	1,60	20,60	20,60	---
MACRO	MCRO	099	4	23,3	23,3	23,3	23,3	---
GLOBAL SUBROUTINE CALL	GSBN	751	2	---	---	---	---	---
GLOBAL SUBROUTINE ENTRY	GRET	752	1	1,26	1,96	---	---	---
GLOBAL SUBROUTINE RETURN	GSBS	750	2	0,86	1,60	---	---	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-18 Interrupt-Steuerungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
SET INTERRUPT MASK	MSKS	690	3	25,6	38,4	39,5	39,5	---
READ INTERRUPT MASK	MSKR	692	3	11,9	11,9	11,9	11,9	---
CLEAR INTERRUPT	CLI	691	3	27,4	41,3	41,3	41,3	---
DISABLE INTERRUPTS	DI	693	1	15,0	16,8	16,8	16,8	---
ENABLE INTERRUPTS	EI	694	1	19,5	21,8	21,8	21,8	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-19 Schritt-Befehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
STEP DEFINE	STEP	008	2	17,4	20,7	27,1	27,1	Schritt-Steuerbit EIN
				11,8	13,7	24,4	24,4	Schritt-Steuerbit AUS
STEP START	SNXT	009	2	6,6	7,3	10,0	10,0	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-20 Befehle für E/A-Baugruppen

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
I/O REFRESH	IORF	097	3	58,5	63,2	81,7	81,7	Aktualisierung von 1 Eingangswort bei C200H E/A-Baugruppen
				62,6	67,0	86,7	86,7	Aktualisierung von 1 Ausgangswort bei C200H E/A-Baugruppen
				15,5	16,4	23,5	23,5	Aktualisierung von 1 Eingangswort bei CS-Serie E/A-Baugruppen
				17,20	18,40	25,6	25,6	Aktualisierung von 1 Ausgangswort bei CS-Serie E/A-Baugruppen
				303,3	343,9	357,1	357,1	Aktualisierung von 10 Eingangsworten bei C200H E/A-Baugruppen
				348,2	376,6	407,5	407,5	Aktualisierung von 10 Ausgangsworten bei C200H E/A-Baugruppen
				319,9	320,7	377,5	377,6	Aktualisierung von 60 Eingangsworten bei CS-Serie E/A-Baugruppen
				358,00	354,40	460,1	460,1	Aktualisierung von 60 Ausgangsworten bei CS-Serie E/A-Baugruppen
7-SEGMENT DECODER	SDEC	078	4	6,5	6,9	14,1	14,1	---
DIGITAL SWITCH INPUT (siehe Hinweis 2)	DSW	210	6	50,7	73,5	---	---	4 Stellen, Dateneingabewert: 0
				51,5	73,4	---	---	4 Stellen, Dateneingabewert: F
				51,3	73,5	---	---	8 Stellen, Dateneingabewert: 0
				50,7	73,4	---	---	8 Stellen, Dateneingabewert: F
TEN KEY INPUT (siehe Hinweis 2)	TKY	211	4	9,7	13,2	---	---	Dateneingabewert: 0
				10,7	14,8	---	---	Dateneingabewert: F
HEXADECIMAL KEY INPUT (siehe Hinweis 2)	HKY	212	5	50,3	70,9	---	---	Dateneingabewert: 0
				50,1	71,2	---	---	Dateneingabewert: F
MATRIX INPUT (siehe Hinweis 2)	MTR	213	5	47,8	68,1	---	---	Dateneingabewert: 0
				48,0	68,0	---	---	Dateneingabewert: F
7-SEGMENT DISPLAY OUTPUT (siehe Hinweis 2)	7SEG	214	5	58,1	83,3	---	---	4 Stellen
				63,3	90,3	---	---	8 Stellen

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
INTELLIGENT I/O READ	IORD	222	4	Die Ausführungszeiten hängen von der jeweiligen von diesem Befehl angesprochenen Spezial-E/A-Baugruppe ab.				---
INTELLIGENT I/O WRITE	IOWR	223	4					---
CPU BUS I/O REFRESH	DLNK	226	4	287,8	315,5	---	---	1 zugeordnetes Wort

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-1-21 Befehle für serielle Kommunikation

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
PROTOCOL MACRO	PMCR	260	5	100,1	142,1	276,8	276,8	0 Worte senden, 0 Worte empfangen
				134,2	189,6	305,9	305,9	249 Worte senden, 249 Worte empfangen
TRANSMIT	TXD	236	4	68,5	98,8	98,8	98,8	1 Byte senden
				734,3	1,10 ms	1,10 ms	1,10 ms	256 Bytes senden
RECEIVE	RXD	235	4	89,6	131,1	131,1	131,1	1 Byte empfangen
				724,2	1,11 ms	1,11 ms	1,11 ms	256 Bytes empfangen
TRANSMIT VIA SERIAL COMMUNICATIONS UNIT	TXDU	256	4	131,5	202,4	---	---	1 Byte senden
RECEIVE VIA SERIAL COMMUNICATIONS UNIT	RXDU	255	4	131	200,8	---	---	1 Byte empfangen
CHANGE SERIAL PORT SETUP	STUP	237	3	341,2	400,0	440,4	440,4	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-22 Netzwerkbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
NETWORK SEND	SEND	090	4	84,4	123,9	123,9	123,9	---
NETWORK RECEIVE	RECV	098	4	85,4	124,7	124,7	124,7	---
DELIVER COMMAND	CMND	490	4	106,8	136,8	136,8	136,8	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
EXPLICIT MESSAGE SEND (siehe Hinweis 2)	EXPLT	720	4	127,6	190,0	---	---	---
EXPLICIT GET ATTRIBUTE (siehe Hinweis 2)	EGATR	721	4	123,9	185,0	---	---	---
EXPLICIT SET ATTRIBUTE (siehe Hinweis 2)	ESATR	722	3	110,0	164,4	---	---	---
EXPLICIT WORD READ (siehe Hinweis 2)	ECHRD	723	4	106,8	158,9	---	---	---
EXPLICIT WORD WRITE (siehe Hinweis 2)	ECHWR	724	4	106,0	158,3	---	---	---

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-1-23 Dateispeicherbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
READ DATA FILE	FREAD	700	5	391,4	632,4	684,1	684,1	Binär dargestellter Ordner- und Dateiname von jeweils 2 Zeichen Länge
				836,1	1,33 ms	1,35 ms	1,35 ms	Binär dargestellter Ordner- und Dateiname von jeweils 73 Zeichen Länge
WRITE DATA FILE	FWRIT	701	5	387,8	627,0	684,7	684,7	Binär dargestellter Ordner- und Dateiname von jeweils 2 Zeichen Länge
				833,3	1,32 ms	1,36 ms	1,36 ms	Binär dargestellter Ordner- und Dateiname von jeweils 73 Zeichen Länge

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-24 Anzeigebefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
DISPLAY MESSAGE	MSG	046	3	10,1	14,2	14,3	14,3	Meldung anzeigen
				8,4	11,3	11,3	11,3	Angezeigte Meldung löschen

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-25 Uhrfunktionsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
CALENDAR ADD	CADD	730	4	38,3	201,9	209,5	209,5	---
CALENDAR SUBTRACT	CSUB	731	4	38,6	170,4	184,1	184,1	---
HOURS TO SECONDS	SEC	065	3	21,4	29,3	35,8	35,8	---
SECONDS TO HOURS	HMS	066	3	22,2	30,9	42,1	42,1	---
CLOCK ADJUSTMENT	DATE	735	2	60,5	87,4	95,9	95,9	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-26 Fehlerbehebungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
Aufzeichnungsspeicher-Erfassung	TRSM	045	1	80,4	120,0	120,0	120,0	Erfassung von einem Bit und 0 Worten
				848,1	1,06 ms	1,06 ms	1,06 ms	Erfassen von 31 Bits und 6 Worten

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-27 Fehlerdiagnosebefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
FAILURE ALARM	FAL	006	3	15,4	16,7	16,7	16,7	Fehler aufzeichnen
				179,8	244,8	244,8	244,8	Fehler löschen (in Reihenfolge der Priorität)
				432,4	657,1	657,1	657,1	Fehler löschen (alle Fehler)
				161,5	219,4	219,4	219,4	Fehler löschen (einzelne Fehler)
SEVERE FAILURE ALARM	FALS	007	3	---	---	---	---	---
FAILURE POINT DETECTION	FPD	269	4	140,9	202,3	202,3	202,3	Bei Ausführung
				163,4	217,6	217,6	217,6	Erstmalige Ausführung
				185,2	268,9	268,9	268,9	Bei Ausführung
				207,5	283,6	283,6	283,6	Erstmalige Ausführung

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-28 Sonstige Befehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
SET CARRY	STC	040	1	0,06	0,06	0,12	0,12	---
CLEAR CARRY	CLC	041	1	0,06	0,06	0,12	0,12	---
SELECT EM BANK	EMBC	281	2	14,0	15,1	15,1	15,1	---
EXTEND MAXIMUM CYCLE TIME	WDT	094	2	15,0	19,7	19,7	19,7	---
SAVE CONDITION FLAGS	CCS	282	1	8,6	12,5	---	---	---
LOAD CONDITION FLAGS	CCL	283	1	9,8	13,9	---	---	---
CONVERT ADDRESS FROM CV	FRMCV	284	3	13,6	19,9	---	---	---
CONVERT ADDRESS TO CV	TOCV	285	3	11,9	17,2	---	---	---
DISABLE PERIPHERAL SERVICING	IOSP	287	---	13,9	19,8	---	---	---
ENABLE PERIPHERAL SERVICING	IORS	288	---	63,6	92,3	---	---	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-29 Blockprogrammierungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
BLOCK PROGRAM BEGIN	BPRG	096	2	12,1	13,0	13,0	13,0	---
BLOCK PROGRAM END	BEND	801	1	9,6	12,3	13,1	13,1	---
BLOCK PROGRAM PAUSE	BPPS	811	2	10,6	12,3	14,9	14,9	---
BLOCK PROGRAM RESTART	BPRS	812	2	5,1	5,6	8,3	8,3	---
CONDITIONAL BLOCK EXIT	(Ausführungsbedingung) EXIT	806	1	10,0	11,3	12,9	12,9	EXIT-Bedingung erfüllt
				4,0	4,9	7,3	7,3	EXIT-Bedingung nicht erfüllt

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
CONDITIONAL BLOCK EXIT	EXIT (Bitadresse)	806	2	6,8	13,5	16,3	16,3	EXIT-Bedingung erfüllt
				4,7	7,2	10,7	10,7	EXIT-Bedingung nicht erfüllt
CONDITIONAL BLOCK EXIT (NOT)	EXIT NOT (Bitadresse)	806	2	12,4	14,0	16,8	16,8	EXIT-Bedingung erfüllt
				7,1	7,6	11,2	11,2	EXIT-Bedingung nicht erfüllt
Verzweigung	IF (Ausführungsbedingung)	802	1	4,6	4,8	7,2	7,2	IF-Bedingung erfüllt
				6,7	7,3	10,9	10,9	IF-Bedingung nicht erfüllt
Verzweigung	IF (Relaisnummer)	802	2	6,8	7,2	10,4	10,4	IF-Bedingung erfüllt
				9,0	9,6	14,2	14,2	IF-Bedingung nicht erfüllt
Verzweigung (NOT)	IF NOT (Relaisnummer)	802	2	7,1	7,6	10,9	10,9	IF-Bedingung erfüllt
				9,2	10,1	14,7	14,7	IF-Bedingung nicht erfüllt
Verzweigung	ELSE	803	1	6,2	6,7	9,9	9,9	IF-Bedingung erfüllt
				6,8	7,7	11,2	11,2	IF-Bedingung nicht erfüllt
Verzweigung	IEND	804	1	6,9	7,7	11,0	11,0	IF-Bedingung erfüllt
				4,4	4,6	7,0	7,0	IF-Bedingung nicht erfüllt
ONE CYCLE AND WAIT	WAIT (Ausführungsbedingung)	805	1	12,6	13,7	16,7	16,7	WAIT-Bedingung erfüllt
				3,9	4,1	6,3	6,3	WAIT-Bedingung nicht erfüllt
ONE CYCLE AND WAIT	WAIT (Relaisnummer)	805	2	12,0	13,4	16,5	16,5	WAIT-Bedingung erfüllt
				6,1	6,5	9,6	9,6	WAIT-Bedingung nicht erfüllt
ONE CYCLE AND WAIT (NOT)	WAIT NOT (Relaisnummer)	805	2	12,2	13,8	17,0	17,0	WAIT-Bedingung erfüllt
				6,4	6,9	10,1	10,1	WAIT-Bedingung nicht erfüllt
COUNTER WAIT	CNTW	814	4	17,9	22,6	27,4	27,4	Standardeinstellung
				19,1	23,9	28,7	28,7	Normale Ausführung
	CNTWX	818	4	17,9	22,6	---	---	Standardeinstellung
				19,1	23,9	---	---	Normale Ausführung
HIGH-SPEED TIMER WAIT	TMHW	815	3	25,8	27,9	34,1	34,1	Standardeinstellung
				20,6	22,7	28,9	28,9	Normale Ausführung
	TMHWX	817	3	25,8	27,9	---	---	Standardeinstellung
				20,6	22,7	---	---	Normale Ausführung
				9,3	10,8	---	---	LEND-Bedingung nicht erfüllt
Schleifensteuerung	LOOP	809	1	7,9	9,1	12,3	12,3	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
Schleifen-steuerung	LEND (Aus-führungsbe-dingung)	810	1	7,7	8,4	10,9	10,9	LEND-Bedingung erfüllt
				6,8	8,0	9,8	9,8	LEND-Bedingung nicht erfüllt
Schleifen-steuerung	LEND (Relaisnum-mer)	810	2	9,9	10,7	14,4	14,4	LEND-Bedingung erfüllt
				8,9	10,3	13,0	13,0	LEND-Bedingung nicht erfüllt
Schleifen-steuerung	LEND NOT (Relaisnum-mer)	810	2	10,2	11,2	14,8	14,8	LEND-Bedingung erfüllt
				9,3	10,8	13,5	13,5	LEND-Bedingung nicht erfüllt
TIMER WAIT	TIMW	813	3	22,3	25,2	33,1	33,1	Standardeinstellung
				24,9	27,8	35,7	35,7	Normale Ausfüh-rung
	TIMWX	816	3	22,3	25,2	---	---	Standardeinstellung
				24,9	27,8	---	---	Normale Ausfüh-rung

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-30 Textzeichenketten-Verarbeitungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
MOV STRING	MOV$	664	3	45,6	66,0	84,3	84,3	1 Zeichen übertragen
CONCATEN-ATE STRING	+$	656	4	86,5	126,0	167,8	167,8	1 Zeichen + 1 Zeichen
GET STRING LEFT	LEFT$	652	4	53,0	77,4	94,3	94,3	1 Zeichen aus einer Zeichenkette von 2 Zeichen
GET STRING RIGHT	RGHT$	653	4	52,2	76,3	94,2	94,2	1 Zeichen aus einer Zeichenkette von 2 Zeichen
GET STRING MIDDLE	MID$	654	5	56,5	84,6	230,2	230,2	1 Zeichen aus einer Zeichenkette von 3 Zeichen
FIND IN STRING	FIND$	660	4	51,4	77,5	94,1	94,1	Suche nach einem Zeichen in einer Zeichenkette von 2 Zeichen
STRING LENGTH	LEN$	650	3	19,8	28,9	33,4	33,4	Zeichenkette von 1 Zeichen
REPLACE IN STRING	RPLC$	661	6	175,1	258,7	479,5	479,5	Erstes von 2 Zeichen durch 1 Zeichen ersetzen
DELETE STRING	DEL$	658	5	63,4	94,2	244,6	244,6	Erstes von 2 Zeichen löschen
EXCHANGE STRING	XCHG$	665	3	60,6	87,2	99,0	99,0	1 Zeichen durch 1 Zeichen ersetzen
CLEAR STRING	CLR$	666	2	23,8	36,0	37,8	37,8	1 Zeichen löschen

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
INSERT INTO STRING	INS$	657	5	136,5	200,6	428,9	428,9	1 Zeichen nach dem ersten von 2 Zeichen einfügen
Zeichenkettenvergleichsbefehle	LD, AND, OR +=$	670	4	48,5	69,8	86,2	86,2	Vergleich von zwei Zeichenketten mit je 1 Zeichen
	LD, AND, OR +<>$	671						
	LD, AND, OR +<$	672						
	LD, AND, OR +>$	674						
	LD, AND, OR +>=$	675						

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-1-31 Task-Steuerbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
TASK ON	TKON	820	2	19,5	26,3	26,3	26,3	---
TASK OFF	TKOF	821	2	13,3	19,0	26,3	26,3	---

4-1-32 Befehle zur Modellkonvertierung (nur CPU-Baugruppen ab Ver. 3.0)

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
BLOCK TRANSFER	XFERC	565	4	6,4	6,5	---	---	Übertragung von 1 Wort
				481,6	791,6	---	---	Übertragung von 1.000 Worten
SINGLE WORD DISTRIBUTE	DISTC	566	4	3,4	3,5	---	---	Datenverteilung
				5,9	7,3	---	---	Stapelverarbeitung
DATA COLLECT	COLLC	567	4	3,5	3,85	---	---	Datenverteilung
				8	9,1	---	---	Stapelverarbeitung
				8,3	9,6	---	---	Stapelverarbeitung 1 Wort lesen (FIFO)
				2.052,3	2.097,5	---	---	Stapelverarbeitung 1.000 Worte lesen (FIFO)
MOVE BIT	MOVBC	568	4	4,5	4,88	---	---	---
BIT COUNTER	BCNTC	621	4	4,9	5	---	---	Zählen von 1 Wort
				1.252,4	1.284,4	---	---	1.000 Worte zählen

4-1-33 Spezielle Funktionsblockbefehle (nur CPU-Baugruppen ab Ver. 3.0)

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)				Bedingungen
				CPU-6□H	CPU-4□H	CPU-6□	CPU-4□	
GET VARIABLE ID	GETID	286	4	14	22,2	---	---	---

Richtlinien für die Programmkapazität nach Konvertierung der für frühere SPS-Systeme geschriebenen Programme

In der folgenden Tabelle sind Richtwerte für die Konvertierung der Programmkapazität (Einheit: Worte) von Programmen, die für frühere OMRON SPS-Systeme (SYSMAC C200HX/HG/HE, CVM1 oder CV-Serie) geschrieben wurden, in die Programmkapazität (Einheit: Steps) der CS-Serie SPS.

