SIEMENS

SIMATIC

Funktionsplan (FUP) für S7-300/400

Referenzhandbuch

04/2017
A5E41654587-AA

Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

⚠GEFAHR
bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

⚠WARNUNG
bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

⚠VORSICHT
bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG
bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

⚠WARNUNG
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
DEUTSCHLAND

A5E41654587-AA
℗ 04/2017 Änderungen vorbehalten

Inhaltsverzeichnis

Vorwort

Zweck des Handbuchs

Dieses Handbuch unterstützt Sie bei der Erstellung von Anwenderprogrammen in der Programmiersprache FUP.

Es beschreibt die Sprachelemente der Programmiersprache FUP, ihre Syntax und Funktionsweise.

Erforderliche Grundkenntnisse

Dieses Handbuch richtet sich an Programmierer von S7-Programmen, Inbetriebsetzer und Servicepersonal.

Zum Verständnis des Handbuchs sind allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik erforderlich.

Außerdem werden Kenntnisse über die Verwendung von Computern oder PC-ähnlichen Arbeitsmitteln (z. B. Programmiergeräten) unter den Betriebssystemen MS Windows XP, MS Windows Server 2003 oder MS Windows 7 vorausgesetzt.

Gültigkeitsbereich des Handbuchs

Das Handbuch ist gültig für die Programmiersoftware STEP 7 ab Version 5.6.

Normerfüllung nach IEC 1131-3

AWL entspricht der in der Norm DIN EN-61131-3 (int. IEC 1131-3) festgelegten Sprache "Funktionsplan". Genaue Aussagen zur Normerfüllung finden Sie in der Normerfüllungstabelle in der NORM.TAB-Datei von STEP 7.

Online-Hilfe

Ergänzend zum Handbuch erhalten Sie bei der Nutzung der Software detaillierte Unterstützung durch die in die Software integrierte Online-Hilfe.

Auf die Inhalte der Online-Hilfe können Sie wie folgt zugreifen:

- Kontext-sensitive Hilfe zum markierten Objekt über Menübefehl **Hilfe > Hilfe zum Kontext,** über die Funktionstaste F1 oder über das Fragezeichen in der Funktionsleiste.
- Hilfe zu STEP 7 über den Menübefehl **Hilfe > Hilfethemen** oder die Schaltfläche "Hilfe zu STEP 7" im Hilfefenster der kontext-sensitiven Hilfe.
- Glossar für alle STEP 7-Applikationen über die Schaltfläche "**Glossar"**.

Wenn Sie Informationen der Online-Hilfe lieber in gedruckter Form lesen möchten, können Sie einzelne Hilfethemen, Bücher oder die gesamte Hilfe auch ausdrucken.

Dieses Handbuch ist ein Auszug der "Hilfe zu FUP". Aufgrund der identischen Gliederungsstruktur von Handbuch und Online-Hilfe können Sie bequem zwischen Handbuch und Online-Hilfe wechseln.

Weitere Unterstützung

Bei Fragen zur Nutzung der im Handbuch beschriebenen Produkte, die Sie hier nicht beantwortet finden, wenden Sie sich Bitte an Ihren Siemens-Ansprechpartner in den für Sie zuständigen Vertretungen und Geschäftsstellen.

Ihren Ansprechpartner finden Sie unter:

http://www.siemens.com/automation/partner

Den Wegweiser zum Angebot an technischen Dokumentationen für die einzelnen SIMATIC Produkte und Systeme finden Sie unter:

http://www.siemens.com/simatic-tech-doku-portal

Den Online-Katalog und das Online-Bestellsystem finden Sie unter:

http://mall.automation.siemens.com/

Trainingscenter

Um Ihnen den Einstieg in das Automatisierungssystem SIMATIC S7 zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich Bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in D 90026 Nürnberg.

Internet: http://www.sitrain.com

Technical Support

Sie erreichen den Technical Support für alle Industry Automation and Drive Technology Produkte über das Web-Formular für den Support Request

http://www.siemens.com/automation/support-request

Weitere Informationen zu unserem Technical Support finden Sie im Internet unter

http://www.siemens.com/automation/service

Service & Support im Internet

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet unser Know-how an.

http://www.siemens.com/automation/service&support

Dort finden Sie:

- den Newsletter, der Sie ständig mit den aktuellsten Informationen zu Ihren Produkten versorgt.
- die für Sie richtigen Dokumente über unsere Suche im Produkt Support.
- ein Forum, in welchem Anwender und Spezialisten weltweit Erfahrungen austauschen.
- Ihren Ansprechpartner für Industry Automation and Drive Technology vor Ort.
- Informationen über Reparaturen, Ersatzteile und Consulting.

Securityhinweise:

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z. B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter

http://www.siemens.com/industrialsecurity

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter

http://www.siemens.com/industrialsecurity.

1 Bitverknüpfung

1.1 Bitverknüpfungsoperationen Übersicht

Beschreibung

Bitverknüpfungsoperationen arbeiten mit den Zahlen "1" und "0". Diese Zahlen bilden die Basis des Dualsystems und werden "Binärziffern" oder kurz "Bits" genannt. Im Zusammenhang mit UND, ODER, XOR und Ausgängen steht eine "1" für "logisch JA" und eine "0" für "logisch NEIN".

Die Bitverknüpfungsoperationen interpretieren die Signalzustände "1" und "0" und verknüpfen sie entsprechend der Booleschen Logik. Die Verknüpfungen liefern ein Ergebnis von "1" oder "0", das sogenannte Verknüpfungsergebnis (VKE).

Folgende Bitverknüpfungsoperationen stehen Ihnen zur Verfügung:

- & UND, >=1 ODER und XOR Exclusiv-ODER: Sie fragen den Signalzustand ab und erzeugen ein Ergebnis, das entweder in das VKE-Bit kopiert oder mit ihm verknüpft wird.
- UND-vor-ODER-Verknüpfung und ODER-vor-UND-Verknüpfung
- = Zuweisung und # Konnektor: Sie weisen das VKE zu oder speichern es vorübergehend.

Folgende Operationen reagieren auf ein VKE von "1":

- S Ausgang setzen
- R Ausgang rücksetzen
- SR Flipflop setzen rücksetzen
- RS Flipflop rücksetzen setzen

Folgende Operationen reagieren auf einen Wechsel im VKE:

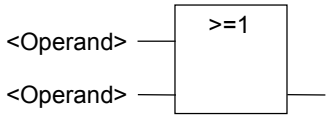
- N Flanke 1 -> 0 abfragen
- P Flanke 0 -> 1 abfragen
- NEG Signalflanke 1 -> 0 abfragen
- POS Signalflanke 0 -> 1 abfragen

Die übrigen Operationen beeinflussen das VKE direkt:

- Binären Eingang einfügen
- Binären Eingang negieren
- SAVE Verknüpfungsergebnis in BIE-Register laden

1.2 >=1 : ODER-Verknüpfung

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, T, Z, D, L	Der Operand gibt das Bit an, dessen Signalzustand abgefragt wird.

Beschreibung

Mit der Operation **ODER** können Sie die Signalzustände zweier oder mehr angegebener Operanden an den Eingängen einer ODER–Box abfragen.

Beträgt der Signalzustand eines der Operanden "1", so ist die Bedingung erfüllt und die Operation liefert das Ergebnis "1". Beträgt der Signalzustand aller Operanden "0", ist die Bedingung nicht erfüllt und die Operation erzeugt das Ergebnis "0".

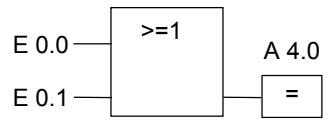
Ist die Operation **ODER** die erste Operation in einer Verknüpfungskette, dann speichert sie das Ergebnis ihrer Signalzustandsabfrage im VKE-Bit.

Jede Operation **ODER**, die nicht die erste Operation in der Verknüpfungskette ist, verknüpft das Ergebnis ihrer Signalzustandsabfrage mit dem im VKE-Bit gespeicherten Wert. Diese Verknüpfung wird entsprechend der ODER–Wahrheitstabelle ausgeführt.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

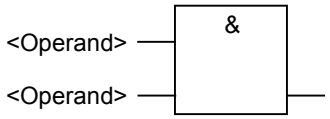
Beispiel



Die Ausgang A 4.0 ist gesetzt, wenn am Eingang E 0.0 ODER am Eingang E 0.1 der Signalzustand "1" ist.

1.3 & : UND-Verknüpfung

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, T, Z, D, L	Der Operand gibt das Bit an, dessen Signalzustand abgefragt wird.

Beschreibung

Mit der Operation **UND** können Sie die Signalzustände zweier oder mehr angegebener Operanden an den Eingängen einer UND–Box abfragen.

Beträgt der Signalzustand aller Operanden "1", so ist die Bedingung erfüllt und die Operation liefert das Ergebnis "1". Beträgt der Signalzustand eines Operanden "0", ist die Bedingung nicht erfüllt und die Operation erzeugt das Ergebnis "0".

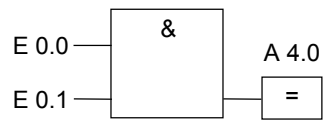
Ist die Operation **UND** die erste Operation in einer Verknüpfungskette, dann speichert sie das Ergebnis ihrer Signalzustandsabfrage im VKE-Bit.

Jede Operation **UND**, die nicht die erste Operation in der Verknüpfungskette ist, verknüpft das Ergebnis ihrer Signalzustandsabfrage mit dem im VKE-Bit gespeicherten Wert. Diese Verknüpfung wird entsprechend der UND–Wahrheitstabelle ausgeführt.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel



Der Ausgang A 4.0 ist gesetzt, wenn am Eingang E 0.0 UND E 0.1 der Signalzustand "1" ist.

1.4 UND-vor-ODER-Verknüpfung und ODER-vor-UND-Verknüpfung

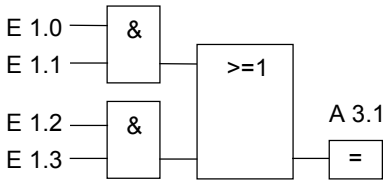
Beschreibung

Mit der Operation **UND-vor-ODER** können Sie das Ergebnis einer Signalzustandsabfrage entsprechend der ODER–Wahrheitstabelle abfragen. Der Signalzustand ist "1", wenn mindestens eine UND-Verknüpfung erfüllt ist.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel



Am Ausgang A 3.1 ist der Signalzustand "1", wenn mindestens eine UND-Verknüpfung erfüllt ist.

Am Ausgang A 3.1 ist der Signalzustand "0", wenn keine UND-Verknüpfung erfüllt ist.

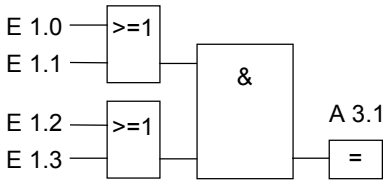
Beschreibung

Mit der Operation **ODER-vor-UND** können Sie das Ergebnis einer Signalzustandsabfrage entsprechend der UND–Wahrheitstabelle abfragen. Der Signalzustand ist "1", wenn alle ODER-Verknüpfungen erfüllt sind.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

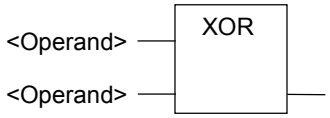


Am Ausgang A 3.1 ist der Signalzustand "1", wenn beide ODER–Verknüpfungen erfüllt sind.

Am Ausgang A 3.1 ist der Signalzustand "0", wenn mindestens eine ODER–Verknüpfung nicht erfüllt ist.

1.5 XOR : EXKLUSIV-ODER-Verknüpfung

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, T, Z, D, L	Der Operand gibt das Bit an, dessen Signalzustand abgefragt wird.

Beschreibung

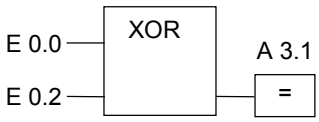
Mit der Operation **EXKLUSIV-ODER** können Sie das Ergebnis einer Signalzustandsabfrage entsprechend der EXKLUSIV-ODER-Wahrheitstabelle abfragen.

Bei einer EXKLUSIV-ODER-Verknüpfung ist der Signalzustand "1", wenn der Signalzustand eines der beiden angegebenen Operanden "1" ist. Bei XOR-Elementen zur Abfrage von mehr als zwei Operanden ist das gemeinsame Verknüpfungsergebnis "1", wenn eine ungerade Anzahl der abgefragten Operanden das Abfrageergebnis "1" liefert.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel



Am Ausgang A 3.1 ist der Signalzustand "1", wenn entweder EXKLUSIV am Eingang E 0.0 ODER am Eingang E 0.2 der Signalzustand "1" ist.

1.6 Binären Eingang einfügen

Symbol

<Operand>

Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, T, Z, D, L	Der Operand gibt das Bit an, dessen Signalzustand abgefragt wird.

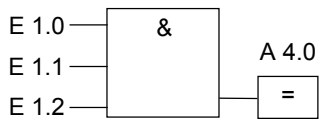
Beschreibung

Die Operation **Binären Eingang einfügen** fügt an einer UND-, ODER- oder XOR–Box hinter der Markierung einen weiteren binären Eingang ein.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	-	1	X	-

Beispiel



Ausgang A 4.0 ist "1", wenn: der Signalzustand an E 1.0 UND E 1.1 UND E 1.2 "1" ist.

1.7 Binären Eingang negieren

Symbol

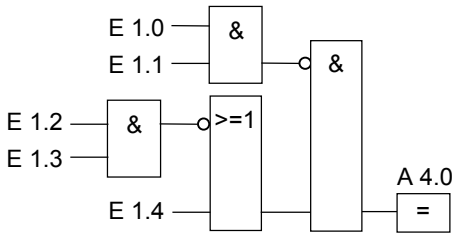
Beschreibung

Die Operation **Binären Eingang negieren** negiert das VKE.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	-	1	X	-

Beispiel

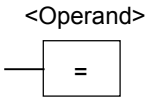


Ausgang A 4.0 ist "1", wenn:

- der Signalzustand an E 1.0 UND E 1.1 NICHT "1" ist
- UND der Signalzustand an E 1.2 UND E 1.3 NICHT "1" ist
- ODER der Signalzustand an E 1.4 "1" ist.

1.8 = : Zuweisung

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, D, L	Der Operand gibt an, welchem Bit der Signalzustand der Verknüpfungskette zugewiesen wird.

Beschreibung

Die Operation **Zuweisung** liefert das Verknüpfungsergebnis. Die Box am Ende der Verknüpfung führt entsprechend den folgenden Kriterien das Signal 1 bzw. 0:

- Der Ausgang führt das Signal 1, wenn die Bedingungen der Verknüpfung vor der Ausgangs-Box erfüllt sind.
- Der Ausgang führt das Signal 0, wenn die Bedingungen der Verknüpfung vor der Ausgangs-Box nicht erfüllt sind.

Die FUP-Verknüpfung weist den Signalzustand dem Ausgang zu, der von der Operation angesprochen wird (um das gleiche zu bewirken, könnte auch der Signalzustand des VKE-Bits dem Operanden zugewiesen werden). Sind die Bedingungen der FUP-Verknüpfungen erfüllt, so ist der Signalzustand an der Ausgangs-Box "1". Andernfalls ist der Signalzustand "0".

Die Operation **Zuweisung** wird vom Master Control Relay beeinflußt.

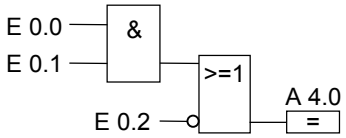
Die Box **Zuweisung** können Sie nur am rechten Ende der Verknüpfungskette anordnen. Sie können allerdings mehrere Boxen **Zuweisung** verwenden.

Eine negierte Zuweisung erzeugen Sie mit der Operation **Binären Eingang negieren**.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	X	-	0

Beispiel

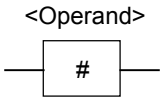


Der Signalzustand an Ausgang A 4.0 ist "1", wenn:

- an den Eingängen E 0.0 UND E 0.1 der Signalzustand "1" ist,
- ODER E 0.2 = 0 ist.

1.9 # : Konnektor

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, D, *L	Der Operand gibt an, welchem Bit das VKE zugewiesen wird.

* Ein Operand im Lokaldaten-Stack kann nur verwendet werden, wenn er in der Variablendeklarationstabelle im Bereich TEMP eines Codebausteins (FC, FB, OB) deklariert wurde.

Beschreibung

Die Operation **Konnektor** ist ein zwischengeschaltetes Zuordnungselement, das das VKE speichert. Genauer gesagt speichert dieses Zuordnungselement die Bitverknüpfung der letzten geöffneten Verzweigung vor dem Zuordnungselement.

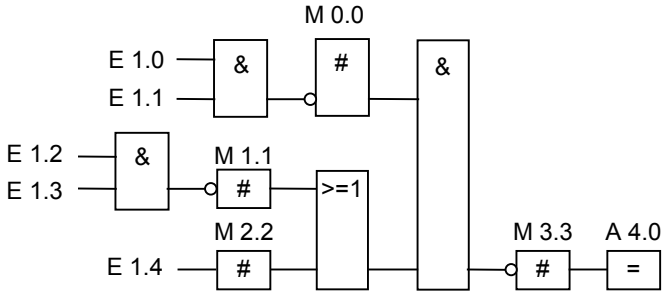
Die Operation **Konnektor** wird vom Master Control Relay beeinflußt.

Einen negierten Konnektor erzeugen Sie, indem Sie den Konnektoreingang negieren.

Statuswort

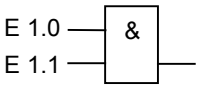
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	X	-	1

Beispiel

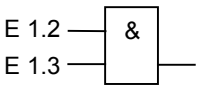


Die Konnektoren speichern die folgenden Verknüpfungsergebnisse:

M 0.0 speichert das negierte VKE von



M 1.1 speichert das negierte VKE von

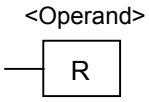


M 2.2 speichert das VKE von E 1.4.

M 3.3 speichert das negierte VKE der gesamten Bitverknüpfung.

1.10 R : Ausgang rücksetzen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL TIMER COUNTER	E, A, M, T, Z, D, L	Der Operand gibt an, welches Bit zurückgesetzt werden soll.

Beschreibung

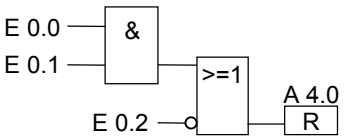
Die Operation **Ausgang rücksetzen** wird nur dann ausgeführt, wenn das VKE = 1 ist. Ist das VKE = 1, wird der angegebene Operand von der Operation auf "0" zurückgesetzt. Ist das VKE = 0, beeinflußt die Operation den angegebenen Operanden nicht. Der Operand bleibt unverändert.

Die Operation **Ausgang rücksetzen** wird vom Master Control Relay beeinflußt.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	X	-	0

Beispiel



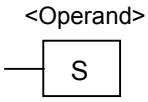
Der Signalzustand an Ausgang A 4.0 wird nur auf "0" zurückgesetzt, wenn:

- an den Eingängen E 0.0 UND E 0.1 Signalzustand "1" ist
- ODER der Signalzustand an Eingang E 0.2 = 0 ist

Wenn das VKE der Verzweigung = 0 ist, wird der Signalzustand an Ausgang A 4.0 nicht verändert.

1.11 S : Ausgang setzen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, D, L	Der Operand gibt an, welches Bit gesetzt werden soll.

Beschreibung

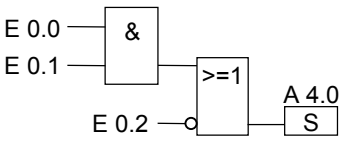
Die Operation **Ausgang setzen** wird nur ausgeführt, wenn das VKE = 1 ist. Ist das VKE = 1, so wird der angegebene Operand von der Operation auf "1" gesetzt. Ist das VKE = 0, beeinflußt die Operation den angegebenen Operanden nicht. Der Operand bleibt unverändert.

Die Operation **Ausgang setzen** wird vom Master Control Relay beeinflußt.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	X	-	0

Beispiel



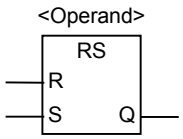
Der Signalzustand an Ausgang A 4.0 wird nur auf "1" gesetzt, wenn:

- an den Eingängen E 0.0 UND E 0.1 der Signalzustand "1" ist
- ODER der Signalzustand an Eingang E 0.2 = 0 ist.

Wenn das VKE der Verzweigung = 0 ist, wird der Signalzustand von A 4.0 nicht verändert.

1.12 RS : Flipflop rücksetzen setzen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, D, L	Der Operand gibt an, welches Bit gesetzt oder zurückgesetzt werden soll.
S	BOOL	E, A, M, D, L, T, Z	Operation Rücksetzen freigegeben
R	BOOL	E, A, M, D, L, T, Z	Operation Setzen freigegeben
Q	BOOL	E, A, M, D, L	Signalzustand des <Operanden>

Beschreibung

Die Operation **Flipflop rücksetzen setzen** führt Operationen wie Setzen (S) oder Rücksetzen (R) nur dann aus, wenn das VKE = 1 ist. Ein VKE von "0" beeinflußt diese Operationen nicht; der in der Operation angegebene Operand wird nicht verändert.

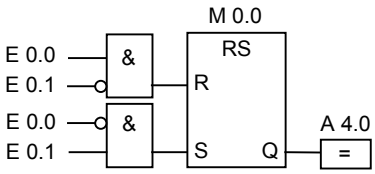
Flipflop rücksetzen setzen wird zurückgesetzt, wenn der Signalzustand an Eingang R = 1 und an Eingang S = 0 ist. Ist Eingang R = 0 und Eingang S = 1, wird das Flipflop gesetzt. Ist das VKE an beiden Eingängen "1", so wird das Flipflop gesetzt.

Die Operation **Flipflop rücksetzen setzen** wird vom Master Control Relay beeinflußt.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

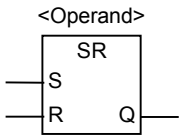


Ist E 0.0 = 1 und E 0.1= 0, wird der Merker M 0.0 rückgesetzt und Ausgang A 4.0 ist "0". Ist E 0.0 = 0 und E 0.1 = 1, wird der Merker M 0.0 gesetzt und Ausgang A 4.0 ist "1".

Wenn beide Signalzustände "0" sind, kommt es zu keiner Änderung. Sind beide Signalzustände "1", dominiert aufgrund der Reihenfolge die Operation Setzen. M 0.0 wird gesetzt und A 4.0 ist "1".

1.13 SR : Flipflop setzen rücksetzen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, D, L	Der Operand gibt an, welches Bit gesetzt oder zurückgesetzt werden soll.
S	BOOL	E, A, M, D, L, T, Z	Operation Setzen freigegeben
R	BOOL	E, A, M, D, L, T, Z	Operation Rücksetzen freigegeben
Q	BOOL	E, A, M, D, L	Signalzustand des <Operanden>

Beschreibung

Die Operation **Flipflop setzen rücksetzen** führt Operationen wie Setzen (S) oder Rücksetzen (R) nur dann aus, wenn das VKE = 1 ist. Ein VKE von "0" beeinflußt diese Operationen nicht; der in der Operation angegebene Operand wird nicht verändert.

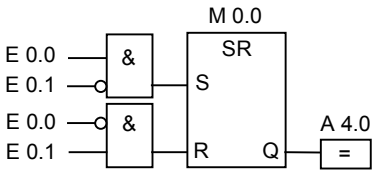
Flipflop setzen rücksetzen wird gesetzt, wenn der Signalzustand an Eingang S = 1 und an Eingang R = 0 ist. Ist Eingang S = 0 und Eingang R = 1, wird das Flipflop zurückgesetzt. Ist das VKE an beiden Eingängen "1", so wird das Flipflop zurückgesetzt.

Die Operation **Flipflop setzen rücksetzen** wird vom Master Control Relay beeinflußt.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

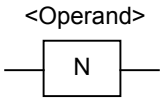


Ist E 0.0 = 1 und E 0.1 = 0, wird der Merker M 0.0 gesetzt und A 4.0 ist "1". Ist E 0.0 = 0 und E 0.1 = 1, wird der Merker M 0.0 zurückgesetzt und A 4.0 ist "0".

Wenn beide Signalzustände "0" sind, kommt es zu keiner Änderung. Sind beide Signalzustände "1", dominiert aufgrund der Reihenfolge die Operation Rücksetzen. M 0.0 wird zurückgesetzt und A 4.0 ist "0".

1.14 N : Flanke 1 -> 0 abfragen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, D, L	Der Operand gibt an, welcher Flankenmerker das vorherige VKE speichert.

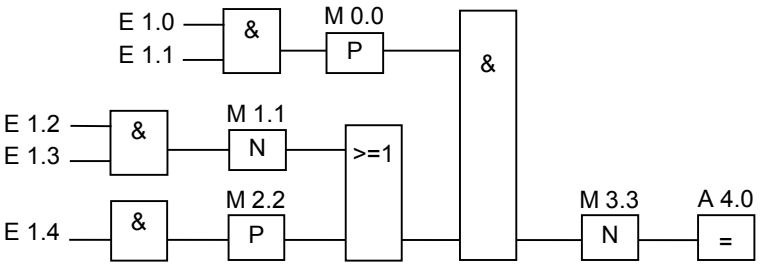
Beschreibung

Die Operation **Flanke 1 → 0 abfragen** erkennt einen Wechsel im angegebenen Operanden von "1" auf "0" (fallende Flanke) und zeigt dies nach der Operation durch VKE = 1 an. Der aktuelle Signalzustand im VKE wird mit dem Signalzustand des Operanden, dem Flankenmerker verglichen. Ist der Signalzustand des Operanden "1" und das VKE vor der Operation "0", so ist das VKE nach der Operation "1" (Impuls), in allen anderen Fällen "0". Das VKE vor der Operation wird im Operanden gespeichert.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	X	X	1

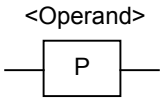
Beispiel



Der Flankenmerker M 3.3 speichert den Signalzustand des vorherigen VKE.

1.15 P : Flanke 0 -> 1 abfragen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand>	BOOL	E, A, M, D, L	Der Operand gibt an, welcher Flankenmerker das vorherige VKE speichert.

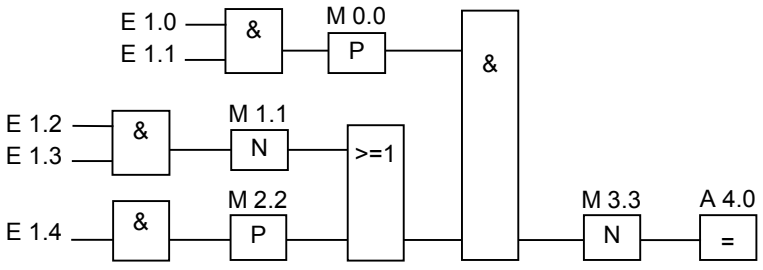
Beschreibung

Die Operation **Flanke 0 → 1 abfragen** erkennt einen Wechsel im angegeben Operanden von "0" auf "1" (steigende Flanke) und zeigt dies nach der Operation durch VKE = 1 an. Der aktuelle Signalzustand im VKE wird mit dem Signalzustand des Operanden, dem Flankenmerker verglichen. Ist der Signalzustand des Operanden "0" und das VKE vor der Operation "1", so ist das VKE nach der Operation "1" (Impuls), in allen anderen Fällen "0". Das VKE vor der Operation wird im Operanden gespeichert

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	X	X	1

Beispiel



Der Flankenmerker M 3.3 speichert den Signalzustand des vorherigen VKE.

1.16 SAVE : Verknüpfungsergebnis in BIE-Register laden

Symbol

Beschreibung

Die Operation **SAVE** speichert das VKE im BIE-Bit des Statusworts. Das Erstabfragebit /ER wird dabei nicht zurückgesetzt.

Aus diesem Grund wird bei einer UND-Verknüpfung im nächsten Netzwerk der Zustand des BIE-Bits mit verknüpft.

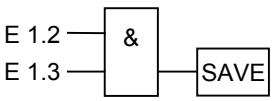
Die Verwendung von **SAVE** und eine nachfolgende Abfrage des BIE-Bits im gleichen Baustein oder in unterlagerten Bausteinen wird nicht empfohlen, da das BIE-Bit durch zahlreiche dazwischen liegende Operationen verändert werden kann. Sinnvoll ist der Einsatz der Operation **SAVE** vor Verlassen eines Bausteins, da damit der ENO-Ausgang (=BIE-Bit) auf den Wert des VKE-Bits gesetzt wird und Sie daran eine Fehlerbehandlung des Bausteins anschließen können.

Mit der Operation **SAVE** kann das VKE eines Netzwerkes in einem untergelagerten Baustein verknüpft werden. Durch die Operation **CALL** im aufrufenden Baustein wird das Erstabfragebit zurückgesetzt.

Statuswort

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**VKE**	**/ER**
schreibt:	X	-	-	-	-	-	-	-	-

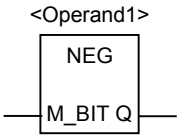
Beispiel



Das Verknüpfungsergebnis (VKE) wird im BIE-Bit gespeichert.

1.17 NEG : Signalflanke 1 -> 0 abfragen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand1>	BOOL	E, A, M, D, L	Abzufragendes Signal für einen negativen Flankenwechsel
M_BIT	BOOL	A, M, D	Der Operand M_BIT gibt an, in welchem Flankenmerker der vorherige Signalzustand von NEG gespeichert ist. Verwenden Sie den Speicherbereich Prozeßabbild der Eingänge E für das M_Bit nur dann, wenn noch keine Eingabebaugruppe diesen Operanden besetzt.
Q	BOOL	E, A, M, D, L	Ausgang des einmaligen Signalwechsels

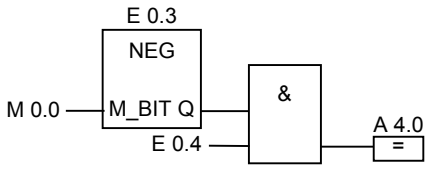
Beschreibung

Die Operation **Signalflanke 1 → 0 abfragen** vergleicht den Signalzustand von <Operand1> mit dem Signalzustand der vorherigen Abfrage, der im Operand M_BIT gespeichert ist. Ist es zu einem Wechsel von "1" auf "0" gekommen, dann ist Ausgang Q = 1, in allen anderen Fällen "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	1	X	1

Beispiel

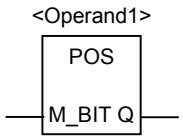


Ausgang A 4.0 ist "1", wenn:

- Eingang E 0.3 eine fallende Flanke hat
- UND an Eingang E 0.4 der Signalzustand "1" ist.

1.18 POS : Signalflanke 0 -> 1 abfragen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
<Operand1>	BOOL	E, A, M, D, L	Abzufragendes Signal für einen positiven Flankenwechsel
M_BIT	BOOL	A, M, D	Der Operand M_BIT gibt an, in welchem Flankenmerker der vorherige Signalzustand von POS gespeichert ist. Verwenden Sie den Speicherbereich Prozeßabbild der Eingänge E für das M_Bit nur dann, wenn noch keine Eingabebaugruppe diesen Operanden besetzt.
Q	BOOL	E, A, M, D, L	Ausgang des einmaligen Signalwechsels

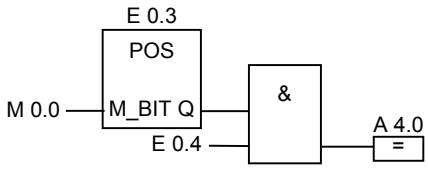
Beschreibung

Die Operation **Signalflanke 0 → 1 abfragen** vergleicht den Signalzustand von <Operand1> mit dem Signalzustand der vorherigen Abfrage, der im Operand M_BIT gespeichert ist. Ist es zu einem Wechsel von "0" auf "1" gekommen, dann ist Ausgang Q = 1, in allen anderen Fällen "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	1	X	1

Beispiel



Ausgang A 4.0 ist "1", wenn:

- Eingang E 0.3 eine steigende Flanke hat
- UND an Eingang E 0.4 der Signalzustand "1" ist.

2 Vergleicher

2.1 Vergleichsoperationen Übersicht

Beschreibung

Verglichen werden die Eingänge IN1 und IN2 entsprechend der folgenden Vergleichsarten:

== IN1 ist gleich IN2
<> IN1 ist ungleich IN2
> IN1 ist größer als IN2
< IN1 ist kleiner als IN2
>= IN1 ist größer als oder gleich IN2
<= IN1 ist kleiner als oder gleich IN2

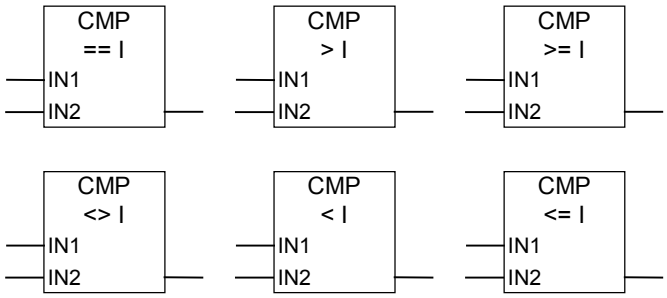
Ergibt der Vergleich die Aussage "wahr", so ist das VKE der Operation "1", ansonsten "0". Es gibt keine Negation des Vergleichsergebnisses, da dies durch die jeweils umgekehrte Vergleichsoperation erreicht werden kann.

Folgende Vergleichsoperationen stehen Ihnen zur Verfügung:

- CMP ? I Ganze Zahlen vergleichen (16 Bit)
- CMP ? D Ganze Zahlen vergleichen (32 Bit)
- CMP ? R Gleitpunktzahlen vergleichen

2.2 CMP ? I : Ganze Zahlen vergleichen (16 Bit)

Symbole



Parameter	Datentyp	Speicherbereich	Beschreibung
IN1	INT	E, A, M, D, L oder Konstante	Erster Vergleichswert
IN2	INT	E, A, M, D, L oder Konstante	Zweiter Vergleichswert
Boxausgang	BOOL	E, A, M, D, L	Ergebnis des Vergleichs

Beschreibung

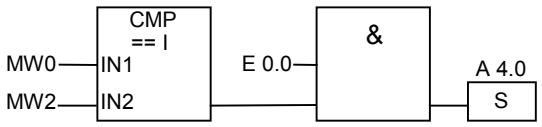
Die Operation **Ganze Zahlen vergleichen (16 Bit)** führt eine Vergleichsoperation auf Basis 16 Bit-Festpunktzahlen aus.