Addieren Sie den folgenden Wert (n) zu der Programmkapazität (Einheit: Worte) für jeden Befehl des für frühere SPS-Systeme geschriebenen Programms, um die Programmkapazität (Einheit: Steps) für SPS-Systeme der CS-Serie zu erhalten.

Steps der CJ-Serie = "a" (Worte) der früheren SPS + n			
Befehle	**Variationen**	**Wert für n bei Konvertierung von C200HX/HG/HE nach CS-Serie**	**Wert für n bei Konvertierung von CV-Serie oder CVM1 nach CS-Serie**
Basisbefehle	Kein	OUT, SET, RSET oder KEEP(011): –1 Sonstige Befehle: 0	0
	Ausführung bei steigender Flanke	Kein	+1
	Direkte Aktualisierung	Kein	0
	Überwachung auf steigende Flanke und direkte Aktualisierung	Kein	+2
Spezialbefehle	Kein	0	–1
	Ausführung bei steigender Flanke	+1	0
	Direkte Aktualisierung	Kein	+3
	Überwachung auf steigende Flanke und direkte Aktualisierung	Kein	+4

Beispielsweise beträgt die Programmkapazität eines C200HX/HG/HE-Programms für jede Verwendung des Befehls OUT mit einer Adresse zwischen CIO 000000 und CIO 25515 zwei Worte pro Befehl. Zur Bestimmung der Programmkapazität (in Steps) bei der CS-Serie ist die Programmkapazität um 1 zu vermindern (2 - 1), beträgt also 1 Step pro Befehl.

Ein weiteres Beispiel: Bei Konvertierung eines CV-Serie-Programms ist für jede Verwendung des Befehls !MOV (MOVE mit direkter Aktualisierung, 4 Worte) die Programmkapazität (in Worten) für die CS-Serie um 3 zu erhöhen, beträgt also 7 (4 + 3) Steps.

4-2 Ausführungszeit und Step-Anzahl der Befehle für die CJ-Serie

In den nachstehenden Tabellen finden Sie die Ausführungszeit und Anzahl der Steps aller Befehle der CJ-Serie CPU-Baugruppen.

Bei der Berechnung der Zykluszeit wird die Programmausführungszeit als Gesamtausführungsdauer aller Befehle des Anwenderprogramms ermittelt (siehe Hinweis).

Hinweis Programme sind Tasks zugeordnet, die als zyklische Tasks oder als Interrupt-Tasks ausgeführt werden.

Die Ausführungszeiten der meisten Befehle unterscheiden sich je nach CPU-Baugruppe (CJ1H-CPU6□H, CJ1H-CPU4□H, CJ1M-CPU□□ und CJ1G-CPU4□) und bestimmten Bedingungen bei der Ausführung des Befehls (Umfang der betroffenen Daten, bestimmte Standardeinstellungen). Wenn in den folgenden Tabellen zu einem Befehl mehrere Zeilen mit Ausführungszeiten gehören, gibt die erste Zeile die minimale Ausführungszeit für diesen Befehl sowie die hierfür angenommenen Bedingungen an, die letzte Zeile entsprechend die maximale Ausführungszeit sowie die hierfür angenommenen Bedingungen.

Nicht erfüllte (AUS) Ausführungsbedingungen tragen ebenfalls zu Abweichungen in der Ausführungszeit bei.

In der Spalte *Länge (Steps)* der folgenden Tabellen wird zudem die Anzahl der Steps der einzelnen Befehle angegeben. Je nach Befehl und Operanden belegt ein CJ-Serie Befehl zwischen ein und sieben Steps im Anwenderprogramm. Die Anzahl der Steps in einem Programm entspricht nicht der Anzahl der Befehle.

Hinweis

1. Anders als bei früheren SPS-Systemen von OMRON, bei denen die Programmkapazität in Worten angegeben wurde, erfolgt bei CJ-Serie SPS-Systemen die Angabe der Programmkapazität in Steps. Einfach gesagt heißt das, 1 Step entspricht 1 Wort. Der für jeden Befehl erforderliche Speicherplatz ist bei einigen Befehlen der CJ-Serie dennoch abweichend und es treten Fehler auf, wenn die Kapazität eines Anwenderprogramms eines anderen SPS-Systems für eine SPS der CJ-Serie auf der Grundlage, dass 1 Wort gleich 1 Step ist, konvertiert wird. Am Ende dieses Kapitels finden Sie im Abschnitt *4-1 Ausführungszeit und Step-Anzahl der Befehle für die CS-Serie* detaillierte Anleitungen zur Konvertierung der Kapazität eines Programms, das für frühere OMRON SPS-Systeme geschrieben wurde.
2. Die meisten Befehle können mit einer Flankenüberwachung (erkennbar an den Symbolen ↑, ↓, @ oder %) kombiniert werden. Dies verlängert die Ausführungszeit des Befehls um den in der nachstehenden Tabelle aufgeführten Wert.

Symbol	CJ1-H		CJ1M	CJ1
	CPU6□H	CPU4□H	CPU□□	CPU4□
↑ oder ↓	+0,24 μs	+0,32 μs	+0,5 μs	+0,45 μs
@ oder %	+0,24 μs	+0,32 μs	+0,5 μs	+0,33 μs

3. Verwenden Sie folgende Zeiten als Richtlinien, wenn Befehle nicht

CJ1-H		CJ1M	CJ1
CPU6□H	CPU4□H	CPU□□	CPU4□
ca. 0,1 μs	ca. 0,2 μs	ca. 0,2 bis 0,5 μs	ca. 0,2 bis 0,4 μs

ausgeführt werden.

4-2-1 Sequenzeingabebefehle

Befehl	AWL	Code	Länge (Steps)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□H	CPU4□H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
LOAD	LD	---	1	0,02	0,04	0,08	0,10	0,10	---
	!LD	---	2	+21,14	+21,16	+21,16	+24,10	+28,07	Erhöhung für direkte Aktualisierung
LOAD NOT	LD NOT	---	1	0,02	0,04	008	0,10	0,10	---
	!LD NOT	---	2	+21,14	+21,16	+21,16	+24,10	+28,07	Erhöhung für direkte Aktualisierung

Befehl	AWL	Code	Länge (Steps)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□H	CPU4□H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
AND	AND	---	1	0,02	0,04	0,08	0,10	0,10	---
	!AND	---	2	+21,14	+21,16	+21,16	+24,10	+28,07	Erhöhung für direkte Aktualisierung
AND NOT	AND NOT	---	1	0,02	0,04	0,08	0,10	0,10	---
	!AND NOT	---	2	+21,14	+21,16	+21,16	+24,10	+28,07	Erhöhung für direkte Aktualisierung
OR	OR	---	1	0,02	0,04	0,08	0,10	0,10	---
	!OR	---	2	+21,14	+21,16	+21,16	+24,10	+28,07	Erhöhung für direkte Aktualisierung
OR NOT	OR NOT	---	1	0,02	0,04	0,08	0,10	0,10	---
	!OR NOT	---	2	+21,14	+21,16	+21,16	+24,10	+28,07	Erhöhung für direkte Aktualisierung
AND LOAD	AND LD	---	1	0,02	0,04	0,08	0,05	0,05	---
OR LOAD	OR LD	---	1	0,02	0,04	0,08	0,05	0,05	---
NOT	NOT	520	1	0,02	0,04	0,08	0,05	0,05	---
CONDITION ON	UP	521	3	0,3	0,42	0,54	0,50	0,50	---
CONDITION OFF	DOWN	522	4	0,3	0,42	0,54	0,50	0,50	---
LOAD BIT TEST	LD TST	350	4	0,14	0,24	0,37	0,35	0,35	---
LOAD BIT TEST NOT	LD TSTN	351	4	0,14	0,24	0,37	0,35	0,35	---
AND BIT TEST NOT	AND TSTN	351	4	0,14	0,24	0,37	0,35	0,35	---
OR BIT TEST	OR TST	350	4	0,14	0,24	0,37	0,35	0,35	---
OR BIT TEST NOT	OR TSTN	351	4	0,14	0,24	0,37	0,35	0,35	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle der folgenden Tabelle (Spalte Länge) um 1.

4-2-2 Sequenzausgabebefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□H	CPU4□H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
OUTPUT	OUT	---	1	0,02	0,04	0,21	0,35	0,35	---
	!OUT	---	2	+21,37	+21,37	+21,37	+23,07	+28,60	Erhöhung für direkte Aktualisierung
OUTPUT NOT	OUT NOT	---	1	0,02	0,04	0,21	0,35	0,35	---
	!OUT NOT	---	2	+21,37	+21,37	+21,37	+23,07	+28,60	Erhöhung für direkte Aktualisierung
KEEP	KEEP	11	1	0,06	0,08	0,29	0,40	0,40	---
DIFFERENTIATE UP	DIFU	13	2	0,24	0,40	0,54	0,50	0,50	---
DIFFERENTIATE DOWN	DIFD	14	2	0,24	0,40	0,54	0,50	0,50	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
SET	SET	---	1	0,02	0,06	0,21	0,30	0,30	---
	!SET	---	2	+21,37	+21,37	+21,37	+23,17	+28,60	Erhöhung für direkte Aktualisierung
RESET	RSET	---	1	0,02	0,06	0,21	0,30	0,30	Bei Spezifikation des Worts
	!RSET	---	2	+21,37	+21,37	+21,37	+23,17	+28,60	Erhöhung für direkte Aktualisierung
MULTIPLE BIT SET	SETA	530	4	5,8	6,1	7,8	11,8	11,8	Setzen von 1 Bit
				25,7	27,2	38,8	64,1	64,1	Setzen von 1.000 Bits
MULTIPLE BIT RESET	RSTA	531	4	5,7	6,1	7,8	11,8	11,8	Rücksetzen von 1 Bit
				25,8	27,1	38,8	64,0	64,0	Rücksetzen von 1.000 Bits
SINGLE BIT SET	SETB	532	2	0,24	0,34	---	0,5	0,5	---
	!SETB		3	+21,44	+21,54	---	+23,31	+23,31	---
SINGLE BIT RESET	RSTB	533	2	0,24	0,34	---	0,5	0,5	---
	!RSTB		3	+21,44	+21,54	---	+23,31	+23,31	---
SINGLE BIT OUTPUT	OUTB	534	2	0,22	0,32	---	0,45	0,45	---
	!OUTB		3	+21,42	+21,52	---	+23,22	+23,22	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-3 Sequenzsteuerungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
END	END	1	1	5,5	6,0	4,0	7,9	7,9	---
NO OPERATION	NOP	0	1	0,02	0,04	0,12	0,05	0,05	---
INTERLOCK	IL	2	1	0,06	0,06	0,12	0,15	0,15	---
INTERLOCK CLEAR	ILC	3	1	0,06	0,06	0,12	0,15	0,15	---
MULTI-INTERLOCK DIFFERENTIATION HOLD (siehe Hinweis 2)	MILH	517	3	6,1	6,5	---	10,3	11,7	Während der Sperre
				7,5	7,9	---	13,3	14,6	Außerhalb der Sperre bei nicht gesetzter Sperre
				8,9	9,7	---	16,6	18,3	Außerhalb der Sperre bei gesetzter Sperre
MULTI-INTERLOCK DIFFERENTIATION RELEASE (siehe Hinweis 2)	MILR	518	3	6,1	6,5	---	10,3	11,7	Während der Sperre
				7,5	7,9	---	13,3	14,6	Außerhalb der Sperre bei nicht gesetzter Sperre
				8,9	9,7	---	16,6	18,3	Außerhalb der Sperre bei gesetzter Sperre
MULTI-INTERLOCK CLEAR (siehe Hinweis 2)	MILC	519	2	5,0	5,6	---	8,3	12,5	Sperre nicht aufgehoben
				5,7	6,2	---	9,6	14,2	Sperre aufgehoben
JUMP	JMP	4	2	0,38	0,48	8,1	0,95	0,95	---
JUMP END	JME	5	2	---	---	---	---	---	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
CONDITIONAL JUMP	CJP	510	2	0,38	0,48	7,4	0,95	0,95	Wenn die Sprungbedingung erfüllt ist
CONDITIONAL JUMP NOT	CJPN	511	2	0,38	0,48	8,5	0,95	0,95	Wenn die Sprungbedingung erfüllt ist
MULTIPLE JUMP	JMP0	515	1	0,06	0,06	0,12	0,15	0,15	---
MULTIPLE JUMP END	JME0	516	1	0,06	0,06	0,12	0,15	0,15	---
FOR LOOP	FOR	512	2	0,21	0,21	0,21	1,00	1,00	Spezifikation einer Konstanten
BREAK LOOP	BREAK	514	1	0,12	0,12	0,12	0,15	0,15	---
NEXT LOOP	NEXT	513	1	0,17	0,17	0,17	0,45	0,45	Wenn die Schleife fortgesetzt wird
				0,12	0,12	0,12	0,55	0,55	Wenn die Schleife beendet wird

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-2-4 Zeitgeber- und Zählerbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
TIMER	TIM	---	3	0,56	0,88	0,42	1,30	1,30	---
	TIMX	550				---		---	
COUNTER	CNT	---	3	0,56	0,88	0,42	1,30	1,30	---
	CNTX	546				---		---	
HIGH-SPEED TIMER	TIMH	15	3	0,88	1,14	0,42	1,80	1,80	---
	TIMHX	551				---		---	
ONE-MS TIMER	TMHH	540	3	0,86	1,12	0,42	1,75	1,75	---
	TMHHX	552				---		---	
ACCUMULATIVE TIMER	TTIM	87	3	16,1	17,0	21,4	27,4	30,9	---
				10,9	11,4	14,8	19,0	21,2	Bei Rücksetzung
				8,5	8,7	10,7	15,0	16,6	Bei Sperrung
	TTIMX	555		16,1	17,0	---	27,4	---	---
				10,9	11,4	---	19,0	---	Bei Rücksetzung
				8,5	8,7	---	15,0	---	Bei Sperrung
LONG TIMER	TIML	542	4	7,6	10,0	12,8	16,3	17,2	---
				6,2	6,5	7,8	13,8	15,3	Bei Sperrung
	TIMLX	553		7,6	10,0	---	16,3	---	---
				6,2	6,5	---	13,8	---	Bei Sperrung

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
MULTI-OUTPUT TIMER	MTIM	543	4	20,9	23,3	26,0	38,55	43,3	---
				5,6	5,8	7,8	12,9	13,73	Bei Rücksetzung
	MTIMX	554		20,9	23,3	---	38,55	---	---
				5,6	5,8	---	12,9	---	Bei Rücksetzung
REVERSIBLE COUNTER	CNTR	12	3	16,9	19,0	20,9	31,8	27,2	---
	CNTRX	548				---		---	
RESET TIMER/ COUNTER	CNR	545	3	9,9	10,6	13,9	14,7	17,93	Rücksetzen eines Worts
				4,16 ms	4,16 ms	5,42 ms	6,21 ms	6,30 ms	Rücksetzen von 1.000 Worten
	CNRX	547		9,9	10,6	---	14,7	17,93	Rücksetzen eines Worts
				4,16 ms	4,16 ms	---	6,21 ms	6,30 ms	Rücksetzen von 1.000 Worten