Verglichen werden die Eingänge IN1 und IN2 entsprechend der Vergleichsart, die Sie ausgewählt haben.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	0	-	0	X	X	1

Beispiel

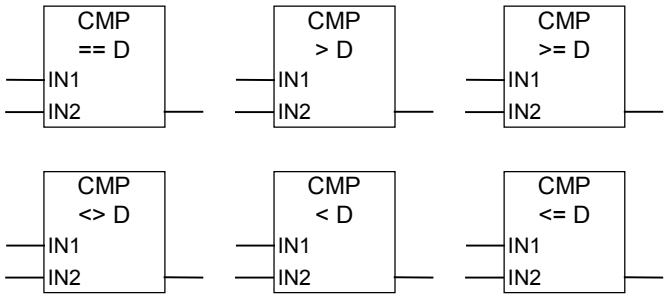


A 4.0 wird gesetzt, wenn:

- MW0 = MW2 ist
- UND an Eingang E 0.0 der Signalzustand "1" ist.

2.3 CMP ? D : Ganze Zahlen vergleichen (32 Bit)

Symbole



Parameter	Datentyp	Speicherbereich	Beschreibung
IN1	DINT	E, A, M, D, L oder Konstante	Erster Vergleichswert
IN2	DINT	E, A, M, D, L oder Konstante	Zweiter Vergleichswert
Boxausgang	BOOL	E, A, M, D, L	Ergebnis des Vergleichs

Beschreibung

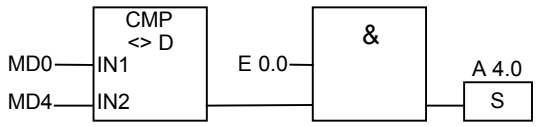
Die Operation **Ganze Zahlen vergleichen (32 Bit)** führt eine Vergleichsoperation auf Basis 32 Bit-Festpunktzahl aus.

Verglichen werden die Eingänge IN1 und IN2 entsprechend der Vergleichsart, die Sie ausgewählt haben.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	X	X	0	-	0	X	X	1

Beispiel

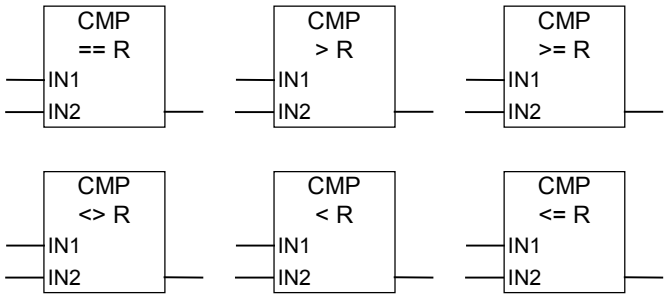


A 4.0 wird gesetzt, wenn:

- MD0 ungleich MD4 ist
- UND an Eingang E 0.0 der Signalzustand "1".

2.4 CMP ? R : Gleitpunktzahlen vergleichen

Symbole



Parameter	Datentyp	Speicherbereich	Beschreibung
IN1	REAL	E, A, M, D, L oder Konstante	Erster Vergleichswert
IN2	REAL	E, A, M, D, L oder Konstante	Zweiter Vergleichswert
Boxausgang	BOOL	E, A, M, D, L	Ergebnis des Vergleichs

Beschreibung

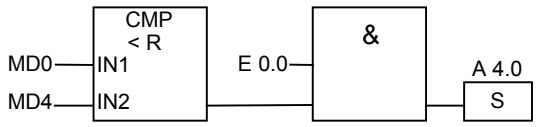
Die Operation **Gleitpunktzahlen vergleichen** führt eine Vergleichsoperation auf Realzahlenbasis aus.

Verglichen werden die Eingänge IN1 und IN2 entsprechend der Vergleichsart, die Sie ausgewählt haben.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	X	X	X	X	0	X	X	1

Beispiel



A 4.0 wird gesetzt, wenn:

- MD0 < MD4 ist
- UND an Eingang E 0.0 der Signalzustand "1" ist

3 Umwandler

3.1 Umwandlungsoperationen Übersicht

Beschreibung

Mit den folgenden Operationen können Sie binär-codierte Dezimalzahlen und Ganzzahlen in andere Zahlenarten umwandeln:

- BCD_I BCD-Zahl in Ganzzahl (16 Bit) wandeln
- I_BCD Ganzzahl (16 Bit) in BCD-Zahl wandeln
- BCD_DI BCD-Zahl in Ganzzahl (32 Bit) wandeln
- I_DI Ganzzahl (16 Bit) in Ganzzahl (32 Bit) wandeln
- DI_BCD Ganzzahl (32 Bit) in BCD-Zahl wandeln
- DI_R Ganzzahl (32 Bit) in Gleitpunktzahl wandeln

Mit den folgenden Operationen können Sie die Komplemente von Ganzzahlen bilden oder das Vorzeichen einer Gleitpunktzahl wechseln:

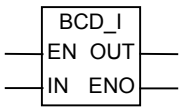
- INV_I Einer-Komplement zu Ganzzahl (16 Bit) erzeugen
- INV_DI Einer-Komplement zu Ganzzahl (32 Bit) erzeugen
- NEG_I Zweier-Komplement zu Ganzzahl (16 Bit) erzeugen
- NEG_DI Zweier-Komplement zu Ganzzahl (32 Bit) erzeugen
- NEG_R Vorzeichen einer Gleitpunktzahl wechseln

Mit den folgenden Operationen können Sie eine Gleitpunktzahl (32 Bit, IEEE 754) in eine Ganzzahl (32 Bit) umwandeln. Die einzelnen Operationen unterscheiden sich in der Art des Rundens.

- ROUND Zahl runden
- TRUNC Ganze Zahl erzeugen
- CEIL Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen
- FLOOR Aus Gleitpunktzahl nächstniedere Ganzzahl erzeugen

3.2 BCD_I : BCD-Zahl in Ganzzahl (16 Bit) wandeln

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	WORD	E, A, M, D, L oder Konstante	BCD-Zahl
OUT	INT	E, A, M, D, L	Ganzzahliger Wert (16 Bit) der BCD-Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

Die Operation **BCD-Zahl in Ganzzahl (16 Bit) wandeln** liest den Inhalt des Eingangsparameters IN als dreistellige binär-codierte Dezimalzahl (BCD, ± 999) und wandelt diese Zahl in einen ganzzahligen Wert (16 Bit) um. Das Ergebnis kann an Ausgang OUT abgefragt werden.

ENO hat immer den gleichen Signalzustand wie EN.

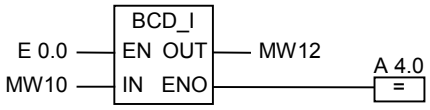
Wenn eine Stelle einer BCD-Zahl im ungültigen Bereich zwischen 10 und 15 liegt, tritt während einer versuchten Umwandlung ein BCD-Fehler auf:

- Die CPU geht in STOP. Im Diagnosespeicher wird ein "BCD-Umwandlungsfehler" der Ereignisnummer 2521 eingetragen.
- Wenn OB 121 programmiert ist, so wird er aufgerufen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	-	-	-	-	0	1	1	1

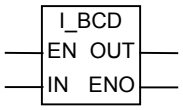
Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerwort MW10 wird als dreistellige BCD-Zahl gelesen und in eine Ganzzahl (16 Bit) umgewandelt. Das Ergebnis wird in MW12 gespeichert. Wird die Umwandlung ausgeführt, ist A 4.0 = 1 (ENO = EN).

3.3 I_BCD : Ganzzahl (16 Bit) in BCD-Zahl wandeln

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	INT	E, A, M, D, L oder Konstante	Ganzzahl (16 Bit)
OUT	WORD	E, A, M, D, L	BCD-Wert der Ganzzahl (16 Bit)
ENO	BOOL	E, A, M, D, L	Freigabeausgang

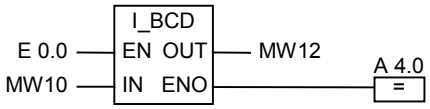
Beschreibung

Die Operation **Ganzzahl (16 Bit) in BCD-Zahl wandeln** liest den Inhalt des Eingangsparameters IN als ganzzahligen Wert (16 Bit) und wandelt ihn in eine dreistellige binär-codierte Dezimalzahl (BCD, ± 999) um. Das Ergebnis kann an Ausgang OUT abgefragt werden. Tritt ein Überlauf auf, ist ENO = 0.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	-	-	X	X	0	X	X	1

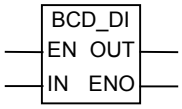
Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerwort MW10 wird als Ganzzahl (16 Bit) gelesen und in eine dreistellige BCD-Zahl umgewandelt. Das Ergebnis wird in MW12 gespeichert. Tritt ein Überlauf auf, ist A 4.0 = 0. Ist der Signalzustand von Eingang EN = 0 (d. h., die Umwandlung wird nicht ausgeführt), dann ist der Signalzustand von Ausgang A 4.0 auch "0".

3.4 BCD_DI : BCD-Zahl in Ganzzahl (32 Bit) wandeln

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	DWORD	E, A, M, D, L oder Konstante	BCD-Zahl
OUT	DINT	E, A, M, D, L	Ganzzahliger Wert (32 Bit) der BCD-Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

Die Operation **BCD-Zahl in Ganzzahl (32 Bit) wandeln** liest den Inhalt des Eingangsparameters IN als siebenstellige binär-codierte Dezimalzahl (BDC, ± 9 999 999) und wandelt diese Zahl in einen ganzzahligen Wert (32 Bit) um. Das Ergebnis kann an Ausgang OUT abgefragt werden.

ENO hat immer den gleichen Signalzustand wie EN.

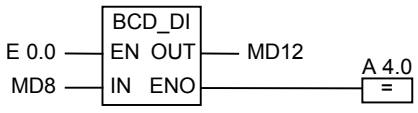
Wenn eine Stelle einer BCD-Zahl im ungültigen Bereich zwischen 10 und 15 liegt, tritt während einer versuchten Umwandlung ein BCD-Fehler auf:

- Die CPU geht in STOP. Im Diagnosespeicher wird ein "BCD-Umwandlungsfehler" der Ereignisnummer 2521 eingetragen.
- Wenn OB 121 programmiert ist, so wird er aufgerufen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	-	-	-	-	0	1	1	1

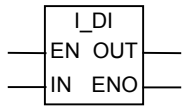
Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerdoppelwort MD8 wird als siebenstellige BCD-Zahl gelesen und in eine Ganzzahl (32 Bit) umgewandelt. Das Ergebnis wird in MD12 gespeichert. Wird die Umwandlung ausgeführt, ist A 4.0 = 1 (ENO = EN).

3.5 I_DI : Ganzzahl (16 Bit) in Ganzzahl (32 Bit) wandeln

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	INT	E, A, M, D, L oder Konstante	Wert, der umgewandelt wird
OUT	DINT	E, A, M, D, L	Ergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

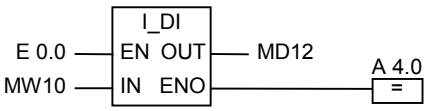
Beschreibung

Die Operation **Ganzzahl (16 Bit) in Ganzzahl (32 Bit) wandeln** liest den Inhalt des Eingangsparameters IN als Ganzzahl (16 Bit) und wandelt diese in eine Ganzzahl (32 Bit) um. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat immer den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	-	-	-	-	0	1	1	1

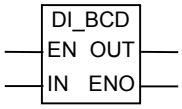
Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerwort MW10 wird als Ganzzahl (16 Bit) gelesen und in eine Ganzzahl (32 Bit) umgewandelt. Das Ergebnis wird in MD12 gespeichert. Wird die Operation ausgeführt, ist A 4.0 = 1 (ENO = EN).

3.6 DI_BCD : Ganzzahl (32 Bit) in BCD-Zahl wandeln

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	DINT	E, A, M, D, L oder Konstante	Ganzzahl (32 Bit)
OUT	DWORD	E, A, M, D, L	BCD-Wert der Ganzzahl (32 Bit)
ENO	BOOL	E, A, M, D, L	Freigabeausgang

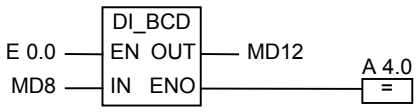
Beschreibung

Die Operation **Ganzzahl (32 Bit) in BCD-Zahl wandeln** liest den Inhalt des Eingangsparameters IN als ganzzahligen Wert (32 Bit) und wandelt diesen Wert in eine siebenstellige binär-codierte Dezimalzahl (BCD, ± 9 999 999) um. Das Ergebnis kann an Ausgang OUT abgefragt werden. Tritt ein Überlauf auf, ist ENO = 0.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	-	-	X	X	0	X	X	1

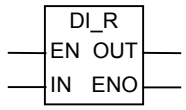
Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerdoppelwort MD8 wird als Ganzzahl (32 Bit) gelesen und in eine siebenstellige BCD-Zahl umgewandelt. Das Ergebnis wird in MD12 gespeichert. Tritt ein Überlauf auf, ist A 4.0 = 0. Ist der Signalzustand von Eingang EN = 0 (d. h., die Umwandlung wird nicht ausgeführt), dann ist der Signalzustand von Ausgang A 4.0 auch "0".

3.7 DI_R : Ganzzahl (32 Bit) in Gleitpunktzahl wandeln

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	DINT	E, A, M, D, L oder Konstante	Wert, der umgewandelt wird
OUT	REAL	E, A, M, D, L	Ergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

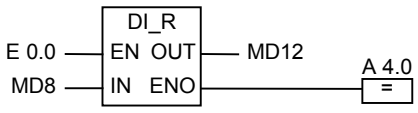
Beschreibung

Die Operation **Ganzzahl (32 Bit) in Gleitpunktzahl wandeln** liest den Inhalt des Eingangsparameters IN als Ganzzahl (32 Bit) und wandelt diese in eine Gleitpunktzahl um. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat immer den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	-	-	-	-	0	1	1	1

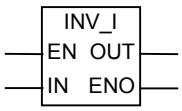
Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerdoppelwort MD8 wird als Ganzzahl (32 Bit) gelesen und in eine Gleitpunktzahl umgewandelt. Das Ergebnis dieser Operation wird in MD12 gespeichert. Wird die Operation nicht ausgeführt, ist A 4.0 = 0 (ENO = EN).

3.8 INV_I : Einer-Komplement zu Ganzzahl (16 Bit) erzeugen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	INT	E, A, M, D, L oder Konstante	Eingangswert
OUT	INT	E, A, M, D, L	Einer-Komplement der Ganzzahl (16 Bit)
ENO	BOOL	E, A, M, D, L	Freigabeausgang

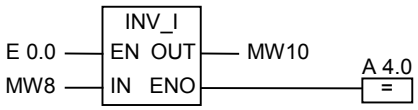
Beschreibung

Die Operation **1er Komplement zu Ganzzahl (16 Bit) erzeugen** liest den Inhalt des Eingangsparameters IN und führt die boolesche Wortverknüpfungsoperation **16 Bit EXKLUSIV ODER verknüpfen** mit der Hexadezimalschablone FFFFH aus. Dadurch wird der Wert jedes Bits umgekehrt. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat immer den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	-	-	-	-	0	1	1	1

Beispiel



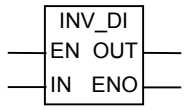
Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Jedes Bit im Merkerdoppelwort MW8 wird umgekehrt:

MW8 = 01000001 10000001 -> MW10 = 10111110 1111110

Die Umwandlung wird nicht ausgeführt, wenn E 0.0 = 0 ist und A 4.0 = 0 (ENO = EN).

3.9 INV_DI : Einer-Komplement zu Ganzzahl (32 Bit) erzeugen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	DINT	E, A, M, D, L oder Konstante	Eingangswert
OUT	DINT	E, A, M, D, L	Einer-Komplement der Ganzzahl (32 Bit)
ENO	BOOL	E, A, M, D, L	Freigabeausgang

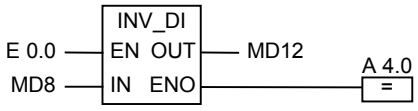
Beschreibung

Die Operation **1er Komplement zu Ganzzahl (32 Bit) erzeugen** liest den Inhalt des Eingangsparameters IN und führt die boolesche Wortverknüpfungsoperation **EXKLUSIV ODER verknüpfen** mit der Hexadezimalschablone FFFF FFFFH aus. Dadurch wird der Wert jedes Bits umgekehrt. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat immer den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	-	-	-	-	0	1	1	1

Beispiel



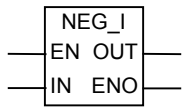
Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Jedes Bit im Merkerdoppelwort MD8 wird umgekehrt:

MD8 = F0FF FFF0 -> MD12 = 0F00 000F

Wird die Umwandlung nicht ausgeführt, ist A 4.0 = 0 (ENO = EN).

3.10 NEG_I : Zweier-Komplement zu Ganzzahl (16 Bit) erzeugen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	INT	E, A, M, D, L oder Konstante	Eingangswert
OUT	INT	E, A, M, D, L	Zweier-Komplement der Ganzzahl (16 Bit)
ENO	BOOL	E, A, M, D, L	Freigabeausgang

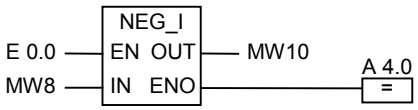
Beschreibung

Die Operation **2er Komplement zu Ganzzahl (16 Bit) erzeugen** liest den Inhalt des Eingangsparameters IN und kehrt das Vorzeichen um (z. B. von einem positiven Wert in einen negativen Wert). Das Ergebnis kann an Ausgang OUT abgefragt werden. Der Signalzustand von EN und ENO ist immer gleich, mit folgender Ausnahme: wenn der Signalzustand von EN gleich "1" ist und ein Überlauf auftritt, ist der Signalzustand von ENO gleich "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

Beispiel



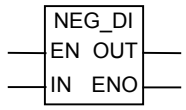
Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Wert von Merkerwort MW8 wird mit umgekehrtem Vorzeichen in MW10 an Ausgang OUT ausgegeben:

MW8 = + 10 -> MW10 = - 10

Wenn EN = 1 ist und ein Überlauf auftritt, ist ENO = 0 und der Signalzustand von A 4.0 ist "0". Wird die Umwandlung nicht ausgeführt, ist A 4.0 = 0 (ENO = EN).

3.11 NEG_DI : Zweier-Komplement zu Ganzzahl (32 Bit) erzeugen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	DINT	E, A, M, D, L oder Konstante	Eingangswert
OUT	DINT	E, A, M, D, L	Zweier-Komplement der Ganzzahl (32 Bit)
ENO	BOOL	E, A, M, D, L	Freigabeausgang

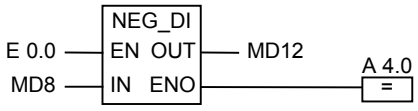
Beschreibung

Die Operation **2er Komplement zu Ganzzahl (32 Bit) erzeugen** liest den Inhalt des Eingangsparameters IN und kehrt das Vorzeichen um (z. B. von einem positiven Wert in einen negativen Wert). Das Ergebnis kann an Ausgang OUT abgefragt werden. Der Signalzustand von EN und ENO ist immer gleich, mit folgender Ausnahme: wenn der Signalzustand von EN gleich "1" ist und ein Überlauf auftritt, ist der Signalzustand von ENO gleich "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

Beispiel



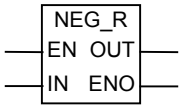
Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Wert von Merkerdoppelwort MD8 wird mit umgekehrtem Vorzeichen in MD12 an Ausgang OUT ausgegeben:

MD8 = + 60.000 -> MD12 = - 60.000

Wenn EN = 1 ist und ein Überlauf auftritt, ist ENO = 0 und der Signalzustand von A 4.0 ist "0". Wird die Umwandlung nicht ausgeführt, ist A 4.0 = 0 (ENO = EN).

3.12 NEG_R : Vorzeichen einer Gleitpunktzahl wechseln

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Eingangswert
OUT	REAL	E, A, M, D, L	Das Ergebnis ist der negierte Eingabewert.
ENO	BOOL	E, A, M, D, L	Freigabeausgang

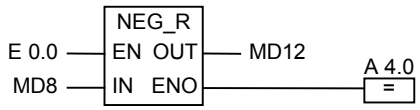
Beschreibung

Die Operation **Vorzeichen einer Gleitpunktzahl wechseln** liest den Inhalt des Eingangsparameters IN und kehrt das Vorzeichenbit um, d. h. die Operation verändert das Vorzeichen der Zahl (z. B. von 0 für positiv in 1 für negativ). Die Bits für Exponent und Mantisse bleiben unverändert. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat immer den gleichen Signalzustand wie EN, mit folgender Ausnahme: wenn der Signalzustand von EN gleich "1" ist und ein Überlauf auftritt, ist der Signalzustand von ENO gleich "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	-	-	-	-	0	X	X	1

Beispiel



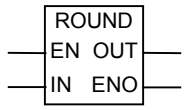
Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Wert von Merkerdoppelwort MD8 wird mit umgekehrtem Vorzeichen in MD12 an Ausgang OUT ausgegeben:

MD8 = + 6,234 -> MD12 = - 6,234

Wird die Umwandlung nicht ausgeführt, ist A 4.0 = 0 (ENO = EN).

3.13 ROUND : Zahl runden

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Wert, der gerundet wird
OUT	DINT	E, A, M, D, L	IN zur nächsten ganzen Zahl gerundet
ENO	BOOL	E, A, M, D, L	Freigabeausgang

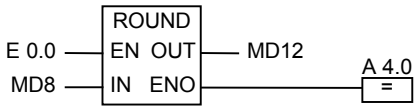
Beschreibung

Die Operation **Zahl runden** liest den Inhalt des Eingangsparameters IN als Gleitpunktzahl und wandelt diese in eine Ganzzahl (32 Bit) um. Das Ergebnis ist die am nächsten liegende ganze Zahl, die vom Ausgangsparameter OUT ausgegeben wird. Ist der gebrochene Anteil = x,5, so wird die gerade Zahl zurückgeliefert (Beispiel: 2,5 -> 2, 1,5 -> 2). Tritt ein Überlauf auf, ist ENO = 0. Ist der Eingang keine Gleitpunktzahl, haben das OV-Bit und OS-Bit den Wert "1" und ENO den Wert "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	-	-	X	X	0	X	X	1

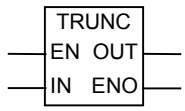
Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerdoppelwort MD8 wird als Gleitpunktzahl gelesen und nach dem Prinzip "round to nearest" in eine Ganzzahl (32 Bit) umgewandelt. Das Ergebnis dieser Operation wird in MD12 gespeichert. Tritt ein Überlauf auf, ist A 4.0 = 0. Ist der Signalzustand von Eingang EN = 0 (d. h. die Umwandlung wird nicht ausgeführt), dann ist der Signalzustand von Ausgang A 4.0 auch "0".

3.14 TRUNC : Ganze Zahl erzeugen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Wert, der gerundet wird
OUT	DINT	E, A, M, D, L	ganzzahliger Anteil von IN
ENO	BOOL	E, A, M, D, L	Freigabeausgang

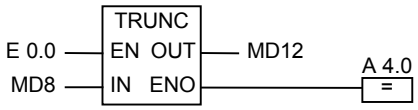
Beschreibung

Die Operation **Ganze Zahl erzeugen** liest den Inhalt des Eingangsparameters IN als Gleitpunktzahl und wandelt diese in eine Ganzzahl (32 Bit) um (Beispiel: aus 1,5 wird 1). Das Ergebnis ist der ganzzahlige Anteil der Gleitpunktzahl, der an Ausgang OUT ausgegeben wird. Tritt ein Überlauf auf, ist ENO = 0. Ist der Eingang keine Gleitpunktzahl, haben das OV-Bit und OS-Bit den Wert "1" und ENO den Wert "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	-	-	X	X	0	X	X	1

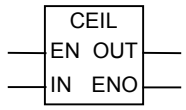
Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerdoppelwort MD8 wird als Gleitpunktzahl gelesen und nach dem Prinzip "round to zero" in eine Ganzzahl (32 Bit) umgewandelt. Das Ergebnis ist der ganzzahlige Teil der Gleitpunktzahl, der in MD12 gespeichert wird. Tritt ein Überlauf auf, ist A 4.0 = 0. Ist der Signalzustand von Eingang EN = 0 (d. h., die Umwandlung wird nicht ausgeführt), dann ist der Signalzustand von Ausgang A 4.0 auch "0".

3.15 CEIL : Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Wert, der umgewandelt wird
OUT	DINT	E, A, M, D, L	Ergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

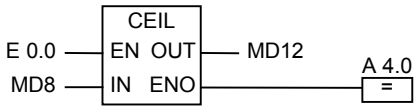
Beschreibung

Die Operation **Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen** liest den Inhalt des Eingangsparameters IN als Gleitpunktzahl und wandelt diese in eine Ganzzahl (32 Bit) um (Beispiel: +1,2 -> +2; -1,5 -> -1). Das Ergebnis ist die kleinste ganze Zahl, die größer gleich der angegebenen Gleitpunktzahl ist. Das Ergebnis kann an Ausgang OUT abgefragt werden. Tritt ein Überlauf auf, ist ENO = 0. Ist der Eingang keine Gleitpunktzahl, haben das OV-Bit und OS-Bit den Wert "1" und ENO den Wert "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	-	-	X	X	0	X	X	1

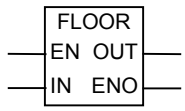
Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerdoppelwort MD8 wird als Gleitpunktzahl gelesen und nach dem Prinzip "round to + infinity" in eine Ganzzahl (32 Bit) umgewandelt. Das Ergebnis dieser Operation wird in MD12 gespeichert. Tritt ein Überlauf auf, ist A 4.0 = 0. Ist der Signalzustand von Eingang EN = 0 (d. h. die Umwandlung wird nicht ausgeführt), dann ist der Signalzustand von Ausgang A 4.0 auch "0".

3.16 FLOOR : Aus Gleitpunktzahl nächstniedere Ganzzahl erzeugen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Wert, der umgewandelt wird
OUT	DINT	E, A, M, D, L	Ergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

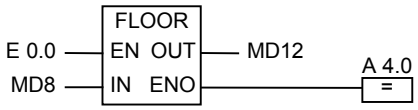
Beschreibung

Die Operation **Aus Gleitpunktzahl nächstniedere Ganzzahl erzeugen** liest den Inhalt des Eingangsparameters IN als Gleitpunktzahl und wandelt diese in eine Ganzzahl (32 Bit) um (Beispiel: +1,5 -> +1; -1,5 -> -2). Das Ergebnis ist die größte ganze Zahl, die kleiner gleich der angegebenen Gleitpunktzahl ist. Das Ergebnis kann an Ausgang OUT abgefragt werden. Tritt ein Überlauf auf, ist ENO = 0. Ist der Eingang keine Gleitpunktzahl, haben das OV-Bit und OS-Bit den Wert "1" und ENO den Wert "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	-	-	X	X	0	X	X	1

Beispiel



Die Umwandlung wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von Merkerdoppelwort MD8 wird als Gleitpunktzahl gelesen und nach dem Prinzip "round to - infinity" in eine Ganzzahl (32 Bit) umgewandelt. Das Ergebnis dieser Operation wird in MD12 gespeichert. Tritt ein Überlauf auf, ist A 4.0 = 0. Ist der Signalzustand von Eingang EN = 0 (d. h. die Umwandlung wird nicht ausgeführt), dann ist der Signalzustand von Ausgang A 4.0 auch "0".

4 Zähler

4.1 Zähloperationen Übersicht

Speicherbereich

Zähler haben einen eigenen reservierten Speicherbereich in Ihrer CPU. Dieser Speicherbereich reserviert ein Wort von 16 Bit für jeden Zähler. Das Programmieren mit FUP unterstützt 256 Zähler.

Zählwert

Die Bits 0 bis 9 des Zählerworts enthalten den Zählwert binär-codiert. Wenn der Zähler gesetzt wird, wird der von Ihnen festgelegte Wert vom Akkumulator in den Zähler übertragen. Der Bereich des Zählwerts liegt zwischen 0 und 999. Sie können den Zählwert innerhalb dieses Bereichs mit folgenden Zähloperationen verändern:

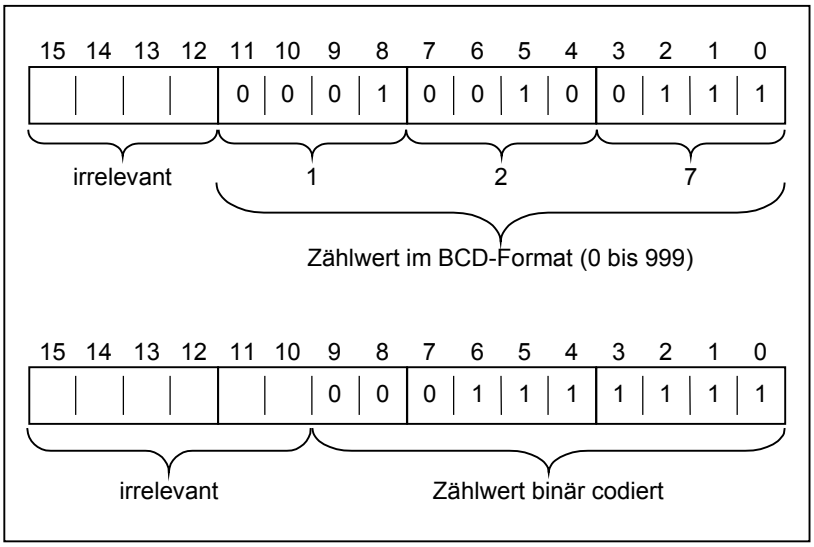
- ZAEHLER Parametrieren und vorwärts-/rückwärtszählen
- Z_VORW Parametrieren und vorwärtszählen
- RUECK Parametrieren und rückwärtszählen
- SZ Zähleranfangswert setzen
- ZV Vorwärtszählen
- ZR Rückwärtszählen

Bit-Konfiguration

Ein Zähler wird auf einen bestimmten Wert gesetzt, indem Sie eine Zahl zwischen 0 und 999 im BCD-Format als Zählwert laden, z. B. C# 127.

Die Bits 0 bis 11 des Zählers enthalten den Zählwert im BCD-Format, d. h. jede Gruppe von 4 Bits enthält jeweils den Binärcode für einen Dezimalwert.

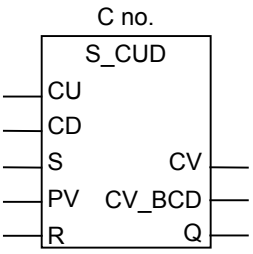
Das folgende Bild zeigt den Inhalt des Zählers, nachdem Sie den Zählwert 127 geladen haben, und den Inhalt des Zählerworts nach dem Setzen des Zählers.


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 0 0 1 0 0 1 0 0 1 1 1
irrelevant
1
2
7
Zählwert im BCD-Format (0 bis 999)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 0 0 1 1 1 1 1 1 1 1
irrelevant
Zählwert binär codiert

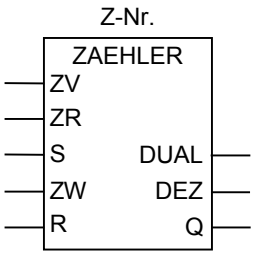
4.2 ZAEHLER : Parametrieren und vorwärts-/rückwärtszählen

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
no.	Nr.	COUNTER	Z	Nummer des Zählers; Bereich ist von der CPU abhängig.
CU	ZV	BOOL	E, A, M, D, L	Eingang ZV: Vorwärtszählen
CD	ZR	BOOL	E, A, M, D, L	Eingang ZR: Rückwärtszählen
S	S	BOOL	E, A, M, D, L, T, Z	Eingang für Voreinstellung des Zählers
PV	ZW	WORD	E, A, M, D, L oder Konstante	Zählwert eingegeben als C#<Wert> im Bereich zwischen 0 und 999
R	R	BOOL	E, A, M, D, L, T, Z	Rücksetzeingang
CV	DUAL	WORD	E, A, M, D, L	Aktueller Zählwert, hexadezimal codiert
CV_BCD	DEZ	WORD	E, A, M, D, L	Aktueller Zählwert, im BCD-Format codiert
Q	Q	BOOL	E, A, M, D, L	Status des Zählers

Beschreibung

Durch einen Flankenwechsel von "0" auf "1" am Eingang S der Operation **Parametrieren und vorwärts-/rückwärtszählen** wird der Zähler mit dem Zählwert ZW vorbesetzt. Der Wert des Zählers wird bei steigender Flanke am Eingang ZV um "1" erhöht, wenn der Zählwert kleiner als 999 ist. Der Wert des Zählers wird bei steigender Flanke am Eingang ZR um "1" vermindert, wenn der Zählwert größer als "0" ist. Haben beide Zähleingänge eine steigende Flanke, werden beide Operationen bearbeitet und der Zählwert bleibt unverändert.

Wird der Zähler gesetzt und ist an den Eingängen ZV/ZR das VKE = 1, so zählt der Zähler einmalig im **nächsten** Zyklus, auch wenn kein Flankenwechsel gegeben war.

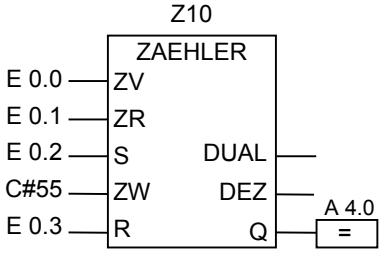
Der Zähler wird zurückgesetzt, wenn am Eingang R eine 1 anliegt. Das Rücksetzen des Zählers setzt den Zählwert auf "0".

Eine Signalzustandsabfrage nach "1" an Ausgang Q ergibt "1", wenn der Zählwert größer als "0" ist. Die Abfrage ergibt "0", wenn der Zählwert gleich "0" ist.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel



Wechselt der Signalzustand an E 0.2 von "0" auf "1", wird der Zähler Z10 mit dem Wert 55 gesetzt. Wechselt der Signalzustand an E 0.0 von "0" auf "1", wird der Wert des Zählers Z10 um "1" erhöht, sofern der Wert von Z10 nicht gleich 999 ist. Wechselt E 0.1 von "0" auf "1", so wird Z10 um "1" vermindert, sofern der Wert von Z10 nicht gleich "0" ist. Wechselt E 0.3 von "0" auf "1", wird der Zählwert von Z10 auf 0 gesetzt. A 4.0 ist "1", wenn Z10 ungleich "0" ist.

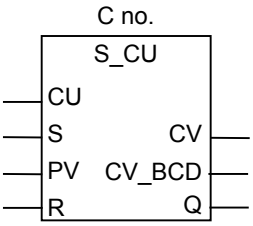
Hinweis

Verwenden Sie einen Zähler nur an einer Stelle im Programm, um Zählfehler zu vermeiden.

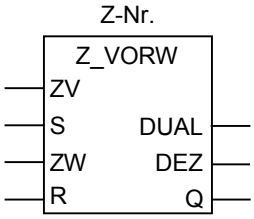
4.3 Z_VORW : Parametrieren und vorwärtszählen

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
no.	Nr.	COUNTER	Z	Nummer des Zählers; Bereich ist von der CPU abhängig.
CU	ZV	BOOL	E, A, M, D, L	Eingang ZV: Vorwärtszählen
S	S	BOOL	E, A, M, D, L, T, Z	Eingang für Voreinstellung des Zählers
PV	ZW	WORD	E, A, M, D, L oder Konstante	Zählwert eingegeben als C#<Wert> im Bereich zwischen 0 und 999
R	R	BOOL	E, A, M, D, L, T, Z	Rücksetzeingang
CV	DUAL	WORD	E, A, M, D, L	Aktueller Zählwert, hexadezimal codiert
CV_BCD	DEZ	WORD	E, A, M, D, L	Aktueller Zählwert, im BCD-Format codiert
Q	Q	BOOL	E, A, M, D, L	Status des Zählers

Beschreibung

Durch einen Flankenwechsel von "0" auf "1" am Eingang S der Operation **Parametrieren und vorwärtszählen** wird der Zähler mit dem Zählwert ZW vorbesetzt. Der Wert des Zählers wird bei steigender Flanke am Eingang ZV um "1" erhöht, wenn der Zählwert kleiner als 999 ist.

Wird der Zähler gesetzt und ist am Eingang ZV das VKE = 1, so zählt der Zähler einmalig im **nächsten** Zyklus, auch wenn kein Flankenwechsel gegeben war.

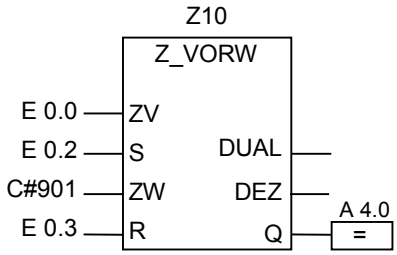
Der Zähler wird zurückgesetzt, wenn am Eingang R eine 1 anliegt. Das Rücksetzen des Zählers setzt den Zählwert auf "0".

Eine Signalzustandsabfrage nach "1" an Ausgang Q ergibt "1", wenn der Zählwert größer als "0" ist. Die Abfrage ergibt "0", wenn der Zählwert gleich "0" ist.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel



Wechselt der Signalzustand an E 0.2 von "0" auf "1", wird der Zähler Z10 mit dem Wert 901 gesetzt. Wechselt der Signalzustand von E 0.0 von "0" auf "1", wird der Wert des Zählers Z10 um "1" erhöht, sofern der Wert von Z10 nicht gleich 999 ist. Wechselt E 0.3 von "0" auf "1", wird der Zählwert von Z10 auf 0 gesetzt. A 4.0 ist "1", wenn Z10 ungleich "0" ist.

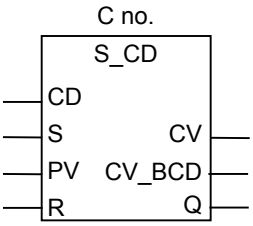
Hinweis

Verwenden Sie einen Zähler nur an einer Stelle im Programm, um Zählfehler zu vermeiden.

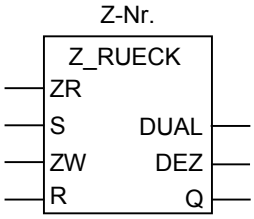
4.4 Z_RUECK : Parametrieren und rückwärtszählen

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
no.	Nr.	COUNTER	Z	Nummer des Zählers; Bereich ist von der CPU abhängig.
CD	ZR	BOOL	E, A, M, D, L	Eingang ZR: Rückwärtszählen
S	S	BOOL	E, A, M, D, L, T, Z	Eingang für Voreinstellung des Zählers
PV	ZW	WORD	E, A, M, D, L oder Konstante	Zählwert eingegeben als C#<Wert> im Bereich zwischen 0 und 999
R	R	BOOL	E, A, M, D, L, T, Z	Rücksetzeingang
CV	DUAL	WORD	E, A, M, D, L	Aktueller Zählwert, hexadezimal codiert
CV_BCD	DEZ	WORD	E, A, M, D, L	Aktueller Zählwert, im BCD-Format codiert
Q	Q	BOOL	E, A, M, D, L	Status des Zählers

Beschreibung

Durch einen Flankenwechsel von "0" auf "1" am Eingang S der Operation **Parametrieren und rückwärtszählen** wird der Zähler mit dem Zählwert ZW vorbesetzt. Der Wert des Zählers wird bei steigender Flanke am Eingang ZR um "1" vermindert, wenn der Zählwert größer als 0 ist.

Wird der Zähler gesetzt und ist am Eingang ZR das VKE = 1, so zählt der Zähler einmalig im **nächsten** Zyklus, auch wenn kein Flankenwechsel gegeben war.

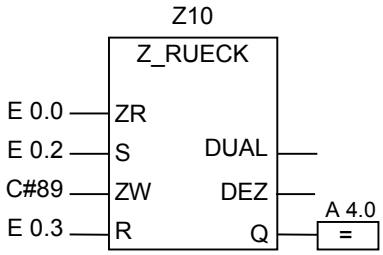
Der Zähler wird zurückgesetzt, wenn am Eingang R eine 1 anliegt. Das Rücksetzen des Zählers setzt den Zählwert auf "0".

Eine Signalzustandsabfrage nach "1" an Ausgang Q ergibt "1", wenn der Zählwert größer als "0" ist. Die Abfrage ergibt "0", wenn der Zählwert gleich "0" ist.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel



Wechselt der Signalzustand an E 0.2 von "0" auf "1", wird der Zähler Z10 mit dem Wert 89 gesetzt. Wechselt der Signalzustand von E 0.0 von "0" auf "1", wird der Wert des Zählers Z10 um "1" vermindert, sofern der Wert von Z10 nicht gleich "0" ist. Wechselt E 0.3 von "0" auf "1", wird der Zählwert von Z10 auf 0 gesetzt.

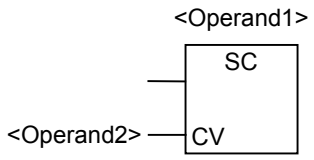
Hinweis

Verwenden Sie einen Zähler nur an einer Stelle im Programm, um Zählfehler zu vermeiden.

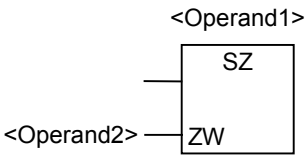
4.5 SZ : Zähleranfangswert setzen

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
Zähler-nummer	Zähler-nummer	COUNTER	Z	Der Operand1 gibt die Nummer des Zählers an, der mit einem Wert voreingestellt werden soll.
CV	ZW	WORD	E, A, M, D, L oder Konstante	Der Wert zum Voreinstellen (Operand2) kann zwischen 0 und 999 liegen. Bei Eingabe einer Konstanten muß vor dem Wert C# stehen, z. B. C#100.

Beschreibung

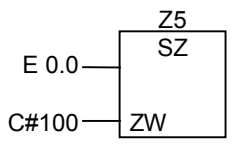
Die Operation **Zähleranfangswert setzen** setzt den Wert des Zählers. Diese Operation wird nur ausgeführt, wenn das VKE eine steigende Flanke aufweist (Wechsel von "0" auf "1" im VKE).

Die Box **Zähleranfangswert setzen** können Sie nur am rechten Ende der Verknüpfungskette anordnen. Sie können allerdings mehrere Boxen **Zähleranfangswert setzen** verwenden.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	-	-	0

Beispiel

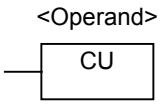


Der Zähler Z5 wird mit dem Wert 100 voreingestellt, wenn der Signalzustand von E 0.0 von "0" auf "1" wechselt (steigende Flanke im VKE). C# gibt an, daß Sie einen Wert im BCD-Format eingeben. Ist keine steigende Flanke vorhanden, wird der Wert des Zählers Z5 nicht verändert.

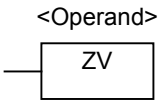
4.6 ZV : Vorwärtszählen

Symbol

Englisch



Deutsch



Parameter	Datentyp	Speicherbereich	Beschreibung
Zählernummer	COUNTER	Z	Der Operand gibt die Nummer des Zählers an, der erhöht werden soll.

Beschreibung

Die Operation **Vorwärtszählen** erhöht den Wert eines angegebenen Zählers um "1", wenn das VKE eine steigende Flanke aufweist (Wechsel von "0" auf "1") und der Wert des Zählers kleiner als 999 ist. Ist im VKE keine steigende Flanke vorhanden oder hat der Zähler bereits den Wert 999, wird er nicht erhöht.

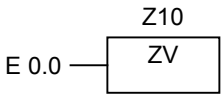
Die Operation **Zähleranfangswert setzen** setzt den Wert des Zählers.

Die Box **Vorwärtszählen** können Sie nur am rechten Ende der Verknüpfungskette anordnen. Sie können allerdings mehrere Boxen **Vorwärtszählen** verwenden.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	-	-	0

Beispiel



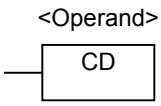
Wechselt der Signalzustand von E 0.0 von "0" auf "1" (steigende Flanke im VKE), wird der Wert des Zählers Z10 um "1" erhöht (es sei denn, der Wert von Z10 ist gleich 999).

Ist keine steigende Flanke vorhanden, wird der Wert von Z10 nicht verändert.

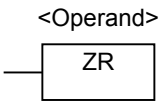
4.7 ZR : Rückwärtszählen

Symbol

Englisch



Deutsch



Parameter	Datentyp	Speicherbereich	Beschreibung
Zählernummer	COUNTER	Z	Der Operand gibt die Nummer des Zählers an, der vermindert werden soll.

Beschreibung

Die Operation **Rückwärtszählen** vermindert den Wert eines angegebenen Zählers um "1", wenn das VKE eine steigende Flanke aufweist (Wechsel von "0" auf "1") und der Wert des Zählers größer als "0" ist. Ist im VKE keine steigende Flanke vorhanden oder hat der Zähler bereits den Wert "0", so wird er nicht vermindert.

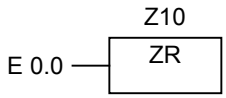
Die Operation **Zähleranfangswert setzen** setzt den Wert des Zählers.

Die Box **Rückwärtszählen** können Sie nur am rechten Ende der Verknüpfungskette anordnen. Sie können allerdings mehrere Boxen **Rückwärtszählen** verwenden.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	-	-	0

Beispiel



Wechselt der Signalzustand von E 0.0 von "0" auf "1" (steigende Flanke im VKE), wird der Wert des Zählers Z10 um "1" vermindert (es sei denn, der Wert von Z10 ist gleich "0").

Ist keine steigende Flanke vorhanden, wird der Wert von Z10 nicht verändert.

5 DB-Aufruf

5.1 OPN: Datenbaustein öffnen

Symbol

<DB-Nummer> oder
<DI-Nummer>

OPN

Parameter	Datentyp	Speicherbereich	Beschreibung
Nummer des DB oder DI	-	-	Nummer des DB oder DI; Bereich ist von der CPU abhängig

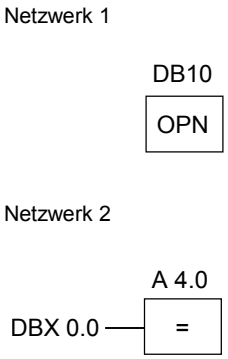
Beschreibung

Mit der Operation **OPN** öffnen Sie einen Datenbaustein als globalen Datenbaustein (DB) oder als Instanz-Datenbaustein (DI). Die Nummer des Datenbausteins wird in das DB- bzw. DI-Register übertragen. Die darauffolgenden DB- und DI-Befehle greifen in Abhängigkeit der Registerinhalte auf die entsprechenden Bausteine zu.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	-	-	-	-

Beispiel



DB 10 ist der aktuell geöffnete Datenbaustein. Daher bezieht sich die Abfrage an DBX 0.0 auf das Bit 0 des Datenbytes 0 in DB 10. Der Signalzustand dieses Bits wird Ausgang A 4.0 zugewiesen.

6 Sprünge

6.1 Sprungoperationen Übersicht

Beschreibung

Sprungoperation können Sie in allen Codebausteinen verwenden, z. B. in Organisationsbausteinen (OBs), Funktionsbausteinen (FBs) und Funktionen (FCs).

Folgende Sprungoperationen stehen Ihnen zur Verfügung:

- JMP Springe im Baustein absolut
- JMP Springe im Baustein wenn 1 (bedingt)
- JMPN Springe im Baustein wenn 0 (bedingt)

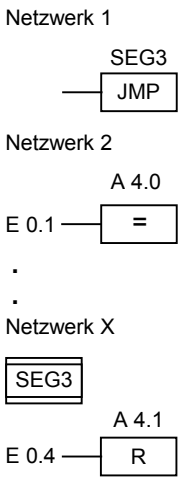
Sprungmarke als Operand

Der Operand einer Sprungoperation ist eine Sprungmarke. Sie gibt das Ziel an, zu dem das Programm springen soll.

Die Sprungmarke geben Sie über der Box JMP ein. Die Sprungmarke besteht aus max. 4 Zeichen. Das erste Zeichen muß ein Buchstabe sein, die anderen Zeichen können Buchstaben oder Zahlen sein (z. B. SEG3).

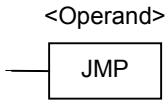
Sprungmarke als Ziel

Die Zielsprungmarke muß am Anfang eines Netzwerks stehen. Sie geben die Zielsprungmarke ein, indem Sie aus der FUP-Auswahlbox LABEL wählen. Es erscheint eine leere Box, in die Sie den Namen der Sprungmarke eingeben.



6.2 JMP : Springe im Baustein absolut

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
Name einer Sprungmarke	-	-	Der Operand gibt die Marke an, zu der absolut gesprungen wird.

Beschreibung

Die Operation **Springe im Baustein absolut** entspricht einer Operation "Gehe zu Sprungmarke". Keine der Operationen zwischen Sprungoperation und Sprungmarke wird ausgeführt.

Diese Operation können Sie in allen Codebausteinen verwenden, z. B. in Organisationsbausteinen (OBs), Funktionsbausteinen (FBs) und Funktionen (FCs).

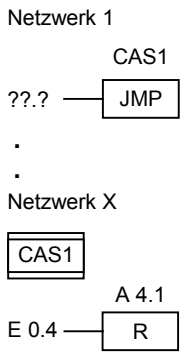
Die Box **Springe im Baustein absolut** darf keine Vorverknüpfung haben.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	-	-	-	-

Die Operation ändert die Bits im Statuswort nicht.

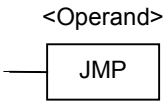
Beispiel



Der Sprung wird immer ausgeführt. Keine der Operationen zwischen Sprungoperation und Sprungmarke wird ausgeführt.

6.3 JMP : Springe im Baustein wenn 1 (bedingt)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
Name einer Sprungmarke	-	-	Der Operand gibt die Marke an, zu der bei VKE = 1 gesprungen wird.

Beschreibung

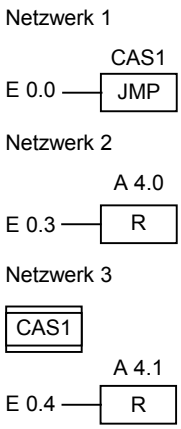
Die Operation **Springe im Baustein, wenn 1** entspricht einer Operation "Gehe zu Sprungmarke" wenn VKE = "1". Verwenden Sie für diese Operation das FUP-Element von **Springe im Baustein absolut**, jedoch mit vorausgehender Verknüpfung. Der bedingte Sprung wird nur dann ausgeführt, wenn das VKE dieser Verknüpfung = 1 ist. Keine der Operationen zwischen Sprungoperation und Sprungmarke wird ausgeführt.

Diese Operation können Sie in allen Codebausteinen verwenden, z. B. in Organisationsbausteinen (OBs), Funktionsbausteinen (FBs) und Funktionen (FCs).

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	1	1	0

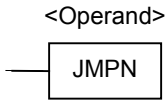
Beispiel



Wenn E 0.0 = 1, wird der Sprung zur Sprungmarke CAS1 ausgeführt. Die Operation, die Ausgang A 4.0 rücksetzt, wird nicht ausgeführt, selbst wenn E 0.3 = 1 ist.

6.4 JMPN : Springe im Baustein wenn 0 (bedingt)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
Name einer Sprungmarke	-	-	Der Operand gibt die Marke an, zu der bei VKE = 0 gesprungen wird.

Beschreibung

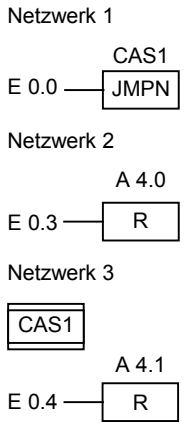
Die Operation **Springe im Baustein, wenn 0** entspricht einer Operation "Gehe zu Sprungmarke", die ausgeführt wird, wenn das VKE = 0 ist.

Diese Operation können Sie in allen Codebausteinen verwenden, z. B. in Organisationsbausteinen (OBs), Funktionsbausteinen (FBs) und Funktionen (FCs).

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	1	1	0

Beispiel



Wenn E 0.0 = 0 ist, wird der Sprung zur Sprungmarke CAS1 ausgeführt. Die Operation, die Ausgang A 4.0 rücksetzt, wird nicht ausgeführt, selbst wenn E 0.3 = 1 ist.

Keine der Operationen zwischen Sprungoperation und Sprungmarke wird ausgeführt.

6.5 LABEL : Sprungmarke

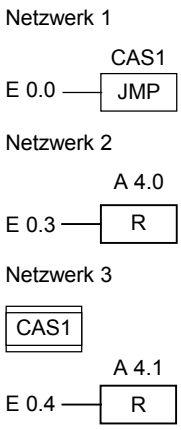
Symbol

Beschreibung

Die Sprungmarke kennzeichnet das Ziel einer Sprungoperation. Es besteht aus 4 Zeichen - erstes Zeichen: Buchstabe; restliche Zeichen: Buchstaben oder Zahlen; z.B. CAS1.

Zu jedem Sprung (**JMP** oder **JMPN**) muß auch eine Sprungmarke (**LABEL**) vorhanden sein.

Beispiel



Wenn E 0.0 = 1 ist, dann wird der Sprung zur Sprungmarke CAS1 ausgeführt.

Wegen des Sprungs wird die Operation "Ausgang rücksetzen" an A 4.0 nicht ausgeführt, auch wenn E 0.3 = 1 ist.

7 Festpunkt-Funktionen

7.1 Festpunkt-Funktionen Übersicht

Beschreibung

Mit den Festpunkt-Funktionen können Sie die folgenden Operationen mit **zwei Ganzzahlen** (16 Bit, 32 Bit) ausführen:

- ADD_I Ganze Zahlen addieren (16 Bit)
- SUB_I Ganze Zahlen subtrahieren (16 Bit)
- MUL_I Ganze Zahlen multiplizieren (16 Bit)
- DIV_I Ganze Zahlen dividieren (16 Bit)

- ADD_DI Ganze Zahlen addieren (32 Bit)
- SUB_DI Ganze Zahlen subtrahieren (32 Bit)
- MUL_DI Ganze Zahlen multiplizieren (32 Bit)
- DIV_DI Ganze Zahlen dividieren (32 Bit)
- MOD_DI Divisionsrest gewinnen (32 Bit)

Auswerten der Bits im Statuswort bei Festpunkt-Funktionen

7.2 Auswerten der Bits im Statuswort bei Festpunkt-Funktionen

Beschreibung

Die Festpunkt-Funktionen beeinflussen die Bits A1, A0, OV und OS im Statuswort.

Die folgenden Tabellen zeigen den Signalzustand der Bits des Statusworts für die Ergebnisse von Operationen mit Festpunktzahlen (16 Bit, 32 Bit).

Gültiger Bereich	A1	A0	OV	OS
0 (Null)	0	0	0	*
16 Bit: -32 768 <= Ergebnis < 0 (negative Zahl) 32 Bit: -2 147 483 648 <=Ergebnis < 0 (negative Zahl)	0	1	0	*
16 Bit: 32 767 >= Ergebnis > 0 (positive Zahl) 32 Bit: 2 147 483 647 >= Ergebnis > 0 (positive Zahl)	1	0	0	*

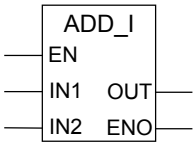
* Das OS-Bit wird vom Ergebnis der Operation nicht beeinflußt.

Ungültiger Bereich	A1	A0	OV	OS
Unterschreitung bei Addition 16 Bit: Ergebnis = -65536 32 Bit: Ergebnis = -4 294 967 296	0	0	1	1
Unterschreitung bei Multiplikation 16 Bit: Ergebnis < -32 768 (negative Zahl) 32 Bit: Ergebnis < -2 147 483 648 (negative Zahl)	0	1	1	1
Überlauf bei Addition, Subtraktion 16 Bit: Ergebnis > 32 767 (positive Zahl) 32 Bit: Ergebnis > 2 147 483 647 (positive Zahl)	0	1	1	1
Überlauf bei Multiplikation, Division 16 Bit: Ergebnis > 32 767 (positive Zahl) 32 Bit: Ergebnis > 2 147 483 647 (positive Zahl)	1	0	1	1
Unterschreitung bei Addition, Subtraktion 16 Bit: Ergebnis < -32 768 (negative Zahl) 32 Bit: Ergebnis < -2 147 483 648 (negative Zahl)	1	0	1	1
Division durch 0	1	1	1	1

Operation	A1	A0	OV	OS
+D: Ergebnis = -4 294 967 296	0	0	1	1
/D oder MOD: Division durch 0	1	1	1	1

7.3 ADD_I : Ganze Zahlen addieren (16 Bit)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	INT	E, A, M, D, L oder Konstante	Erster Summand
IN2	INT	E, A, M, D, L oder Konstante	Zweiter Summand
OUT	INT	E, A, M, D, L	Additionsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

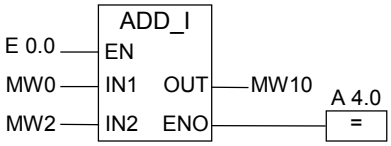
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Ganze Zahlen addieren (16 Bit)**. Diese Operation addiert die Eingänge IN1 und IN2. Das Ergebnis kann an Ausgang OUT abgefragt werden. Liegt das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (16 Bit), haben das OV–Bit und OS–Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Festpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

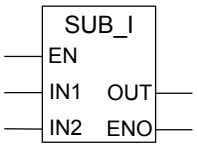
Beispiel



Die Box ADD_I wird aktiviert, wenn E 0.0 = 1 ist. Das Additionsergebnis aus MW0 + MW2 wird in Merkerwort MW10 abgelegt. Befindet sich das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (16 Bit) oder ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

7.4 SUB_I : Ganze Zahlen subtrahieren (16 Bit)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	INT	E, A, M, D, L oder Konstante	Minuend
IN2	INT	E, A, M, D, L oder Konstante	Subtrahend
OUT	INT	E, A, M, D, L	Subtraktionsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

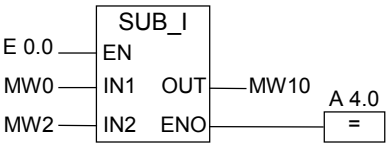
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Ganze Zahlen subtrahieren (16 Bit).** Diese Operation subtrahiert Eingang IN2 von IN1. Das Ergebnis kann an Ausgang OUT abgefragt werden. Liegt das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (16 Bit), haben das OV–Bit und OS–Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Festpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

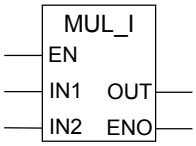
Beispiel



Die Box SUB_I wird aktiviert, wenn E 0.0 = 1 ist. Das Subtraktionsergebnis aus MW0 - MW2 wird in Merkerwort MW10 abgelegt. Befindet sich das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (16 Bit) oder ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

7.5 MUL_I : Ganze Zahlen multiplizieren (16 Bit)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	INT	E, A, M, D, L oder Konstante	Multiplikand
IN2	INT	E, A, M, D, L oder Konstante	Multiplikator
OUT	INT	E, A, M, D, L	Multiplikationsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

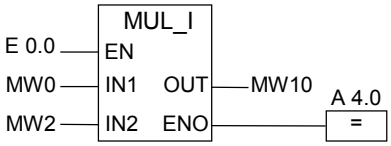
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Ganze Zahlen multiplizieren (16 Bit**). Diese Operation multipliziert Eingang IN1 mit IN2. Das Ergebnis kann an Ausgang OUT abgefragt werden. Liegt das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (16 Bit), haben das OV–Bit und OS–Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Festpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

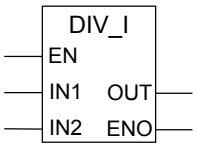
Beispiel



Die Box MUL_I wird aktiviert, wenn E 0.0 = 1 ist. Das Multiplikationsergebnis aus MW0 x MW2 wird in Merkerwort MW10 abgelegt. Befindet sich das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (16 Bit) oder ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

7.6 DIV_I : Ganze Zahlen dividieren (16 Bit)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	INT	E, A, M, D, L oder Konstante	Dividend
IN2	INT	E, A, M, D, L oder Konstante	Divisor
OUT	INT	E, A, M, D, L	Divisionsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

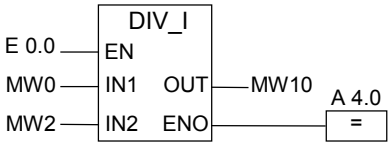
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Ganze Zahlen dividieren (16 Bit)**. Diese Operation dividiert Eingang IN1 durch IN2. Der Quotient dieser Division (ganzzahliger Anteil) kann an Ausgang OUT abgefragt werden. Der Divisionsrest kann nicht abgefragt werden. Liegt der Quotient außerhalb des zulässigen Bereichs für Ganzzahlen (16 Bit), haben das OV–Bit und OS–Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Festpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

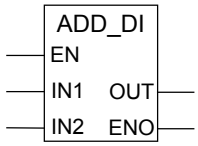
Beispiel



Die Box DIV_I wird aktiviert, wenn E 0.0 = 1 ist. Der Quotient der Division MW0 durch MW2 wird in Merkerwort MW10 abgelegt. Befindet sich der Quotient außerhalb des zulässigen Bereichs für Ganzzahlen (16 Bit) oder ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

7.7 ADD_DI : Ganze Zahlen addieren (32 Bit)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	DINT	E, A, M, D, L oder Konstante	Erster Summand
IN2	DINT	E, A, M, D, L oder Konstante	Zweiter Summand
OUT	DINT	E, A, M, D, L	Additionsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

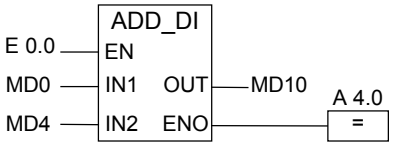
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Ganze Zahlen addieren (32 Bit)**. Diese Operation addiert die Eingänge IN1 und IN2. Das Ergebnis kann an Ausgang OUT abgefragt werden. Liegt das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit), haben das OV–Bit und OS–Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Festpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

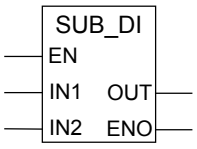
Beispiel



Die Box ADD_DI wird aktiviert, wenn E 0.0 = 1 ist. Das Additionsergebnis aus MD0 + MD4 wird in Merkerdoppelwort MD10 abgelegt. Befindet sich das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit) oder ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

7.8 SUB_DI : Ganze Zahlen subtrahieren (32 Bit)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	DINT	E, A, M, D, L oder Konstante	Minuend
IN2	DINT	E, A, M, D, L oder Konstante	Subtrahend
OUT	DINT	E, A, M, D, L	Subtraktionsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

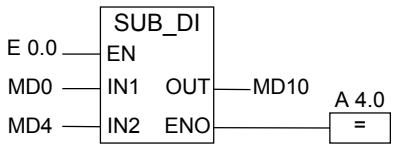
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Ganze Zahlen subtrahieren (32 Bit**). Diese Operation subtrahiert Eingang IN2 von IN1. Das Ergebnis kann an Ausgang OUT abgefragt werden. Liegt das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit), haben das OV–Bit und OS–Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Festpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

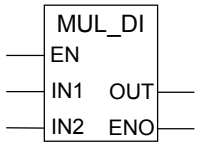
Beispiel



Die Box SUB_DI wird aktiviert, wenn E 0.0 = 1 ist. Das Subtraktionsergebnis aus MD0 - MD4 wird in Merkerdoppelwort MD10 abgelegt. Befindet sich das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit) oder ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

7.9 MUL_DI : Ganze Zahlen multiplizieren (32 Bit)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	DINT	E, A, M, D, L oder Konstante	Multiplikand
IN2	DINT	E, A, M, D, L oder Konstante	Multiplikator
OUT	DINT	E, A, M, D, L	Multiplikationsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

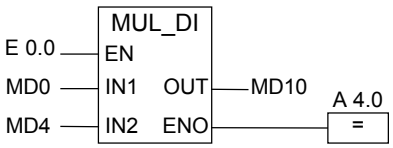
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Ganze Zahlen multiplizieren (32 Bit)**. Diese Operation multipliziert Eingang IN1 mit IN2. Das Ergebnis kann an Ausgang OUT abgefragt werden. Liegt das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit), haben das OV–Bit und OS–Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Festpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

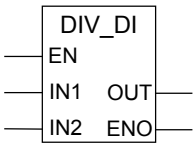
Beispiel



Die Box MUL_DI wird aktiviert, wenn E 0.0 = 1 ist. Das Multiplikationsergebnis aus MD0 x MD4 wird in Merkerdoppelwort MD10 abgelegt. Befindet sich das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit) oder ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

7.10 DIV_DI : Ganze Zahlen dividieren (32 Bit)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	DINT	E, A, M, D, L oder Konstante	Dividend
IN2	DINT	E, A, M, D, L oder Konstante	Divisor
OUT	DINT	E, A, M, D, L	Divisionsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

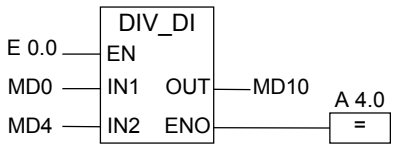
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Ganze Zahlen dividieren (32 Bit)**. Diese Operation dividiert Eingang IN1 durch IN2. Der Quotient dieser Division (ganzzahliger Anteil) kann an Ausgang OUT abgefragt werden. Die Operation Ganze Zahlen dividieren (32 Bit) legt den Quotienten als einfachen 32–Bit–Wert im DINT-Format ab und erzeugt keinen Divisionsrest. Liegt der Quotient außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit), haben das OV–Bit und OS–Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Festpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

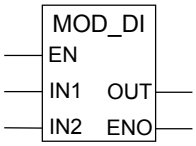
Beispiel



Die Box DIV_DI wird aktiviert, wenn E 0.0 = 1 ist. Der Quotient der Division MD0 durch MD4 wird in Merkerdoppelwort MD10 abgelegt. Befindet sich der Quotient außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit) oder ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

7.11 MOD_DI : Divisionsrest gewinnen (32 Bit)

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	DINT	E, A, M, D, L oder Konstante	Dividend
IN2	DINT	E, A, M, D, L oder Konstante	Divisor
OUT	DINT	E, A, M, D, L	Divisionsrest
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

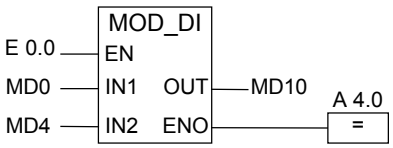
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Divisionsrest gewinnen (32 Bit)**. Diese Operation dividiert Eingang IN1 durch IN2. Der Divisionsrest kann an Ausgang OUT abgefragt werden. Liegt das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit) haben das OV–Bit und OS–Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Festpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

Beispiel



Die Box MOD_DI wird aktiviert, wenn E 0.0 = 1 ist. Der Divisionsrest aus der Division MD0 durch MD4 wird in Merkerdoppelwort MD10 abgelegt. Befindet sich das Ergebnis außerhalb des zulässigen Bereichs für Ganzzahlen (32 Bit) oder ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

8 Gleitpunkt-Funktionen

8.1 Gleitpunkt-Funktionen Übersicht

Beschreibung

Die Gleitpunktzahlen gehören zum Datentyp REAL. Mit den Gleitpunkt-Funktionen können Sie folgende arithmetische Operationen mit **zwei Gleitpunktzahlen** (32 Bit, IEEE 754) ausführen:

- ADD_R Addieren
- SUB_R Subtrahieren
- MUL_R Multiplizieren
- DIV_R Dividieren

Folgende Funktionen können Sie mit **einer Gleitpunktzahl** (32 Bit, IEEE 754) ausführen:

- Bilden des Absolutwertes (ABS) einer Gleitpunktzahl
- Bilden des Quadrats (SQR) bzw. der Quadratwurzel (SQRT) einer Gleitpunktzahl
- Bilden des Exponentialwertes einer Gleitpunktzahl (EXP) auf der Basis e (= 2,71828...)
- Bilden des natürlichen Logarithmus (LN) einer Gleitpunktzahl
- Bilden von trigonometrischen Funktionen von einem Winkel, der als Gleitpunktzahl dargestellt ist:
  - Sinus (SIN) und Arcussinus (ASIN)
  - Cosinus (COS) und Arcuscosinus (ACOS)
  - Tangens (TAN) und Arcustangens (ATAN)
  -

Auswerten der Bits im Statuswort bei Gleitpunkt-Funktionen

8.2 Auswerten der Bits im Statuswort bei Gleitpunkt-Funktionen

Beschreibung

Die Gleitpunkt-Funktionen beeinflussen die Bits A1, A0, OV und OS im Statuswort.

Die folgenden Tabellen zeigen den Signalzustand der Bits im Statuswort für die Ergebnisse von Operationen mit Gleitpunktzahlen (32 Bit).

Gültiger Bereich	**A1**	**A0**	**OV**	**OS**
+0, -0 (Null)	0	0	0	*
-3.402823E+38 < Ergebnis < -1.175494E-38 (negative Zahl)	0	1	0	*
+1.175494E-38 < Ergebnis < 3.402824E+38 (positive Zahl)	1	0	0	*

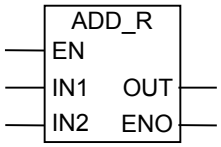
* Das OS-Bit wird vom Ergebnis der Operation nicht beeinflußt.