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-5 Vergleichsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
Symbolvergleichsbefehle (vorzeichenlos)	LD, AND, OR +=	300	4	0,10	0,16	0,37	0,35	0,35	---
	LD, AND, OR + <>	305							
	LD, AND, OR + <	310							
	LD, AND, OR +<=	315							
	LD, AND, OR + >	320							
	LD, AND, OR +>=	325							
Symbolvergleichsbefehle (Doppelwort, vorzeichenlos)	LD, AND, OR +=+L	301	4	0,10	0,16	0,54	0,35	0,35	---
	LD, AND, OR +<>+L	306							---
	LD, AND, OR +<+L	311							---
	LD, AND, OR +<=+L	316							---
	LD, AND, OR +>+L	321							---
	LD, AND, OR +>=+L	326							---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
Symbolvergleichsbefehle (vorzeichenbehaftet)	LD, AND, OR +=+S	302	4	0,10	0,16	6,50	0,35	0,35	---
	LD, AND, OR +<>+S	307							
	LD, AND, OR +<+S	312							
	LD, AND, OR +<=	317							
	LD, AND, OR +>+S	322							
	LD, AND, OR +>=+S	327							
Symbolvergleichsbefehle (Doppelwort, vorzeichenbehaftet)	LD, AND, OR +=+SL	303	4	0,10	0,16	6,50	0,35	0,35	---
	LD, AND, OR +<>+SL	308							
	LD, AND, OR +<+SL	313							
	LD, AND, OR +<=+SL	318							
	LD, AND, OR +>+SL	323							
	LD, AND, OR +>=+SL	328							
Zeitvergleichsbefehle (siehe Hinweis 2)	LD, AND, OR +DT	341	4	25,1	36,4	---	18,8	39,6	---
	LD, AND, OR +<>DT	342	4	25,2	36,4	---	45,6	40,6	---
	LD, AND, OR +<DT	343	4	25,2	36,4	---	45,6	40,7	---
	LD, AND, OR +<=DT	344	4	25,2	36,4	---	18,8	39,6	---
	LD, AND, OR +>DT	345	4	25,1	36,4	---	45,6	41,1	---
	LD, AND, OR +>=DT	346	4	25,2	36,4	---	18,8	39,6	---
COMPARE	CMP	20	3	0,04	0,04	0,29	0,10	0,10	---
	!CMP	20	7	42,1	42,1	42,4	+45,2	45,2	Erhöhung für direkte Aktualisierung
DOUBLE COMPARE	CMPL	60	3	0,08	0,08	0,46	0,50	0,50	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
SIGNED BINARY COMPARE	CPS	114	3	0,08	0,08	6,50	0,30	0,30	---
	!CPS	114	7	35,9	35,9	42,4	+45,2	45,2	Erhöhung für direkte Aktualisierung
DOUBLE SIGNED BINARY COMPARE	CPSL	115	3	0,08	0,08	6,50	0,50	0,50	---
TABLE COMPARE	TCMP	85	4	14,0	15,2	21,9	29,77	32,13	---
MULTIPLE COMPARE	MCMP	19	4	20,5	22,8	31,2	45,80	48,67	---
UNSIGNED BLOCK COMPARE	BCMP	68	4	21,5	23,7	32,6	47,93	51,67	---
EXPANDED BLOCK COMPARE	BCMP2	502	4	---	---	---	13,20	19,33	Anzahl der Datenworte: 1
				---	---	---	650,0	754,67	Anzahl der Datenworte: 255
AREA RANGE COMPARE	ZCP	88	3	5,3	5,4	---	11,53	12,43	---
DOUBLE AREA RANGE COMPARE	ZCPL	116	3	5,5	6,7	---	11,28	11,90	---

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-2-6 Datenübertragungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
MOVE	MOV	21	3	0,18	0,20	0,29	0,30	0,30	---
	!MOV	21	7	21,38	21,40	42,36	+35,1	43,0	Erhöhung für direkte Aktualisierung
DOUBLE MOVE	MOVL	498	3	0,32	0,34	0,50	0,60	0,60	---
MOVE NOT	MVN	22	3	0,18	0,20	0,29	0,35	0,35	---
DOUBLE MOVE NOT	MVNL	499	3	0,32	0,34	0,50	0,60	0,60	---
MOVE BIT	MOVB	82	4	0,24	0,34	7,5	0,50	0,50	---
MOVE DIGIT	MOVD	83	4	0,24	0,34	7,3	0,50	0,50	---
MULTIPLE BIT TRANSFER	XFRB	62	4	10,1	10,8	13,6	20,9	22,1	Übertragung von 1 Bit
				186,4	189,8	269,2	253,3	329,7	Übertragung von 255 Bits
BLOCK TRANSFER	XFER	70	4	0,36	0,44	11,2	0,8	0,8	Übertragung von 1 Wort
				300,1	380,1	633,5	650,2	650,2	Übertragung von 1.000 Worten

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
BLOCK SET	BSET	71	4	0,26	0,28	8,5	0,55	0,55	Setzen von 1 Wort
				200,1	220,1	278,3	400,2	400,2	Setzen von 1.000 Worten
DATA EXCHANGE	XCHG	73	3	0,40	0,56	0,7	0,80	0,80	---
DOUBLE DATA EXCHANGE	XCGL	562	3	0,76	1,04	1,3	1,5	1,5	---
SINGLE WORD DISTRIBUTE	DIST	80	4	5,1	5,4	7,0	6,6	12,47	---
DATA COLLECT	COLL	81	4	5,1	5,3	7,1	6,5	12,77	---
MOVE TO REGISTER	MOVR	560	3	0,08	0,08	0,50	0,60	0,60	---
MOVE TIMER/COUNTER PV TO REGISTER	MOVRW	561	3	0,42	0,50	0,50	0,60	0,60	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-7 Datenverschiebungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
SHIFT REGISTER	SFT	10	3	7,4	10,4	10,4	11,9	15,3	Verschieben von 1 Wort
				433,2	488,0	763,1	1,39 ms	1,43 ms	Verschieben von 1.000 Worten
REVERSIBLE SHIFT REGISTER	SFTR	84	4	6,9	7,2	9,6	11,4	15,5	Verschieben von 1 Wort
				615,3	680,2	859,6	1,43 ms	1,55 ms	Verschieben von 1.000 Worten
ASYNCHRONOUS SHIFT REGISTER	ASFT	17	4	6,2	6,4	7,7	13,4	14,2	Verschieben von 1 Wort
				1,22 ms	1,22 ms	2,01 ms	2,75 ms	2,99 ms	Verschieben von 1.000 Worten
WORD SHIFT	WSFT	16	4	4,5	4,7	7,8	9,6	12,3	Verschieben von 1 Wort
				171,5	171,7	781,7	928,0	933,3	Verschieben von 1.000 Worten
ARITHMETIC SHIFT LEFT	ASL	25	2	0,22	0,32	0,37	0,45	0,45	---
DOUBLE SHIFT LEFT	ASLL	570	2	0,40	0,56	0,67	0,80	0,80	---
ARITHMETIC SHIFT RIGHT	ASR	26	2	0,22	0,32	0,37	0,45	0,45	---
DOUBLE SHIFT RIGHT	ASRL	571	2	0,40	0,56	0,67	0,80	0,80	---
ROTATE LEFT	ROL	27	2	0,22	0,32	0,37	0,45	0,45	---
DOUBLE ROTATE LEFT	ROLL	572	2	0,40	0,56	0,67	0,80	0,80	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
ROTATE LEFT WITHOUT CARRY	RLNC	574	2	0,22	0,32	0,37	0,45	0,45	---
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL	576	2	0,40	0,56	0,67	0,80	0,80	---
ROTATE RIGHT	ROR	28	2	0,22	0,32	0,37	0,45	0,45	---
DOUBLE ROTATE RIGHT	RORL	573	2	0,40	0,56	0,67	0,80	0,80	---
ROTATE RIGHT WITHOUT CARRY	RRNC	575	2	0,22	0,32	0,37	0,45	0,45	---
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL	577	2	0,40	0,56	0,67	0,80	0,80	---
ONE DIGIT SHIFT LEFT	SLD	74	3	5,9	6,1	8,2	7,6	12,95	Verschieben von 1 Wort
				561,1	626,3	760,7	1,15 ms	1,27 ms	Verschieben von 1.000 Worten
ONE DIGIT SHIFT RIGHT	SRD	75	3	6,9	7,1	8,7	8,6	15,00	Verschieben von 1 Wort
				760,5	895,5	1,07 ms	1,72 ms	1,82 ms	Verschieben von 1.000 Worten
SHIFT N-BIT DATA LEFT	NSFL	578	4	7,5	8,3	10,5	14,8	16,0	Verschieben von 1 Bit
				40,3	45,4	55,5	86,7	91,3	Verschieben von 1.000 Bits
SHIFT N-BIT DATA RIGHT	NSFR	579	4	7,5	8,3	10,5	14,7	15,9	Verschieben von 1 Bit
				50,5	55,3	69,3	114,1	119,6	Verschieben von 1.000 Bits
SHIFT N-BITS LEFT	NASL	580	3	0,22	0,32	0,37	0,45	0,45	---
DOUBLE SHIFT N-BITS LEFT	NSLL	582	3	0,40	0,56	0,67	0,80	0,80	---
SHIFT N-BITS RIGHT	NASR	581	3	0,22	0,32	0,37	0,45	0,45	---
DOUBLE SHIFT N-BITS RIGHT	NSRL	583	3	0,40	0,56	0,67	0,80	0,80	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-8 Inkrementier-/Dekrementierbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
INCREMENT BINARY	++	590	2	0,22	0,32	0,37	0,45	0,45	---
DOUBLE INCREMENT BINARY	++L	591	2	0,40	0,56	0,67	0,80	0,80	---
DECREMENT BINARY	– –	592	2	0,22	0,32	0,37	0,45	0,45	---
DOUBLE DECREMENT BINARY	– –L	593	2	0,40	0,56	0,67	0,80	0,80	---
INCREMENT BCD	++B	594	2	6,4	4,5	7,4	12,3	14,7	---
DOUBLE INCREMENT BCD	++BL	595	2	5,6	4,9	6,1	9,24	10,8	---
DECREMENT BCD	– –B	596	2	6,3	4,6	7,2	11,9	14,9	---
DOUBLE DECREMENT BCD	– –BL	597	2	5,3	4,7	7,1	9,0	10,7	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-9 Symbolarithmetische Befehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
SIGNED BINARY ADD WITHOUT CARRY	+	400	4	0,18	0,20	0,37	0,30	0,30	---
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	401	4	0,32	0,34	0,54	0,60	0,60	---
SIGNED BINARY ADD WITH CARRY	+C	402	4	0,18	0,20	0,37	0,40	0,40	---
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	403	4	0,32	0,34	0,54	0,60	0,60	---
BCD ADD WITHOUT CARRY	+B	404	4	8,2	8,4	14,0	18,9	21,5	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DOUBLE BCD ADD WITHOUT CARRY	+BL	405	4	13,3	14,5	19,0	24,4	27,7	---
BCD ADD WITH CARRY	+BC	406	4	8,9	9,1	14,5	19,7	22,6	---
DOUBLE BCD ADD WITH CARRY	+BCL	407	4	13,8	15,0	19,6	25,2	28,8	---
SIGNED BINARY SUBTRACT WITHOUT CARRY	–	410	4	0,18	0,20	0,37	0,3	0,3	---
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	–L	411	4	0,32	0,34	0,54	0,60	0,60	---
SIGNED BINARY SUBTRACT WITH CARRY	–C	412	4	0,18	0,20	0,37	0,3	0,3	---
DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	–CL	413	4	0,32	0,34	0,54	0,60	0,60	---
BCD SUBTRACT WITHOUT CARRY	–B	414	4	8,0	8,2	13,1	18,1	20,5	---
DOUBLE BCD SUBTRACT WITHOUT CARRY	–BL	415	4	12,8	14,0	18,2	23,2	26,7	---
BCD SUBTRACT WITH CARRY	–BC	416	4	8,5	8,6	13,8	19,1	21,6	---
DOUBLE BCD SUBTRACT WITH CARRY	–BCL	417	4	13,4	14,7	18,8	24,3	27,7	---
SIGNED BINARY MULTIPLY	*	420	4	0,38	0,40	0,58	0,65	0,65	---
DOUBLE SIGNED BINARY MULTIPLY	*L	421	4	7,23	8,45	11,19	13,17	15,0	---
UNSIGNED BINARY MULTIPLY	*U	422	4	0,38	0,40	0,58	0,75	0,75	---
DOUBLE UNSIGNED BINARY MULTIPLY	*UL	423	4	7,1	8,3	10,63	13,30	15,2	---
BCD MULTIPLY	*B	424	4	9,0	9,2	12,8	17,5	19,7	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DOUBLE BCD MULTIPLY	*BL	425	4	23,0	24,2	35,2	36,3	45,7	---
SIGNED BINARY DIVIDE	/	430	4	0,40	0,42	0,83	0,70	0,70	---
DOUBLE SIGNED BINARY DIVIDE	/L	431	4	7,2	8,4	9,8	13,7	15,5	---
UNSIGNED BINARY DIVIDE	/U	432	4	0,40	0,42	0,83	0,8	0,8	---
DOUBLE UNSIGNED BINARY DIVIDE	/UL	433	4	6,9	8,1	9,1	12,8	14,7	---
BCD DIVIDE	/B	434	4	8,6	8,8	15,9	19,3	22,8	---
DOUBLE BCD DIVIDE	/BL	435	4	17,7	18,9	26,2	27,1	34,7	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-10 Konvertierungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
BCD-TO-BINARY	BIN	023	3	0,22	0,24	0,29	0,40	0,40	---
DOUBLE BCD-TO-DOUBLE BINARY	BINL	058	3	6,5	6,8	9,1	12,3	13,7	---
BINARY-TO-BCD	BCD	024	3	0,24	0,26	8,3	7,62	9,78	---
DOUBLE BINARY-TO-DOUBLE BCD	BCDL	059	3	6,7	7,0	9,2	10,6	12,8	---
2'S COMPLEMENT	NEG	160	3	0,18	0,20	0,29	0,35	0,35	---
DOUBLE 2'S COMPLEMENT	NEGL	161	3	0,32	0,34	0,5	0,60	0,60	---
16-BIT TO 32-BIT SIGNED BINARY	SIGN	600	3	0,32	0,34	0,50	0,60	0,60	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DATA DECODER	MLPX	076	4	0,32	0,42	8,8	0,85	0,85	Dekodieren einer Stelle (4 zu 16)
				0,98	1,20	12,8	1,60	1,60	Dekodieren von vier Stellen (4 zu 16)
				3,30	4,00	20,3	4,70	4,70	Dekodieren einer Stelle (8 zu 256)
				6,50	7,90	33,4	8,70	8,70	Dekodieren von zwei Stellen (8 zu 256)
DATA ENCODER	DMPX	077	4	7,5	7,9	10,4	9,4	13,9	Kodieren einer Stelle (16 zu 4)
				49,6	50,2	59,1	57,3	71,73	Kodieren von vier Stellen (16 zu 4)
				18,2	18,6	23,6	56,8	82,7	Kodieren einer Stelle (256 zu 8)
				55,1	57,4	92,5	100,0	150,7	Kodieren von zwei Stellen (256 zu 8)
ASCII CONVERT	ASC	086	4	6,8	7,1	9,7	8,3	14,6	Konvertieren einer Stelle zu ASCII
				11,2	11,7	15,1	19,1	21,8	Konvertieren von vier Stellen zu ASCII
ASCII TO HEX	HEX	162	4	7,1	7,4	10,1	12,1	15,6	Konvertieren einer Stelle
COLUMN TO LINE	LINE	063	4	19,0	23,1	29,1	37,0	40,3	---
LINE TO COLUMN	COLM	064	4	23,2	27,5	37,3	45,7	48,2	---
SIGNED BCD-TO-BINARY	BINS	470	4	8,0	8,3	12,1	16,2	17,0	Datenformateinstellung 0
				8,0	8,3	12,1	16,2	17,1	Datenformateinstellung 1
				8,3	8,6	12,7	16,5	17,7	Datenformateinstellung 2
				8,5	8,8	13,0	16,5	17,6	Datenformateinstellung 3
DOUBLE SIGNED BCD-TO-BINARY	BISL	472	4	9,2	9,6	13,6	18,4	19,6	Datenformateinstellung 0
				9,2	9,6	13,7	18,5	19,8	Datenformateinstellung 1
				9,5	9,9	14,2	18,6	20,1	Datenformateinstellung 2
				9,6	10,0	14,4	18,7	20,1	Datenformateinstellung 3
SIGNED BINARY-TO-BCD	BCDS	471	4	6,6	6,9	10,6	13,5	16,4	Datenformateinstellung 0
				6,7	7,0	10,8	13,8	16,7	Datenformateinstellung 1
				6,8	7,1	10,9	13,9	16,8	Datenformateinstellung 2
				7,2	7,5	11,5	14,0	17,1	Datenformateinstellung 3