Ungültiger Bereich	**A1**	**A0**	**OV**	**OS**
Unterschreitung -1.175494E-38 < Ergebnis < - 1.401298E-45 (negative Zahl)	0	0	1	1
Unterschreitung +1.401298E-45 < Ergebnis < +1.175494E-38 (positive Zahl)	0	0	1	1
Überlauf Ergebnis < -3.402823E+38 (negative Zahl)	0	1	1	1
Überlauf Ergebnis > 3.402823E+38 (positive Zahl)	1	0	1	1
keine gültige Gleitpunktzahl oder unzulässige Operation (Eingangswert außerhalb des gültigen Wertebereichs)	1	1	1	1

8.3 Grundoperationen

8.3.1 ADD_R : Gleitpunktzahlen addieren

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	REAL	E, A, M, D, L oder Konstante	Erster Summand
IN2	REAL	E, A, M, D, L oder Konstante	Zweiter Summand
OUT	REAL	E, A, M, D, L	Additionsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

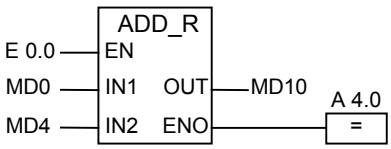
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Gleitpunktzahlen addieren**. Diese Operation addiert Eingang IN1 und IN2. Das Ergebnis kann an Ausgang OUT abgefragt werden. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl, haben das OV-Bit und OS-Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Gleitpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

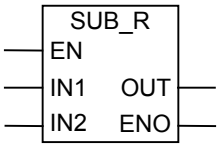
Beispiel



Die Box ADD_R wird aktiviert, wenn E 0.0 = 1 ist. Das Additionsergebnis aus MD0 + MD4 wird in Merkerdoppelwort MD10 abgelegt. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl und ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

8.3.2 SUB_R : Gleitpunktzahlen subtrahieren

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	REAL	E, A, M, D, L oder Konstante	Minuend
IN2	REAL	E, A, M, D, L oder Konstante	Subtrahend
OUT	REAL	E, A, M, D, L	Subtraktionsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

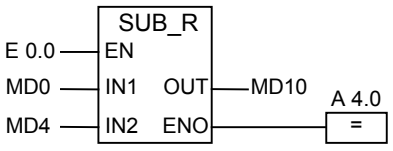
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Gleitpunktzahlen subtrahieren**. Diese Operation subtrahiert Eingang IN2 von IN1. Das Ergebnis kann an Ausgang OUT abgefragt werden. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl, haben das OV-Bit und das OS-Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Gleitpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

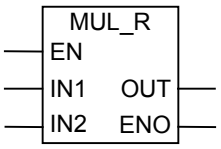
Beispiel



Die Box SUB_R wird aktiviert, wenn E 0.0 = 1 ist. Das Subtraktionsergebnis aus MD0 - MD4 wird in Merkerdoppelwort MD10 abgelegt. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl und ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

8.3.3 MUL_R : Gleitpunktzahlen multiplizieren

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	REAL	E, A, M, D, L oder Konstante	Multiplikand
IN2	REAL	E, A, M, D, L oder Konstante	Multiplikator
OUT	REAL	E, A, M, D, L	Multiplikationsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

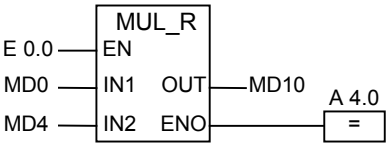
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Gleitpunktzahlen multiplizieren**. Diese Operation multipliziert Eingang IN1 mit IN2. Das Ergebnis kann an Ausgang OUT abgefragt werden. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl, haben das OV-Bit und das OS-Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Gleitpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

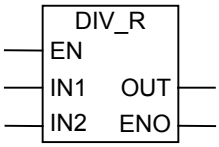
Beispiel



Die Box MUL_R wird aktiviert, wenn E 0.0 = 1 ist. Das Multiplikationsergebnis aus MD0 x MD4 wird in Merkerdoppelwort MD10 abgelegt. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl und ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

8.3.4 DIV_R : Gleitpunktzahlen dividieren

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	REAL	E, A, M, D, L oder Konstante	Dividend
IN2	REAL	E, A, M, D, L oder Konstante	Divisor
OUT	REAL	E, A, M, D, L	Divisionsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

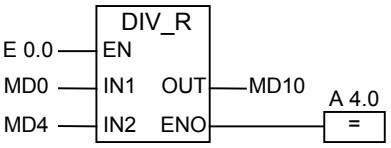
Ein Signalzustand von "1" am Freigabeeingang EN aktiviert die Operation **Gleitpunktzahlen dividieren**. Diese Operation dividiert Eingang IN1 durch IN2. Das Ergebnis kann an Ausgang OUT abgefragt werden. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl, haben das OV-Bit und das OS-Bit den Wert "1" und ENO den Wert "0".

Siehe auch Auswerten der Bits im Statuswort bei Gleitpunkt-Funktionen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

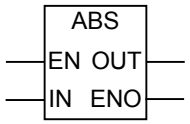
Beispiel



Die Box DIV_R wird aktiviert, wenn E 0.0 = 1 ist. Das Ergebnis der Division MD0 durch MD4 wird in Merkerdoppelwort MD10 abgelegt. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl und ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen und die Operation wird nicht ausgeführt.

8.3.5 ABS : Bilden des Absolutwertes einer Gleitpunktzahl

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Eingangswert: Gleitpunktzahl
OUT	REAL	E, A, M, D, L	Ausgangswert: Absolutwert der Gleitpunktzahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

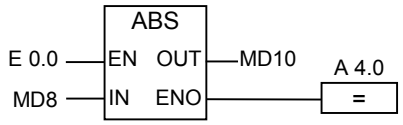
Beschreibung

Mit der Operation **Bilden des Absolutwertes einer Gleitpunktzahl** können Sie den Absolutwert einer Gleitpunktzahl bilden.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	-	-	-	-	0	X	X	1

Beispiel



Ist E 0.0 = 1, dann wird der Absolutwert von MD8 an MD12 ausgegeben:

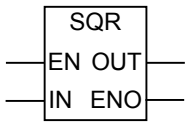
MD8 = - 6,234 -> MD12 = 6,234

Ausgang A 4.0 ist "0", wenn die Umwandlung nicht ausgeführt wird (ENO = EN = 0).

8.4 Erweiterte Operationen

8.4.1 SQR : Bilden des Quadrats einer Gleitpunktzahl

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Quadrat der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

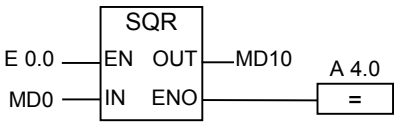
Beschreibung

Mit der Operation **Bilden des Quadrats einer Gleitpunktzahl** können Sie eine Gleitpunktzahl quadrieren. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl, haben das OV-Bit und OS-Bit den Wert "1" und ENO den Wert "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

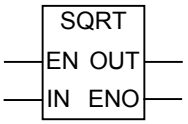
Beispiel



Die Box SQR wird aktiviert, wenn E 0.0 = 1 ist. Das Ergebnis von SQR (MD0) wird in Merkerdoppelwort MD10 abgelegt. Ist MD0 < 0 oder ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl und ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen.

8.4.2 SQRT : Bilden der Quadratwurzel einer Gleitpunktzahl

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Quadratwurzel der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

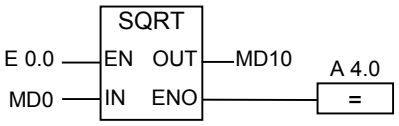
Beschreibung

Mit der Operation **Bilden der Quadratwurzel einer Gleitpunktzah**l können Sie die Quadratwurzel einer Gleitpunktzahl ziehen. Diese Operation gibt ein positives Ergebnis aus, wenn der Operand größer als "0" ist. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl, haben das OV-Bit und OS-Bit den Wert "1" und ENO den Wert "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

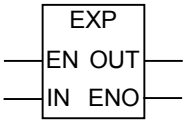
Beispiel



Die Box SQRT wird aktiviert, wenn E 0.0 = 1 ist. Das Ergebnis von SQRT (MD0) wird in Merkerdoppelwort MD10 abgelegt. Ist MD0 < 0 oder ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl und ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen.

8.4.3 EXP : Bilden des Exponentialwerts einer Gleitpunktzahl

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Exponent der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

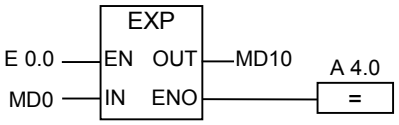
Beschreibung

Mit der Operation **Bilden des Exponentialwerts einer Gleitpunktzahl** können Sie den Exponentialwert einer Gleitpunktzahl auf der Basis e (= 2,71828...) bilden. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl, haben das OV-Bit und OS-Bit den Wert "1" und ENO den Wert "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

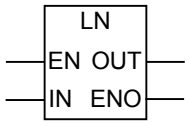
Beispiel



Die Box EXP wird aktiviert, wenn E 0.0 = 1 ist. Das Ergebnis von EXP (MD0) wird in Merkerdoppelwort MD10 abgelegt. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl und ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen.

8.4.4 LN : Bilden des natürlichen Logarithmus einer Gleitpunktzahl

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Natürlicher Logarithmus der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

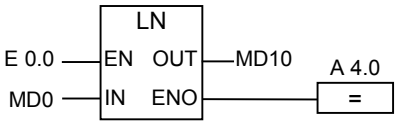
Beschreibung

Mit der Operation **Bilden des natürlichen Logarithmus einer Gleitpunktzahl** können Sie den natürlichen Logarithmus einer Gleitpunktzahl bilden. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl, haben das OV-Bit und OS-Bit den Wert "1" und ENO den Wert "0".

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

Beispiel



Die Box LN wird aktiviert, wenn E 0.0 = 1 ist. Das Ergebnis von LN (MD0) wird in Merkerdoppelwort MD10 abgelegt. Ist MD0 < 0 oder ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl und ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen.

8.4.5 Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen

Beschreibung

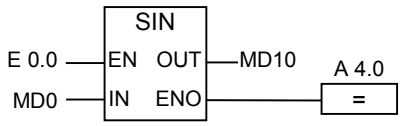
Mit den folgenden Operationen können Sie trigonometrische Funktionen von Winkeln, die als Gleitpunktzahlen (32 Bit, IEEE 754) dargestellt sind, bilden:

Operation	Bedeutung
SIN	Bilden des Sinus einer Gleitpunktzahl von einem Winkel, der im Bogenmaß angegeben wird.
ASIN	Bilden des Arcussinus einer Gleitpunktzahl. Das Ergebnis ist ein Winkel, der im Bogenmaß angegeben wird. Der Wert liegt in dem folgenden Bereich: - π / 2 <= Arcussinus <= + π / 2, wobei π = 3,14...
COS	Bilden des Cosinus einer Gleitpunktzahl von einem Winkel, der im Bogenmaß angegeben wird.
ACOS	Bilden des Arcuscosinus einer Gleitpunktzahl. Das Ergebnis ist ein Winkel, der im Bogenmaß angegeben wird. Der Wert liegt in dem folgenden Bereich: 0 <= Arcuscosinus <= + π, wobei π = 3,14...
TAN	Bilden des Tangens einer Gleitpunktzahl von einem Winkel, der im Bogenmaß angegeben wird.
ATAN	Bilden des Arcustangens einer Gleitpunktzahl. Das Ergebnis ist ein Winkel, der im Bogenmaß angegeben wird. Der Wert liegt in dem folgenden Bereich: - π / 2 <= Arcustangens <= + π / 2, wobei π = 3,14...

Statuswort

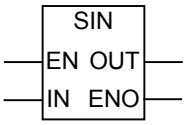
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	X	0	X	X	1

Beispiel



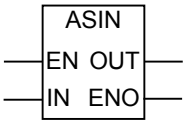
Die Box SIN wird aktiviert, wenn E 0.0 = 1 ist. Das Ergebnis von SIN (MD0) wird in Merkerdoppelwort MD10 abgelegt. Ist einer der Eingänge oder das Ergebnis keine Gleitpunktzahl und ist der Signalzustand von E 0.0 = 0, wird dem Ausgang A 4.0 Signal "0" zugewiesen.

Symbol



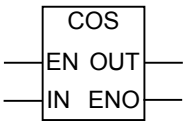
Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Sinus der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Symbol



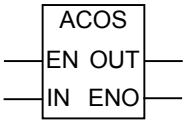
Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Arcussinus der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Symbol



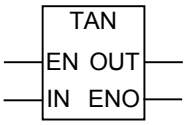
Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Cosinus der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Symbol



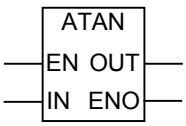
Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Arcuscosinus der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Tangens der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Symbol

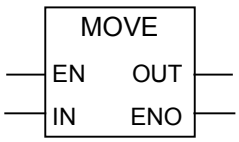


Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	REAL	E, A, M, D, L oder Konstante	Zahl
OUT	REAL	E, A, M, D, L	Arcustangens der Zahl
ENO	BOOL	E, A, M, D, L	Freigabeausgang

9 Verschieben

9.1 MOVE : Wert übertragen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN	Alle elementaren Datentyen mit einer Länge von 8, 16 oder 32 Bit	E, A, M, D, L oder Konstante	Quellwert
OUT	Alle elementaren Datentypen mit einer Länge von 8, 16 oder 32 Bit	E, A, M, D, L	Zieladresse
ENO	BOOL	E, A, M, D, L	Freigabeausgang

Beschreibung

Mit der Operation **Wert übertragen** können Sie Variablen mit spezifischen Werten vorbelegen.

Der Wert, der an Eingang IN angegeben ist, wird in den Operanden kopiert, der an Ausgang OUT angegeben ist. ENO hat den gleichen Signalzustand wie EN.

Die Operation **Wert übertragen** kann mit der Box MOVE alle elementaren Datentypen kopieren, die eine Länge von 8, 16 oder 32 Bits haben. Anwenderdefinierte Datentypen wie Felder oder Strukturen müssen mit der Systemfunktion SFC 20 "BLKMOV" kopiert werden.

Die Operation **Wert übertragen** wird vom Master Control Relay beeinflußt.

Statuswort

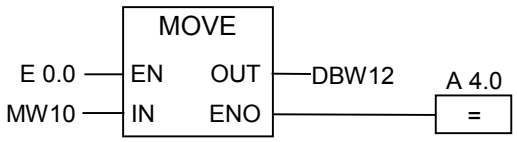
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	-	-	-	-	0	1	1	1

Hinweis

Bei der Übertragung eines Wertes in einen Datentyp anderer Länge werden höherwertige Bytes bei Bedarf abgeschnitten oder mit Nullen aufgefüllt. Beispiele:

Doppelwort	**1111 1111**	**0000 1111**	**1111 0000**	**0101 0101**
Übertragung	**Ergebnis**			
in ein Doppelwort:	1111 1111	0000 1111	1111 0000	0101 0101
in ein Byte:				0101 0101
in ein Wort:			1111 0000	0101 0101
Byte				**1111 0000**
Übertragung	**Ergebnis**			
in ein Byte:				1111 0000
in ein Wort:			0000 0000	1111 0000
in ein Doppelwort:	0000 0000	0000 0000	0000 0000	1111 0000

Beispiel



Die Operation wird ausgeführt, wenn E 0.0 = 1 ist. Der Inhalt von MW10 wird in Datenwort 12 des geöffneten DB kopiert.

Wird die Operation ausgeführt, ist A 4.0 = 1.

10 Programmsteuerung

10.1 Programmsteuerungsoperationen Übersicht

Beschreibung

Folgende Operationen stehen Ihnen zur Programmsteuerung zur Verfügung:

- CALL FC/SFC aufrufen ohne Parameter
- CALL_FB FB als Box aufrufen
- CALL_FC FC als Box aufrufen
- CALL_SFB System-FB als Box aufrufen
- CALL_SFC System-FC als Box aufrufen
- Multiinstanzen aufrufen
- Baustein aus einer Bibliothek aufrufen

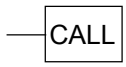
- Funktionen des Master Control Relay
- Wichtige Hinweise zur MCR-Funktionalität
- MCR< Master Control Relay einschalten
- MCR> Master Control Relay ausschalten
- MCRA Master Control Relay Anfang
- MCRD Master Control Relay Ende

- RET Springe zurück

10.2 CALL : FC/SFC aufrufen ohne Parameter

Symbol

<FC-/SFC-Nummer>



Parameter	Datentyp	Speicher-bereich	Beschreibung
Nummer	BLOCK_FC	-	Nummer der FC oder SFC (z. B. FC 10 oder SFC 59). Welche SFCs zur Verfügung stehen, hängt von Ihrer CPU ab. Ein bedingter Aufruf mit einem Parameter vom Datentyp BLOCK_FC als Operand ist nur im FB möglich, nicht in der FC.

Beschreibung

Mit der Operation **CALL** (FC/SFC aufrufen ohne Parameter) können Sie eine Funktion (FC) oder Systemfunktion (SFC) aufrufen, die keine Parameter hat. Abhängig von der vorangehenden Verknüpfung handelt es sich um einen absoluten oder bedingten Aufruf (siehe Beispiel).

Im Anweisungsteil einer Funktion (FC) können Sie bei einem bedingten Aufruf als Operand keinen Parameter vom Datentyp BLOCK_FC angeben. Einem Funktionsbaustein (FB) jedoch können Sie als Operanden einen Parameter vom Typ BLOCK_FC angeben.

Ein bedingter Aufruf wird nur dann ausgeführt, wenn das VKE "1" ist. Wird ein bedingter Aufruf nicht ausgeführt, ist das VKE nach der Aufrufoperation "0". Wird die Operation ausgeführt, arbeitet sie folgendermaßen:

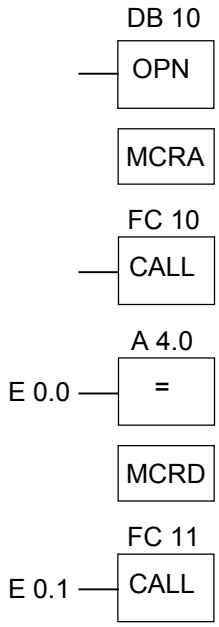
- Sie speichert die Rücksprungadresse des aufrufenden Bausteins.
- Sie speichert die beiden Datenbaustein-Register (Datenbaustein und Instanz-Datenbaustein).
- Sie aktualisiert den Lokaldatenbereich für die aufgerufene FC oder SFC.
- Sie schiebt das MA-Bit (aktives MCR-Bit) in den Baustein-Stack (B-Stack).

Anschließend wird die Programmbearbeitung in der aufgerufenen Funktion oder Systemfunktion fortgesetzt.

Statuswort

		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Bedingt	schreibt:	-	-	-	-	0	0	1	1	0
Absolut	schreibt:	-	-	-	-	0	0	1	-	0

Beispiel



Wird der absolute Aufruf von FC 10 ausgeführt, arbeitet die Operation CALL wie folgt:

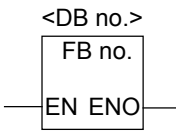
- Sie speichert die Rücksprungadresse des aktuellen FB.
- Sie speichert die Selektoren für DB 10 und den Instanz-DB des FB.
- Sie schiebt das MA-Bit, das von der Operation MCRA auf "1" gesetzt wurde, in den Baustein-Stack (B-Stack) und setzt es für die aufgerufene FC 10 auf "0" zurück.

Die Programmbearbeitung wird in FC 10 fortgesetzt. Möchten Sie die Funktion MCR in FC 10 verwenden, müssen Sie sie dort neu aktivieren. Ist FC 10 beendet, kehrt die Programmbearbeitung zum aufrufenden FB zurück. Das MA-Bit wird wiederhergestellt. DB 10 und der Instanz-DB des anwenderdefinierten FB sind wieder die aktuellen DBs, und zwar unabhängig davon, welche DBs von FC 10 verwendet wurden.

Nach Rücksprung von FC 10 wird der Signalzustand von E 0.0 dem Ausgang A 4.0 zugewiesen. Bei dem Aufruf von FC 11 handelt es sich um einen bedingten Aufruf, der nur ausgeführt wird, wenn Eingang E 0.1 = 1 ist. Wird der Aufruf ausgeführt, ist die Funktion die gleiche wie beim Aufruf von FC 10.

10.3 CALL_FB : FB als Box aufrufen

Symbol



Das Symbol ist von dem Funktionsbaustein abhängig (je nachdem, ob bzw. wie viele Parameter vorhanden sind). EN, ENO und der Name bzw. die Nummer des FB müssen vorhanden sein.

Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D	Freigabeeingang
ENO	BOOL	E, A, M, L, D	Freigabeausgang
FB no. DB no.	BLOCK_FB BLOCK_DB	- -	Nummer des FB/DB, Bereich ist von der CPU abhängig

Beschreibung

CALL_FB (FB als Box aufrufen) wird ausgeführt, wenn EN = 1 ist. Die Operation **CALL_FB** arbeitet folgendermaßen:

- Sie speichert die Rücksprungadresse des aufrufenden Bausteins.
- Sie speichert die Auswahldaten für die beiden aktuellen Datenbausteine (DB und Instanz-DB).
- Sie aktualisiert den Lokaldatenbereich für den aufgerufenen Funktionsbaustein.
- Sie schiebt das MA-Bit (aktives MCR-Bit) in den Baustein-Stack (B-Stack).

Statuswort

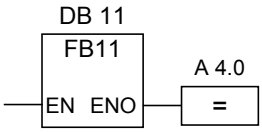
		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Bedingt	schreibt:	X	-	-	-	0	0	X	X	X
Absolut	schreibt:	-	-	-	-	0	0	X	X	X

Beispiel

Netzwerk 1

Netzwerk 2

Netzwerk 3



Netzwerk 4

Bei den oben dargestellten Netzwerken handelt es sich um Programmteile eines vom Anwender geschriebenen Funktionsbausteins. DB 10 wird in diesem Baustein geöffnet und das MCR aktiviert. Wenn der absolute Aufruf von FB 11 ausgeführt wird, geschieht folgendes:

Die Rücksprungadresse des aufrufenden Funktionsbausteins und die Auswahldaten für DB 10 und den Instanz-Datenbaustein des aufrufenden Funktionsbausteins werden gespeichert. Das MA-Bit, das von der Funktion MCRA auf "1" gesetzt wurde, wird in den B-Stack geschoben und dann für den aufgerufenen Funktionsbaustein FB 11 auf "0" gesetzt. Die Programmbearbeitung wird in FB 11 fortgesetzt. Benötigt FB 11 das MCR, muß das MCR im Funktionsbaustein wieder aktiviert werden. Der Zustand des VKE muß durch die Operation [SAVE] im BIE-Bit gespeichert werden, um eine Fehlerauswertung im aufrufenden FB vornehmen zu können. Ist die Bearbeitung des FB 11 beendet, geht die Programmbearbeitung zurück zum aufrufenden Funktionsbaustein. Das MA-Bit wird wiederhergestellt und der Instanz-Datenbaustein des vom Anwender geschriebenen Funktionsbausteins wird wieder zum geöffneten DB. Wird der FB 11 korrekt bearbeitet, ist ENO = 1 und somit A 4.0 = 1.

Hinweis

Bei FB/SFB-Aufrufen geht die Nummer des zuvor geöffneten Datenbausteins verloren. Der benötigte DB muß erneut geöffnet werden.

10.4 CALL_FC : FC als Box aufrufen

Symbol

Das Symbol ist von der Funktion abhängig (je nachdem, ob bzw. wie viele Parameter vorhanden sind). EN, ENO und der Name bzw. die Nummer der FC müssen vorhanden sein.

Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D	Freigabeeingang
ENO	BOOL	E, A, M, L, D	Freigabeausgang
FC no.	BLOCK_FC	-	Nummer der FC, Bereich ist von der CPU abhängig

Beschreibung

CALL_FC (FC als Box aufrufen) ruft eine Funktion (FC) auf, wenn EN = 1 ist. Die Funktion **CALL_FC** arbeitet folgendermaßen:

- Sie speichert die Rücksprungadresse des aufrufenden Bausteins.
- Sie aktualisiert den Lokaldatenbereich für die aufgerufene Funktion.
- Sie schiebt das MA-Bit (aktives MCR-Bit) in den Baustein-Stack (B-Stack).

Anschließend wird die Programmbearbeitung in der aufgerufenen Funktion fortgesetzt.

Zur Ermittlung des ENO wird das BIE-Bit abgefragt, diesem muß vom Anwender im aufgerufenen Baustein mit [SAVE] der gewünschte Zustand (Fehlerauswertung) zugewiesen werden.

Wenn Sie eine FC aufrufen und die Variablendeklarationstabelle des aufgerufenen Bausteins über Deklarationen vom Typ IN, OUT und IN_OUT verfügt, werden diese Variablen im Programm des aufrufenden Bausteins als Liste der Formalparameter angezeigt.

Beim Aufruf der FCs **müssen Sie zwingend** den Formalparametern Aktualparameter an der Aufrufstelle zuordnen. Eventuelle Anfangswerte in der FC-Deklaration sind ohne Bedeutung.

Statuswort

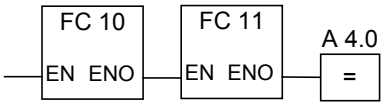
		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Bedingt	schreibt:	X	-	-	-	0	0	X	X	X
Absolut	schreibt:	-	-	-	-	0	0	X	X	X

Beispiel

Netzwerk 1

Netzwerk 2

Netzwerk 3



Bei den oben dargestellten Netzwerken handelt es sich um Programmteile eines vom Anwender geschriebenen Funktionsbausteins. DB 10 wird in diesem Baustein geöffnet und das MCR aktiviert. Wird der absolute Aufruf von FC 10 ausgeführt, geschieht folgendes:

Die Rücksprungadresse des aufrufenden Funktionsbausteins und die Auswahldaten für DB 10 und den Instanz-Datenbaustein des aufrufenden Funktionsbausteins werden gespeichert. Das MA-Bit, das von der Operation MCRA auf "1" gesetzt wurde, wird in den B-Stack geschoben und dann für den aufgerufenen Baustein FC 10 auf "0" gesetzt. Die Programmbearbeitung wird in FC 10 fortgesetzt. Benötigt FC 10 das MCR, dann muß das MCR in FC 10 wieder aktiviert werden. Der Zustand des VKE muß durch die Operation [SAVE] im BIE-Bit gespeichert werden, um eine Fehlerauswertung im aufrufenden FB vornehmen zu können. Ist die Bearbeitung von FC 10 beendet, geht die Programmbearbeitung zurück zum aufrufenden Funktionsbaustein. Das MA-Bit wird wiederhergestellt. Nach Bearbeitung der FC 10 wird in Abhängigkeit vom ENO das Programm im aufrufenden FB fortgesetzt:

ENO = 1 FC 11 wird bearbeitet

ENO = 0 Bearbeitung beginnt im nächsten Netzwerk

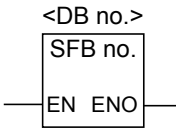
Wird auch FC 11 korrekt bearbeitet, ist ENO = 1 und somit A 4.0 = 1.

Hinweis

Nach dem Rücksprung in den aufrufenden Baustein ist nicht immer sichergestellt, daß der zuvor geöffnete DB wieder geöffnet ist. Beachten Sie Bitte den Hinweis in der Liesmich-Datei.

10.5 CALL_SFB : System-FB als Box aufrufen

Symbol



Das Symbol ist von dem Systemfunktionsbaustein abhängig (je nachdem, ob bzw. wie viele Parameter vorhanden sind). EN, ENO und der Name bzw. die Nummer des SFB müssen vorhanden sein.

Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D	Freigabeeingang
ENO	BOOL	E, A, M, L, D	Freigabeausgang
SFB no. DB no.	BLOCK_SFB BLOCK_DB	- -	Nummer des SFB/DB, Bereich ist von der CPU abhängig

Beschreibung

CALL_SFB (SFB als Box aufrufen) wird ausgeführt, wenn EN = 1 ist. Die Operation **CALL_SFB** arbeitet folgendermaßen:

- Sie speichert die Rücksprungadresse des aufrufenden Bausteins.
- Sie speichert die Auswahldaten für die beiden aktuellen Datenbausteine (DB und Instanz-DB).
- Sie aktualisiert den Lokaldatenbereich für den aufgerufenen Systemfunktionsbaustein.
- Sie schiebt das MA-Bit (aktives MCR-Bit) in den Baustein-Stack (B-Stack).

Anschließend wird die Programmbearbeitung in dem aufgerufenen Systemfunktionsbaustein fortgesetzt. ENO ist "1", wenn der Systemfunktionsbaustein aufgerufen wurde (EN = 1) und keine Fehler aufgetreten sind.

Statuswort

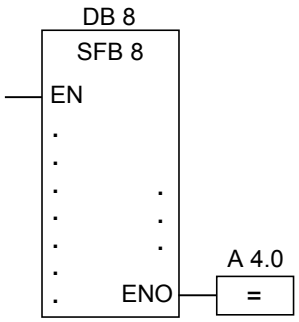
		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Bedingt	schreibt:	X	-	-	-	0	0	X	X	X
Absolut	schreibt:	-	-	-	-	0	0	X	X	X

Beispiel

Netzwerk 1

Netzwerk 2

Netzwerk 3



Netzwerk 4

Bei den oben dargestellten Netzwerken handelt es sich um Programmteile eines vom Anwender geschriebenen Funktionsbausteins. DB 10 wird in diesem Baustein geöffnet und das MCR aktiviert. Wenn der absolute Aufruf von SFB 8 ausgeführt wird, geschieht folgendes:

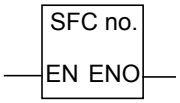
Die Rücksprungadresse des aufrufenden Funktionsbausteins und die Auswahldaten für DB 10 und den Instanz-Datenbaustein des aufrufenden Funktionsbausteins werden gespeichert. Das MA-Bit, das von der Funktion MCRA auf "1" gesetzt wurde, wird in den B-Stack geschoben und dann für den aufgerufenen Systemfunktionsbaustein SFB 8 auf "0" gesetzt. Die Programmbearbeitung wird in SFB 8 fortgesetzt. Ist die Bearbeitung von SFB 8 beendet, geht die Programmbearbeitung zurück zum aufrufenden Funktionsbaustein. Das MA-Bit wird wiederhergestellt, und der Instanz-Datenbaustein des vom Anwender geschriebenen Funktionsbausteins wird wieder zum aktuellen Instanz-DB. Wird der SFB 8 korrekt bearbeitet, ist ENO = 1 und somit A 4.0 = 1.

Hinweis

Bei FB/SFB-Aufrufen geht die Nummer des zuvor geöffneten Datenbausteins verloren. Der benötigte DB muß erneut geöffnet werden.

10.6 CALL_SFC : System-FC als Box aufrufen

Symbol



Das Symbol ist von der Systemfunktion abhängig (je nachdem, ob bzw. wie viele Parameter vorhanden sind). EN, ENO und der Name bzw. die Nummer der SFC müssen vorhanden sein.

Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D	Freigabeeingang
ENO	BOOL	E, A, M, L, D	Freigabeausgang
SFC no.	BLOCK_SFC	-	Nummer der SFC, Bereich ist von der CPU abhängig

Beschreibung

CALL_SFC (System-FC als Box aufrufen) ruft eine Systemfunktion auf, wenn EN = 1 ist. Die Operation **CALL_SFC** arbeitet folgendermaßen:

- Sie speichert die Rücksprungadresse des aufrufenden Bausteins.
- Sie aktualisiert den Lokaldatenbereich für die aufgerufene Funktion.
- Sie schiebt das MA-Bit (aktives MCR-Bit) in den Baustein-Stack (B-Stack).

Anschließend wird die Programmbearbeitung in der aufgerufenen Systemfunktion fortgesetzt. ENO ist "1", wenn die Funktion aufgerufen wurde (EN = 1) und keine Fehler aufgetreten sind.

Statuswort

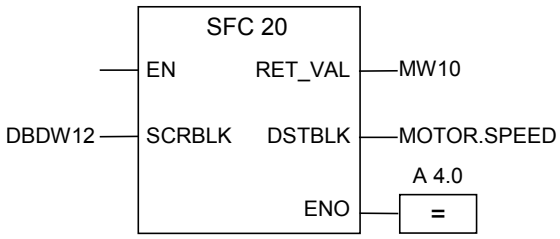
		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
Bedingt	schreibt:	X	-	-	-	0	0	X	X	X
Absolut	schreibt:	-	-	-	-	0	0	X	X	X

Beispiel

Netzwerk 1

Netzwerk 2

Netzwerk 3



Bei den oben dargestellten Netzwerken handelt es sich um Programmteile eines vom Anwender geschriebenen Funktionsbausteins. DB 10 wird in diesem Baustein geöffnet und das MCR aktiviert. Wird der absolute Aufruf von SFC 20 ausgeführt, geschieht folgendes:

Die Rücksprungadresse des aufrufenden Funktionsbausteins und die Auswahldaten für DB 10 und den Instanz-Datenbaustein des aufrufenden Funktionsbausteins werden gespeichert. Das MA-Bit, das von der Funktion MCRA auf "1" gesetzt wurde, wird in den B-Stack geschoben und dann für den aufgerufenen Baustein SFC 20 auf "0" gesetzt. Die Programmbearbeitung wird in SFC 20 fortgesetzt. Wenn die Bearbeitung der SFC 20 beendet ist, geht die Programmbearbeitung zurück zum aufrufenden Funktionsbaustein. Das MA-Bit wird wiederhergestellt.

Nach Bearbeitung der SFC 20 wird in Abhängigkeit von ENO das Programm im aufrufenden FB fortgesetzt:

ENO = 1 A 4.0 = 1

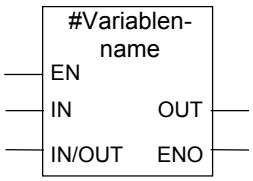
ENO = 0 A 4.0 = 0

Hinweis

Nach dem Rücksprung in den aufrufenden Baustein ist nicht immer sichergestellt, daß der zuvor geöffnete DB wieder geöffnet ist. Beachten Sie Bitte den Hinweis in der Liesmich-Datei.