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DOUBLE SIGNED BINARY-TO-BCD	BDSL	473	4	8,1	8,4	11,6	11,4	12,5	Datenformateinstellung 0
				8,2	8,6	11,8	11,7	12,73	Datenformateinstellung 1
				8,3	8,7	12,0	11,8	12,8	Datenformateinstellung 2
				8,8	9,2	12,5	11,9	13,0	Datenformateinstellung 3
GRAY CODE CONVERSION (siehe Hinweis 2)	GRY	474	4	46,9	72,1	---	80,0	71,2	8 Bit im Binärformat
				49,6	75,2	---	83,0	75,6	8 Bit im BCD-Format
				57,7	87,7	---	95,9	86,4	8 Bit im Winkelformat
				61,8	96,7	---	104,5	91,6	15 Bit im Binärformat
				64,5	99,6	---	107,5	96,1	15 Bit im BCD-Format
				72,8	112,4	---	120,4	107,3	15 Bit im Winkelformat
				52,3	87,2	---	88,7	82,4	360° Binärwert
				55,1	90,4	---	91,7	86,8	360° BCD-Wert
				64,8	98,5	---	107,3	98,1	360° Winkelwert

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-2-11 Logikbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
LOGICAL AND	ANDW	034	4	0,18	0,20	0,37	0,30	0,30	---
DOUBLE LOGICAL AND	ANDL	610	4	0,32	0,34	0,54	0,60	0,60	---
LOGICAL OR	ORW	035	4	0,22	0,32	0,37	0,45	0,45	---
DOUBLE LOGICAL OR	ORWL	611	4	0,32	0,34	0,54	0,60	0,60	---
EXCLUSIVE OR	XORW	036	4	0,22	0,32	0,37	0,45	0,45	---
DOUBLE EXCLUSIVE OR	XORL	612	4	0,32	0,34	0,54	0,60	0,60	---
EXCLUSIVE NOR	XNRW	037	4	0,22	0,32	0,37	0,45	0,45	---
DOUBLE EXCLUSIVE NOR	XNRL	613	4	0,32	0,34	0,54	0,60	0,60	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
COMPLEMENT	COM	029	2	0,22	0,32	0,37	0,45	0,45	---
DOUBLE COMPLEMENT	COML	614	2	0,40	0,56	0,67	0,80	0,80	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-12 Spezielle arithmetische Befehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
BINARY ROOT	ROTB	620	3	49,6	50,0	530,7	56,5	82,7	---
BCD SQUARE ROOT	ROOT	072	3	13,7	13,9	514,5	59,3	88,4	---
ARITHMETIC PROCESS	APR	069	4	6,7	6,9	32,3	14,0	15,0	Sinus und Cosinus
				17,2	18,4	78,3	32,2	37,9	Geradensegment-Annäherung
FLOATING POINT DIVIDE	FDIV	079	4	116,6	176,6	176,6	246,0	154,7	---
BIT COUNTER	BCNT	067	4	0,3	0,38	22,1	0,65	0,65	Zählen von 1 Wort

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-13 Gleitkomma-Arithmetikbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
FLOATING TO 16-BIT	FIX	450	3	10,6	10,8	14,5	16,2	19,5	---
FLOATING TO 32-BIT	FIXL	451	3	10,8	11,0	14,6	16,6	21,7	---
16-BIT TO FLOATING	FLT	452	3	8,3	8,5	11,1	12,2	14,6	---
32-BIT TO FLOATING	FLTL	453	3	8,3	8,5	10,8	14,0	15,8	---
FLOATING-POINT ADD	+F	454	4	8,0	9,2	10,2	13,3	15,7	---
FLOATING-POINT SUBTRACT	–F	455	4	8,0	9,2	10,3	13,3	15,8	---
FLOATING-POINT DIVIDE	/F	457	4	8,7	9,9	12,0	14,0	17,6	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
FLOATING-POINT MULTIPLY	*F	456	4	8,0	9,2	10,5	13,2	15,8	---
DEGREES TO RADIANS	RAD	458	3	10,1	10,2	14,9	15,9	20,6	---
RADIANS TO DEGREES	DEG	459	3	9,9	10,1	14,8	15,7	20,4	---
SINE	SIN	460	3	42,0	42,2	61,1	47,9	70,9	---
COSINE	COS	461	3	31,5	31,8	44,1	41,8	51,0	---
TANGENT	TAN	462	3	16,3	16,6	22,6	20,8	27,6	---
ARC SINE	ASIN	463	3	17,6	17,9	24,1	80,3	122,9	---
ARC COSINE	ACOS	464	3	20,4	20,7	28,0	25,3	33,5	---
ARC TAN-GENT	ATAN	465	3	16,1	16,4	16,4	45,9	68,9	---
SQUARE ROOT	SQRT	466	3	19,0	19,3	28,1	26,2	33,2	---
EXPONENT	EXP	467	3	65,9	66,2	96,7	68,8	108,2	---
LOGARITHM	LOG	468	3	12,8	13,1	17,4	69,4	103,7	---
EXPONEN-TIAL POWER	PWR	840	4	125,4	126,0	181,7	134,0	201,0	---
Symbolver-gleichsbefehle mit Gleitkom-maoperanden	LD, AND, OR +=F	329	3	6,6	8,3	---	12,6	15,37	---
	LD, AND, OR +<>F	330							
	LD, AND, OR +<F	331							
	LD, AND, OR +<=F	332							
	LD, AND, OR +>F	333							
	LD, AND, OR +>=F	334							
FLOATING-POINT TO ASCII	FSTR	448	4	48,5	48,9	---	58,4	85,7	---
ASCII TO FLOATING-POINT	FVAL	449	3	21,1	21,3	---	31,1	43.773	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-14 Gleitkommaarithmetikbefehle doppelter Genauigkeit

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DOUBLE SYMBOL COMPARISON	LD, AND, OR +=D	335	3	8,5	10,3	---	16,2	19,9	---
	LD, AND, OR +<>D	336							
	LD, AND, OR +<D	337							
	LD, AND, OR +<=D	338							
	LD, AND, OR +>D	339							
	LD, AND, OR +>=D	340							
DOUBLE FLOATING TO 16-BIT BINARY	FIXD	841	3	11,7	12,1	---	16,1	21,6	---
DOUBLE FLOATING TO 32-BIT BINARY	FIXLD	842	3	11,6	12,1	---	16,4	21,7	---
16-BIT BINARY TO DOUBLE FLOATING	DBL	843	3	9,9	10,0	---	14,3	16,5	---
32-BIT BINARY TO DOUBLE FLOATING	DBLL	844	3	9,8	10,0	---	16,0	17,7	---
DOUBLE FLOATING-POINT ADD	+D	845	4	11,2	11,9	---	18,3	23,6	---
DOUBLE FLOATING-POINT SUBTRACT	–D	846	4	11,2	11,9	---	18,3	23,6	---
DOUBLE FLOATING-POINT MULTIPLY	*D	847	4	12,0	12,7	---	19,0	25,0	---
DOUBLE FLOATING-POINT DIVIDE	/D	848	4	23,5	24,2	---	30,5	44,3	---
DOUBLE DEGREES TO RADIANS	RADD	849	3	27,4	27,8	---	32,7	49,1	---
DOUBLE RADIANS TO DEGREES	DEGD	850	3	11,2	11,9	---	33,5	48,4	---
DOUBLE SINE	SIND	851	3	45,4	45,8	---	67,9	76,7	---
DOUBLE COSINE	COSD	852	3	43,0	43,4	---	70,9	72,3	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DOUBLE TANGENT	TAND	853	3	20,1	20,5	---	97,9	157,0	---
DOUBLE ARC SINE	ASIND	854	3	21,5	21,9	---	32,3	37,3	---
DOUBLE ARC COSINE	ACOSD	855	3	24,7	25,1	---	29,9	42,5	---
DOUBLE ARC TANGENT	ATAND	856	3	19,3	19,7	---	24,0	34,4	---
DOUBLE SQUARE ROOT	SQRTD	857	3	47,4	47,9	---	52,9	81,9	---
DOUBLE EXPONENT	EXPD	858	3	121,0	121,4	---	126,3	201,3	---
DOUBLE LOGARITHM	LOGD	859	3	16,0	16,4	---	21,6	29,3	---
DOUBLE EXPONENTIAL POWER	PWRD	860	4	223,9	224,2	---	232,3	373,4	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-15 Tabellendaten-Verarbeitungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
SET STACK	SSET	630	3	8,0	8,3	8,5	14,2	20,3	Spezifikation von 5 Worten im Stapelbereich
				231,6	251,8	276,8	426,5	435,3	Spezifikation von 1.000 Worten im Stapelbereich
PUSH ONTO STACK	PUSH	632	3	6,5	8,6	9,1	15,7	16,4	---
FIRST IN FIRST OUT	FIFO	633	3	6,9	8,9	10,6	15,8	16,8	Spezifikation von 5 Worten im Stapelbereich
				352,6	434,3	1,13 ms	728,0	732,0	Spezifikation von 1.000 Worten im Stapelbereich
LAST IN FIRST OUT	LIFO	634	3	7,0	9,0	9,9	16,6	17,2	---
DIMENSION RECORD TABLE	DIM	631	5	15,2	21,6	142,1	27,8	27,1	---
SET RECORD LOCATION	SETR	635	4	5,4	5,9	7,0	12,8	13,2	---
GET RECORD NUMBER	GETR	636	4	7,8	8,4	11,0	16,1	18,3	---
DATA SEARCH	SRCH	181	4	15,5	19,5	19,5	29,1	26,4	Suche nach einem Wort
				2,42 ms	3,34 ms	3,34 ms	4,41 ms	3,60 ms	Suche nach 1.000 Worten

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
SWAP BYTES	SWAP	637	3	12,2	13,6	13,6	21,0	18,4	Austausch von 1 Wort
				1,94 ms	2,82 ms	2,82 ms	3,65 ms	3,15 ms	Austausch von 1.000 Worten
FIND MAXIMUM	MAX	182	4	19,2	24,9	24,9	35,3	32,0	Suche in einem Wort
				2,39 ms	3,36 ms	3,36 ms	4,39 ms	3,57 ms	Suche in 1.000 Worten
FIND MINIMUM	MIN	183	4	19,2	25,3	25,3	35,4	31,9	Suche in einem Wort
				2,39 ms	3,33 ms	3,33 ms	4,39 ms	3,58 ms	Suche in 1.000 Worten
SUM	SUM	184	4	28,2	38,5	38,3	49,5	44,1	Addition von 1 Wort
				1,42 ms	1,95 ms	1,95 ms	2,33 ms	2,11 ms	Addition von 1.000 Worten
FRAME CHECKSUM	FCS	180	4	20,0	28,3	28,3	34,8	31,5	Tabelle mit 1 Wort
				1,65 ms	2,48 ms	2,48 ms	3,11 ms	2,77 ms	Tabelle mit 1.000 Worten
STACK SIZE READ	SNUM	638	3	6,0	6,3	---	12,1	13,7	---
STACK DATA READ	SREAD	639	4	8,0	8,4	---	18,1	20,6	---
STACK DATA OVERWRITE	SWRIT	640	4	7,2	7,6	---	16,9	18,8	---
STACK DATA INSERT	SINS	641	4	7,8	9,9	---	18,2	20,5	---
				354,0	434,8	---	730,7	732,0	Tabelle mit 1.000 Worten
STACK DATA DELETE	SDEL	642	4	8,6	10,6	---	19,3	22,0	---
				354,0	436,0	---	732,0	744,0	Tabelle mit 1.000 Worten

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-16 Datensteuerungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
PID CONTROL	PID	190	4	436,2	678,2	678,2	612,0	552,6	Erstmalige Ausführung
				332,3	474,9	474,9	609,3	548,0	Abtastung
				97,3	141,3	141,3	175,3	162,0	Keine Abtastung
LIMIT CONTROL	LMT	680	4	16,1	22,1	22,1	27,1	26,1	---
DEAD BAND CONTROL	BAND	681	4	17,0	22,5	22,5	27,4	26,6	---
DEAD ZONE CONTROL	ZONE	682	4	15,4	20,5	20,5	28,0	26,4	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
TIME-PROPORTIONAL OUTPUT (siehe Hinweis 2)	TPO	685	4	10,6	14,8	---	20,2	19,8	AUS-Ausführungszeit
				54,5	82,0	---	92,7	85,1	EIN-Ausführungszeit mit Tastverhältnis-Spezifikation oder angezeigter Ausgabegrenze
				61,0	91,9	---	102,5	95,3	EIN-Ausführungszeit mit Ausgangsstellwert-Spezifikation und aktivierter Ausgabegrenze
SCALING	SCL	194	4	37,1	53,0	56,8	25,0	32,8	---
SCALING 2	SCL2	486	4	28,5	40,2	50,7	22,3	29,1	---
SCALING 3	SCL3	487	4	33,4	47,0	57,7	25,6	30,0	---
AVERAGE	AVG	195	4	36,3	52,6	53,1	62,9	59,1	Mittelwertbildung über einen Zyklus
				291,0	419,9	419,9	545,3	492,7	Mittelwertbildung über 64 Zyklen
PID CONTROL WITH AUTOTUNING	PIDAT	191	4	446,3	712,5	---	765,3	700,0	Erstmalige Ausführung
				339,4	533,9	---	620,7	558,0	Abtastung
				100,7	147,1	---	180,0	166,1	Keine Abtastung
				189,2	281,6	---	233,7	225,1	Erstmalige Ausführung des Autotunings
				535,2	709,8	---	575,3	558,2	Autotuning während der Abtastung

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-2-17 Unterprogrammbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
SUBROUTINE CALL	SBS	91	2	1,26	1,96	17,0	2,04	2,04	---
SUBROUTINE ENTRY	SBN	92	2	---	---	---	---	---	---
SUBROUTINE RETURN	RET	93	1	0,86	1,60	20,60	1,80	1,80	---
MACRO	MCRO	99	4	23,3	23,3	23,3	47,9	50,3	---
GLOBAL SUBROUTINE CALL	GSBN	751	2	---	---	---	---	---	---
GLOBAL SUBROUTINE ENTRY	GRET	752	1	1,26	1,96	---	2,04	2,04	---
GLOBAL SUBROUTINE RETURN	GSBS	750	2	0,86	1,60	---	1,80	1,80	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-18 Interrupt-Steuerungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
SET INTERRUPT MASK	MSKS	690	3	25,6	38,4	39,5	44,7	42,9	---
READ INTERRUPT MASK	MSKR	692	3	11,9	11,9	11,9	16,9	15,9	---
CLEAR INTERRUPT	CLI	691	3	27,4	41,3	41,3	42,7	44,5	---
DISABLE INTERRUPTS	DI	693	1	15,0	16,8	16,8	30,3	28,5	---
ENABLE INTERRUPTS	EI	694	1	19,5	21,8	21,8	37,7	34,4	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-19 Befehle für schnelle Zähler und Impulsausgabe

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
MODE CONTROL	INI	880	4	---	---	---	77,00	80,4	Start des Zählervergleichs
				---	---	---	43,00	43,0	Stopp des Zählervergleichs
				---	---	---	43,40	48,8	Ändern des Impulsausgang-Istwerts
				---	---	---	51,80	50,8	Ändern des Zähler-Istwerts
				---	---	---	31,83	28,5	Ändern des Zähler-Istwerts im Interrupt-Eingangs-Modus
				---	---	---	45,33	49,8	Stoppen der Impulsausgabe
				---	---	---	36,73	30,5	Stoppen der PWM(891)-Ausgabe