10.7 Multiinstanzen aufrufen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
ENO	BOOL	E, A, M, D, L	Freigabeausgang
# Variablenname	FB/SFB	-	Name der Multiinstanz

Beschreibung

Eine Multiinstanz entsteht durch die Deklaration einer statischen Variablen vom Datentyp eines Funktionsbausteins. Nur bereits deklarierte Multiinstanzen werden im Programmelementekatalog aufgeführt. Das Symbol einer Multiinstanz verändert sich, je nachdem, ob und wie viele Parameter vorhanden sind. EN, ENO und der Variablenname sind immer vorhanden.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	0	0	X	X	X

10.8 Baustein aus einer Bibliothek aufrufen

Die im SIMATIC Manager bekannten Bibliotheken werden Ihnen zur Auswahl angeboten.

Aus diesen Bibliotheken können Sie Bausteine auswählen,

- die im Betriebssystem Ihrer CPU integriert sind (Bibliotheken "Standard Library" für STEP 7-Projekte der Stufe 3 und "stdlibs (V2)" für STEP 7-Projekte der Version 2),
- die Sie selbst in Bibliotheken abgelegt haben, weil Sie sie mehrfach verwenden wollen.

10.9 Funktionen des Master Control Relay

Wichtige Hinweise zur MCR-Funktionalität

Definition des Master Control Relay (MCR)

Das Master Control Relay wird für das Aktivieren und Deaktivieren des Signalflusses verwendet. Ein deaktivierter Signalfluß entspricht einer Operationsfolge, die einen Nullwert statt des errechneten Wertes schreibt bzw. einer Operationsfolge, die den bestehenden Speicherwert unverändert läßt.

Die Operationen **Zuweisung** und **Konnektor** schreiben eine "0" in den Speicher, wenn das MCR "0" ist. Die Operationen **Ausgang setzen** und **Ausgang rücksetzen** verändern den bestehenden Wert nicht.

Folgende Operationen hängen vom MCR ab:

- # Konnektor
- = Zuweisung
- S Ausgang setzen
- R Ausgang rücksetzen
- SR Flipflop setzen rücksetzen
- RS Flipflop rücksetzen setzen
- MOVE Wert übertragen

Reaktionen der Operationen auf den Signalzustand des MCR

Signalzustand des MCR	Zuweisung, Konnektor	Operand setzen oder rücksetzen	Wert übertragen
0	Schreibt "0" (Imitiert ein Relais, das bei Spannungsabfall in den Ruhezustand geht)	Schreibt nicht (Imitiert ein Relais, das bei Spannungsabfall in seinem aktuellen Zustand bleibt)	Schreibt "0" (Imitiert eine Komponente, die bei Spannungsausfall einen Wert von "0" liefert)
1	Normale Bearbeitung	Normale Bearbeitung	Normale Bearbeitung

10.10 Wichtige Hinweise zur MCR-Funktionalität

Vorsicht bei Bausteinen, in denen mit MCRA das Master Control Relay aktiviert wurde:

- Wenn das MCR abgeschaltet ist, wird in Programmabschnitten zwischen MCR einschalten und MCR ausschalten durch alle Zuweisungen der Wert 0 geschrieben! Das betrifft dann natürlich auch **alle** Boxen, die eine Zuweisung enthalten, einschließlich der Parameterübergabe an Bausteine!
- Das MCR ist genau dann abgeschaltet, wenn vor dem Befehl "MCR einschalten" das VKE = 0 war.

Gefahr: STOP der AS oder undefiniertes Laufzeitverhalten

Der Compiler greift für Adreßberechnungen auch schreibend auf Lokaldaten hinter den in VAR_TEMP definierten temporären Variablen zu. Daher setzen folgende Befehlssequenzen die AS in STOP oder führen zu undefiniertem Laufzeitverhalten:

Formalparameterzugriffe

- Zugriffe auf Komponenten komplexer FC-Parameter vom Typ STRUCT, UDT, ARRAY, STRING
- Zugriffe auf Komponenten komplexer FB-Parameter vom Typ STRUCT, UDT, ARRAY, STRING aus dem Bereich IN_OUT in einem multiinstanzfähigen Baustein (Bausteinversion 2).
- Zugriffe auf Parameter eines multiinstanzfähigen FB (Bausteinversion 2), wenn ihre Adresse größer als 8180.0 ist.
- Zugriff im multiinstanzfähigen FB (Bausteinversion 2) auf einen Parameter vom Typ BLOCK_DB schlägt den DB 0 auf. Nachfolgende Datenzugriffe bringen die CPU in STOP. Bei TIMER, COUNTER, BLOCK_FC, BLOCK_FB wird auch immer T 0, Z 0, FC 0 bzw. FB 0 verwendet.

Parameterübergabe

- Calls, bei denen Parameter übergeben werden.

KOP/FUP

- T-Abzweige und Konnektoren in KOP oder FUP starten mit VKE = 0.

Abhilfe

Lösen Sie die genannten Befehle aus der MCR-Abhängigkeit:

1. **Deaktivieren** Sie das MCR mit Master Control Relay Ende **vor** der betreffenden Anweisung bzw. vor dem betreffenden Netzwerk.
2. **Aktivieren** Sie das MCR mit Master Control Relay Anfang erneut **nach** der betreffenden Anweisung bzw. nach dem betreffenden Netzwerk.

10.11 MCR< / MCR> : Master Control Relay einschalten/ausschalten

Wichtige Hinweise zur MCR-Funktionalität

Symbol

MCR einschalten

Mit der Operation **Master Control Relay einschalten** (MCR<) wird das VKE in den MCR-Stack gespeichert und eine MCR-Zone geöffnet. Die unter "Funktionen des MCR" aufgeführten Operationen werden innerhalb dieser Zone von dem VKE beeinflußt, das beim Öffnen der MCR-Zone in den MCR-Stack gespeichert wird.

Der MCR-Stack kann maximal 8 Einträge enthalten und arbeitet wie ein LIFO-Zwischenspeicher: **l**ast **i**n, **f**irst **o**ut. Ist der Stack voll, erzeugt die Operation **MCR<** eine Fehlermeldung (MCRF).

Symbol

MCR ausschalten

Mit der Operation **Master Control Relay ausschalten** (MCR>) schließen Sie die zuletzt geöffnete MCR-Zone. Dies geschieht durch Löschen des VKE-Eintrags aus dem MCR-Stack. Der freiwerdende Eintrag wird auf "1" gesetzt.

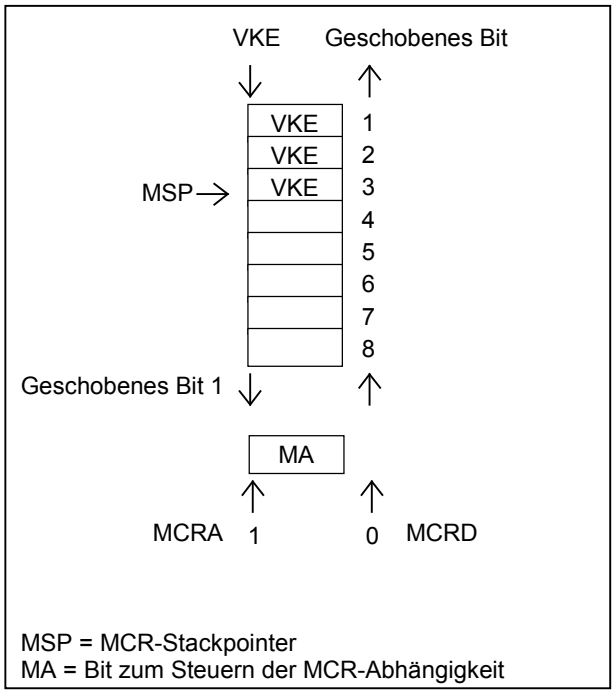
Ist der Stack bereits leer, erzeugt die Operation **Master Control Relay ausschalten** eine Fehlermeldung (MCRF).

MCR-Stack

Das MCR wird von einem Stack gesteuert, der ein Bit breit und acht Einträge tief ist. Das MCR wird so lange aktiviert, wie alle acht Einträge in dem Stack gleich "1" sind. Der MCR-Stack arbeitet wie ein LIFO-Zwischenspeicher: **l**ast **i**n, **f**irst **o**ut. Die Operation **MCR<** kopiert das VKE in den MCR-Stack. Die Operation **MCR>** löscht den letzten Eintrag aus dem Stack und setzt die freigewordene Stackadresse auf "1".

Die Fehlermeldung MCRF wird ausgelöst, z. B. wenn mehr als acht Operationen **MCR>** aufeinander folgen oder versucht wird, bei leerem MCR-Stack die Operation **MCR>** auszuführen.

Die Überwachung des MCR-Stacks folgt dem Stackpointer: 0 = leer, 1 = ein Eintrag, 2 = zwei Einträge, ... 8 = acht Einträge.



Die Operation **MCR<** übernimmt den Signalzustand des VKE und kopiert ihn in das MCR-Bit.

Die Operation **MCR>** setzt das MCR-Bit absolut auf "1". Aufgrund dieser Eigenschaft arbeitet jede weitere Operation zwischen den Operationen MCRA und MCRD unabhängig vom MCR-Bit.

Schachteln der Operationen MCR< und MCR>

Sie können die Operationen **MCR<** und **MCR>** schachteln. Sie können maximal acht Operationen MCR< hintereinander schreiben, bevor Sie eine Operation MCR> einfügen. Die Operationen MCR< und MCR> dürfen Sie in Ihrem Programm immer nur paarweise verwenden.

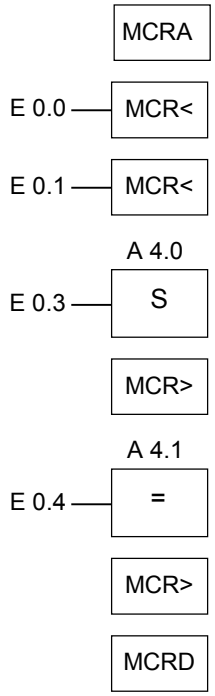
Sind die Operationen MCR< geschachtelt, wird das MCR-Bit der tieferen Schachtelungsebene gebildet. Dann verknüpft die Operation MCR< das aktuelle VKE mit dem aktuellen MCR-Bit entsprechend der UND-Wahrheitstabelle.

Wenn eine Operation MCR> eine Schachtelungsebene beendet, holt sie das MCR-Bit aus der nächsthöheren Ebene.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	1	-	0

Beispiel



Wenn die Operation **MCRA** die Funktion MCR aktiviert, können Sie bis zu acht geschachtelte MCR-Zonen erstellen. In unserem Beispiel gibt es zwei MCR-Zonen. Die erste Operation MCR> arbeitet mit der zweiten Operation MCR< zusammen. Alle Operationen zwischen der zweiten Gruppe von MCR-Klammern (MCR< MCR>) gehören zur zweiten MCR-Zone. Die Operationen werden wie folgt ausgeführt:

- E 0.0 = 1: Der Signalzustand von E 0.4 wird Ausgang A 4.1 zugewiesen.
- E 0.0 = 0: Ausgang A 4.1 ist "0", unabhängig vom Signalzustand an E 0.4
  Ausgang A 4.0 wird nicht verändert, unabhängig vom Signalzustand an E 0.3.
- E 0.0 und E 0.1 = 1: Ausgang A 4.0 wird auf "1" gesetzt, wenn E 0.3 = 1 ist, und A 4.1 = E 0.4.
- E 0.1 = 0: Ausgang A 4.0 wird nicht geändert, unabhängig vom Signalzustand an E 0.3 und E 0.0.

10.12 MCRA / MCRD : Master Control Relay Anfang/Ende

Wichtige Hinweise zur MCR-Funktionalität

Symbol

MCR-Anfang

Mit der Operation **Master Control Relay Anfang** schalten Sie die MCR-Abhängigkeit der nachfolgenden Befehle ein. Nach diesem Befehl können Sie mit den Operationen MCR einschalten/ausschalten die MCR-Zonen programmieren. Wenn Ihr Programm einen MCR-Bereich aktiviert, hängen alle MCR-Aktionen vom Inhalt des MCR-Stack ab.

Symbol

MCR-Ende

Mit der Operation **Master Control Relay Ende** schalten Sie die MCR-Abhängigkeit der nachfolgenden Befehle aus. Nach dieser Operation können Sie keine MCR-Zonen mehr programmieren. Wenn Ihr Programm einen MCR-Bereich deaktiviert, ist das MCR immer stromführend - unabhängig von den Einträgen im MCR-Stack.

Der MCR-Stack und das Bit, das seine Abhängigkeit steuert (das MA-Bit), beziehen sich auf die jeweilige Ebene und müssen jedesmal gesichert und geholt werden, wenn in der Sequenzebene umgeschaltet wird. Zu Beginn jeder Sequenzebene werden die MCR-Eingabebits 1 bis 8 auf "1" gesetzt, MCR-Stackpointer = 0, und MA-Bit = 0.

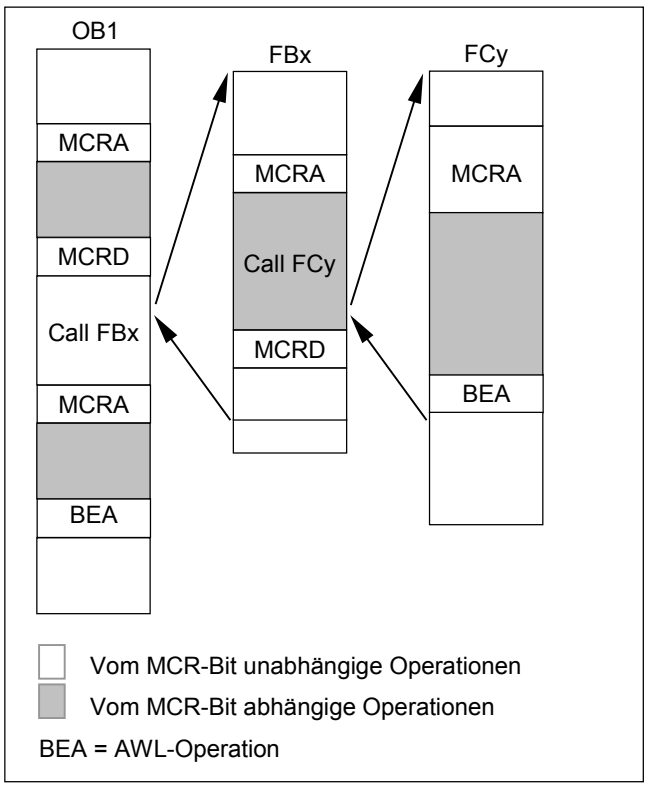
Der MCR-Stack wird von Baustein zu Baustein weitergegeben und das MA-Bit wird bei jedem Bausteinaufruf gesichert und auf "0" gesetzt. Am Ende des Bausteins wird es wiedergeholt.

Das MCR kann so implementiert werden, daß es die Laufzeit code-generierender CPUs optimiert. Grund hierfür ist, daß die Abhängigkeit vom MCR nicht zum Baustein weitergeleitet wird, sondern explizit durch eine MCRA-Operation aktiviert werden muß. Eine code-generierende CPU erkennt diese Operation und generiert den zusätzlichen Code, der für die Auswertung des MCR-Stack notwendig ist, bis sie eine MCRD-Operation erkennt oder das Bausteinende erreicht ist. Für Operationen außerhalb des MCRA/MCRD-Bereichs kommt es zu keiner Zunahme der Laufzeit.

Die Operationen **MCRA** und **MCRD** dürfen Sie in Ihrem Programm immer nur paarweise verwenden.

Aktivieren und Deaktivieren eines MCR-Bereichs

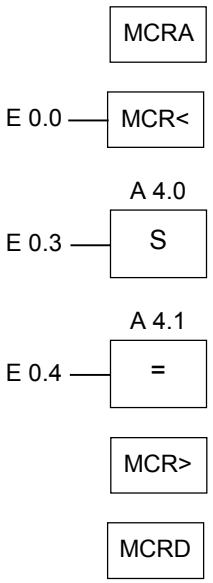
Nur die Operationen, die zwischen MCRA und MCRD programmiert sind, hängen vom Signalzustand des MCR-Bits ab. Fehlt eine MCRD-Operation, dann hängen die Operationen, die zwischen den Operationen MCRA und BEA programmiert sind, vom MCR-Bit ab.



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	-	-	-	-

Beispiel



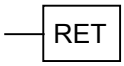
Die Operation MCRA aktiviert die Funktion MCR bis zum nächstfolgenden MCRD. Die Operationen zwischen MCR< und MCR> werden in Abhängigkeit des MA-Bits (hier E 0.0) bearbeitet:

- Wenn E 0.0 = 1 ist,
  - wird A 4.0 auf "1" gesetzt, sofern E 0.3 = 1 ist.
  - wird A 4.0 nicht geändert, sofern E 0.3 = 0 ist.
  - wird der Signalzustand von E 0.4 Ausgang A 4.1 zugewiesen.
- Wenn E 0.0 = 0 ist,
  - wird A 4.0 unabhängig vom Signalzustand an E 0.3 nicht geändert.
  - hat A 4.1 unabhängig vom Signalzustand an E 0.4 den Wert "0".

Die Abhängigkeit der Funktionen (FC) und Funktionsbausteine (FB) vom MCR müssen Sie in den Bausteinen selbst programmieren. Wird diese Funktion bzw. dieser Funktionsbaustein aus einer MCRA-MCRD-Sequenz heraus aufgerufen, dann sind nicht automatisch alle Anweisungen innerhalb dieser Sequenz vom MCR-Bit abhängig. Hierzu verwenden Sie die Operation MCRA des aufgerufenen Bausteins.

10.13 RET : Springe zurück

Symbol



Beschreibung

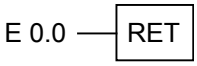
Mit der Operation **RET** können Sie Bausteine verlassen. Sie können einen Baustein bedingt verlassen.

Statuswort

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**VKE**	**/ER**
schreibt:	*	-	-	-	0	0	1	1	0

* Die Operation **RET** wird intern auf die Sequenz "SAVE; BEB;" abgebildet. Das bewirkt, daß auch das BIE-Bit beinflußt wird.

Beispiel



Der Baustein wird verlassen, wenn E 0.0 = 1 ist.

11 Schieben/Rotieren

11.1 Schiebeoperationen

11.1.1 Schiebeoperationen Übersicht

Beschreibung

Mit den Schiebeoperationen können Sie den Inhalt von Eingang IN Bitweise nach links oder rechts schieben (siehe auch CPU-Register). Ein Schieben um n Bits nach links multipliziert den Inhalt von Eingang IN mit 2 hoch n; ein Schieben um n Bits nach rechts dividiert den Inhalt von Eingang IN durch 2 hoch n. Wenn Sie also beispielsweise das binäre Äquivalent des Dezimalwerts 3 um 3 Bits nach links schieben, so ergibt sich das binäre Äquivalent des Dezimalwerts 24. Schieben Sie das binäre Äquivalent des Dezimalwerts 16 um 2 Bits nach rechts, so ergibt sich das binäre Äquivalent des Dezimalwerts 4.

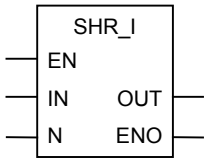
Am Eingang N können Sie angeben, um wie viele Bits geschoben werden soll. Die Stellen, die durch die Schiebeoperation frei werden, werden entweder mit Nullen oder mit dem Signalzustand des Vorzeichenbits aufgefüllt ("0" steht für positiv, "1" steht für negativ). Das zuletzt geschobene Bit wird in das Bit A1 des Statusworts geladen. Die Bits A0 und OV werden auf "0" zurückgesetzt. Mit den Sprungoperationen können Sie das Bit A1 im Statuswort auswerten.

Folgende Schiebeoperationen stehen Ihnen zur Verfügung:

- SHR_I Ganzzahl (16 Bit) rechts schieben
- SHR_DI Ganzzahl (32 Bit) rechts schieben
- SHL_W 16 Bit Links schieben
- SHR_W 16 Bit Rechts schieben
- SHL_DW 32 Bit Links schieben
- SHR_DW 32 Bit Rechts schieben

11.1.2 SHR_I : Ganzzahl (16 Bit) rechts schieben

Symbol

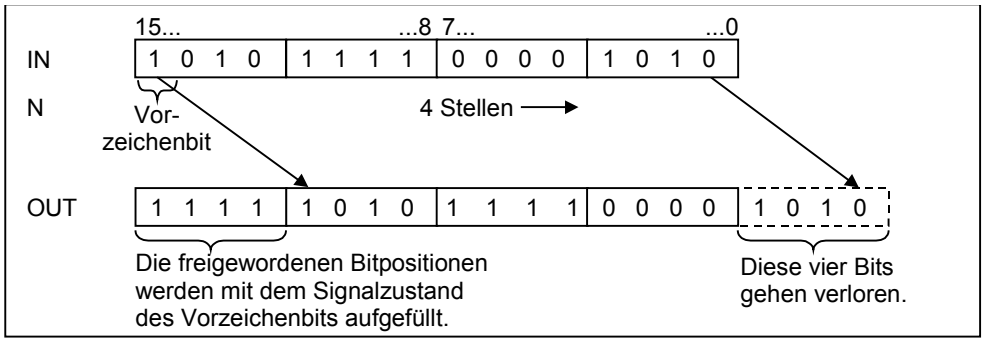


Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D, T, Z	Freigabeeingang
IN	INT	E, A, M, L, D	Wert, der geschoben wird
N	WORD	E, A, M, L, D	Anzahl der Bitpositionen, um die geschoben wird
OUT	INT	E, A, M, L, D	Ergebnis der Schiebeoperation
ENO	BOOL	E, A, M, L, D	Freigabeausgang

Beschreibung

Die Operation **Ganzzahl (16 Bit) rechts schieben** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und schiebt die Bits 0 bis 15 von Eingang IN Bitweise nach rechts. Eingang N gibt an, um wie viele Bits geschoben wird. Ist N größer als 16, arbeitet der Befehl so, als ob N = 16 wäre. Die Bitpositionen links werden mit dem Signalzustand von Bit 15 (Vorzeichen der Ganzzahl) belegt, d.h. mit Null, sofern die Zahl positiv ist, und mit 1, sofern die Zahl negativ ist. Das Ergebnis der Schiebeoperation kann am Ausgang OUT abgefragt werden.

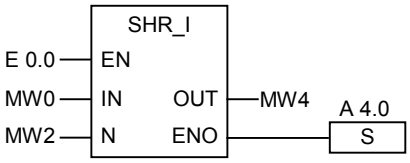
Die ausgelöste Operation setzt bei N ungleich Null das A0- und OV-Bit des Statusworts auf "0" zurück. ENO hat den gleichen Signalzustand wie EN.



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	-	X	X	X	1

Beispiel



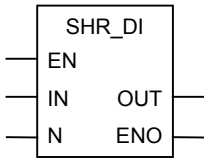
Die Operation wird aktiviert, wenn E 0.0 = 1 ist.

Merkerwort MW0 wird um die Anzahl an Bits nach rechts geschoben, die in MW2 angegeben ist.

Das Ergebnis wird in MW4 abgelegt. Ausgang A 4.0 wird auf 1 gesetzt.

11.1.3 SHR_DI : Ganzzahl (32 Bit) rechts schieben

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D, T, Z	Freigabeeingang
IN	DINT	E, A, M, L, D	Wert, der geschoben wird
N	WORD	E, A, M, L, D	Anzahl der Bitpositionen, um die geschoben wird
OUT	DINT	E, A, M, L, D	Ergebnis der Schiebeoperation
ENO	BOOL	E, A, M, L, D	Freigabeausgang

Beschreibung

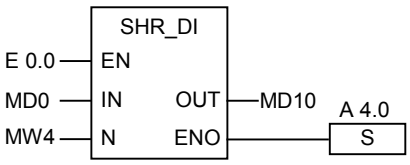
Die Operation **Ganzzahl (32 Bit) rechts schieben** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und schiebt den gesamten Inhalt von Eingang IN Bitweise nach rechts. Eingang N gibt an, um wie viele Bits geschoben wird. Ist N größer als 32 arbeitet der Befehl so, als ob N = 32 wäre. Die Bitpositionen links werden mit dem Signalzustand von Bit 31 (Vorzeichen der Ganzzahl) belegt, d.h. mit Null, sofern die Zahl positiv ist, und mit 1, sofern die Zahl negativ ist. Das Ergebnis der Schiebeoperation kann am Ausgang OUT abgefragt werden.

Die ausgelöste Operation setzt bei N ungleich Null das A0- und OV-Bit des Statusworts auf "0" zurück. ENO hat den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	-	X	X	X	1

Beispiel



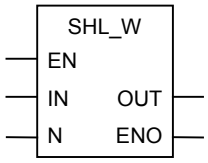
Die Operation wird aktiviert, wenn E 0.0 = 1 ist.

Merkerdoppelwort MD0 wird um die Anzahl an Bits nach rechts geschoben, die in MW4 angegeben ist.

Das Ergebnis wird in MD10 abgelegt. Ausgang A 4.0 wird auf 1 gesetzt.

11.1.4 SHL_W : 16 Bit Links schieben

Symbol



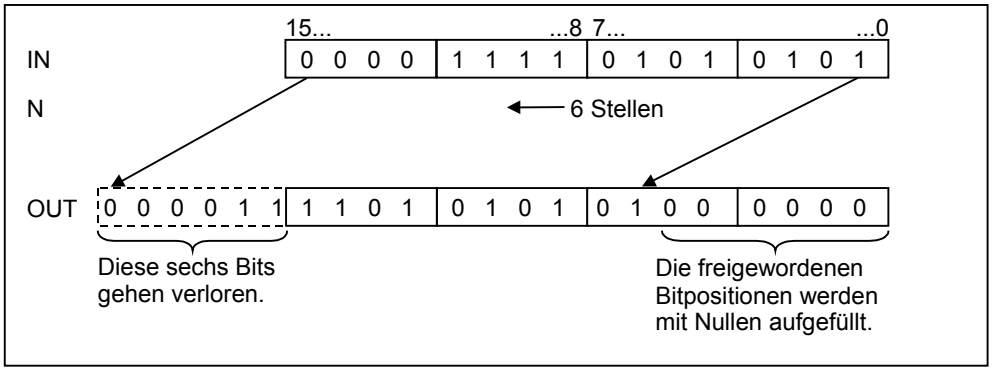
Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D, T, Z	Freigabeeingang
IN	WORD	E, A, M, L, D	Wert, der geschoben wird
N	WORD	E, A, M, L, D	Anzahl der Bitpositionen, um die geschoben wird
OUT	WORD	E, A, M, L, D	Ergebnis der Schiebeoperation
ENO	BOOL	E, A, M, L, D	Freigabeausgang

Beschreibung

Die Operation **16 Bit links schieben** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und schiebt die Bits 0 bis 15 von Eingang IN Bitweise nach links.

Eingang N gibt an, um wie viele Bits geschoben wird. Ist N größer als 16, schreibt der Befehl in Ausgang OUT eine 0 und setzt die Bits A0 und OV des Statusworts auf "0". Die rechts frei werdenden Bitpositionen werden mit Nullen aufgefüllt. Das Ergebnis der Schiebeoperation kann am Ausgangsparameter OUT abgefragt werden.

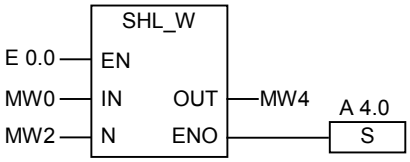
Die ausgelöste Operation setzt bei N ungleich Null das A0- und OV-Bit des Statusworts auf "0" zurück. ENO hat den gleichen Signalzustand wie EN.



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	-	X	X	X	1

Beispiel



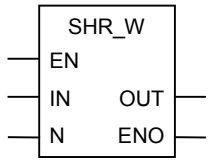
Die Operation wird aktiviert, wenn E 0.0 = 1 ist.

Merkerwort MW0 wird um die Anzahl an Bits nach links geschoben, die in MW2 angegeben ist.

Das Ergebnis wird in MW4 abgelegt. Ausgang A 4.0 wird auf 1 gesetzt.

11.1.5 SHR_W : 16 Bit Rechts schieben

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D, T, Z	Freigabeeingang
IN	WORD	E, A, M, L, D	Wert, der geschoben wird
N	WORD	E, A, M, L, D	Anzahl der Bitpositionen, um die geschoben wird
OUT	WORD	E, A, M, L, D	Ergebnis der Schiebeoperation
ENO	BOOL	E, A, M, L, D	Freigabeausgang

Beschreibung

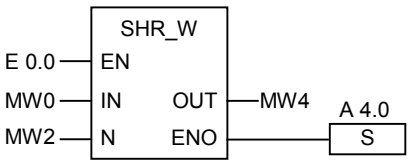
Die Operation **16 Bit rechts schieben** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und schiebt die Bits 0 bis 15 von Eingang IN Bitweise nach rechts. Die Bits 16 bis 31 werden nicht beeinflußt. Eingang N gibt an, um wie viele Bits geschoben wird. Ist N größer als 16, schreibt der Befehl eine 0 in Ausgang OUT und setzt die Bits A0 und OV auf "0". Die links frei werdenden Bitpositionen werden mit Nullen aufgefüllt. Das Ergebnis der Schiebeoperation kann am Ausgang OUT abgefragt werden.

Die ausgelöste Operation setzt bei N ungleich Null das A0- und OV-Bit des Statusworts immer auf "0" zurück. ENO hat den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	-	X	X	X	1

Beispiel



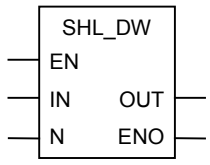
Die Operation wird aktiviert, wenn E 0.0 = 1 ist.

Merkerwort MW0 wird um die Anzahl an Bits nach rechts geschoben, die in MW2 angegeben ist.

Das Ergebnis wird in MW4 abgelegt. Ausgang A 4.0 wird auf 1 gesetzt.

11.1.6 SHL_DW : 32 Bit Links schieben

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D, T, Z	Freigabeeingang
IN	DWORD	E, A, M, L, D	Wert, der geschoben wird
N	WORD	E, A, M, L, D	Anzahl der Bitpositionen, um die geschoben wird
OUT	DWORD	E, A, M, L, D	Ergebnis der Schiebeoperation
ENO	BOOL	E, A, M, L, D	Freigabeausgang

Beschreibung

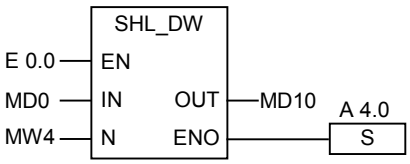
Die Operation **32 Bit links schieben** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und schiebt die Bits 0 bis 31 von Eingang IN Bitweise nach links. Der Parameter N gibt an, um wie viele Bits geschoben wird. Ist N größer als 32, schreibt der Befehl eine 0 in Ausgang OUT und setzt die Bits A0 und OV auf "0". Die rechts frei werdenden Bitpositionen werden mit Nullen aufgefüllt. Das Ergebnis der Schiebeoperation kann am Ausgang OUT abgefragt werden.

Die ausgelöste Operation setzt bei N ungleich Null das A0- und OV-Bit des Statusworts immer auf "0" zurück. ENO hat den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	-	X	X	X	1

Beispiel



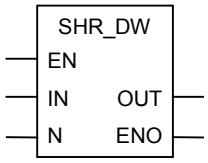
Die Operation wird aktiviert, wenn E 0.0 = 1 ist.

Merkerdoppelwort MD0 wird um die Anzahl an Bits nach links geschoben, die in MW4 angegeben ist.

Das Ergebnis wird in MD10 abgelegt. Ausgang A 4.0 wird auf 1 gesetzt.

11.1.7 SHR_DW : 32 Bit Rechts schieben

Symbol

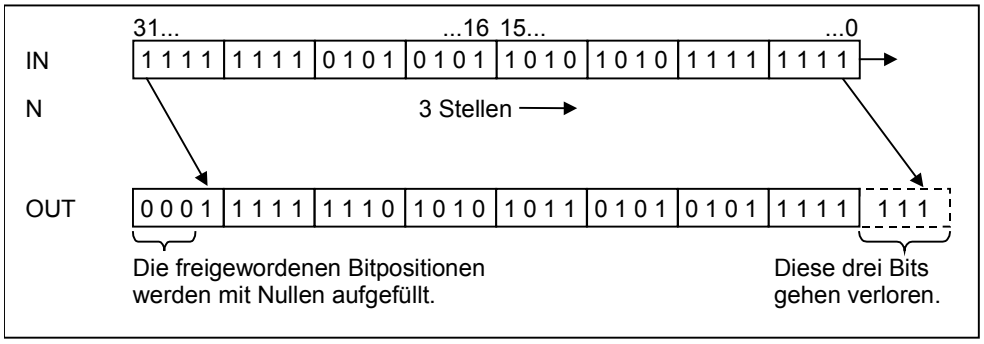


Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D, T, Z	Freigabeeingang
IN	DWORD	E, A, M, L, D	Wert, der geschoben wird
N	WORD	E, A, M, L, D	Anzahl der Bitpositionen, um die geschoben wird
OUT	DWORD	E, A, M, L, D	Ergebnis der Schiebeoperation
ENO	BOOL	E, A, M, L, D	Freigabeausgang

Beschreibung

Die Operation **32 Bit rechts schieben** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und schiebt die Bits 0 bis 31 von Eingang IN Bitweise nach rechts. Eingang N gibt an, um wie viele Bits geschoben wird. Ist N größer als 32, schreibt der Befehl eine 0 in Ausgang OUT und setzt die Bits A0 und OV auf "0". Die links frei werdenden Bitpositionen werden mit Nullen aufgefüllt. Das Ergebnis der Schiebeoperation kann am Ausgang OUT abgefragt werden.

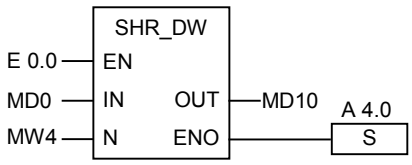
Die ausgelöste Operation setzt bei N ungleich Null das A0- und OV-Bit des Statusworts auf "0" zurück. ENO hat den gleichen Signalzustand wie EN.



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	-	X	X	X	1

Beispiel



Die Operation wird aktiviert, wenn E 0.0 = 1 ist.

Merkerdoppelwort MD0 wird um die Anzahl an Bits nach rechts geschoben, die in MW4 angegeben ist.

Das Ergebnis wird in MD10 abgelegt. Ausgang A 4.0 wird auf 1 gesetzt.

11.2 Rotieroperationen

11.2.1 Rotieroperationen Übersicht

Beschreibung

Mit den Rotieroperationen können Sie den gesamten Inhalt von Eingang IN Bitweise nach rechts oder links rotieren (siehe auch CPU-Register). Die frei gewordenen Stellen werden mit den Signalzuständen der Bits aufgefüllt, die aus dem Eingang IN geschoben wurden.

Am Eingang N können Sie angeben, um wie viele Bits rotiert werden soll.

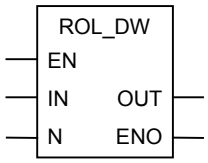
Je nach der gewählten Operation wird über das Bit A1 rotiert. Das Bit A0 im Statuswort wird auf "0" zurückgesetzt.

Folgende Rotieroperationen stehen Ihnen zur Verfügung:

- ROL_W 32 Bit Links rotieren
- ROR_W 32 Bit Rechts rotieren

11.2.2 ROL_DW : 32 Bit Links rotieren

Symbol

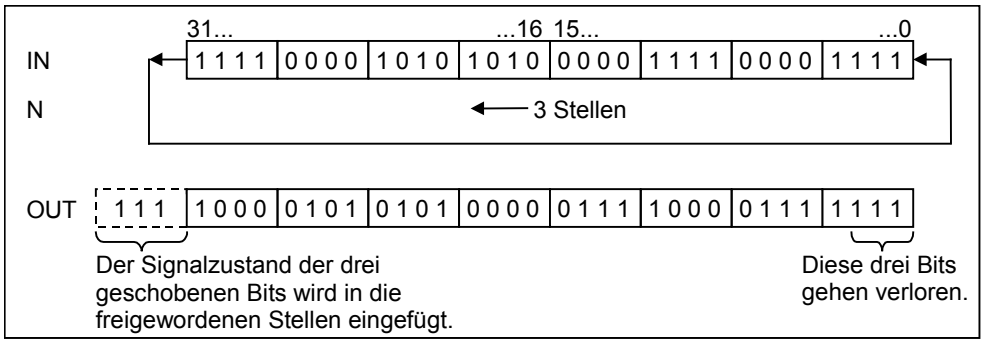


Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D, T, Z	Freigabeeingang
IN	DWORD	E, A, M, L, D	Wert, der rotiert wird
N	WORD	E, A, M, L, D	Anzahl der Bitpositionen, um die rotiert wird
OUT	DWORD	E, A, M, L, D	Ergebnis der Rotieroperation
ENO	BOOL	E, A, M, L, D	Freigabeausgang

Beschreibung

Die Operation **32 Bit links rotieren** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und rotiert den gesamten Inhalt von Eingang IN Bitweise nach links. Eingang N gibt an, um wie viele Bits rotiert wird. Ist N größer als 32, wird das Doppelwort mit [(N-1) Modul 32) +1] rotiert. Die rechts frei werdenden Bitpositionen werden mit den Signalzuständen der rotierten Bits aufgefüllt. Das Ergebnis der Rotieroperation kann am Ausgang OUT abgefragt werden.

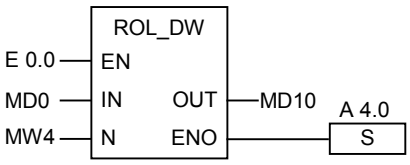
Die ausgelöste Operation setzt bei N ungleich Null das A0- und OV-Bit des Statusworts auf "0" zurück. ENO hat den gleichen Signalzustand wie EN.



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	-	X	X	X	1

Beispiel



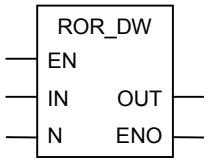
Die Operation wird aktiviert, wenn E 0.0 = 1 ist.

Merkerdoppelwort MD0 wird um die Anzahl an Bits nach links rotiert, die in MW4 angegeben ist.

Das Ergebnis wird in MD10 abgelegt. Ausgang A 4.0 wird auf 1 gesetzt.

11.2.3 ROR_DW : 32 Bit Rechts rotieren

Symbol

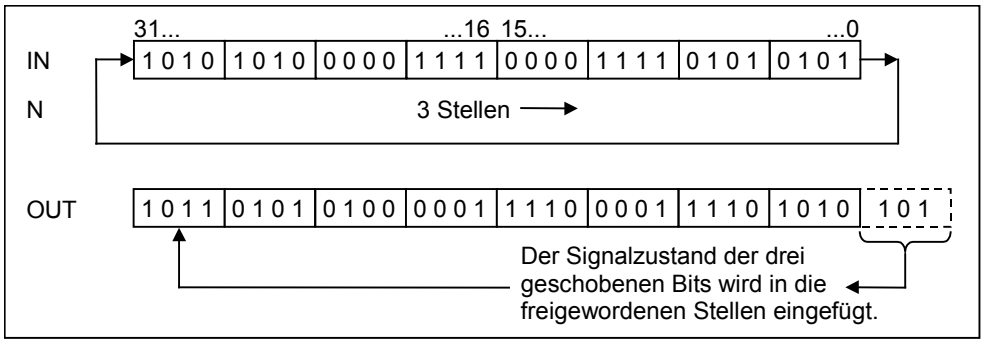


Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, L, D, T, Z	Freigabeeingang
IN	DWORD	E, A, M, L, D	Wert, der rotiert wird
N	WORD	E, A, M, L, D	Anzahl der Bitpositionen, um die rotiert wird
OUT	DWORD	E, A, M, L, D	Ergebnis der Rotieroperation
ENO	BOOL	E, A, M, L, D	Freigabeausgang

Beschreibung

Die Operation **32 Bit rechts rotieren** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und rotiert den gesamten Inhalt von Eingang IN Bitweise nach rechts. Eingang N gibt an, um wie viele Bits rotiert wird. Der Wert von N kann zwischen 0 und 31 liegen. Ist N größer als 32, wird das Doppelwort mit [((N-1) Modul 32) +1] rotiert. Die links frei werdenden Bitpositionen werden mit den Signalzuständen der rotierten Bits aufgefüllt. Das Ergebnis der Rotieroperation kann am Ausgang OUT abgefragt werden.

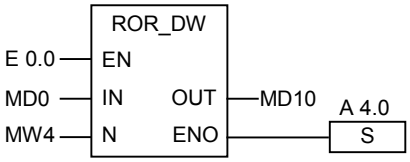
Die ausgelöste Operation setzt bei N ungleich Null das A0- und OV-Bit des Statusworts auf "0" zurück. ENO hat den gleichen Signalzustand wie EN.



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	X	X	X	X	-	X	X	X	1

Beispiel



Die Operation wird aktiviert, wenn E 0.0 = 1 ist.

Merkerdoppelwort MD0 wird um die Anzahl an Bits nach rechts rotiert, die in MW4 angegeben ist.

Das Ergebnis wird in MD10 abgelegt. Ausgang A 4.0 wird auf 1 gesetzt.

12 Statusbits

12.1 Statusbitoperationen Übersicht

Beschreibung

Statusbitoperationen sind Bitverknüpfungsoperationen, die mit den Bits des Statusworts arbeiten. Diese Operationen reagieren auf eine der folgenden Bedingungen, die von einem oder mehreren Bits angezeigt werden:

- Das Binärergebnis-Bit (BIE) wird gesetzt (d. h. es hat einen Signalzustand von "1").
- Die Beziehung des Ergebnisses einer arithmetischen Operation zu 0 ist: == 0, <> 0, > 0, < 0, >= 0, <= 0.
- Das Ergebnis einer arithmetischen Operation ist ungültig (UO).
- In einer arithmetischen Operation trat ein Überlauf (OV) oder ein speichernder Überlauf (OS) auf.

In einer UND–Operation verknüpfen die Statusbitoperationen das Ergebnis ihrer Signalzustandsabfrage mit dem vorherigen Verknüpfungsergebnis VKE entsprechend der UND–Wahrheitstabelle. In einer ODER–Operation geschieht dies entsprechend der ODER–Wahrheitstabelle.

Statuswort

Das Statuswort ist ein Register im Speicher Ihrer CPU. Es enthält Bits, die Sie in den Operanden von Bit– und Wortverknüpfungsoperationen ansprechen können. Aufbau des Statusworts:

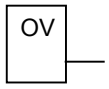
2^{15}... ...2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER

Sie können die Bits im Statuswort auswerten

- bei Festpunkt-Funktionen,
- bei Gleitpunkt-Funktionen.

12.2 OV : Störungsbit Überlauf

Symbol



Beschreibung

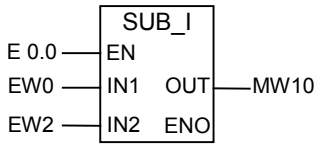
Mit der Operation **Störungsbit Überlauf** können Sie erkennen, ob in der zuletzt bearbeiteten arithmetischen Operation ein Überlauf (OV) auftrat. Befindet sich das Ergebnis nach der arithmetischen Operation außerhalb des zulässigen negativen oder außerhalb des zulässigen positiven Bereichs, so wird das OV–Bit im Statuswort gesetzt. Die Operation fragt den Signalzustand dieses Bits ab. Dieses Bit wird von fehlerfrei durchlaufenen arithmetischen Operationen zurückgesetzt.

Statuswort

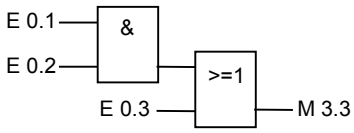
	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**VKE**	**/ER**
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

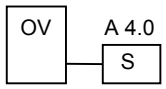
Netzwerk 1



Netzwerk 2



Netzwerk 3



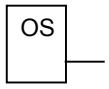
Die Box SUB_I wird aktiviert, wenn E 0.0 = 1 ist. Liegt das Ergebnis der arithmetischen Operation EW0 – EW2 außerhalb des zulässigen Bereichs für eine Ganzzahl, wird das OV–Bit gesetzt.

Die Signalzustandsabfrage an OV ergibt "1". Ausgang A 4.0 wird gesetzt, wenn die Abfrage bei OV "1" beträgt und das VKE von Netzwerk 2 "1" ist (d. h. wenn das VKE vor Ausgang A 4.0 = 1 ist).

Ist der Signalzustand von Eingang E 0.0 = 0 (nicht aktiviert), dann ist der Signalzustand von EN und ENO "0". Ist der Signalzustand von EN = 1 (aktiviert) und liegt das Ergebnis der arithmetischen Operation außerhalb des Bereichs, dann ist der Signalzustand von ENO = 0.

12.3 OS : Störungsbit Überlauf gespeichert

Symbol



Beschreibung

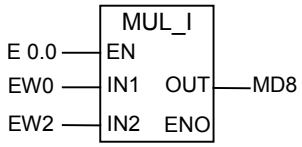
Mit der Operation **Störungsbit Überlauf gespeichert** können Sie erkennen, ob in einer arithmetischen Operation ein speichernder Überlauf (OS) auftrat. Befindet sich das Ergebnis nach einer arithmetischen Operation außerhalb des zulässigen negativen oder außerhalb des zulässigen positiven Bereichs, so wird das OS–Bit im Statuswort gesetzt. Die Operation fragt den Signalzustand dieses Bits ab. Anders als beim OV–Bit (Überlauf) bleibt das OS–Bit bei fehlerfrei durchlaufenen arithmetischen Operationen gesetzt.

Statuswort

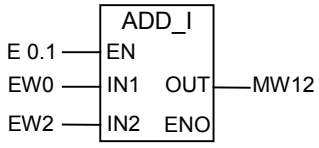
	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**VKE**	**/ER**
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

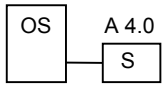
Netzwerk 1



Netzwerk 2



Netzwerk 3



Die Box MUL_I wird aktiviert, wenn E 0.0 = 1 ist und die Box ADD_I wird aktiviert, wenn E 0.1 = 1 ist. Liegt eines der beiden Ergebnisse außerhalb des zulässigen Bereichs für eine Ganzzahl, wird das OS–Bit gesetzt.

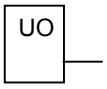
Die Signalzustandsabfrage an OS ergibt "1" und Ausgang A 4.0 wird gesetzt.

Netzwerk 1: Ist der Signalzustand von Eingang E 0.0 = 0 (nicht aktiviert), dann ist der Signalzustand von EN und ENO "0". Ist der Signalzustand von EN = 1 (aktiviert) und liegt das Ergebnis der arithmetischen Operation außerhalb des Bereichs, dann ist der Signalzustand von ENO = 0.

Netzwerk 2: Ist der Signalzustand von Eingang E 0.1 = 0 (nicht aktiviert), dann ist der Signalzustand von EN und ENO "0". Ist der Signalzustand von EN = 1 (aktiviert) und liegt das Ergebnis der arithmetischen Operation außerhalb des Bereichs, dann ist der Signalzustand von ENO = 0.

12.4 UO : Störungsbit Ungültige Operation

Symbol



Beschreibung

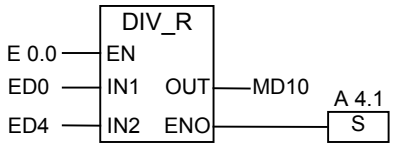
Mit der Operation **Störungsbit Ungültige Operation** können Sie abfragen, ob das Ergebnis einer arithmetischen Operation mit Gleitpunktzahlen ungültig ist (d. h. ob einer der Werte in der arithmetischen Operation keine gültige Gleitpunktzahl ist). Hierfür werden die Anzeigenbits A1 und A0 im Statuswort ausgewertet. Ist das Ergebnis einer arithmetischen Operation ungültig (UO), so ergibt die Signalzustandsabfrage "1". Gibt die Verknüpfung in A1 und A0 nicht "ungültig" an, dann ist das Ergebnis der Signalzustandsabfrage "0".

Statuswort

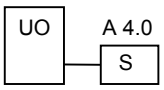
	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**VKE**	**/ER**
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

Netzwerk 1



Netzwerk 2



Die Box DIV_R wird aktiviert, wenn E 0.0 = 1 ist. Ist der Wert von ED0 oder ED4 keine gültige Gleitpunktzahl, so ist die arithmetische Operation ungültig.

Ist der Signalzustand von EN = 1 (aktiviert) und tritt während der Bearbeitung der Funktion DIV_R ein Fehler auf, dann ist der Signalzustand von ENO = 0.

Ausgang A 4.0 wird gesetzt, wenn die Operation DIV_R ausgeführt wird, jedoch einer der Werte in der arithmetischen Operation keine gültige Gleitpunktzahl ist. Ist der Signalzustand an Eingang E 0.0 = 0 (nicht aktiviert), dann ist der Signalzustand von EN und ENO "0".

12.5 BIE : Störungsbit BIE-Register

Symbol

Englisch	Deutsch
BR	BIE

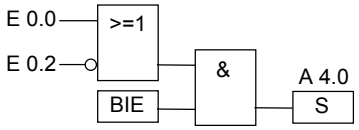
Beschreibung

Mit der Operation **Störungsbit BIE–Register** können Sie den Signalzustand des BIE–Bits (Binärergebnis-Bit) abfragen.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel



Ausgang A 4.0 wird gesetzt, wenn E 0.0 = 1 ist ODER E 0.2 = 0 ist und, zusätzlich zu diesem VKE, der Signalzustand des BIE–Bits = 1 ist.

12.6 <> 0 : Ergebnisbits

Symbole

== 0	Die Operation **Ergebnisbit bei gleich 0** bestimmt, ob das Ergebnis einer arithmetischen Operation gleich 0 ist.
<> 0	Die Operation **Ergebnisbit bei ungleich 0** bestimmt, ob das Ergebnis einer arithmetischen Operation ungleich 0 ist.
> 0	Die Operation **Ergebnisbit bei größer als 0** bestimmt, ob das Ergebnis einer arithmetischen Operation größer als 0 ist.
< 0	Die Operation **Ergebnisbit bei kleiner als 0** bestimmt, ob das Ergebnis einer arithmetischen Operation kleiner als 0 ist.
>= 0	Die Operation **Ergebnisbit bei größer gleich 0** bestimmt, ob das Ergebnis einer arithmetischen Operation größer als oder gleich 0 ist.
<= 0	Die Operation **Ergebnisbit bei kleiner gleich 0** bestimmt, ob das Ergebnis einer arithmetischen Operation kleiner als oder gleich 0 ist.

Beschreibung

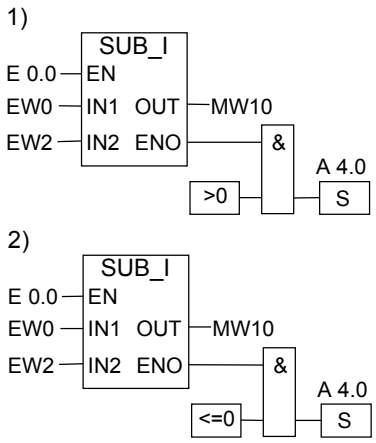
Mit den Operationen **Ergebnisbit ...** können Sie abfragen, wie sich das Ergebnis einer arithmetischen Operation zu Null verhält, d. h. ob das Ergebnis == 0, <> 0, > 0, < 0, >= 0 oder <= 0 ist. Hierzu werden die Anzeigenbits A1 und A0 des Statusworts ausgewertet. Ist die in dem Operanden angegebene Vergleichsbedingung erfüllt, so ergibt die Signalzustandsabfrage "1".

In einer UND–Operation verknüpfen die Statusbitoperationen das Ergebnis ihrer Signalzustandsabfrage mit dem vorherigen Verknüpfungsergebnis entsprechend der UND–Wahrheitstabelle. In einer ODER–Operation geschieht dies entsprechend der ODER–Wahrheitstabelle.

Statuswort

	BIE	**A1**	**A0**	**OV**	**OS**	**OR**	**STA**	**VKE**	**/ER**
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel



Die Box SUB_I wird aktiviert, wenn E 0.0 = 1 ist. Ist der Wert von EW0 größer als der Wert von EW2, ist das Ergebnis der arithmetischen Operation EW0 - EW2 größer als 0. Ist der Signalzustand von EN = 1 und tritt während der Bearbeitung der Funktion SUB_I ein Fehler auf, dann ist der Signalzustand von ENO = 0.

1) Ausgang A 4.0 wird gesetzt, wenn die Funktion fehlerfrei ausgeführt wurde und das Ergebnis größer als 0 ist. Ist der Signalzustand an Eingang E 0.0 = 0 (nicht aktiviert), dann ist der Signalzustand von EN und ENO "0".

2) Ausgang A 4.0 wird gesetzt, wenn die Funktion fehlerfrei ausgeführt wurde und das Ergebnis kleiner als oder gleich 0 ist. Ist der Signalzustand an Eingang E 0.0 = 0 (nicht aktiviert), dann ist der Signalzustand von EN und ENO "0".

12.6 <> 0 : Ergebnisbits

13 Zeiten

13.1 Zeitoperationen Übersicht

Beschreibung

Unter "Speicherbereiche und Komponenten einer Zeit" finden Sie Informationen zum Einstellen und zur Auswahl der richtigen Zeit.

Folgende Zeitoperationen stehen Ihnen zur Verfügung:

- S_IMPULS Zeit als Impuls parametrieren und starten
- S_VIMP Zeit als verlängerten Impuls parametrieren und starten
- S_EVERZ Zeit als Einschaltverzögerung parametrieren und starten
- S_SEVERZ Zeit als speichernde Einschaltverzögerung parametrieren und starten
- S_AVERZ Zeit als Ausschaltverzögerung parametrieren und starten

- SI Zeit als Impuls starten
- SV Zeit als verlängerten Impuls starten
- SE Zeit als Einschaltverzögerung starten
- SS Zeit als speichernde Einschaltverzögerung starten
- SA Zeit als Ausschaltverzögerung starten

13.2 Speicherbereiche und Komponenten einer Zeit

Speicherbereich

Zeiten haben einen eigenen reservierten Speicherbereich in Ihrer CPU. Dieser Speicherbereich reserviert ein 16-Bit-Wort für jeden Zeitoperanden. Das Programmieren mit FUP unterstützt 256 Zeiten. Wie viele Zeitworte in Ihrer CPU zur Verfügung stehen, entnehmen Sie Bitte deren technischen Daten.

Folgende Funktionen greifen auf den Speicherbereich der Zeiten zu:

- Zeitoperationen
- Aktualisieren der Timerwörter über Zeitimpulsgeber. Diese Funktion Ihrer CPU im RUN-Zustand vermindert einen bestimmten Wert um jeweils eine Einheit in einem Intervall, das von der Zeitbasis festgelegt wurde, bis der Zeitwert gleich "0" ist.

Zeitwert

Die Bits 0 bis 9 des Timerworts enthalten den Zeitwert binär-codiert. Der Zeitwert gibt eine Anzahl von Einheiten an. Das Aktualisieren der Zeit vermindert den Zeitwert um jeweils eine Einheit in einem Intervall, der von der Zeitbasis festgelegt wurde. Der Zeitwert wird solange vermindert, bis er gleich "0" ist.

Mit der folgenden Syntax können Sie einen vordefinierten Zeitwert laden:

- S5T#a**H**_b**M**_c**S**_d**MS**
  - H (Stunden), M (Minuten), S (Sekunden), MS (Millisekunden);
    a, b, c, d werden vom Anwender definiert.
  - Die Zeitbasis wird automatisch gewählt und der Wert zur nächstniederen Zahl mit dieser Zeitbasis gerundet

Sie können einen Zeitwert von max. 9 990 Sekunden bzw. 2H_46M_30S eingeben. Beispiele:

S5TIME#4S = 4 Sekunden
s5t#2h_15m = 2 Stunden und 15 Minuten
S5T#1H_12M_18S = 1 Stunde, 12 Minuten und 18 Sekunden

Zeitbasis

Die Bits 12 und 13 des Timerworts enthalten die Zeitbasis binär-codiert. Die Zeitbasis definiert das Intervall, in dem der Zeitwert um eine Einheit vermindert wird. Die kleinste Zeitbasis beträgt 10 ms, die größte 10 s.

Zeitbasis	Binärcode für Zeitbasis
10 ms	00
100 ms	01
1 s	10
10 s	11

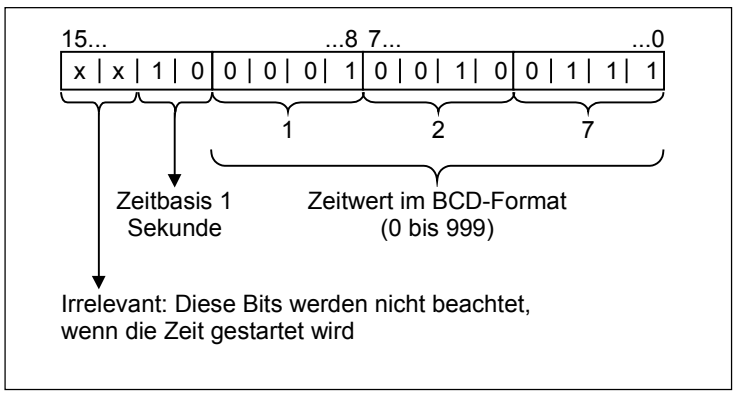
Die Werte dürfen 2H_46M_30S nicht überschreiten. Werte, die für einen Bereich oder für eine Auflösung zu groß sind, werden gerundet. Das allgemeine Format für den Datentyp S5TIME hat folgende Grenzwerte:

Auflösung	Bereich
0,01 Sekunde	10MS bis 9S_990MS
0,1 Sekunde	100MS bis 1M_39S_900MS
1 Sekunde	1S bis 16M_39S
10 Sekunden	10S bis 2H_46M_30S

Bit-Konfiguration in der Zeitzelle

Wird eine Zeit gestartet, so wird der Inhalt der Zeitzelle als Zeitwert verwendet. Die Bits 0 bis 11 der Zeitzelle enthalten den Zeitwert im binär-codierten Dezimalformat (BCD-Format: jede Gruppe von vier Bits enthält den Binärcode für einen Dezimalwert). Die Bits 12 und 13 enthalten die Zeitbasis im Binärcode.

Folgendes Bild zeigt den Inhalt der Zeitzelle, nachdem Sie den Zeitwert 127 mit der Zeitbasis 1 Sekunde geladen haben:

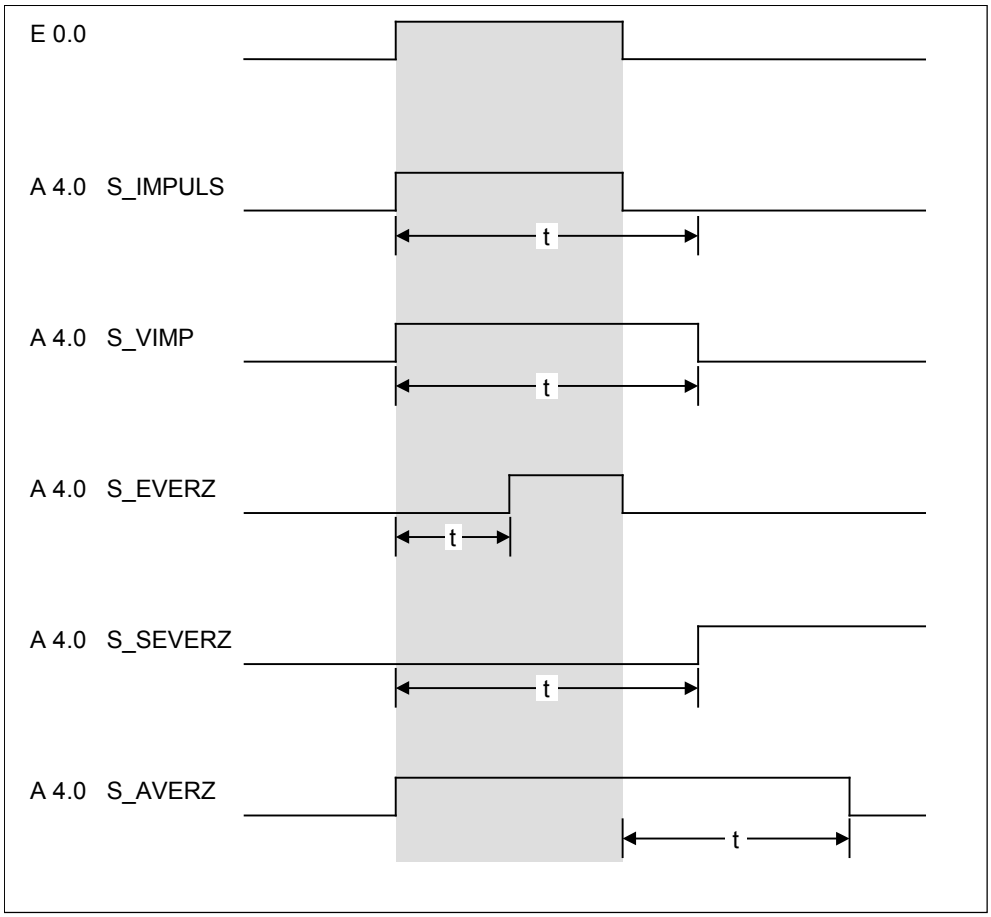


Lesen der Zeit und der Zeitbasis

Jede Timerbox liefert zwei Ausgänge, DUAL und DEZ, für die Sie eine Wortadresse angeben können. Am Ausgang DUAL ist der Zeitwert binär-codiert, die Zeitbasis wird nicht angezeigt. Am Ausgang DEZ sind Zeitbasis und Zeitwort BCD-codiert.

Auswahl der richtigen Zeit

Die Übersicht über die 5 verschiedenen Zeiten soll Ihnen helfen, die für Ihre Zwecke adäquate Zeit auszuwählen.



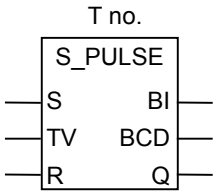
Zeiten	Erklärung
S_IMPULS Zeit als Impuls	Die maximale Zeit, in der das Ausgangssignal auf "1" bleibt, ist gleich dem programmierten Zeitwert t. Das Ausgangssignal bleibt für eine kürzere Zeit auf "1", wenn das Eingangssignal auf "0" wechselt.
S_VIMP Zeit als verlängerter Impuls	Das Ausgangssignal bleibt für die programmierte Zeit auf "1", unabhängig davon, wie lange das Eingangssignal auf "1" bleibt.
S_EVERZ Zeit als Einschaltverzögerung	Das Ausgangssignal ist nur "1", wenn die programmierte Zeit abgelaufen ist und das Eingangssignal noch immer "1" beträgt.
S_SEVERZ Zeit als speichernde Einschaltverzögerung	Das Ausgangssignal wechselt nur von "0" auf "1", wenn die programmierte Zeit abgelaufen ist, unabhängig davon, wie lange das Eingangssignal auf "1" bleibt.

Zeiten	Erklärung
S_AVERZ Zeit als Ausschaltverzögerung	Das Ausgangssignal ist "1", wenn das Eingangssignal "1" ist oder die Zeit läuft. Die Zeit wird gestartet wenn das Eingangssignal von "1" auf "0" wechselt.

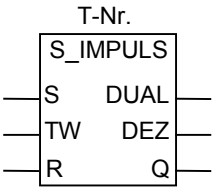
13.3 S_IMPULS : Zeit als Impuls parametrieren und starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
no.	Nr.	TIMER	T	Nummer der Zeit; Bereich ist von der CPU abhängig.
S	S	BOOL	E, A, M, D, L, T, Z	Starteingang
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Voreingestellter Zeitwert (Bereich 0-9999)
R	R	BOOL	E, A, M, D, L, T, Z	Rücksetzeingang
BI	DUAL	WORD	E, A, M, D, L	Rest-Zeitwert (Ganzzahlenformat)
BCD	DEZ	WORD	E, A, M, D, L	Rest-Zeitwert (BCD-Format)
Q	Q	BOOL	E, A, M, D, L	Status der Zeit

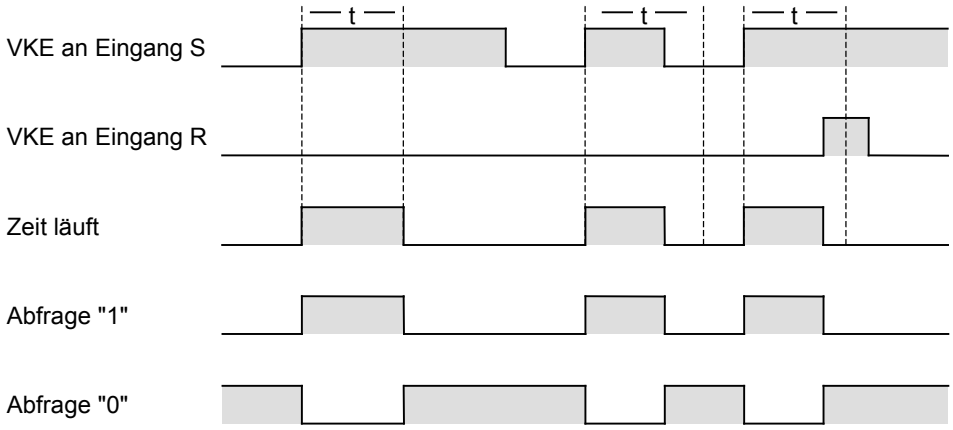
Beschreibung

Die Operation **Zeit als Impuls parametrieren und starten** startet eine angegebene Zeit, wenn der Starteingang (S) eine steigende Flanke aufweist (d.h. wenn der Signalzustand von "0" auf "1" wechselt). Um die Zeit freizugeben, ist immer ein Signalwechsel erforderlich. Die Zeit läuft solange mit dem Wert weiter, der an Eingang TW angegeben ist, bis die programmierte Zeit abgelaufen ist und der Eingang S = 1 ist. Solange die Zeit läuft, ergibt eine Signalzustandsabfrage nach "1" an Ausgang Q das Ergebnis "1". Wechselt Eingang S von "1" auf "0", bevor der Zeitwert abgelaufen ist, wird die Zeit angehalten. In diesem Fall ergibt eine Signalzustandsabfrage an Ausgang Q das Ergebnis "0"

Die Zeit wird zurückgesetzt, wenn der Rücksetzeingang (R) von "0" auf "1" wechselt, während die Zeit läuft. Durch diesen Wechsel werden auch der Zeitwert und die Zeitbasis auf Null zurückgesetzt. Der Signalzustand "1" an Eingang R hat keinen Einfluß, wenn die Zeit nicht läuft.

Der aktuelle Zeitwert kann an den Ausgängen DUAL und DEZ abgefragt werden. Der Zeitwert an Ausgang DUAL ist binärcodiert, der Zeitwert an Ausgang DEZ ist BCD-codiert.

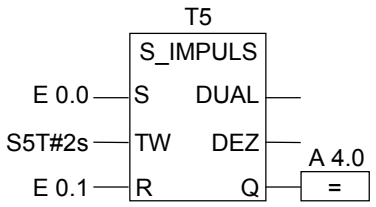
Impulsdiagramm



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

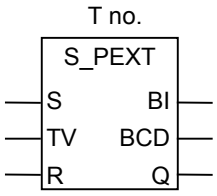


Wechselt der Signalzustand an Eingang E 0.0 von "0" auf "1" (steigende Flanke im VKE), wird die Zeit T5 gestartet. Sie läuft mit dem angegebenen Wert von zwei Sekunden (2 s) ab, solange E 0.0 = 1 ist. Wechselt der Signalzustand an E 0.0 vor Ablauf der zwei Sekunden von "1" auf "0", wird die Zeit angehalten. Wenn der Signalzustand an E 0.1 von "0" auf "1" wechselt, während die Zeit läuft, wird sie zurückgesetzt. Ausgang A 4.0 ist "1", solange die Zeit läuft.