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
HIGH-SPEED COUNTER PV READ	PRV	881	4	---	---	---	42,40	43,9	Auslesen des Impulsausgabe-Istwerts
				---	---	---	53,40	65,9	Auslesen des Zähler-Istwerts
				---	---	---	33,60	30,5	Auslesen des Zähler-Istwerts im Interrupt-Eingangs-Modus
				---	---	---	38,80	40,0	Auslesen des Impulsausgabe-Status
				---	---	---	39,30	66,9	Auslesen des Zähler-Status
				---	---	---	38,30	34,5	Auslesen des PWM(891)-Status
				---	---	---	117,73	145,7	Auslesen der Zähler-Bereichvergleichs-ergebnisse
				---	---	---	48,20	48,5	Auslesen der Zählerfrequenz des schnellen Zählers 0
COMPARISON TABLE LOAD	CTBL	882	4	---	---	---	238,0	235,0	Registrieren der Zielwertetabelle und Beginn des Vergleichs mit einem Zielwert
				---	---	---	14,42 ms	9,97 ms	Registrieren der Zielwertetabelle und Beginn des Vergleichs mit 48 Zielwerten
				---	---	---	289,0	276,0	Registrieren der Bereichstabelle und Beginn des Vergleichs
				---	---	---	198,0	183,0	Nur Registrieren der Zielwertetabelle für einen Zielwert (ohne Vergleich)
				---	---	---	14,40 ms	9,61 ms	Nur Registrieren der Zielwertetabelle für 48 Zielwerte (ohne Vergleich)
				---	---	---	259,0	239,0	Nur Registrieren der Bereichstabelle (ohne Vergleich)
COUNTER FREQUENCY CONVERT	PRV2	883	4	---	---	---	23,03	22,39	---
SPEED OUTPUT	SPED	885	4	---	---	---	56,00	89,3	Fortlaufender Modus
				---	---	---	62,47	94,9	Einmaliger Modus
SET PULSES	PULS	886	4	---	---	---	26,20	32,9	---
PULSE OUTPUT	PLS2	887	5	---	---	---	100,80	107,5	---
ACCELERATION CONTROL	ACC	888	4	---	---	---	90,80	114,8	Fortlaufender Modus
				---	---	---	80,00	122,1	Einmaliger Modus

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
ORIGIN SEARCH	ORG	889	3	---	---	---	106,13	116,0	Nullpunktsuche
				---	---	---	52,00	102,1	Nullpunktrückkehr
PULSE WITH VARIABLE DUTY FACTOR	PWM	891	4	---	---	---	25,80	33,0	---

Hinweis Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-2-20 Schritt-Befehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
STEP DEFINE	STEP	008	2	17,4	20,7	27,1	35,9	37,1	Schritt-Steuerbit EIN
				11,8	13,7	24,4	13,8	18,3	Schritt-Steuerbit AUS
STEP START	SNXT	009	2	6,6	7,3	10,0	12,1	14,0	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-21 Befehle für E/A-Baugruppen

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
I/O REFRESH	IORF	097	3	15,5	16,4	23,5	26,7	30,4	1 Eingangswort von E/A-Baugruppen
				17,20	18,40	25,6	29,7	35,0	1 Ausgangswort von E/A-Baugruppen
				319,9	320,7	377,6	291,0	100,0	60 Eingangsworte von E/A-Baugruppen
				358,00	354,40	460,1	325,0	134,7	60 Ausgangsworte von E/A-Baugruppen
7-SEGMENT DECODER	SDEC	078	4	6,5	6,9	14,1	8,1	15,7	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DIGITAL SWITCH INPUT (siehe Hinweis 2)	DSW	210	6	50,7	73,5	---	77,7	77,6	4 Stellen, Dateneingabewert: 0
				51,5	73,4	---	77,9	77,6	4 Stellen, Dateneingabewert: F
				51,3	73,5	---	83,2	80,0	8 Stellen, Dateneingabewert: 0
				50,7	73,4	---	77,9	77,7	8 Stellen, Dateneingabewert: F
TEN KEY INPUT (siehe Hinweis 2)	TKY	211	4	9,7	13,2	---	18,7	18,6	Dateneingabewert: 0
				10,7	14,8	---	20,2	19,1	Dateneingabewert: F
HEXADECIMAL KEY INPUT (siehe Hinweis 2)	HKY	212	5	50,3	70,9	---	77,3	78,1	Dateneingabewert: 0
				50,1	71,2	---	76,8	77,3	Dateneingabewert: F
MATRIX INPUT (siehe Hinweis 2)	MTR	213	5	47,8	68,1	---	76,4	77,7	Dateneingabewert: 0
				48,0	68,0	---	77,7	76,9	Dateneingabewert: F
7-SEGMENT DISPLAY OUTPUT (siehe Hinweis 2)	7SEG	214	5	58,1	83,3	---	89,6	89,9	4 Stellen
				63,3	90,3	---	98,3	99,2	8 Stellen
INTELLIGENT I/O READ	IORD	222	4	Die Ausführungszeiten hängen von der jeweiligen von diesem Befehl angesprochenen Spezial-E/A-Baugruppe ab.			225,3	217,7	Erstmalige Ausführung
							232,0	241,7	Wenn beschäftigt
							223,0	215,3	Am Ende
INTELLIGENT I/O WRITE	IOWR	223	4				245,3	219,7	Erstmalige Ausführung
							231,0	225,7	Wenn beschäftigt
							244,0	218,7	Am Ende
CPU BUS I/O REFRESH	DLNK	226	4	287,8	315,5	---	321,3	458,7	1 zugeordnetes Wort

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-2-22 Befehle für serielle Kommunikation

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
PROTOCOL MACRO	PMCR	260	5	100,1	142,1	276,8	158,4	206,0	0 Worte senden, 0 Worte empfangen
				134,2	189,6	305,9	210,0	256,7	249 Worte senden, 249 Worte empfangen
TRANSMIT	TXD	236	4	68,5	98,8	98,8	109,3	102,9	1 Byte senden
				734,3	1,10 ms	1,10 ms	1,23 ms	1,16 ms	256 Bytes senden
RECEIVE	RXD	235	4	89,6	131,1	131,1	144,0	132,1	1 Byte empfangen
				724,2	1,11 ms	1,11 ms	1,31 ms	1,22 ms	256 Bytes empfangen

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
TRANSMIT VIA SERIAL COMMUNICATIONS UNIT	TXDU	256	4	131,5	202,4	---	213,4	208,6	1 Byte senden
RECEIVE VIA SERIAL COMMUNICATIONS UNIT	RXDU	255	4	131	200,8	---	211,8	206,8	1 Byte empfangen
CHANGE SERIAL PORT SETUP	STUP	237	3	341,2	400,0	440,4	504,7	524,7	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-23 Netzwerkbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
NETWORK SEND	SEND	090	4	84,4	123,9	123,9	141,6	195,0	---
NETWORK RECEIVE	RECV	098	4	85,4	124,7	124,7	142,3	196,7	---
DELIVER COMMAND	CMND	490	4	106,8	136,8	136,8	167,7	226,7	---
EXPLICIT MESSAGE SEND (siehe Hinweis 2)	EXPLT	720	4	127,6	190,0	---	217,0	238,0	---
EXPLICIT GET ATTRIBUTE (siehe Hinweis 2)	EGATR	721	4	123,9	185,0	---	210,0	232,7	---
EXPLICIT SET ATTRIBUTE (siehe Hinweis 2)	ESATR	722	3	110,0	164,4	---	188,3	210,3	---
EXPLICIT WORD READ (siehe Hinweis 2)	ECHRD	723	4	106,8	158,9	---	176,3	220,3	---
EXPLICIT WORD WRITE (siehe Hinweis 2)	ECHWR	724	4	106,0	158,3	---	175,7	205,3	---

Hinweis
1. Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.
2. Nur von CPU-Baugruppen ab Baugruppenversion 2.0 unterstützt.

4-2-24 Dateispeicherbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
READ DATA FILE	FREAD	700	5	391,4	632,4	684,1	657,3	641,3	Binär dargestellter Ordner- und Dateiname von jeweils 2 Zeichen Länge
				836,1	1,33 ms	1,35 ms	1,45 ms	1,16 ms	Binär dargestellter Ordner- und Dateiname von jeweils 73 Zeichen Länge
WRITE DATA FILE	FWRIT	701	5	387,8	627,0	684,7	650,7	637,3	Binär dargestellter Ordner- und Dateiname von jeweils 2 Zeichen Länge
				833,3	1,32 ms	1,36 ms	1,44 ms	1,16 ms	Binär dargestellter Ordner- und Dateiname von jeweils 73 Zeichen Länge

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-25 Anzeigebefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DISPLAY MESSAGE	MSG	046	3	10,1	14,2	14,3	16,8	17,3	Meldung anzeigen
				8,4	11,3	11,3	14,7	14,7	Angezeigte Meldung löschen

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-26 Uhrfunktionsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
CALENDAR ADD	CADD	730	4	38,3	201,9	209,5	217,0	194,0	---
CALENDAR SUBTRACT	CSUB	731	4	38,6	170,4	184,1	184,7	167,0	---
HOURS TO SECONDS	SEC	065	3	21,4	29,3	35,8	36,1	35,4	---
SECONDS TO HOURS	HMS	066	3	22,2	30,9	42,1	45,1	45,7	---
CLOCK ADJUSTMENT	DATE	735	2	216,0	251,5	120,0	118,7	128,3	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-27 Fehlerbehebungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
TRACE MEMORY SAMPLING	TRSM	045	1	80,4	120,0	120,0	207,0	218,3	Erfassung von einem Bit und 0 Worten
				848,1	1,06 ms	1,06 ms	1,16 ms	1,10 ms	Erfassen von 31 Bits und 6 Worten

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-28 Fehlerdiagnosebefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
FAILURE ALARM	FAL	006	3	15,4	16,7	16,7	26,1	24,47	Fehler aufzeichnen
				179,8	244,8	244,8	294,0	264,0	Fehler löschen (in Reihenfolge der Priorität)
				432,4	657,1	657,1	853,3	807,3	Fehler löschen (alle Fehler)
				161,5	219,4	219,4	265,7	233,0	Fehler löschen (einzelne Fehler)
SEVERE FAILURE ALARM	FALS	007	3	---	---	---	---	---	---
FAILURE POINT DETECTION	FPD	269	4	140,9	202,3	202,3	220,7	250,0	Bei Ausführung
				163,4	217,6	217,6	250,3	264,3	Erstmalige Ausführung
				185,2	268,9	268,9	220,7	321,7	Bei Ausführung
				207,5	283,6	283,6	320,7	336,0	Erstmalige Ausführung

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-29 Sonstige Befehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
SET CARRY	STC	040	1	0,06	0,06	0,12	0,15	0,15	---
CLEAR CARRY	CLC	041	1	0,06	0,06	0,12	0,15	0,15	---
SELECT EM BANK	EMBC	281	2	14,0	15,1	15,1	---	---	---
EXTEND MAXIMUM CYCLE TIME	WDT	094	2	15,0	19,7	19,7	23,6	22,0	---
SAVE CONDITION FLAGS	CCS	282	1	8,6	12,5	---	14,2	12,9	---
LOAD CONDITION FLAGS	CCL	283	1	9,8	13,9	---	16,3	15,7	---
CONVERT ADDRESS FROM CV	FRMCV	284	3	13,6	19,9	---	23,1	31,8	---
CONVERT ADDRESS TO CV	TOCV	285	3	11,9	17,2	---	22,5	31,4	---
DISABLE PERIPHERAL SERVICING	IOSP	287	---	13,9	19,8	---	21,5	21,5	---
ENABLE PERIPHERAL SERVICING	IORS	288	---	63,6	92,3	---	22,2	22,2	---

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-30 Blockprogrammierungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
BLOCK PROGRAM BEGIN	BPRG	096	2	12,1	13,0	13,0	27,5	30,4	---
BLOCK PROGRAM END	BEND	801	1	9,6	12,3	13,1	23,2	27,1	---
BLOCK PROGRAM PAUSE	BPPS	811	2	10,6	12,3	14,9	16,0	21,7	---
BLOCK PROGRAM RESTART	BPRS	812	2	5,1	5,6	8,3	9,0	10,2	---
CONDITIONAL BLOCK EXIT	(Ausführungsbedingung) EXIT	806	1	10,0	11,3	12,9	23,8	26,0	EXIT-Bedingung erfüllt
				4,0	4,9	7,3	7,2	8,4	EXIT-Bedingung nicht erfüllt
CONDITIONAL BLOCK EXIT	EXIT (Bitadresse)	806	2	6,8	13,5	16,3	28,4	30,6	EXIT-Bedingung erfüllt
				4,7	7,2	10,7	11,4	13,1	EXIT-Bedingung nicht erfüllt
CONDITIONAL BLOCK EXIT (NOT)	EXIT NOT (Bitadresse)	806	2	12,4	14,0	16,8	28,4	31,2	EXIT-Bedingung erfüllt
				7,1	7,6	11,2	11,8	13,5	EXIT-Bedingung nicht erfüllt

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
Verzweigung	IF (Ausführungsbedingung)	802	1	4,6	4,8	7,2	6,8	8,5	IF-Bedingung erfüllt
				6,7	7,3	10,9	12,2	13,9	IF-Bedingung nicht erfüllt
Verzweigung	IF (Relaisnummer)	802	2	6,8	7,2	10,4	11,0	12,7	IF-Bedingung erfüllt
				9,0	9,6	14,2	16,5	18,5	IF-Bedingung nicht erfüllt
Verzweigung (NOT)	IF NOT (Relaisnummer)	802	2	7,1	7,6	10,9	11,5	13,1	IF-Bedingung erfüllt
				9,2	10,1	14,7	16,8	18,9	IF-Bedingung nicht erfüllt
Verzweigung	ELSE	803	1	6,2	6,7	9,9	11,4	12,6	IF-Bedingung erfüllt
				6,8	7,7	11,2	13,4	15,0	IF-Bedingung nicht erfüllt
Verzweigung	IEND	804	1	6,9	7,7	11,0	13,5	15,4	IF-Bedingung erfüllt
				4,4	4,6	7,0	6,93	8,1	IF-Bedingung nicht erfüllt
ONE CYCLE AND WAIT	WAIT (Ausführungsbedingung)	805	1	12,6	13,7	16,7	28,6	34,0	WAIT-Bedingung erfüllt
				3,9	4,1	6,3	5,6	6,9	WAIT-Bedingung nicht erfüllt
ONE CYCLE AND WAIT	WAIT (Relaisnummer)	805	2	12,0	13,4	16,5	27,2	30,0	WAIT-Bedingung erfüllt
				6,1	6,5	9,6	10,0	11,4	WAIT-Bedingung nicht erfüllt
ONE CYCLE AND WAIT (NOT)	WAIT NOT (Relaisnummer)	805	2	12,2	13,8	17,0	27,8	30,6	WAIT-Bedingung erfüllt
				6,4	6,9	10,1	10,5	11,8	WAIT-Bedingung nicht erfüllt
COUNTER WAIT	CNTW	814	4	17,9	22,6	27,4	41,0	43,5	Erstmalige Ausführung
				19,1	23,9	28,7	42,9	45,7	Normale Ausführung
	CNTWX	818	4	17,9	22,6	---	41,0	43,5	Erstmalige Ausführung
				19,1	23,9	---	42,9	45,7	Normale Ausführung
HIGH-SPEED TIMER WAIT	TMHW	815	3	25,8	27,9	34,1	47,9	53,7	Erstmalige Ausführung
				20,6	22,7	28,9	40,9	46,2	Normale Ausführung
	TMHWX	817	3	25,8	27,9	---	47,9	53,7	Erstmalige Ausführung
				20,6	22,7	---	40,9	46,2	Normale Ausführung
Schleifensteuerung	LOOP	809	1	7,9	9,1	12,3	15,6	17,6	---

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
Schleifensteuerung	LEND (Ausführungsbedingung)	810	1	7,7	8,4	10,9	13,5	15,5	LEND-Bedingung erfüllt
				6,8	8,0	9,8	17,5	19,8	LEND-Bedingung nicht erfüllt
Schleifensteuerung	LEND (Relaisnummer)	810	2	9,9	10,7	14,4	17,5	19,9	LEND-Bedingung erfüllt
				8,9	10,3	13,0	21,6	24,5	LEND-Bedingung nicht erfüllt
Schleifensteuerung	LEND NOT (Relaisnummer)	810	2	10,2	11,2	14,8	21,9	24,9	LEND-Bedingung erfüllt
				9,3	10,8	13,5	17,8	20,4	LEND-Bedingung nicht erfüllt
TIMER WAIT	TIMW	813	3	22,3	25,2	33,1	47,4	52,0	Standardeinstellung
				24,9	27,8	35,7	46,2	53,4	Normale Ausführung
	TIMWX	816	3	22,3	25,2	33,1	47,4	52,0	Standardeinstellung
				24,9	27,8	35,7	46,2	53,4	Normale Ausführung

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-31 Textzeichenketten-Verarbeitungsbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
MOV STRING	MOV$	664	3	45,6	66,0	84,3	79,3	72,7	1 Zeichen übertragen
CONCATENATE STRING	+$	656	4	86,5	126,0	167,8	152,0	137,0	1 Zeichen + 1 Zeichen
GET STRING LEFT	LEFT$	652	4	53,0	77,4	94,3	93,6	84,8	1 Zeichen aus einer Zeichenkette von 2 Zeichen
GET STRING RIGHT	RGHT$	653	4	52,2	76,3	94,2	92,1	83,3	1 Zeichen aus einer Zeichenkette von 2 Zeichen
GET STRING MIDDLE	MID$	654	5	56,5	84,6	230,2	93,7	84,0	1 Zeichen aus einer Zeichenkette von 3 Zeichen
FIND IN STRING	FIND$	660	4	51,4	77,5	94,1	89,1	96,7	Suche nach einem Zeichen in einer Zeichenkette von 2 Zeichen
STRING LENGTH	LEN$	650	3	19,8	28,9	33,4	33,8	30,1	Zeichenkette von 1 Zeichen
REPLACE IN STRING	RPLC$	661	6	175,1	258,7	479,5	300,7	267,7	Erstes von 2 Zeichen durch 1 Zeichen ersetzen

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DELETE STRING	DEL$	658	5	63,4	94,2	244,6	11,3	99,3	Erstes von 2 Zeichen löschen
EXCHANGE STRING	XCHG$	665	3	60,6	87,2	99,0	105,2	95,3	1 Zeichen durch 1 Zeichen ersetzen
CLEAR STRING	CLR$	666	2	23,8	36,0	37,8	42,0	36,8	1 Zeichen löschen
INSERT INTO STRING	INS$	657	5	136,5	200,6	428,9	204,0	208,0	1 Zeichen nach dem ersten von 2 Zeichen einfügen
Zeichenketten-vergleichsbefehle	LD, AND, OR +=$	670	4	48,5	69,8	86,2	79,9	68,5	Vergleich von zwei Zeichenketten mit je 1 Zeichen
	LD, AND, OR +<>$	671							
	LD, AND, OR +<$	672							
	LD, AND, OR +>$	674							
	LD, AND, OR +>=$	675							

Hinweis Bei Verwendung eines Operanden doppelter Länge erhöht sich die Länge der Befehle in der folgenden Tabelle (Spalte Länge) um 1.

4-2-32 Task-Steuerbefehle

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
TASK ON	TKON	820	2	19,5	26,3	26,3	33,1	32,5	---
TASK OFF	TKOF	821	2	13,3	19,0	26,3	19,7	20,2	---

4-2-33 Befehle zur Modellkonvertierung (nur CPU-Baugruppen ab Ver. 3.0)

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
BLOCK TRANSFER	XFERC	565	4	6,4	6,5	---	33,1	31,1	Übertragung von 1 Wort
				481,6	791,6	---	3.056,1	2.821,1	Übertragung von 1.000 Worten
SINGLE WORD DISTRIBUTE	DISTC	566	4	3,4	3,5	---	19	18,1	Datenverteilung
				5,9	7,3	---	39,5	38,5	Stapelverarbeitung

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
DATA COLLECT	COLLC	567	4	3,5	3,85	---	24,9	29,7	Datenverteilung
				8	9,1	---	22,1	25,3	Stapelverarbeitung
				8,3	9,6	---	25,5	31	Stapelverarbeitung 1 Wort lesen (FIFO)
				2.052,3	2.097,5	---	8.310,1	7.821,1	Stapelverarbeitung 1.000 Worte lesen (FIFO)
MOVE BIT	MOVBC	568	4	4,5	4,88	---	28,1	22,1	---
BIT COUNTER	BCNTC	621	4	4,9	5	---	30,6	28,8	Zählen von 1 Wort
				1.252,4	1.284,4	---	5.814,1	5.223,8	1.000 Worte zählen

4-2-34 Spezielle Funktionsblockbefehle (nur CPU-Baugruppen ab Ver. 3.0)

Befehl	AWL	Code	Länge (Steps, siehe Hinweis)	EIN-Ausführungszeit (μs)					Bedingungen
				CPU6□ H	CPU4□ H	CPU4□	CJ1M, ausgenommen CPU11/21	CJ1M CPU11/21	
GET VARIABLE ID	GETID	286	4	14	22,2	---	23,4	21,3	

4-2-35 Anzahl Programm-Steps von Funktionsblock-Instanzen (CPU-Baugruppen ab Version 3.0)

Mithilfe der nachstehenden Formel können Sie die Anzahl der Programm-Steps bei Verwendung von Funktionsblock-Instanzen (CS/CJ-Serie CPU-Baugruppen ab Baugruppenversion 3.0) berechnen.

Anzahl der Programm-Steps

= Anzahl der Instanzen × (Funktionsblockaufruf [m] + Übergabe der E/A-Parameter [n] × Anzahl der Parameter) + Anzahl der Befehls-Steps in der Funktionsblockdefinition [p] (siehe Hinweis).

Hinweis Wird dieselbe Funktionsblockdefinition mehrfach verwendet und werden somit mehrere Instanzen des Funktionsblocks ausgeführt, erhöht sich die Anzahl der Befehls-Steps in der Funktionsblockdefinition dennoch nicht. Daher wird in der obigen Gleichung die Anzahl der Befehls-Steps in der Funktionsblockdefinition [p] nicht mit der Anzahl der Instanzen multipliziert.

Inhalt			CS/CJ-Serie CPU-Baugruppen ab Version 3.0
m	Funktions-block-aufruf		57 Steps
n	Übergabe der E/A-Parameter (In Klammern: Datentyp)	1-Bit-E/A-Variable (BOOL)	6 Steps
		1-Wort-E/A-Variable (INT, UINT, WORD)	6 Steps
		2-Wort-E/A-Variable (DINT, UDINT, DWORD, REAL)	6 Steps
		4-Wort-E/A-Variable (LINT, ULINT, LWORD, LREAL)	12 Steps
p	Anzahl der Befehls-Steps in der Funktions-blockdefinition	Gesamtzahl der Befehls-Steps (wie bei einem normalen Anwenderprogramm) + 27 Steps	

Beispiel:
Eingabevariablen (1-Wort-Datentyp, INT): 5
Ausgabevariablen (1-Wort-Datentyp, INT): 5
Funktionsblockdefinition: 100 Steps
Anzahl der Steps für eine Instanz = 57 + (5 + 5) × 6 Steps + 100 Steps + 27 Steps = 244 Steps

4-2-36 Bestimmung der Programmkapazität bestehender Programme

In der folgenden Tabelle sind Richtwerte für die Konvertierung der Programmkapazität (Einheit: Worte) von Programmen, die für frühere OMRON SPS-Systeme (SYSMAC C200HX/HG/HE, CVM1 oder CV-Serie) geschrieben wurden, in die Programmkapazität (Einheit: Steps) der CJ-Serie SPS.

Addieren Sie den folgenden Wert (n) zu der Programmkapazität (Einheit: Worte) für jeden Befehl des für frühere SPS-Systeme geschriebenen Programms, um die Programmkapazität (Einheit: Steps) für SPS-Systeme der CJ-Serie zu erhalten.

Steps der CJ-Serie = "a" (Worte) der früheren SPS + n			
Befehle	**Variationen**	**Wert für n bei Konvertierung von C200HX/HG/HE nach CJ-Serie**	**Wert für n bei Konvertierung von CV-Serie oder CVM1 nach CJ-Serie**
Basisbefehle	Kein	OUT, SET, RSET oder KEEP(011): –1 Sonstige Befehle: 0	0
	Ausführung bei steigender Flanke	Kein	+1
	Direkte Aktualisierung	Kein	0
	Überwachung auf steigende Flanke und direkte Aktualisierung	Kein	+2
Spezialbefehle	Kein	0	–1
	Ausführung bei steigender Flanke	+1	0
	Direkte Aktualisierung	Kein	+3
	Überwachung auf steigende Flanke und direkte Aktualisierung	Kein	+4

Beispielsweise ist bei Konvertierung eines C200HX/HG/HE-Programms zur Bestimmung der Programmkapazität (in Steps) für jede Verwendung des Befehls OUT mit einer Adresse zwischen CIO 000000 und CIO 25515 die

Programmkapazität (in Worten) um 1 zu vermindern, sodass sich für die CJ-Serie 1 (2 – 1) ergibt.

Ein weiteres Beispiel: Bei Konvertierung eines CV-Serie-Programms ist für jede Verwendung des Befehls !MOV (MOVE mit direkter Aktualisierung, 4 Worte) die Programmkapazität (in Worten) für die CJ-Serie um 3 zu erhöhen, beträgt also 7 (4 + 3) Steps.

4-2-37 Ausführungszeiten von Funktionsblock-Instanzen (CPU-Baugruppen ab Version 3.0)

Mithilfe der nachstehenden Formel können Sie den Einfluss der Ausführung einer Funktionsblock-Instanz (CS/CJ-Serie CPU-Baugruppen ab Baugruppenversion 3.0) auf die Zykluszeit berechnen.

Einfluss der Ausführung einer Funktionsblock-Instanz auf die Zykluszeit
= Startzeit (A)
+ Zeit für die Übertragung der Ein- und Ausgabeparameter (B)
+ Ausführungszeit für die Befehle in der Funktionsblockdefinition (C)

Die Zeiten A, B und C der verschiedenen CPU-Baugruppen können der folgenden Tabelle entnommen werden.

Funktion			CPU-Baugruppenmodell		
			CS1H-CPU6□H CJ1H-CPU6□H	**CS1G-CPU4□H CJ1G-CPU4□H**	**CJ1M-CPU□□**
A	Startzeit	Startzeit ohne Berücksichtigung der Übertragung der Ein- und Ausgabeparameter	6,8 μs	8,8 μs	15,0 μs
B	Zeit für die Übertragung der Ein- und Ausgabeparameter (B) (In Klammern: Datentyp)	1-Bit-E/A-Variable (BOOL)	0,4 μs	0,7 μs	1,0 μs
		1-Wort-E/A-Variable (INT, UINT, WORD)	0,3 μs	0,6 μs	0,8 μs
		2-Wort-E/A-Variable (DINT, UDINT, DWORD, REAL)	0,5 μs	0,8 μs	1,1 μs
		4-Wort-E/A-Variable (LINT, ULINT, LWORD, LREAL)	1,0 μs	1,6 μs	2,2 μs
C	Ausführungszeit für die Befehle in der Funktionsblockdefinition	Gesamte Befehlsverarbeitungszeit (wie bei einem normalen Anwenderprogramm)			

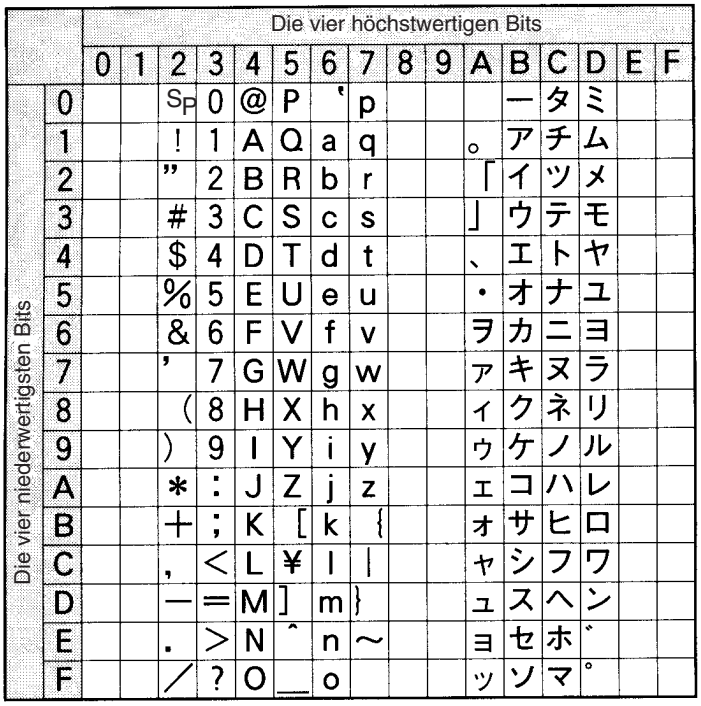
Beispiel: CS1H-CPU63H
Eingabevariablen (1-Wort-Datentyp, INT): 3
Ausgabevariablen (1-Wort-Datentyp, INT): 2
Ausführungszeit für die Befehle in der Funktionsblockdefinition: 10 μs
Ausführungszeit für eine Instanz = 6,8 μs + (3 + 2) × 0,3 μs + 10 μs = 18,3 μs

Hinweis Wird dieselbe Funktionsblockdefinition mehrfach verwendet, erhöht sich die Ausführungszeit des Programms entsprechend der Anzahl der Instanzen.

Anhang A

ASCII-Codetabelle

ASCII



Die vier niederwertigsten Bits	Die vier höchstwertigen Bits															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	‘	p				ー	タ	ミ		
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			”	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			$	4	D	T	d	t			、	エ	ト	ヤ		
5			%	5	E	U	e	u			・	オ	ナ	ユ		
6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7			’	7	G	W	g	w			ァ	キ	ヌ	ラ		
8			(	8	H	X	h	x			ィ	ク	ネ	リ		
9			)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ェ	コ	ハ	レ		
B			+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C			,	<	L	¥	l	|			ャ	シ	フ	ワ		
D			−	=	M	]	m	}			ュ	ス	ヘ	ン		
E			.	>	N	^	n	~			ョ	セ	ホ	゛		
F			/	?	O	_	o				ッ	ソ	マ	°		

Index

A

Adressierung
  Operanden 6
  Siehe auch Indexregister
  Zählernummern 272
  Zeitgebernummern 272
Aktualisierung
  Aktualisierungsbefehle mit Flankenausführung 169
  Befehle mit direkter Aktualisierung 169
  mit IORF(097) 885
Anwendungen
  Sicherheitshinweise xxvi
Anzeigebefehle
  Ausführungszeiten 1245, 1276
Arithmetik
  Addition eines Wortbereichs 697
  Arithmetische Befehle 49
  Exponenten 595, 651
  Gleitkommaaddition 570, 626
  Gleitkomma-Arithmetikbefehle 63, 558–600, 613–657
  Gleitkommadaten-Division 552, 576
  Gleitkommamultiplikation 574, 630
  Gleitkommasubtraktion 572, 628
  lineare Extrapolation 542
  Logarithmus 597, 653
  Mittelwertbildung 769
  Quadratwurzel 534, 536, 593, 649
  Siehe auch Trigonometrische Funktionen
  Spezielle arithmetische Befehle 62, 534–1214
  Suchen des Maximalwerts in einem Bereich 689
  Suchen des Minimalwerts in einem Bereich 693
  Symbolarithmetische Befehle 409–465
  trigonometrische Funktionen 540
Arithmetikbefehle für Gleitkommadaten doppelter Genauigkeit 613
ASCII
  Konvertierung in Gleitkommadaten 609
  Konvertierung von Gleitkommadaten 604
  Konvertierung, ASCII in hexadezimal 490
  Konvertierung, hexadezimal in ASCII 486
  Textzeichenkettenverarbeitung 1158
  Zeichentabelle 10
Aufzeichnung
  Flags und Steuerbits 1077
Ausführungsbedingung
  Ausgabe 195
Ausführungszeiten 1219, 1221–1284
Ausgabebefehle
  Ausführungszeiten 1223, 1252
Ausschalt-Interrupts 815–816
Ausschalt-Interrupt-Verarbeitung
  Deaktivieren 814