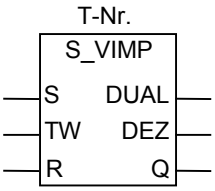
13.4 S_VIMP : Zeit als verlängerten Impuls parametrieren und starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
no.	Nr.	TIMER	T	Nummer der Zeit; Bereich ist von der CPU abhängig.
S	S	BOOL	E, A, M, D, L, T, Z	Starteingang
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Voreingestellter Zeitwert (Bereich 0-9999)
R	R	BOOL	E, A, M, D, L, T, Z	Rücksetzeingang
BI	DUAL	WORD	E, A, M, D, L	Rest-Zeitwert (Ganzzahlenformat)
BCD	DEZ	WORD	E, A, M, D, L	Rest-Zeitwert (BCD-Format)
Q	Q	BOOL	E, A, M, D, L	Status der Zeit

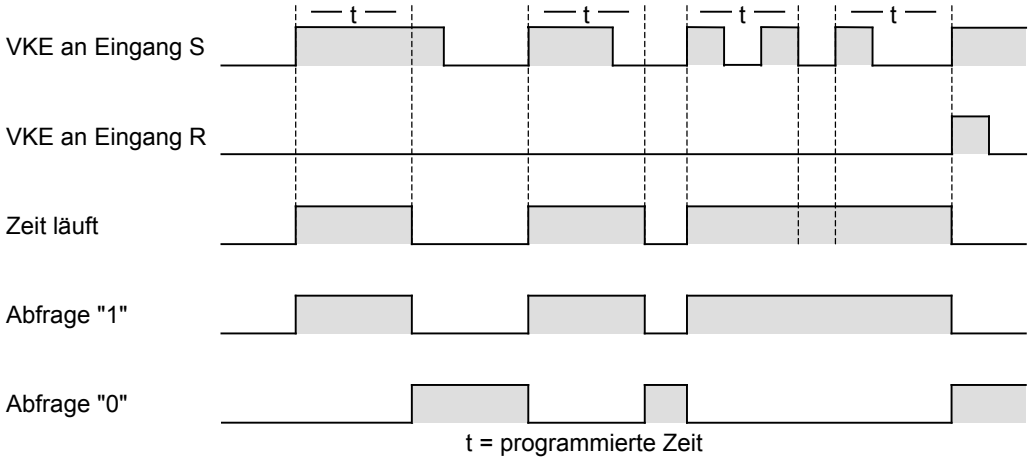
Beschreibung

Die Operation **Zeit als verlängerten Impuls parametrieren und starten** startet die angegebene Zeit, wenn der Starteingang (S) eine steigende Flanke aufweist (d.h. wenn der Signalzustand von "0" auf "1" wechselt). Es ist immer ein Signalwechsel erforderlich, um die Zeit freizugeben. Die Zeit läuft auch dann mit dem Wert weiter, der an Eingang TW angegebenen ist, wenn der Signalzustand an Eingang S noch vor Ablauf des Zeitwerts auf "0" wechselt. Solange die Zeit läuft, ergibt eine Signalzustandsabfrage nach "1" an Ausgang Q das Ergebnis "1". Die Zeit wird mit dem angegebenen Zeitwert neu gestartet, wenn der Signalzustand an Eingang S von "0" auf "1" wechselt, während die Zeit läuft.

Die Zeit wird zurückgesetzt, wenn der Rücksetzeingang (R) von "0" auf "1" wechselt, während die Zeit läuft. Durch diesen Wechsel werden auch der Zeitwert und die Zeitbasis auf Null zurückgesetzt.

Der aktuelle Zeitwert kann an den Ausgängen DUAL und DEZ abgefragt werden. Der Zeitwert an Ausgang DUAL ist binär-codiert, der Zeitwert an Ausgang DEZ ist BCD-codiert.

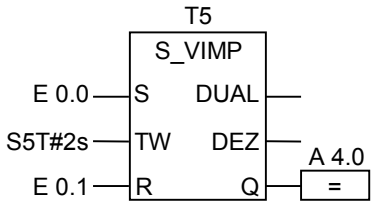
Impulsdiagramm



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

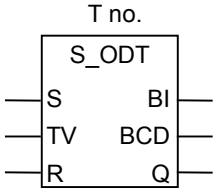


Wechselt der Signalzustand an Eingang E 0.0 von "0" auf "1" (steigende Flanke im VKE), wird die Zeit T5 gestartet. Die Zeit läuft unabhängig von einer fallenden Flanke an Eingang S mit dem angegebenen Wert von zwei Sekunden (2 s) weiter. Wechselt der Signalzustand an E 0.0 vor Ablauf dieser zwei Sekunden von "0" auf "1", wird die Zeit neu gestartet. Wechselt der Signalzustand an E 0.1 von "0" auf "1", während die Zeit abläuft, wird die Zeit neu gestartet. Ausgang A 4.0 ist "1", solange die Zeit läuft.

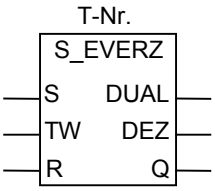
13.5 S_EVERZ : Zeit als Einschaltverzögerung parametrieren und starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
no.	Nr.	TIMER	T	Nummer der Zeit; Bereich ist von der CPU abhängig.
S	S	BOOL	E, A, M, D, L, T, Z	Starteingang
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Voreingestellter Zeitwert (Bereich 0-9999)
R	R	BOOL	E, A, M, D, L, T, Z	Rücksetzeingang
BI	DUAL	WORD	E, A, M, D, L	Rest-Zeitwert (Ganzzahlenformat)
BCD	DEZ	WORD	E, A, M, D, L	Rest-Zeitwert (BCD-Format)
Q	Q	BOOL	E, A, M, D, L	Status der Zeit

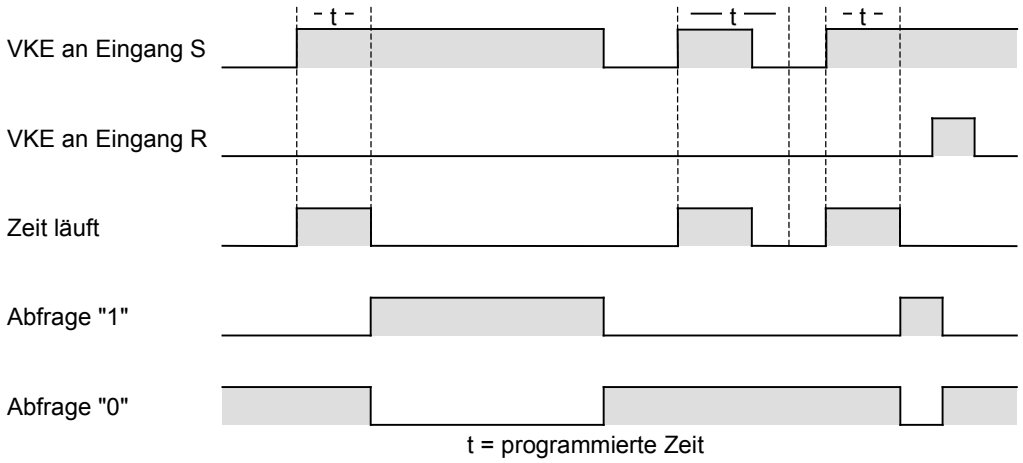
Beschreibung

Die Operation **Zeit als Einschaltverzögerung parametrieren und starten** startet die angegebene Zeit, wenn der Starteingang (S) eine steigende Flanke aufweist (d.h. wenn der Signalzustand von "0" auf "1" wechselt). Es ist immer ein Signalwechsel erforderlich, um die Zeit freizugeben. Die Zeit läuft mit dem Wert weiter, der an Eingang TW angegebenen ist, solange der Signalzustand an Eingang S = 1 ist. Eine Signalzustandsabfrage nach "1" an Ausgang Q ergibt "1", wenn die Zeit abgelaufen ist, Eingang S noch immer "1" ist und Rücksetzeingang (R) "0" bleibt. Wechselt der Signalzustand an Eingang S von "1" auf "0", während die Zeit läuft, wird sie angehalten. In diesem Fall ergibt eine Signalzustandsabfrage nach "1" immer "0".

Die Zeit wird zurückgesetzt, wenn der Rücksetzeingang (R) von "0" auf "1" wechselt, während die Zeit läuft. Durch diesen Wechsel werden auch der Zeitwert und die Zeitbasis auf Null zurückgesetzt. Die Zeit wird auch dann zurückgesetzt, wenn R = 1 ist, während die Zeit nicht läuft.

Der aktuelle Zeitwert kann an den Ausgängen DUAL und DEZ abgefragt werden. Der Zeitwert an Ausgang DUAL ist binär-codiert, der Zeitwert an Ausgang DEZ ist BCD-codiert.

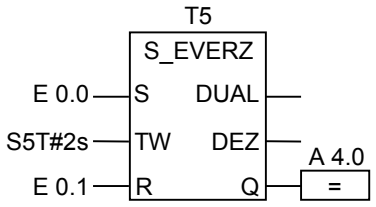
Impulsdiagramm



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

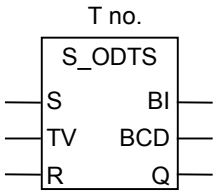


Wechselt der Signalzustand an Eingang E 0.0 von "0" auf "1" (steigende Flanke im VKE), so wird die Zeit T5 gestartet. Ist die angegebene Zeit von zwei Sekunden (2s) abgelaufen und beträgt der Signalzustand an E 0.0 noch immer "1", dann ist der Signalzustand von Ausgang A 4.0 = 1. Wechselt der Signalzustand an E 0.0 von "1" auf "0", wird die Zeit angehalten und A 4.0 ist "0". Wechselt der Signalzustand an E 0.0 von "0" auf "1", während die Zeit abläuft, wird die Zeit neu gestartet.

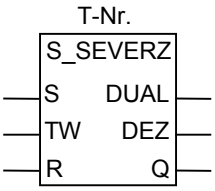
13.6 S_SEVERZ : Zeit als speichernde Einschaltverzögerung parametrieren und starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
no.	Nr.	TIMER	T	Nummer der Zeit; Bereich ist von der CPU abhängig.
S	S	BOOL	E, A, M, D, L, T, Z	Starteingang
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Voreingestellter Zeitwert
R	R	BOOL	E, A, M, D, L, T, Z	Rücksetzeingang
BI	DUAL	WORD	E, A, M, D, L	Rest-Zeitwert
BCD	DEZ	WORD	E, A, M, D, L	Rest-Zeitwert (BCD-Format)
Q	Q	BOOL	E, A, M, D, L	Status der Zeit

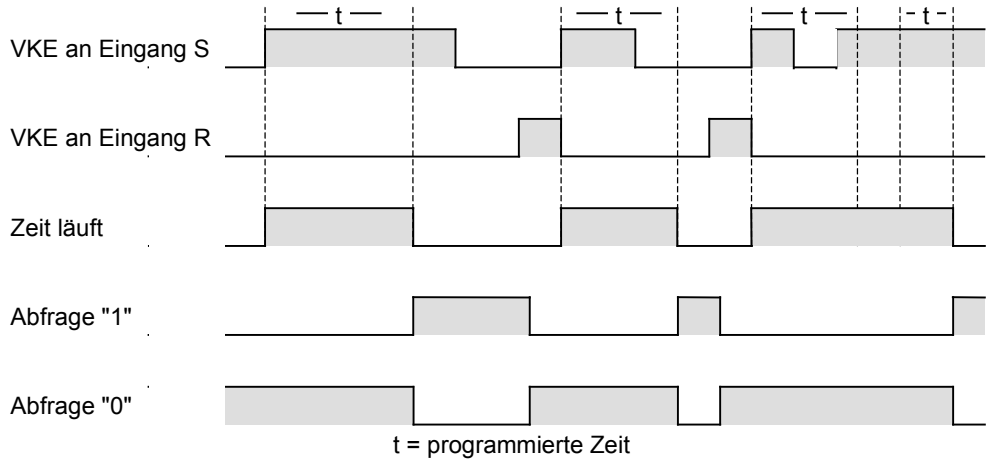
Beschreibung

Die Operation **Zeit als speichernde Einschaltverzögerung parametrieren und starten** startet die angegebene Zeit, wenn der Starteingang (S) eine steigende Flanke aufweist (d.h. wenn der Signalzustand von "0" auf "1" wechselt). Es ist immer ein Signalwechsel erforderlich, um die Zeit freizugeben. Die Zeit läuft auch dann mit dem Wert weiter, der an Eingang TW angegeben ist, wenn der Signalzustand an Eingang S noch vor Ablauf der Zeit auf "0" wechselt. Eine Signalzustandsabfrage nach "1" an Ausgang Q ergibt nach Ablauf der Zeit unabhängig vom Signalzustand an Eingang S das Ergebnis "1" wenn Rücksetzeingang (R) "0" bleibt. Die Zeit wird mit dem angegebenen Wert neu gestartet, wenn Eingang S von "0" auf "1" wechselt, während die Zeit läuft.

Wechselt der Rücksetzeingang (R) von "0" auf "1", wird die Zeit unabhängig vom VKE an Eingang S zurückgesetzt.

Der aktuelle Zeitwert kann an den Ausgängen DUAL und DEZ abgefragt werden. Der Zeitwert an Ausgang DUAL ist binär-codiert, der Zeitwert an Ausgang DEZ ist BCD-codiert

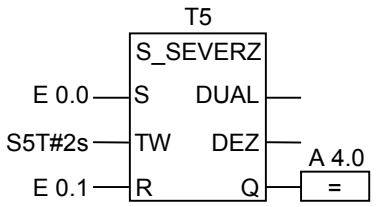
Impulsdiagramm



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

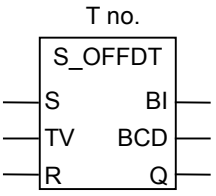


Wechselt der Signalzustand an Eingang E 0.0 von "0" auf "1" (steigende Flanke im VKE), wird die Zeit T5 gestartet. Die Zeit läuft weiter, unabhängig von einem Signalwechsel an E 0.0 von "1" auf "0". Wechselt der Signalzustand an E 0.0 vor Ablauf des angegebenen Werts von "0" auf "1", wird die Zeit neu gestartet. Wechselt der Signalzustand an E 0.0 von "0" auf "1", während die Zeit abläuft, wird die Zeit neu gestartet. Ausgang A 4.0 ist "1", nachdem die Zeit abgelaufen ist und E 0.1 auf "0" bleibt.

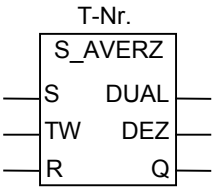
13.7 S_AVERZ : Zeit als Ausschaltverzögerung parametrieren und starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
no.	Nr.	TIMER	T	Nummer der Zeit; Bereich ist von der CPU abhängig.
S	S	BOOL	E, A, M, D, L, T, Z	Starteingang
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Voreingestellter Zeitwert
R	R	BOOL	E, A, M, D, L, T, Z	Rücksetzeingang
BI	DUAL	WORD	E, A, M, D, L	Rest-Zeitwert
BCD	DEZ	WORD	E, A, M, D, L	Rest-Zeitwert (BCD-Format)
Q	Q	BOOL	E, A, M, D, L	Status der Zeit

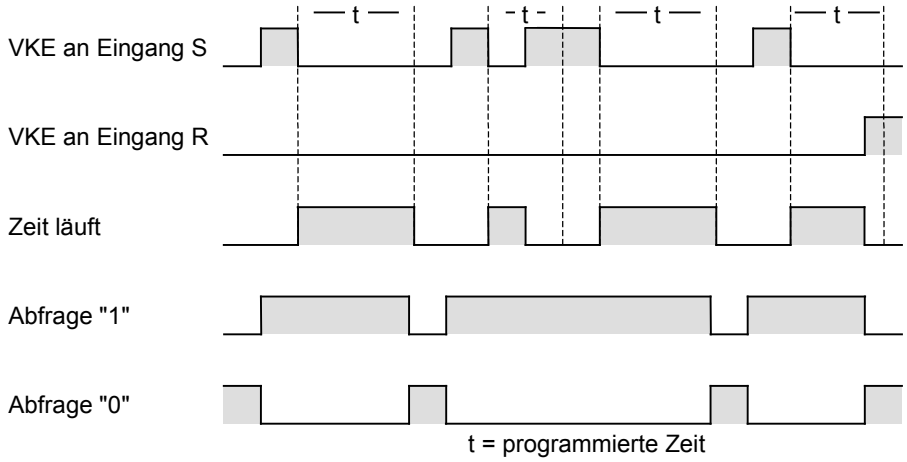
Beschreibung

Die Operation **Zeit als Ausschaltverzögerung parametrieren und starten** startet die angegebene Zeit, wenn der Starteingang (S) eine fallende Flanke aufweist (d.h. wenn der Signalzustand von "1" auf "0" wechselt). Es ist immer ein Signalwechsel erforderlich, um die Zeit freizugeben. Eine Signalzustandsabfrage nach "1" an Ausgang Q ergibt "1", wenn der Signalzustand an Eingang S = 1 ist oder die Zeit läuft. Die Zeit wird zurückgesetzt, wenn der Signalzustand an Eingang S von "0" auf "1" wechselt, während die Zeit läuft. Die Zeit wird erst dann wieder neu gestartet, wenn der Signalzustand an Eingang S von "1" auf "0" wechselt.

Wechselt der Rücksetzeingang (R) von "0" auf "1", während die Zeit läuft, wird die Zeit zurückgesetzt.

Der aktuelle Zeitwert kann an den Ausgängen DUAL und DEZ abgefragt werden. Der Zeitwert an Ausgang DUAL ist binär-codiert, der Zeitwert an Ausgang DEZ ist BCD-codiert.

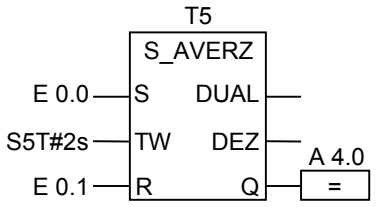
Impulsdiagramm



Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	X	X	X	1

Beispiel

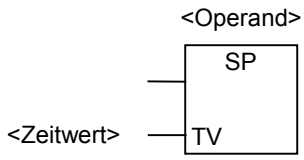


Wechselt der Signalzustand an Eingang E 0.0 von "1" auf "0", wird die Zeit gestartet. Ausgang A 4.0 ist "1", wenn E 0.0 = 1 ist oder die Zeit läuft. Wechselt der Signalzustand an E 0.1 von "0" auf "1", während die Zeit abläuft, wird die Zeit zurückgesetzt.

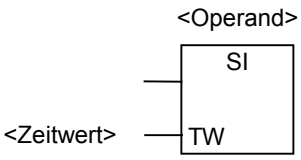
13.8 SI : Zeit als Impuls starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
Nummer der Zeit	Nummer der Zeit	TIMER	T	Der Operand gibt die Nummer der Zeit an, die gestartet werden soll.
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Zeitwert (Format S5TIME)

Beschreibung

Die Operation **Zeit als Impuls starten** startet eine Zeit mit einem angegebenen Wert, wenn das VKE eine steigende Flanke aufweist (Wechsel von "0" auf "1"). Solange das VKE positiv ist, läuft die Zeit mit dem angegebenen Wert weiter. Eine Signalzustandsabfrage nach "1" ergibt "1", solange die Zeit läuft. Wechselt das VKE vor Ablauf der Zeit von "1" auf "0", wird die Zeit angehalten. In diesem Fall erzeugt die Signalzustandsabfrage nach "1" das Ergebnis "0".

Die Box **Zeit als Impuls starten** können Sie nur am rechten Ende der Verknüpfungskette anordnen. Sie können allerdings mehrere Boxen **Zeit als Impuls starten** verwenden.

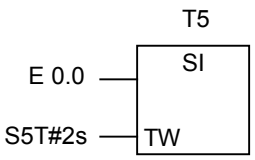
Siehe auch "Speicherbereiche und Komponenten einer Zeit".

Statuswort

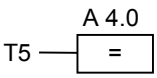
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	-	-	0

Beispiel

Netzwerk 1



Netzwerk 2

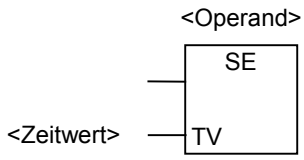


Wechselt der Signalzustand an E 0.0 von "0" auf "1" (steigende Flanke im VKE), wird die Zeit T5 gestartet. Solange der Signalzustand = 1 ist, läuft die Zeit mit dem angegebenen Wert von 2 Sekunden ab. Wechselt der Signalzustand an E 0.0 vor Ablauf dieser Zeit von "1" auf "0", wird die Zeit angehalten. Solange die Zeit läuft, ist der Signalzustand an Ausgang A 4.0 = 1.

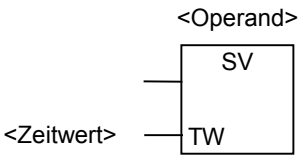
13.9 SV : Zeit als verlängerten Impuls starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
Nummer der Zeit	Nummer der Zeit	TIMER	T	Der Operand gibt die Nummer der Zeit an, die gestartet werden soll.
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Zeitwert (Format S5TIME)

Beschreibung

Die Operation **Zeit als verlängerten Impuls starten** startet eine Zeit mit einem angegebenen Wert, wenn das VKE eine steigende Flanke aufweist (Wechsel von "0" auf "1"). Die Zeit läuft auch dann noch mit dem angegebenen Wert weiter, wenn das VKE vor Ablauf dieser Zeit auf "0" wechselt. Eine Signalzustandsabfrage nach "1" ergibt "1", solange die Zeit läuft. Die Zeit wird mit dem angegebenen Zeitwert neu gestartet (nachgetriggert), wenn das VKE, noch während die Zeit läuft, von "0" auf "1" wechselt.

Die Box **Zeit als verlängerten Impuls starten** können Sie nur am rechten Ende der Verknüpfungskette anordnen. Sie können allerdings mehrere Boxen **Zeit als verlängerten Impuls starten** verwenden.

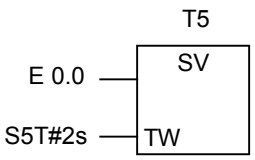
Siehe auch "Speicherbereiche und Komponenten einer Zeit".

Statuswort

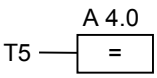
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	-	-	0

Beispiel

Netzwerk 1



Netzwerk 2



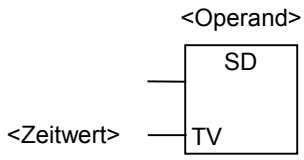
Wechselt der Signalzustand an E 0.0 von "0" auf "1" (steigende Flanke im VKE), wird die Zeit T5 gestartet. Die Zeit läuft weiter, ohne von einer fallenden Flanke im VKE beeinträchtigt zu werden. Wechselt der Signalzustand an E 0.0 von "0" auf "1", bevor der angegebene Zeitwert abgelaufen ist, wird die Zeit nachgetriggert.

Solange die Zeit läuft, ist der Signalzustand an Ausgang A 4.0 = 1.

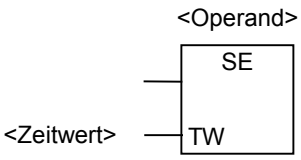
13.10 SE : Zeit als Einschaltverzögerung starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
Nummer der Zeit	Nummer der Zeit	TIMER	T	Der Operand gibt die Nummer der Zeit an, die gestartet werden soll.
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Zeitwert (Format S5TIME)

Beschreibung

Die Operation **Zeit als Einschaltverzögerung starten** startet eine angegebene Zeit, wenn das VKE eine steigende Flanke aufweist (Wechsel von "0" auf "1"). Eine Signalzustandsabfrage nach "1" ergibt "1", wenn die angegebene Zeit fehlerfrei abgelaufen ist und das VKE noch immer "1" beträgt. Wechselt das VKE von "1" auf "0", während die Zeit läuft, wird diese angehalten. In diesem Fall liefert eine Signalzustandsabfrage nach "1" immer das Ergebnis "0".

Die Box **Zeit als Einschaltverzögerung starten** können Sie nur am rechten Ende der Verknüpfungskette anordnen. Sie können allerdings mehrere Boxen **Zeit als Einschaltverzögerung starten** verwenden.

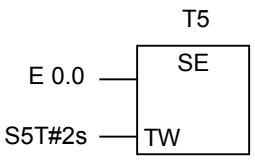
Siehe auch "Speicherbereiche und Komponenten einer Zeit".

Statuswort

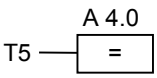
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	-	-	0

Beispiel

Netzwerk 1



Netzwerk 2

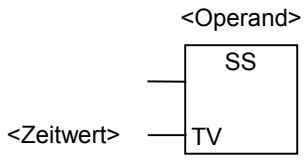


Wechselt der Signalzustand an E 0.0 von "0" auf "1" (steigende Flanke im VKE), wird die Zeit T5 gestartet. Wenn die Zeit abläuft und der Signalzustand noch immer "1" beträgt, ist der Ausgang A 4.0 = 1. Wechselt der Signalzustand von "1" auf "0", wird die Zeit angehalten.

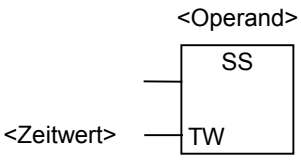
13.11 SS : Zeit als speichernde Einschaltverzögerung starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
Nummer der Zeit	Nummer der Zeit	TIMER	T	Der Operand gibt die Nummer der Zeit an, die gestartet werden soll.
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Zeitwert (Format S5TIME)

Beschreibung

Die Operation **Zeit als speichernde Einschaltverzögerung starten** startet die angegebene Zeit, wenn das VKE eine steigende Flanke aufweist (Wechsel von "0" auf "1"). Die Zeit läuft auch dann mit dem angegebenen Zeitwert weiter, wenn das VKE vor Ablauf der Zeit auf "0" wechselt. Eine Signalzustandsabfrage nach "1" ergibt unabhängig vom VKE das Ergebnis "1", wenn die Zeit abgelaufen ist. Wechselt das VKE von "0" auf "1", während die Zeit läuft, wird diese mit dem angegebenen Wert neu gestartet (nachgetriggert).

Die Box **Zeit als speichernde Einschaltverzögerung starten** können Sie nur am rechten Ende der Verknüpfungskette anordnen. Sie können allerdings mehrere Boxen **Zeit als speichernde Einschaltverzögerung starten** verwenden.

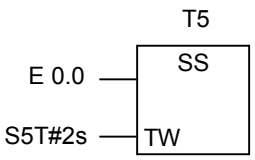
Siehe auch "Speicherbereiche und Komponenten einer Zeit".

Statuswort

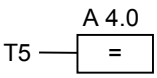
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	-	-	0

Beispiel

Netzwerk 1



Netzwerk 2



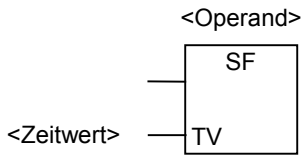
Wechselt der Signalzustand an E 0.0 von "0" auf "1" (steigende Flanke im VKE), wird die Zeit T5 gestartet. Die Zeit läuft weiter, unabhängig davon, ob der Signalzustand an E 0.0 von "1" auf "0" wechselt. Wechselt der Signalzustand von "0" auf "1", bevor der Zeitwert abgelaufen ist, wird die Zeit nachgetriggert.

Ausgang A 4.0 = 1, wenn die Zeit abgelaufen ist.

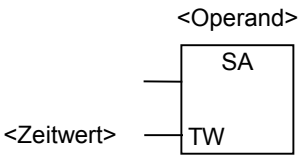
13.12 SA : Zeit als Ausschaltverzögerung starten

Symbol

Englisch



Deutsch



Parameter Englisch	Parameter Deutsch	Datentyp	Speicher-bereich	Beschreibung
Nummer der Zeit	Nummer der Zeit	TIMER	T	Der Operand gibt die Nummer der Zeit an, die gestartet werden soll.
TV	TW	S5TIME	E, A, M, D, L oder Konstante	Zeitwert (Format S5TIME)

Beschreibung

Die Operation **Zeit als Ausschaltverzögerung starten** startet die angegebene Zeit, wenn das VKE eine fallende Flanke aufweist (Wechsel von "1" auf "0"). Eine Signalzustandsabfrage nach "1" ergibt "1", wenn das VKE = 1 ist oder wenn die Zeit läuft. Die Zeit wird zurückgesetzt, wenn das VKE von "0" auf "1" wechselt, während die Zeit läuft. Die Zeit wird erst dann wieder neu gestartet, wenn das VKE von "1" auf "0"

Die Box **Zeit als Ausschaltverzögerung starten** können Sie nur am rechten Ende der Verknüpfungskette anordnen. Sie können allerdings mehrere Boxen **Zeit als Ausschaltverzögerung starten** verwenden.

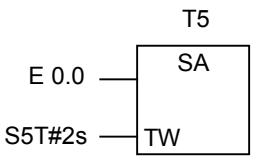
Siehe auch "Speicherbereiche und Komponenten einer Zeit".

Statuswort

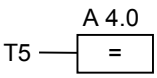
	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	-	-	-	-	-	0	-	-	0

Beispiel

Netzwerk 1



Netzwerk 2



Die Zeit wird gestartet, wenn der Signalzustand an E 0.0 von "1" auf "0" wechselt.

Wechselt Signalzustand von "0" auf "1", wird die Zeit rückgesetzt.

Der Signalzustand an Ausgang A 4.0 ist "1", wenn der Signalzustand an Eingang E 0.0 = 1 ist oder die Zeit läuft.

14 Wortverknüpfung

14.1 Wortverknüpfungsoperationen Übersicht

Beschreibung

Wortverknüpfungsoperationen verknüpfen die beiden digitalen Werte der Eingänge IN1 und IN2 entsprechend der Booleschen Logik. Sie werden jeweils durch den Signalzustand "1" am Freigabeeingang EN aktiviert. Die Werte werden als reine Bitmuster interpretiert. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat den gleichen Signalzustand wie EN.

Ist das Ergebnis an Ausgang OUT gleich "0", wird Bit A1 des Statusworts auf "0" gesetzt.
Ist das Ergebnis an Ausgang OUT ungleich "0", wird Bit A1 des Statusworts auf "1" gesetzt.

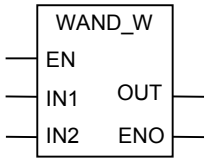
Folgende Operationen stehen Ihnen für Wortverknüpfungen zur Verfügung:

- WAND_W 16 Bit UND verknüpfen
- WOR_W 16 Bit ODER verknüpfen
- WXOR_W 16 Bit EXKLUSIV ODER verknüpfen

- WAND_DW 32 Bit UND verknüpfen
- WOR_DW 32 Bit ODER verknüpfen
- WXOR_DW 32 Bit EXKLUSIV ODER verknüpfen

14.2 WAND_W : 16 Bit UND verknüpfen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	WORD	E, A, M, D, L oder Konstante	Erster Wert der Verknüpfung
IN2	WORD	E, A, M, D, L oder Konstante	Zweiter Wert der Verknüpfung
OUT	WORD	E, A, M, D, L	Verknüpfungsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

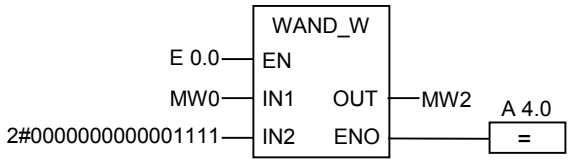
Beschreibung

Die Operation **16 Bit UND verknüpfen** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und verknüpft die beiden digitalen Werte der Eingänge IN1 und IN2 Bitweise entsprechend der UND-Wahrheitstabelle. Die Werte werden als reine Bitmuster interpretiert. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	X	0	0	-	X	1	1	1

Beispiel



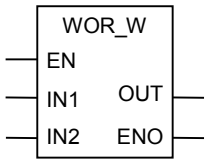
Die Operation wird aktiviert, wenn E 0.0 = 1 ist. Nur die Bits 0 bis 3 sind von Bedeutung, alle anderen Bits von MW0 sind maskiert.

IN1 = 0101010101010101
IN2 = 0000000000001111
OUT = 0000000000000101

A 4.0 ist "1", wenn die Verknüpfung ausgeführt wird.

14.3 WOR_W : 16 Bit ODER verknüpfen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	WORD	E, A, M, D, L oder Konstante	Erster Wert der Verknüpfung
IN2	WORD	E, A, M, D, L oder Konstante	Zweiter Wert der Verknüpfung
OUT	WORD	E, A, M, D, L	Verknüpfungsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

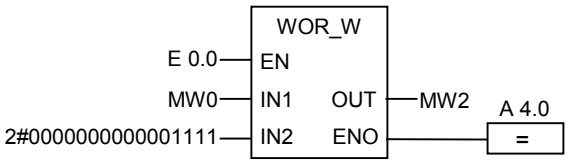
Beschreibung

Die Operation **16 Bit ODER verknüpfen** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und verknüpft die beiden digitalen Werte der Eingänge IN1 und IN2 Bitweise entsprechend der ODER-Wahrheitstabelle. Die Werte werden als reine Bitmuster interpretiert. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	X	0	0	-	X	1	1	1

Beispiel



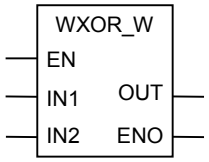
Ist E 0.0 = 1, wird die Operation aktiviert. Die Bits in MW0 und in der Konstanten werden mit ODER verknüpft, wobei die Bits 0 bis 3 auf "1" gesetzt werden, alle anderen Bits von MW0 werden unverändert in MW2 übernommen

IN1 = 0101010101010101
IN2 = 0000000000001111
OUT = 0101010101011111

A 4.0 ist "1", wenn die Verknüpfung ausgeführt wird.

14.4 WXOR_W : 16 Bit EXKLUSIV ODER verknüpfen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	WORD	E, A, M, D, L oder Konstante	Erster Wert der Verknüpfung
IN2	WORD	E, A, M, D, L oder Konstante	Zweiter Wert der Verknüpfung
OUT	WORD	E, A, M, D, L	Verknüpfungsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

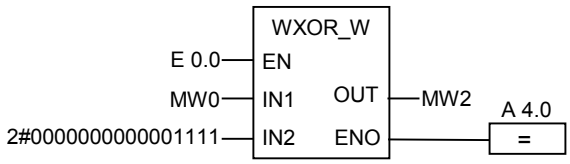
Beschreibung

Die Operation **16 Bit EXKLUSIV ODER verknüpfen** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und verknüpft die beiden digitalen Werte der Eingänge IN1 und IN2 Bitweise entsprechend der EXKLUSIV ODER-Wahrheitstabelle. Die Werte werden als reine Bitmuster interpretiert. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	X	0	0	-	X	1	1	1

Beispiel



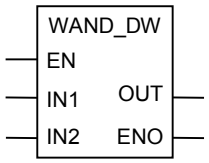
Die Operation wird aktiviert, wenn Eingang E 0.0 = 1 ist.

IN1 = 0101010101010101
IN2 = 0000000000001111
OUT = 0101010101011010

A 4.0 ist "1", wenn die Verknüpfung ausgeführt wird.

14.5 WAND_DW : 32 Bit UND verknüpfen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	DWORD	E, A, M, D, L oder Konstante	Erster Wert der Verknüpfung
IN2	DWORD	E, A, M, D, L oder Konstante	Zweiter Wert der Verknüpfung
OUT	DWORD	E, A, M, D, L	Verknüpfungsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

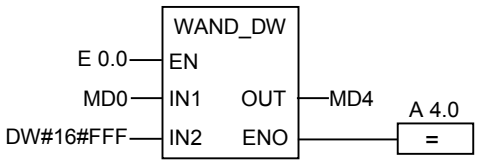
Beschreibung

Die Operation **32 Bit UND verknüpfen** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und verknüpft die beiden digitalen Werte der Eingänge IN1 und IN2 Bitweise entsprechend der UND-Wahrheitstabelle. Die Werte werden als reine Bitmuster interpretiert. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	X	0	0	-	X	1	1	1

Beispiel



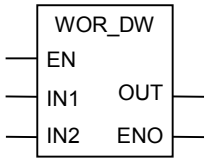
Die Operation wird aktiviert, wenn E 0.0 = 1 ist. Nur die Bits 0 bis 11 sind von Bedeutung, alle anderen Bits von MD4 sind maskiert.