B

BCD-Daten 11
Bedienung der Peripheriegeräte
  Aktivieren 1123
  Deaktivieren 1121
Bedingungs-Flags
  Status laden 1112
  Status speichern 1110
Befehle 141–274
  alphabetische Liste 108
  Anzahl der Programm-Steps 1219
  Anzahl der Steps 1221
  Anzeigebefehle 92, 1058–1285
  Arithmetische Befehle 49
  Ausführungszeiten 1221, 1250
  Ausgabebefehle 26, 177–191
  Befehle für E/A-Baugruppen 84, 885–921
  Befehle für schnelle Zähler 823
  Befehle für serielle Kommunikation 87, 926–972
  Befehle mit Flankendifferenzierung 3
  Befehlsausführungszeiten 1219
  Befehlsvariationen 4
  Blockprogrammierbefehle 96, 1124–1157
  Dateispeicherbefehle 91, 1042–1045
  Datensteuerungsbefehle 720–772
  Datenübertragungsbefehle 41, 315
  Datenverschiebebefehle 44, 344–392
  Datum- und Uhrzeitbefehle 92, 1061–1121
  Dekrementierbefehle 48, 393–408
  Eingabebefehle 24
  Fehlerdiagnosebefehle 94, 1079–1104
  Fehlersuch-Befehle 93
  Fehlersuchebefehle 1075–1078
  Gleitkomma-Arithmetikbefehle 63, 558–600, 613–657
  Impulsausgabe-Befehle 823
  Inkrementierbefehle 48, 393–408
  Interrupt-Steuerbefehle 80
  Interrupt-Steuerungsbefehle 798–822
  Klassifiziert nach Funktion 16
  Konvertierungsbefehle 54, 465–510
  Liste nach Funktionscode 125
  Logikbefehle 60, 517–534
  Netzwerkbefehle 88, 973–1013
  Regelungsbefehle 75
  Schritt-Befehle 867–884
  Schrittbefehle 84
  Sequenzeingabebefehle 153–177
  Sequenz-Steuerbefehle 29
  Sequenzsteuerbefehle 197–233
  Spezielle arithmetische Befehle 62, 534–1214
  Steps pro Befehl 1250
  Steuerung von Ausführungsbedingungen UP(521) und DOWN(522) 173
  Steuerung von schnellen Zählern und Impulsausgaben 823
  Symbolarithmetische Befehle 409–465
  Symbolvergleichsbefehle 275–281, 600, 657
  Tabellendaten-Verarbeitungsbefehle 67, 71, 660–704, 1237, 1267

Task-Steuerbefehle 105–107, 1192–1199
Textverarbeitungsbefehle 102, 1158–1192
Unterprogrammbefehle 79, 773–797
Vergleichsbefehle 37, 275–310
Zählerbefehle 33, 233–274
Zeichenketten-Vergleichsbefehle 1187–1192
Zeitgeberbefehle 33, 233–274
Befehle für serielle Kommunikation
Ausführungszeiten 1243, 1274
Befehlsausführungszeiten 1221–1284
Befehlssätze
-(410) 424
--(592) 397
*(420) 443
*B(424) 450
*BL(425) 452
*D(847) 630
*F(456) 574, 630
*L(421) 445
*U(422) 447
*UL(423) 449
+$(656) 1161
+(400) 410
++(590) 393
++B(594) 401
++BL(595) 403
++L(591) 395
+B(404) 418
+BC(406) 421
+BCL(407) 423
+BL(405) 419
+C(402) 414
+CL(403) 416
+D(845) 626
+F(454) 570, 626
+L(401) 412
/(430) 454
/B(434) 462
/BL(435) 464
/D(848) 632
/F(457) 576
/L(431) 456
/U(432) 458
/UL(433) 460
7SEG(214) 908
ACC(888) 855
ACOS(464) 589, 645
ACOSD(855) 645
AND 157
AND LD 164
AND NOT 159
ANDL(610) 519
ANDW(034) 517
APR(069) 540
ASC(086) 486
ASIN(463) 587, 643
ASIND(854) 643
ATAN(465) 591, 647
ATAND(856) 647
AVG(195) 769
-B(414) 435
--B(596) 405
BAND(681) 743
-BC(416) 440
BCD(024) 469
BCDL(059) 470
BCDS(471) 505
-BCL(417) 441
BCMP(068) 304
BCNT(067) 556
BDSL(473) 507
BIN(023) 466
BINL(058) 467
BINS(470) 499
BISL(472) 502
-BL(415) 436
--BL(597) 407
BPPS(811) 1131
BPRS(812) 1131
BREAK(514) 232
BSET(071) 331
-C(412) 430
CADD(730) 1061
CCL(283) 1112
CCS(282) 1110
CJP(510) 223
CJPN(511) 223
-CL(413) 432
CLC(041) 1105
CLI(691) 809
CLR$(666) 1182
CMND(490) 973
CMP(020) 287
CMPL(060) 290
CNR(545) 267
CNT 260
CNTR(012) 263
CNTRX(548) 263
CNTW(814) 1147
CNTWX(818) 1147
CNTX(546) 260
COLL(081) 338, 1206
COLM(064) 496
COM(029) 531
COML(614) 533
COS(461) 583, 639
COSD(852) 639
CPS(114) 293
CPSL(115) 296
CSUB(731) 1065
CTBL(882) 837
–D(846) 628
DBL(843) 623
DBLL(844) 624
DEG(459) 579, 636
DEGD(850) 636
DEL$(658) 1178
DI(693) 814
DIFD(014) 184–186
Verwendung in Sprüngen 222, 226, 228
Verwendung in Verriegelungen 203

DIFU(013) 184–186
Verwendung in Sprüngen 222, 226, 228
Verwendung in Verriegelungen 203
DIM(631) 678
DIST(080) 336
DLNK(226) 921
DMPX(077) 482
DOWN(522) 173
DSW(210) 890
EI(694) 816
ELSE(803) 1133
END(001) 197
EXIT(806) 1137
EXP(467) 595, 651
EXPD(858) 651
-F(455) 572, 628
FAL(006) 1079
FALS(007) 1087
FCS(180) 700
FDIV(079) 552
FIFO(633) 672
FIND$(660) 1171
FIX(450) 563, 620
FIXD(841) 620
FIXL(451) 565, 604, 621
FIXLD(842) 621
FLT(452) 566, 623
FLTL(453) 568, 624
FOR(512) 229
FREAD(700) 1045
FRMCV(284) 1113
FSTR(448) 604
FVAL(449) 609
FWRIT(701) 1052
GETR(636) 683
GRET(752) 797
GSBN(751) 794
GSBS(750) 786
HEX(162) 490
HKY(212) 899
HMS(066) 1070
IEND(804) 1133
IF(802) 1133, 1140
IL(002) 201–219
ILC(003) 201–219
INI(880) 823
INS$(657) 1184
IORD(222) 913
IORF(097) 885
IORS(288) 1123
IOSP(287) 1121
IOWR(223) 917
JME(005) 219
JME0(516) 227
JMP(004) 219
JMP0(515) 227
KEEP(011) 180
-L(411) 426
--L(593) 399
LD 153
LD NOT 155
LEFT$(652) 1164
LEN$(650) 1173
LEND(810) 1153
LIFO(634) 675
LINE(063) 494
LMT(680) 741
LOG(468) 597, 653
LOGD(859) 653
LOOP(809) 1153
MAX(182) 689
MCMP(019) 299, 313
MCRO(099) 779
MID$(654) 1168
MIN(183) 693
MLPX(076) 477
MOV$(664) 1159
MOV(021) 315
MOVB(082) 321
MOVD(083) 323
MOVL(498) 318
MOVR(560) 340
MOVRW(561) 342
MSG(046) 1058
MSKR(692) 804
MSKS(690) 798
MTIM(543) 254
MTIMX(554) 254
MVN(022) 317
MVNL(499) 320
NEG(160) 472
NEGL(161) 474
NEXT(513) 229
NOP(000) 198
NOT(520) 172
OR 161
OR LD 166
OR NOT 163
ORG(889) 862
ORW(035) 520
ORWL(611) 522
OUT 177
OUT NOT 178
OUTB(534) 195
PID(190) 720, 731, 1113, 1117, 1121, 1123
PIDAT(191) 731
PLS2(887) 849
PMCR(260) 928
PRV(881) 827, 833
PULS(886) 846
PUSH(632) 669
PWM(891) 865
PWRD(860) 655
RAD(458) 578, 634
RADD(849) 634
RECV(098) 973
RET(093) 786, 797
RGHT$(653) 1166
ROOT(072) 536
ROTB(620) 534
RPLC$(661) 1175
RSET 187

RSTA(531) 189–192, 195
RSTB(533) 192
RXD(235) 944
SBN(092) 783, 794
SBS(091) 773, 786, 921
SCL(194) 757
SCL2(486) 762
SCL3(487) 766
SDEC(078) 888
SDEL(642) 716
SEC(065) 1068
SEND(090) 973, 991
SET 187
SETA(530) 189–192, 195
SETB(532) 192
SETR(635) 681
SIGN(600) 476
SIN(460) 581, 637
SIND(851) 637
SINS(641) 713
SNUM(638) 704
SNXT(009) 868
SPED(885) 841
SQRT(466) 593, 649
SQRTD(857) 649
SRCH(181) 685
SREAD(639) 707
SSET(630) 666
STEP(008) 868
STUP(237) 968
SUM(184) 697
SWAP(637) 687, 704, 707, 710, 713, 716
SWRIT(640) 710
Symbolvergleichsbefehle für Gleitkommadaten doppelter Genauigkeit (335 bis 340) 657
Symbolvergleichsbefehle für Gleitkommadaten einfacher Genauigkeit (329 bis 334) 600
TAN(462) 585
TAND(853) 641
TCMP(085) 301
Testen des Bit-Status 174
TIM 235
TIMH(015) 240
TIMHWX(817) 1150
TIMHX(551) 240
TIML(542) 251
TIMLX(553) 251
TIMW(813) 1144
TIMWX(816) 1144
TIMX(550) 235
TKOF(821) 1196
TKON(820) 1192
TKY(211) 896
TMHH(540) 244
TMHHX(552) 244
TMHW(815) 1150
TOCV(285) 1117
TRSM(045) 1075
TST(350) 174
TSTN(351) 174
TTIM(087) 247
TTIMX(555) 247
TXD(236) 937
UP(521) 173
WDT(094) 1108
XCGL(562) 334
XCHG$(665) 1180
XCHG(073) 333
XFER(070) 328
XFRB(062) 326
XNRL(613) 529
XNRW(037) 528
XORL(612) 526
XORW(036) 524
ZCP(088) 310
ZCPL(116) 313
ZONE(682) 746
Bereichsvergleich 310, 313, 840
Betriebsumgebung
  Sicherheitshinweise xxvi
Binärdaten mit Vorzeichen 11
Binärdaten ohne Vorzeichen 11
Bits
  Setzen und Rücksetzen 192
Blockprogramme
  Anhalten und Neustart 1131
  Befehlsausführungszeiten 1246, 1278
  Beschreibung 1124–1128
  Blockprogrammierbefehle 96, 1124–1157
  Verzweigung 1133, 1140, 1144, 1147, 1150, 1153

C

CJ1 CPU-Baugruppen 3
CJ1-H CPU-Baugruppen 3
CJ1M CPU-Baugruppen 3
CJ-Serie
  Definition xix
CPU-Bus-Baugruppen
  Aktualisierung 921
CS1 CPU-Baugruppen 2
CS1-H CPU-Baugruppen 2
CS-Serie
  Definition xix

D

Dateispeicher
  Befehlsausführungszeiten 1244, 1276
  Dateispeicherbefehle 91, 1042–1045
Dateispeicherbefehle
  Ausführungszeiten 1244, 1276
Daten
  konvertieren
    Radianten und Grad 578–579, 634, 636
  Suchen 685
Datenaufzeichnung
  Siehe auch Aufzeichnung

Datendateien
Lesen 1045
Schreiben 1052
Datenformat
Gleitkommadaten 614
Datenformate 11
Datensteuerungsbefehle
Ausführungszeiten 1240, 1269
Datenübertragungsbefehle
Ausführungszeiten 1228, 1258
Datenverschiebebefehle
Ausführungszeiten 1229, 1258
Datum- und Uhrzeitbefehle
Ausführungszeiten 1245, 1276
Dekrementierbefehle
Ausführungszeiten 1230, 1260
DM-Bereich
Verwenden des DM-Bereichs in Ausführungsbedingungen 174
Drehzahlausgabe 841

E

E/A-Baugruppen
Befehle für E/A-Baugruppen 84, 885–921
E/A-Speicheradresse
Siehe auch Interne E/A-Speicheradresse
Eingabebefehle
Ausführungszeiten 1222, 1251
EM-Bereich
Verwenden des EM-Bereichs in Ausführungsbedingungen 174
EU-Richtlinien xxx
Exponenten 595, 651

F

Fehler
Anwenderdefinierte Fehler 1079, 1087
Befehlsverarbeitungsfehler 13
Codes
Programmierung 1079, 1087
Kommunikationsschnittstellen-Fehler-Flags 982
Kommunikationsschnittstellen-Fehlermerker 957, 965
Meldungen
Programmierung 1058
Nicht schwerwiegende Fehler
Erzeugen 1079
Löschen 1079
Programmfehler 13
Programmieren von Meldungen 1058
schwerwiegend
Erzeugen 1087
Löschen 1087
UM-Überlauf 13
Unzulässiger-Befehl-Fehler 13
Zugriffsfehler 13
Fehlerbehebungsbefehle
Ausführungszeiten 1245, 1277
Fehlerdiagnosebefehle
Ausführungszeiten 1245, 1277
Fehlerprotokoll
Deaktivieren der Speicherung von anwenderdefinierten Fehlern 1083
Fehlersuche
Fehlerdiagnosebefehle 94, 1079–1104
Fehlersuch-Befehle 93
Fehlersuchebefehle 1075–1078
FINS-Befehle 1003
Senden von Befehlen an die lokale CPU-Baugruppe 1010
Flags
AER-Flag 13
Anwenderprogrammspeicherüberlauf-Fehlerflag 13
Aufzeichnung-abgeschlossen-Flag 1077
Aufzeichnung-beschäftigt-Flag 1077
Aufzeichnungstrigger-Überwachung-Flag 1077
CY
Löschen 1105
ER-Flag 13
Unzulässiger-Befehl-Fehlerflag 13
Funktionscodes
Befehle nach Funktionscode aufgelistet 125

G

Gleitkomma-Arithmetikbefehle
Ausführungszeiten 1236, 1265
Gleitkommadaten 559, 614
arithmetische Funktionen 613
Division 552
Exponenten 595, 651
Format 614
Gleitkomma-Arithmetikbefehle 63, 558–600, 613–657
Gleitkomma-Arithmetikbefehle mit doppelter Genauigkeit 67
Konvertierung 613
Konvertierung in ASCII 604, 609
Logarithmen 597, 653
Quadratwurzeln 593, 649
trigonometrische Funktionen 613
Vergleich 600
Gleitkomma-Dezimaldaten 12
Grad
Konvertierung von Grad in Radianten 578, 634

H

Halterelais
Verwendung von KEEP(011) 180

I

Impulsausgaben 823
  Steuerung 823, 855
Indexregister
  Adressierung 8
  Einstellen einer Wort-/Bitadresse in einem Indexregister 340
  Einstellen einer Zeitgeber-/Zähler-Istwertadresse in einem Indexregister 342
Inkrementierbefehle
  Ausführungszeiten 1230, 1260
Installation
  Sicherheitshinweise xxvi
Interne E/A-Speicheradresse
  Einstellen einer Wort-/Bitadresse in einem Indexregister 340
  Einstellen einer Zeitgeber-/Zähler-Istwertadresse in einem Indexregister 342
Interrupts
  alle aktivieren 816
  alle deaktivieren 814
  Lese des Maskierungsstatus 804
  Löschen 809
  Maskierung 798
  Übersicht über die Interrupt-Steuerung 818
  zeitgesteuert
    Lesen des Intervalls 804
Interrupt-Steuerungsbefehle
  Ausführungszeiten 1241, 1271

K

Kommunikation
  Ausführungszeiten von Netzwerkbefehlen 1243, 1275
  Befehle für serielle Kommunikation 87, 926–972
  Befehlsausführungszeiten 1243, 1274
  Beschreibung der seriellen Kommunikation 926
  Empfangen über RS-232C-Schnittstelle 944
  Übertragung über RS-232C-Schnittstelle 937
Kontaktplanprogramme
  Steuerung des Bit-Status
    Verwendung von DIFU(013) und DIFD(014) 184–186
    Verwendung von KEEP(011) 180–184
    Verwendung von SET und RSET 187–189
    Verwendung von SETA(530) und RSTA(531) 189–192, 195
Konvertieren von Speicheradressen 1117
Konvertierung
  siehe auch Daten, konvertieren
Konvertierungsbefehle
  Ausführungszeiten 1234

L

Laufzeitüberwachung
  verlängern 1108
Logarithmus 597, 653
Logikbefehle
  Ausführungszeiten 1235, 1264

M

Maximale Zykluszeit
  verlängern 1108
Meldungen
  Programmierung 1058
Multipoint-E/A-Baugruppen der Gruppe 2
  Aktualisierung mit IORF(097) 886

N

Netzwerkbefehle
  Ausführungszeiten 1243, 1275
Netzwerke
  Netzwerkbefehle 88, 973–1013
Nicht schwerwiegende Systemfehler
  Erzeugen und Löschen 1079

O

Operanden 5
  Eingabe von Daten 5

P

PID-Regelung 720, 731, 1113, 1117, 1121, 1123
Programmierung
  Anhalten/Neustart von Blockprogrammen 1131
  Befehlsausführungszeiten 1221, 1250
  Erstellen von Schrittprogrammen 867
  Konvertieren von Programmen 1250, 1283
  Programmfehler 13
  Programmieren von Meldungen 1058
  Programmkapazität 2
  Verwendung von TR-Bits 170
  Vorbereitung von Daten in Datenbereichen 331
Programmkapazität 2
Protokoll-Makro 928
Prüfsumme
  Berechnung 700
Prüfsummenbefehle 660

Q

Quadratwurzel
BCD-Daten 536
Binärdaten mit Vorzeichen
Siehe auch Arithmetik
Gleitkommadaten 593, 649

R

Radianten
Konvertierung von Radianten in Grad 579, 636
Rahmenprüfsumme
Berechnung 700
RS-232C-Schnittstelle
Empfangen über RS-232C-Schnittstelle 944
Übertragung über RS-232C-Schnittstelle 937
Rücksetzen von Bits 192

S

Schleifen
BREAK(514) 232
FOR(512) und NEXT(513) 229
Schnelle Zähler
Istwert lesen 827, 833
schnelle Zähler und Impulsausgaben, Befehle 823
Schritt-Befehle
Ausführungszeiten 1241, 1271, 1273
Schrittprogramme
Erstellen 867
Schwerwiegende Funktionsfehler
Erzeugen und Löschen 1087
Selbsthaltende Bits
Verwendung von KEEP(011) 181
Sequenzsteuerbefehle
Ausführungszeiten 1224, 1253
Serielle Kommunikation
Beschreibung 926
Setzen von Bits 192
Sicherheitshinweise
Allgemein xxiv
Anwendungen xxvi
Betriebsumgebung xxvi
Siehe auch Vorsichtsmaßnahmen
Sieben-Segment-Anzeigen
Konvertierung von Daten 888
Speicheradresse konvertieren 1113
Speicherkarten
Vorsichtsmaßnahmen 1042
Spezial-E/A-Baugruppen
Lesen des Baugruppenspeichers 913
Schreiben in Baugruppenspeicher 917
Spezielle arithmetische Befehle
Ausführungszeiten 1235, 1265
Sprungbefehle 219, 227
CJP(510) und CJPN(511) 223
SPS der CV-Serie
Konvertieren von Speicheradressen 1117
Speicheradresse konvertieren 1113
SPS-Speicheradresse
Siehe auch Interne E/A-Speicheradresse
Stapel
Stapelbefehle 660
Stapelbefehle 660
Ausführungszeiten 1238, 1268
Stapelverarbeitung
Ausführungszeiten 1238, 1268
Steuerbits
Aufzeichnungsstartbit 1077
Erfassungsstartbit 1077
Suchbefehle 660
Symbolarithmetische Befehle
Ausführungszeiten 1231, 1260
Symbolvergleichsbefehle für Gleitkommadaten doppelter Genauigkeit 657
Symbolvergleichsbefehle für Gleitkommadaten einfacher Genauigkeit 600
SYSMAC LINK-System
Kommunikation 973–979
SYSMAC NET Link-System
Kommunikation 973–979
Systemfehler
Deaktivieren der Speicherung im Fehlerprotokoll 1081
Systemfehlersimulation 1079–1080, 1087
Systemuhr
Addition zur Uhrzeit 1061
Datum- und Uhrzeitbefehle ??–1075, ??–1116, ??–1117, ??–1120, ??–1121
Subtrahieren von der Uhrzeit 1065

T

Tasks
Befehlsausführungszeiten 1249, 1281
Blockprogramme innerhalb von Tasks 1125
Task-Steuerbefehle 105–107, 1192–1199
Task-Steuerbefehle
Ausführungszeiten 1249, 1281
Tastverhältnis
Impulsausgabe mit variablem Tastverhältnis 865
Textzeichenketten
Befehlsausführungszeiten 1249, 1281
Textverarbeitungsbefehle 102, 1158–1192
Trigonometrische Funktionen
Arkuskosinus 589, 645
Arkussinus 587, 643
Arkustangens 591, 647
Konvertierung von Grad in Radianten 578, 634

Konvertierung von Radianten in Grad 579, 636
Kosinus 583, 639
Sinus 581, 637
Tangens 585, 641

U

Uhr
Datum- und Uhrzeitbefehle 92, 1061
Uhrzeit
Konvertierung der Zeitnotation 1068, 1070
Unterprogrammbefehle
Ausführungszeiten 1241, 1270
Unterprogramme
Ausführungszeiten 1241, 1270

V

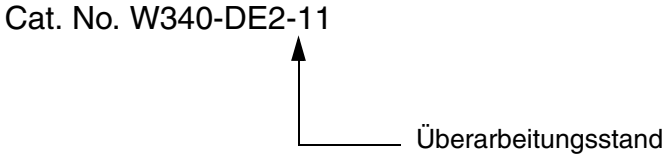
Vergleich 837
Vergleichsbefehle
Ausführungszeiten 1227, 1255, 1257
Vergleichstabellen 837
Verriegelungen 201–219
Vorsichtsmaßnahmen
Sicherheit xxiv
Vorzeichenbehaftete Binärdaten
Vorzeichen entfernen 476

Z

Zähler 233–274
Ausführungszeiten 1225, 1254
Beispielanwendungen 269
Rücksetzung mit CNR(545) 267
Umkehrbarer Zähler 263
Zeitgeber 233–274
Ausführungszeiten 1225, 1254
Beispielanwendungen 269
Blockprogramm-Verzögerungszeitgeber 1150
Rücksetzung mit CNR(545) 267
Zyklische Zusatztasks 1192, 1196
Zykluszeit
Befehlsausführungszeiten 1219
Verlängern der maximalen Zykluszeit 1108

Revisionshistorie

Der Revisionsnummer des Handbuchs bildet das Ende der Katalognummer, die auf der vorderen Umschlagseite des Handbuchs angegeben ist.



In der folgenden Tabelle sind die mit den einzelnen Überarbeitungen vorgenommenen Änderungen aufgeführt. Seitenzahlen beziehen sich stets auf die vorherige Version.

Überarbeitungs-stand	Datum	Änderungen
01	Februar 1999	Erstveröffentlichung
02	Oktober 1999	Überarbeitungen und Hinzufügungen zu CPU-Baugruppen der Version 1. Siehe Liste auf Seite 118.
03	Mai 2000	Überarbeitungen und Änderungen wie folgt: **Seite xiii:** Neue Sicherheitshinweise. **Seite 8:** Hinweis entfernt. **Seiten 162, 166, 177, 180, 183, 189, 196, 198, 262, 531, 560 und 705:** Indexregister aus den Operanden-Spezifikationen entfernt. **Seite 170:** Satz, der mit "Es tritt ein Fehler auf, wenn ein JMP0(515)" beginnt, entfernt. **Seiten 178, 181 und 184:** Vorsichtsmaßnahmen über Zeitgebernummern und über die Verwendung in geänderten Programmsprüngen hinzugekommen. **Seite 181:** Vorsichtmaßnahmen zu Ablaufbit hinzugefügt. **Seiten 179, 182, 184:** Vorsichtmaßnahmen zur Aktualisierung geändert. **Seite 554:** In Klammern gesetzte Informationen aus den Vorsichtmaßnahmen entfernt. **Seiten 576, 577, 579, 581 und 583:** Beschreibung geändert, beinhaltet jetzt auch CS1W–INT01. **Seite 578:** Hinweis zur Verwendung von CLI mit MSKS hinzugefügt. **Seiten 578 und 583:** Vorsichtmaßnahmen für Interrupt-Priorität geändert. **Seiten 639, 647, 651 und 655:** Bezeichnung der seriellen Schnittstellen geändert. **Seite 642:** Handbuchreferenz hinzugefügt. **Seite 675:** Informationen über Dateistruktur hinzugefügt. **Seite 709:** Vorsichtmaßnahmen über lange Zykluszeiten hinzugefügt.
04	November 2000	Überarbeitungen und Änderungen wie folgt: **Seiten 169 und 170:** Vorsichtmaßnahmen zur Zykluszeit gelöscht. **Seiten 176, 180, 183, 186, 196, 199, 743, 746 und 749:** Zeitgebernummer, Zählernummer und Einstellwertanzeigen korrigiert. **Seiten 189 und 192:** Anzeigen für PV- und SV-Bereich korrigiert. **Seiten 209 und 210:** Kontaktplanprogramm geändert und Warnhinweis gelöscht. **Seite 342:** Beschreibung über den Befehl CLEAR CARRY aus den Vorsichtmaßnahmen gestrichen. **Seite 395:** EIN-Bedingung des Fehler-Flags umgeschrieben. **Seite 531:** Aktualisierungszeitpunkt der PID-Konstante zur Abbildung hinzugefügt. **Seiten 533 und 534:** Beschreibung zu PID und Beispiel am Ende der Beschreibung hinzugefügt. **Seite 536:** Bit 01 von C+5 zur Tabelle hinzugefügt. **Seiten 567, 572, 730, 732, 788 und 791:** Hinweis unter der Flag-Tabelle entfernt. **Seite 580:** Hinweis 1 im oberen Bereich der Seite geändert. **Seite 613:** CIO-Adressen geändert. **Seite 704:** FAL-Nummern in Operandentabelle geändert.
05	Mai 2001	Name des Handbuchs geändert, "CS1-Serie" wurde zu "CS-Serie" oder SPS der "CS/CJ-Serie," SPS der CJ-Serie hinzugefügt und "nur CS-Serie" zu spezifizierten beschränkten Funktionen hinzugefügt. Zusätzliche Änderungen und Hinzufügungen erfolgten auf folgenden Seiten: xv, 2, 661, 667, 678, **Seite 116:** Abschnitt 3-2 entfernt. **Seiten 589, 590, 594 und 595:** Informationen zu S und D hinzugefügt. **Seite 598:** Überschriften geändert.
06	Oktober 2001	Neue Produkte wurden in das Handbuch aufgenommen, so z.B. die neuen Hochgeschwindigkeits-CPU-Baugruppen (CS1-H und CJ1-H) und die Befehle, die diese unterstützen. (Umfangreiche Änderungen, die hier nicht alle aufgelistet werden können.)
06A	Februar 2002	**Seite 666:** Bit-Spezifikationen in der Spalte *Steuerdaten* für Bits 04 bis 07 von C+6 und Bits 00 bis 03 von C+6 wurden umgekehrt.

Überarbeitungs-stand	Datum	Änderungen
07	Juli 2002	Handbuch wurde überarbeitet und CJ1M CPU-Baugruppen sowie neue Befehle, die von ihnen unterstützt werden (einschließlich Unterstützung für binäre Aktualisierung für Zeitgeber-/Zähler-Istwert) hinzugefügt. (Umfangreiche Änderungen, die hier nicht alle aufgelistet werden können.) Neue Befehle für Zeitgeber und Zähler hinzugefügt: TIMX, TIMHX, TMHHX, TTIMX, TIMLX, MTIMX, CNTX, CNTRX und CNRX. BCMP2 hinzugefügt. "PC" global zu "SPS" geändert, wenn es sich um eine speicherprogrammierbare Steuerung handelt. **Seite x:** Handbuch hinzugefügt und Produktversionen aktualisiert. **Seiten 379 und 389:** Programmierbeispiel geändert. **Seite 489:** "Kleiner als"-Symbol zu "Kleiner als oder gleich"-Symbol geändert. **Seite 490:** Grafik geändert. **Seite 628:** Operand im Beispiel geändert und Hinweis hinzugefügt. **Seiten 648 und 651:** Ersten Eintrag für Fehler-Flag geändert. **Seite 666:** Bit-Nummern in Tabelle korrigiert. **Seite 701:** Grafik für R+1 geändert. **Seiten 728 bis 748:** Anleitungen überarbeitet. **Seite 787, 814, 816 to 832:** Informationen für automatische Schnittstellenzuordnung hinzugefügt. **Seiten 820 und 825:** Neue Vorsichtmaßnahmen. **Seite 833:** Vorsichtmaßnahmen für Speicherkarten hinzugefügt. **Seite 873:** Untere Hälfte der Seite geändert.
08	September 2002	Handbuch zur Aufnahme von CS1D CPU-Baugruppen geändert. Folgende Änderungen wurden ebenfalls durchgeführt. **Seite xiii:** Zusätzlicher Warnhinweis. **Seiten xvi bis xviii:** Anwendungshinweise durch den selben Abschnitt aus dem *Programmierhandbuch* ersetzt. **Seite 4:** Beschreibung des Vorgangs der direkten Aktualisierung geändert. **Seite 9:** Datentypen hinzugefügt. **Seiten 222 und 225:** "Nicht verwenden" zur Grafik hinzugefügt. **Seite 683:** Rampenreaktionsgrafik korrigiert.
09	Juni 2003	**Seiten 10 und 11:** Hinweis mit Beispielen über Befehle hinzugefügt, die bei auf AUS gesetzten Eingangsbedingungen ausführbar sind. **Seite 24:** Tabelle aktualisiert und Hinweis hinzugefügt für Befehle, die von CS1D CPU-Baugruppen und CS1 CPU-Baugruppen mit -V1 Suffix nicht unterstützt werden. **Seiten 26 bis 28:** Tabelle aktualisiert und Hinweis hinzugefügt für Befehle, die von CS1D CPU-Baugruppen nicht unterstützt werden. **Seiten 36 und 37:**Tabelle aktualisiert und Hinweis hinzugefügt für Befehle, die von CS1D CPU-Baugruppen nicht unterstützt werden. **Seiten 144, 148 und 152:** Tabellen aktualisiert und Hinweise hinzugefügt für neue CPU-Baugruppenmodelle. **Seite 233:** Neuer Hinweis mit Informationen über das Hinzufügen von Zählern mit Hilfe der Online-Bearbeitung. **Seite 293:** Informationen über den Status des ersten Zielworts entfernt. **Seite 679:** Neue Informationen zur Grafik hinzugefügt. **Seiten 681 und 691:** Bedingungen zur Tabelle hinzugefügt, um die Bedeutung der Parametereinstellungen zu verdeutlichen. **Seite 692:** Bit-Nummern korrigiert (ausgetauscht) für Ausgabebereich sowie für integrale und sekundäre Baugruppe. **Seite 710:** Informationen über die Ausgabe von negativen Werten bei Skalierergebnissen geändert. **Seite 781:** Bedingungen für Fehler-Flag zur Tabelle hinzugefügt. **Seite 791:** Informationen über die Ausführung von PLS2(887) zum Hinweis hinzugefügt. **Seite 794:** Tabelle korrigiert. **Seite 797:** Informationen über die Ausführung von PLS2(887) zum Hinweis hinzugefügt. **Seite 824:** Kontaktplanprogrammierung für Prozess B korrigiert. **Seite 831:** "E/A-Baugruppen" in "Spezial-E/A-Baugruppen" geändert. **Seiten 844 und 845:** Informationen über zuerst gesendete und gelesene Wörter/Adressen geändert. **Seite 894:** Referenzhandbuch geändert. **Seite 899:** Informationen über Dateistruktur von Seite 912 im vorigen Handbuch auf diese Seite verschoben. **Page:** Informationen über Dateistruktur von Seiten 912 und 913 im vorigen Handbuch auf diese Seite verschoben. **Seite 1110:** ASCII-Codetabelle von Seite 916 hinzugefügt.

Überarbeitungs-stand	Datum	Änderungen
10	Dezember 2003	Informationen zu von CPU-Baugruppenversionen unterstützten, neuen Funktionen hinzugefügt (eine genaue Liste entfällt hier aufgrund der Vielzahl der Änderungen). **Seiten xi bis xx:** Sicherheitshinweise aktualisiert.
11	Juli 2004	Handbuch für CPU-Baugruppe Version 3.0 sowie für die neuen Befehle, die unterstützt werden, überarbeitet. (Umfangreiche Änderungen, die hier nicht alle aufgelistet werden können.) Neue Befehle: TXDU, RXDU, XFERC, DISTC, COLLC, MOVBC, BCNTC und GETID Überarbeitete Befehle: TXD, RXD, PRV, PRV2, Netzwerkbefehle CPU-Baugruppe hinzugefügt: CJ1H-CPU67H Folgende Korrekturen wurden ebenfalls durchgeführt. **Seite 99:** Funktionscodes für CNTWX und TWHWX korrigiert. **Seiten 183 und 229:** Neue Sicherheitshinweise. **Seite 271:** AWL-Kürzel in der Tabelle korrigiert. **Seite 428:** Überschrift geändert. **Seite 676:** Sicherheitshinweise ersetzt. **Seite 677:** Datensatznummern korrigiert. **Seite 857:** Schnittstellen-Spezifikatortabelle ersetzt.