IN1	=	0101010101010101	0101010101010101
IN2	=	0000000000000000	0000111111111111
OUT	=	0000000000000000	0000010101010101

A 4.0 ist "1", wenn die Verknüpfung ausgeführt wird.

14.6 WOR_DW : 32 Bit ODER verknüpfen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	DWORD	E, A, M, D, L oder Konstante	Erster Wert der Verknüpfung
IN2	DWORD	E, A, M, D, L oder Konstante	Zweiter Wert der Verknüpfung
OUT	DWORD	E, A, M, D, L	Verknüpfungsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

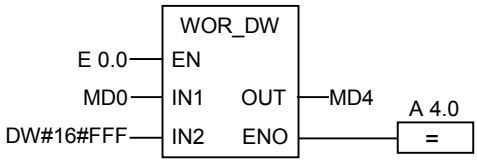
Beschreibung

Die Operation **32 Bit ODER verknüpfen** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und verknüpft die beiden digitalen Werte der Eingänge IN1 und IN2 Bitweise entsprechend der ODER-Wahrheitstabelle. Die Werte werden als reine Bitmuster interpretiert. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	X	0	0	-	X	1	1	1

Beispiel



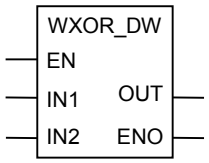
Die Operation wird aktiviert, wenn E 0.0 = 1 ist. Die Bits in MW0 und in der Konstanten werden mit ODER verknüpft, wobei die Bits 0 bis 11 auf "1" gesetzt werden, alle anderen Bits von MW0 werden unverändert in MW2 übernommen.

IN1	=	0101010101010101	0101010101010101
IN2	=	0000000000000000	0000111111111111
OUT	=	0101010101010101	0101111111111111

A 4.0 ist "1", wenn die Verknüpfung ausgeführt wird.

14.7 WXOR_DW : 32 Bit EXKLUSIV ODER verknüpfen

Symbol



Parameter	Datentyp	Speicherbereich	Beschreibung
EN	BOOL	E, A, M, D, L, T, Z	Freigabeeingang
IN1	WORD	E, A, M, D, L oder Konstante	Erster Wert der Verknüpfung
IN2	WORD	E, A, M, D, L oder Konstante	Zweiter Wert der Verknüpfung
OUT	WORD	E, A, M, D, L	Verknüpfungsergebnis
ENO	BOOL	E, A, M, D, L	Freigabeausgang

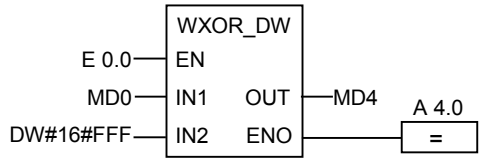
Beschreibung

Die Operation **32 Bit EXKLUSIV ODER verknüpfen** wird durch den Signalzustand "1" am Freigabeeingang EN aktiviert und verknüpft die beiden digitalen Werte der Eingänge IN1 und IN2 Bitweise entsprechend der EXKLUSIV ODER-Wahrheitstabelle. Die Werte werden als reine Bitmuster interpretiert. Das Ergebnis kann an Ausgang OUT abgefragt werden. ENO hat den gleichen Signalzustand wie EN.

Statuswort

	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER
schreibt:	1	X	0	0	-	X	1	1	1

Beispiel



Die Operation wird aktiviert, wenn Eingang E 0.0 = 1 ist.

IN1	=	0101010101010101	0101010101010101
IN2	=	0000000000000000	0000111111111111
OUT	=	0101010101010101	0101101010101010

A 4.0 ist "1", wenn die Verknüpfung ausgeführt wird.

A FUP-Operationen Übersicht

A.1 FUP-Operationen sortiert nach deutscher Mnemonik (SIMATIC)

Deutsche Mnemonik	Englische Mnemonik	Operation/Funktion	Beschreibung
&	&	Bitverknüpfung	UND-Verknüpfung
>=1	>=1	Bitverknüpfung	ODER-Verknüpfung
=	=	Bitverknüpfung	Zuweisung
#	#	Bitverknüpfung	Konnektor
---|	---|	Bitverknüpfung	Binären Eingang einfügen
---o|	---o|	Bitverknüpfung	Binären Eingang negieren
== 0	== 0	Statusbits	Ergebnisbits
<> 0	<> 0	Statusbits	Ergebnisbits
> 0	> 0	Statusbits	Ergebnisbits
< 0	< 0	Statusbits	Ergebnisbits
>= 0	>= 0	Statusbits	Ergebnisbits
<= 0	<= 0	Statusbits	Ergebnisbits
ABS	ABS	Gleitpunkt-Funktion	Bilden des Absolutwertes einer Gleitpunktzahl
ACOS	ACOS	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
ADD_DI	ADD_DI	Festpunkt-Funktion	Ganze Zahlen addieren (32 Bit)
ADD_I	ADD_I	Festpunkt-Funktion	Ganze Zahlen addieren (16 Bit)
ADD_R	ADD_R	Gleitpunkt-Funktion	Gleitpunktzahlen addieren
ASIN	ASIN	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
ATAN	ATAN	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
BCD_DI	BCD_DI	Umwandler	BCD-Zahl in Ganzzahl (32 Bit) wandeln
BCD_I	BCD_I	Umwandler	BCD-Zahl in Ganzzahl (16 Bit) wandeln
BIE	BR	Statusbits	Störungsbit BIE-Register
CALL	CALL	Programmsteuerung	FC/SFC aufrufen ohne Parameter
CALL_FB	CALL_FB	Programmsteuerung	FB als Box aufrufen
CALL_FC	CALL_FC	Programmsteuerung	FC als Box aufrufen
CALL_SFB	CALL_SFB	Programmsteuerung	System-FB als Box aufrufen
CALL_SFC	CALL_SFC	Programmsteuerung	System-FC als Box aufrufen
CEIL	CEIL	Umwandler	Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen
CMP ? D	CMP ? D	Vergleicher	Ganze Zahlen vergleichen (32 Bit)
CMP ? I	CMP ? I	Vergleicher	Ganze Zahlen vergleichen (16 Bit)
CMP ? R	CMP ? R	Vergleicher	Gleitpunktzahlen vergleichen
COS	COS	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
DI_BCD	DI_BCD	Umwandler	Ganzzahl (32 Bit) in BCD-Zahl wandeln

Deutsche Mnemonik	Englische Mnemonik	Operation/Funktion	Beschreibung
DI_R	DI_R	Umwandler	Ganzzahl (32 Bit) in Gleitpunktzahl wandeln
DIV_DI	DIV_DI	Festpunkt-Funktion	Ganze Zahlen dividieren (32 Bit)
DIV_I	DIV_I	Festpunkt-Funktion	Ganze Zahlen dividieren (16 Bit)
DIV_R	DIV_R	Gleitpunkt-Funktion	Gleitpunktzahlen dividieren
EXP	EXP	Gleitpunkt-Funktion	Bilden des Exponentialwerts einer Gleitpunktzahl
FLOOR	FLOOR	Umwandler	Aus Gleitpunktzahl nächstniedere Ganzzahl erzeugen
I_BCD	I_BCD	Umwandler	Ganzzahl (16 Bit) in BCD-Zahl wandeln
I_DI	I_DI	Umwandler	Ganzzahl (16 Bit) in Ganzzahl (32 Bit) wandeln
INV_I	INV_I	Umwandler	Einer-Komplement zu Ganzzahl (16 Bit) erzeugen
INV_DI	INV_DI	Umwandler	Einer-Komplement zu Ganzzahl (32 Bit) erzeugen
JMP	JMP	Sprünge	Springe im Baustein absolut
JMP	JMP	Sprünge	Springe im Baustein wenn 1 (bedingt)
JMPN	JMPN	Sprünge	Springe im Baustein wenn 0 (bedingt)
LABEL	LABEL	Sprünge	Sprungmarke
LN	LN	Gleitpunkt-Funktion	Bilden des natürlichen Logarithmus einer Gleitpunktzahl
MCR>	MCR>	Programmsteuerung	Master Control Relay ausschalten
MCR<	MCR<	Programmsteuerung	Master Control Relay einschalten
MCRA	MCRA	Programmsteuerung	Master Control Relay Anfang
MCRD	MCRD	Programmsteuerung	Master Control Relay Ende
MOD_DI	MOD_DI	Festpunkt-Funktion	Divisionsrest gewinnen (32 Bit)
MOVE	MOVE	Verschieben	Wert übertragen
MUL_DI	MUL_DI	Festpunkt-Funktion	Ganze Zahlen multiplizieren (32 Bit)
MUL_I	MUL_I	Festpunkt-Funktion	Ganze Zahlen multiplizieren (16 Bit)
MUL_R	MUL_R	Gleitpunkt-Funktion	Gleitpunktzahlen multiplizieren
N	N	Bitverknüpfung	Flanke 1 -> 0 abfragen
NEG	NEG	Bitverknüpfung	Signalflanke 1 -> 0 abfragen
NEG_DI	NEG_DI	Umwandler	Zweier-Komplement zu Ganzzahl (32 Bit) erzeugen
NEG_I	NEG_I	Umwandler	Zweier-Komplement zu Ganzzahl (16 Bit) erzeugen
NEG_R	NEG_R	Umwandler	Vorzeichen einer Gleitpunktzahl wechseln
OPN	OPN	DB-Aufruf	Datenbaustein öffnen
OS	OS	Statusbits	Störungsbit Überlauf gespeichert
OV	OV	Statusbits	Störungsbit Überlauf
P	P	Bitverknüpfung	Flanke 0 -> 1 abfragen
POS	POS	Bitverknüpfung	Signalflanke 0 -> 1 abfragen
R	R	Bitverknüpfung	Ausgang rücksetzen
RET	RET	Programmsteuerung	Springe zurück
ROL_DW	ROL_DW	Schieben/Rotieren	Links rotieren 32 Bit
ROR_DW	ROR_DW	Schieben/Rotieren	Rechts rotieren 32 Bit
ROUND	ROUND	Umwandler	Zahl runden
RS	RS	Bitverknüpfung	Flipflop rücksetzen setzen
S	S	Bitverknüpfung	Ausgang setzen
SA	SF	Zeiten	Zeit als Ausschaltverzögerung starten

Deutsche Mnemonik	Englische Mnemonik	Operation/Funktion	Beschreibung
SAVE	SAVE	Bitverknüpfung	Verknüpfungsergebnis in BIE-Register laden
S_AVERZ	S_OFFDT	Zeiten	Zeit als Ausschaltverzögerung parametrieren und starten
SE	SD	Zeiten	Zeit als Einschaltverzögerung starten
S_EVERZ	S_ODT	Zeiten	Zeit als Einschaltverzögerung parametrieren und starten
SHL_DW	SHL_DW	Schieben/Rotieren	Links schieben 32 Bit
SHL_W	SHL_W	Schieben/Rotieren	Links schieben 16 Bit
SHR_DI	SHR_DI	Schieben/Rotieren	Ganzzahl rechts schieben (32 Bit)
SHR_DW	SHR_DW	Schieben/Rotieren	Rechts schieben 32 Bit
SHR_I	SHR_I	Schieben/Rotieren	Ganzzahl rechts schieben (16 Bit)
SHR_W	SHR_W	Schieben/Rotieren	Rechts schieben 16 Bit
SI	SP	Zeiten	Zeit als Impuls starten
S_IMPULS	S_PULSE	Zeiten	Zeit als Impuls parametrieren und starten
SIN	SIN	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
SQR	SQR	Gleitpunkt-Funktion	Bilden des Quadrats einer Gleitpunktzahl
SQRT	SQRT	Gleitpunkt-Funktion	Bilden der Quadratwurzel einer Gleitpunktzahl
SR	SR	Bitverknüpfung	Flipflop setzen rücksetzen
SS	SS	Zeiten	Zeit als speichernde Einschaltverzögerung starten
S_SEVERZ	S_ODTS	Zeiten	Zeit als speichernde Einschaltverzögerung parametrieren und starten
SUB_DI	SUB_DI	Festpunkt-Funktion	Ganze Zahlen subtrahieren (32 Bit)
SUB_I	SUB_I	Festpunkt-Funktion	Ganze Zahlen subtrahieren (16 Bit)
SUB_R	SUB_R	Gleitpunkt-Funktion	Gleitpunktzahlen subtrahieren
SV	SE	Zeiten	Zeit als verlängerten Impuls starten
S_VIMP	S_PEXT	Zeiten	Zeit als verlängerten Impuls parametrieren und starten
SZ	SC	Zähler	Zähleranfangswert setzen
TAN	TAN	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
TRUNC	TRUNC	Umwandler	Ganze Zahl erzeugen
UO	UO	Statusbits	Störungsbit Ungültige Operation
WAND_DW	WAND_DW	Wortverknüpfung	UND verknüpfen 32 Bit
WAND_W	WAND_W	Wortverknüpfung	UND verknüpfen 16 Bit
WOR_DW	WOR_DW	Wortverknüpfung	ODER verknüpfen 32 Bit
WOR_W	WOR_W	Wortverknüpfung	ODER verknüpfen 16 Bit
WXOR_DW	WXOR_DW	Wortverknüpfung	EXKLUSIV ODER verknüpfen 32 Bit
WXOR_W	WXOR_W	Wortverknüpfung	EXKLUSIV ODER verknüpfen 16 Bit
XOR	XOR	Bitverknüpfung	EXKLUSIV-ODER-Verknüpfung
ZAEHLER	S_CUD	Zähler	Parametrieren und vorwärts-/rückwärtszählen
ZR	CD	Zähler	Rückwärtszählen
Z_RUECK	S_CD	Zähler	Parametrieren und rückwärtszählen
ZV	CU	Zähler	Vorwärtszählen

Deutsche Mnemonik	Englische Mnemonik	Operation/Funktion	Beschreibung
Z_VORW	S_CU	Zähler	Parametrieren und vorwärtszählen

A.2 FUP-Operationen sortiert nach englischer Mnemonik (International)

Englische Mnemonik	Deutsche Mnemonik	Operation/Funktion	Beschreibung
&	&	Bitverknüpfung	UND-Verknüpfung
>=1	>=1	Bitverknüpfung	ODER-Verknüpfung
=	=	Bitverknüpfung	Zuweisung
#	#	Bitverknüpfung	Konnektor
---|	---|	Bitverknüpfung	Binären Eingang einfügen
---o|	---o|	Bitverknüpfung	Binären Eingang negieren
== 0	== 0	Statusbits	Ergebnisbits
<> 0	<> 0	Statusbits	Ergebnisbits
> 0	> 0	Statusbits	Ergebnisbits
< 0	< 0	Statusbits	Ergebnisbits
>= 0	>= 0	Statusbits	Ergebnisbits
<= 0	<= 0	Statusbits	Ergebnisbits
ABS	ABS	Gleitpunkt-Funktion	Bilden des Absolutwertes einer Gleitpunktzahl
ACOS	ACOS	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
ADD_DI	ADD_DI	Festpunkt-Funktion	Ganze Zahlen addieren (32 Bit)
ADD_I	ADD_I	Festpunkt-Funktion	Ganze Zahlen addieren (16 Bit)
ADD_R	ADD_R	Gleitpunkt-Funktion	Gleitpunktzahlen addieren
ASIN	ASIN	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
ATAN	ATAN	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
BCD_DI	BCD_DI	Umwandler	BCD-Zahl in Ganzzahl (32 Bit) wandeln
BCD_I	BCD_I	Umwandler	BCD-Zahl in Ganzzahl (16 Bit) wandeln
BR	BIE	Statusbits	Störungsbit BIE-Register
CALL	CALL	Programmsteuerung	FC/SFC aufrufen ohne Parameter
CALL_FB	CALL_FB	Programmsteuerung	FB als Box aufrufen
CALL_FC	CALL_FC	Programmsteuerung	FC als Box aufrufen
CALL_SFB	CALL_SFB	Programmsteuerung	System-FB als Box aufrufen
CALL_SFC	CALL_SFC	Programmsteuerung	System-FC als Box aufrufen
CD	ZR	Zähler	Rückwärtszählen
CEIL	CEIL	Umwandler	Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen
CMP ? D	CMP ? D	Vergleicher	Ganze Zahlen vergleichen (32 Bit)
CMP ? I	CMP ? I	Vergleicher	Ganze Zahlen vergleichen (16 Bit)
CMP ? R	CMP ? R	Vergleicher	Gleitpunktzahlen vergleichen
COS	COS	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
CU	ZV	Zähler	Vorwärtszählen
DI_BCD	DI_BCD	Umwandler	Ganzzahl (32 Bit) in BCD-Zahl wandeln
DI_R	DI_R	Umwandler	Ganzzahl (32 Bit) in Gleitpunktzahl wandeln
DIV_DI	DIV_DI	Festpunkt-Funktion	Ganze Zahlen dividieren (32 Bit)

Englische Mnemonik	Deutsche Mnemonik	Operation/Funktion	Beschreibung
DIV_I	DIV_I	Festpunkt-Funktion	Ganze Zahlen dividieren (16 Bit)
DIV_R	DIV_R	Gleitpunkt-Funktion	Gleitpunktzahlen dividieren
EXP	EXP	Gleitpunkt-Funktion	Bilden des Exponentialwerts einer Gleitpunktzahl
FLOOR	FLOOR	Umwandler	Aus Gleitpunktzahl nächstniedere Ganzzahl erzeugen
I_BCD	I_BCD	Umwandler	Ganzzahl (16 Bit) in BCD-Zahl wandeln
I_DI	I_DI	Umwandler	Ganzzahl (16 Bit) in Ganzzahl (32 Bit) wandeln
INV_I	INV_I	Umwandler	Einer-Komplement zu Ganzzahl (16 Bit) erzeugen
INV_DI	INV_DI	Umwandler	Einer-Komplement zu Ganzzahl (32 Bit) erzeugen
JMP	JMP	Sprünge	Springe im Baustein absolut
JMP	JMP	Sprünge	Springe im Baustein wenn 1 (bedingt)
JMPN	JMPN	Sprünge	Springe im Baustein wenn 0 (bedingt)
LABEL	LABEL	Sprünge	Sprungmarke
LN	LN	Gleitpunkt-Funktion	Bilden des natürlichen Logarithmus einer Gleitpunktzahl
MCR>	MCR>	Programmsteuerung	Master Control Relay ausschalten
MCR<	MCR<	Programmsteuerung	Master Control Relay einschalten
MCRA	MCRA	Programmsteuerung	Master Control Relay Anfang
MCRD	MCRD	Programmsteuerung	Master Control Relay Ende
MOD_DI	MOD_DI	Festpunkt-Funktion	Divisionsrest gewinnen (32 Bit)
MOVE	MOVE	Verschieben	Wert übertragen
MUL_DI	MUL_DI	Festpunkt-Funktion	Ganze Zahlen multiplizieren (32 Bit)
MUL_I	MUL_I	Festpunkt-Funktion	Ganze Zahlen multiplizieren (16 Bit)
MUL_R	MUL_R	Gleitpunkt-Funktion	Gleitpunktzahlen multiplizieren
N	N	Bitverknüpfung	Flanke 1 -> 0 abfragen
NEG	NEG	Bitverknüpfung	Signalflanke 1 -> 0 abfragen
NEG_DI	NEG_DI	Umwandler	Zweier-Komplement zu Ganzzahl (32 Bit) erzeugen
NEG_I	NEG_I	Umwandler	Zweier-Komplement zu Ganzzahl (16 Bit) erzeugen
NEG_R	NEG_R	Umwandler	Vorzeichen einer Gleitpunktzahl wechseln
OPN	OPN	DB-Aufruf	Datenbaustein öffnen
OS	OS	Statusbits	Störungsbit Überlauf gespeichert
OV	OV	Statusbits	Störungsbit Überlauf
P	P	Bitverknüpfung	Flanke 0 -> 1 abfragen
POS	POS	Bitverknüpfung	Signalflanke 0 -> 1 abfragen
R	R	Bitverknüpfung	Ausgang rücksetzen
RET	RET	Programmsteuerung	Springe zurück
ROL_DW	ROL_DW	Schieben/Rotieren	Links rotieren 32 Bit
ROR_DW	ROR_DW	Schieben/Rotieren	Rechts rotieren 32 Bit
ROUND	ROUND	Umwandler	Zahl runden
RS	RS	Bitverknüpfung	Flipflop rücksetzen setzen
S	S	Bitverknüpfung	Ausgang setzen
SAVE	SAVE	Bitverknüpfung	Verknüpfungsergebnis in BIE-Register laden
SC	SZ	Zähler	Zähleranfangswert setzen
S_CD	Z_RUECK	Zähler	Parametrieren und rückwärtszählen

Englische Mnemonik	Deutsche Mnemonik	Operation/Funktion	Beschreibung
S_CU	Z_VORW	Zähler	Parametrieren und vorwärtszählen
S_CUD	ZAEHLER	Zähler	Parametrieren und vorwärts-/rückwärtszählen
SD	SE	Zeiten	Zeit als Einschaltverzögerung starten
SE	SV	Zeiten	Zeit als verlängerten Impuls starten
SF	SA	Zeiten	Zeit als Ausschaltverzögerung starten
SHL_DW	SHL_DW	Schieben/Rotieren	Links schieben 32 Bit
SHL_W	SHL_W	Schieben/Rotieren	Links schieben 16 Bit
SHR_DI	SHR_DI	Schieben/Rotieren	Ganzzahl rechts schieben (32 Bit)
SHR_DW	SHR_DW	Schieben/Rotieren	Rechts schieben 32 Bit
SHR_I	SHR_I	Schieben/Rotieren	Ganzzahl rechts schieben (16 Bit)
SHR_W	SHR_W	Schieben/Rotieren	Rechts schieben 16 Bit
SIN	SIN	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
S_ODT	S_EVERZ	Zeiten	Zeit als Einschaltverzögerung parametrieren und starten
S_ODTS	S_SEVERZ	Zeiten	Zeit als speichernde Einschaltverzögerung parametrieren und starten
S_OFFDT	S_AVERZ	Zeiten	Zeit als Ausschaltverzögerung parametrieren und starten
SP	SI	Zeiten	Zeit als Impuls starten
S_PEXT	S_VIMP	Zeiten	Zeit als verlängerten Impuls parametrieren und starten
S_PULSE	S_IMPULS	Zeiten	Zeit als Impuls parametrieren und starten
SQR	SQR	Gleitpunkt-Funktion	Bilden des Quadrats (SQR) einer Gleitpunktzahl
SQRT	SQRT	Gleitpunkt-Funktion	Bilden der Quadratwurzel (SQRT) einer Gleitpunktzahl
SR	SR	Bitverknüpfung	Flipflop setzen rücksetzen
SS	SS	Zeiten	Zeit als speichernde Einschaltverzögerung starten
SUB_DI	SUB_DI	Festpunkt-Funktion	Ganze Zahlen subtrahieren (32 Bit)
SUB_I	SUB_I	Festpunkt-Funktion	Ganze Zahlen subtrahieren (16 Bit)
SUB_R	SUB_R	Gleitpunkt-Funktion	Gleitpunktzahlen subtrahieren
TAN	TAN	Gleitpunkt-Funktion	Bilden von trigonometrischen Funktionen von Winkeln als Gleitpunktzahlen
TRUNC	TRUNC	Umwandler	Ganze Zahl erzeugen
UO	UO	Statusbits	Störungsbit Ungültige Operation
WAND_DW	WAND_DW	Wortverknüpfung	UND verknüpfen 32 Bit
WAND_W	WAND_W	Wortverknüpfung	UND verknüpfen 16 Bit
WOR_DW	WOR_DW	Wortverknüpfung	ODER verknüpfen 32 Bit
WOR_W	WOR_W	Wortverknüpfung	ODER verknüpfen 16 Bit
WXOR_DW	WXOR_DW	Wortverknüpfung	EXKLUSIV ODER verknüpfen 32 Bit
WXOR_W	WXOR_W	Wortverknüpfung	EXKLUSIV ODER verknüpfen 16 Bit
XOR	XOR	Bitverknüpfung	EXKLUSIV-ODER-Verknüpfung

B Programmierbeispiele

B.1 Programmierbeispiele Übersicht

Praktische Anwendungen

Jede FUP–Operation löst eine bestimmte Funktion aus. Durch Kombination der Operationen in einem Programm können Sie eine breite Palette von Automatisierungsaufgaben ausführen. Hier einige Beispiele für praktische Anwendungen:

- Steuern eines Förderbandes durch Bitverknüpfungsoperationen
- Feststellen der Bewegungsrichtung auf einem Förderband durch Bitverknüpfungsoperationen
- Generieren eines Taktimpulses durch Zeitoperationen
- Überwachen des Lagerbereichs durch Zähl– und Vergleichsoperationen
- Berechnungen mit arithmetischen Operationen für Ganzzahlen
- Einstellen der Zeitdauer für das Beheizen eines Ofens

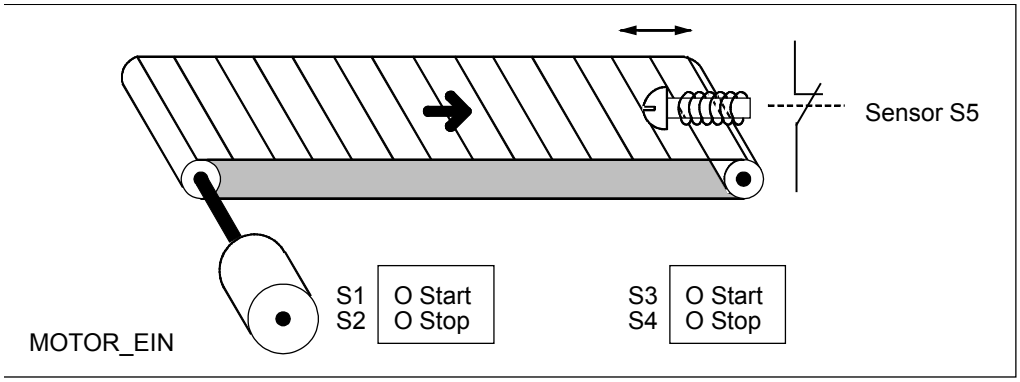
Verwendete Operationen

Mnemonik	Operation	Beschreibung
WAND_W	Wortverknüpfung	16 Bit UND verknüpfen
WOR_W	Wortverknüpfung	16 Bit ODER verknüpfen
Z_RUECK	Zähler	Rückwärtszählen
Z_VORW	Zähler	Vorwärtszählen
R	Bitverknüpfung	Ausgang rücksetzen
S	Bitverknüpfung	Ausgang setzen
P	Bitverknüpfung	Flanke 0 → 1 abfragen
ADD_I	Festpunkt-Funktion	Ganze Zahlen addieren (16 Bit)
DIV_I	Festpunkt-Funktion	Ganze Zahlen dividieren (16 Bit)
MUL_I	Festpunkt-Funktion	Ganze Zahlen multiplizieren (16 Bit)
CMP >=I	Vergleicher	Ganze Zahlen vergleichen (16 Bit)
CMP <=I	Vergleicher	Ganze Zahlen vergleichen (16 Bit)
&	Bitverknüpfung	UND–Verknüpfung
>=1	Bitverknüpfung	ODER–Verknüpfung
=	Bitverknüpfung	Zuweisung
JMPN	Sprünge	Springe im Baustein wenn 0 (bedingt)
RET	Programmsteuerung	Springe zurück
MOVE	Verschieben	Wert übertragen
SV	Zeiten	Zeit als verlängerten Impuls starten

B.2 Bitverknüpfungsoperationen Beispiel

Beispiel 1: Steuern eines Förderbandes

Das folgende Bild zeigt ein Förderband, das elektrisch in Gang gesetzt werden kann. Am Anfang des Bandes befinden sich zwei Druckschalter, S1 für START und S2 für STOP. Am Ende des Bandes befinden sich ebenfalls zwei Druckschalter, S3 für START und S4 für STOP. Das Band kann von beiden Enden aus gestartet oder gestoppt werden. Außerdem stoppt der Sensor S5 das Band, wenn ein Gegenstand auf dem Band dessen Ende erreicht.



Absolute und symbolische Programmierung

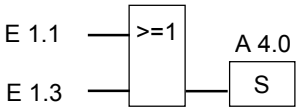
Sie können ein Programm zum Steuern des Förderbandes schreiben, indem Sie die verschiedenen Komponenten des Förderbandsystems mit Hilfe von **absoluten Adressen** oder **Symbolen** darstellen.

Die von Ihnen gewählten Symbole setzen Sie in der Symboltabelle mit den absoluten Adressen in Beziehung (siehe Online–Hilfe zu STEP 7).

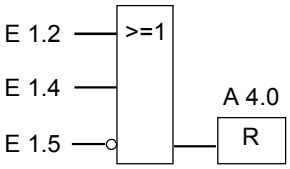
Systemkomponente	Absolute Adresse	Symbol	Symboltabelle
Startschalter	E 1.1	S1	E 1.1 S1
Stoppschalter	E 1.2	S2	E 1.2 S2
Startschalter	E 1.3	S3	E 1.3 S3
Stoppschalter	E 1.4	S4	E 1.4 S4
Sensor	E 1.5	S5	E 1.5 S5
Motor	A 4.0	MOTOR_EIN	A 4.0 MOTOR_EIN

Funktionsplan zum Steuern des Förderbandes

Netzwerk 1: Der Motor wird durch Betätigen eines der beiden Startschalters eingeschaltet.

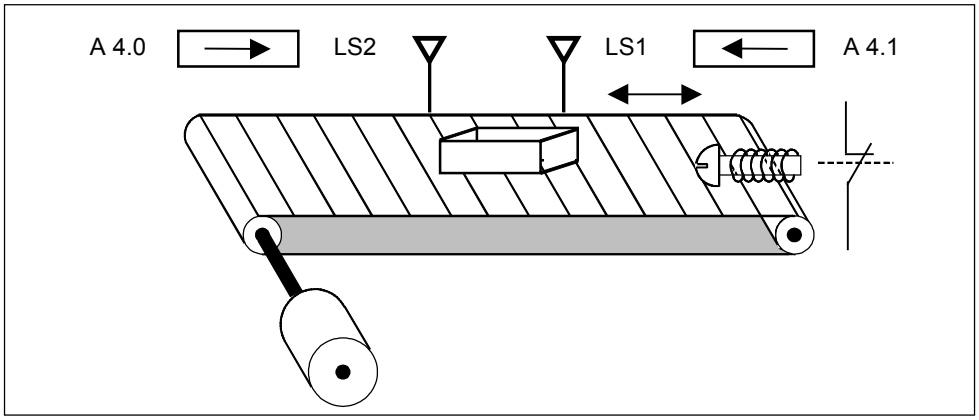


Netzwerk 2: Der Motor wird durch Betätigen eines der beiden Stoppschalter oder durch Ansprechen des Sensors am Ende des Bandes ausgeschaltet.



Beispiel 2: Erfassen der Richtung eines Förderbandes

Das folgende Bild zeigt ein Förderband, das mit zwei Lichtschranken (LS1, LS2) ausgestattet ist. Die Lichtschranken sollen feststellen, in welche Richtung sich ein Paket auf dem Band bewegt.



Absolute und symbolische Programmierung

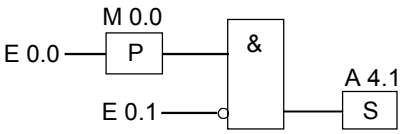
Sie können ein Programm schreiben, das die Richtungsanzeige für das Förderbandsystem aktiviert, indem Sie die verschiedenen Komponenten des Fördersystems mit Hilfe von **absoluten Adressen** oder **Symbolen** darstellen.

Die von Ihnen gewählten Symbole setzen Sie in der Symboltabelle mit den absoluten Adressen in Beziehung (siehe Online–Hilfe zu STEP 7).

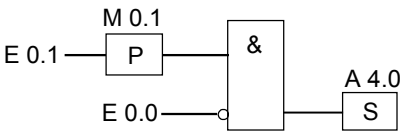
Systemkomponente	Absolute Adresse	Symbol	Symboltabelle
Lichtschranke 1	E 0.0	LS1	E 0.0 LS1
Lichtschranke 2	E 0.1	LS2	E 0.1 LS2
Anzeige für Bewegung nach rechts	A 4.0	RECHTS	A 4.0 RECHTS
Anzeige für Bewegung nach links	A 4.1	LINKS	A 4.1 LINKS
Taktmerker 1	M 0.0	TM1	M 0.0 TM1
Taktmerker 2	M 0.1	TM2	M 0.1 TM2

Funktionsplan zur Richtungserfassung eines Förderbandes

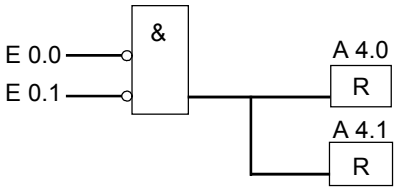
Netzwerk 1: Wenn an E 0.0 ein Wechsel des Signalzustands von "0" auf "1" auftritt (positive Flanke) und gleichzeitig der Signalzustand an E 0.1 "0" ist, dann bewegt sich das Paket auf dem Band nach links.



Netzwerk 2: Wenn an E 0.1 ein Wechsel des Signalzustands von "0" auf "1" auftritt (positive Flanke) und gleichzeitig der Signalzustand an E 0.0 "0" ist, dann bewegt sich das Paket auf dem Band nach rechts.



Netzwerk 3: Ist eine der Lichtschranken unterbrochen, dann befindet sich ein Paket zwischen den Schranken. Die Richtungsanzeiger sind ausgeschaltet.



B.3 Zeitoperationen Beispiel

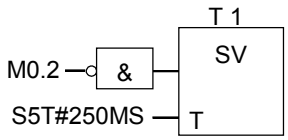
Taktgeber

Zur Erzeugung eines sich periodisch wiederholenden Signals können Sie einen Taktgeber oder ein Blinkrelais verwenden. Taktgeber finden sich häufig in Meldesystemen, die das Blinken von Anzeigeleuchten steuern.

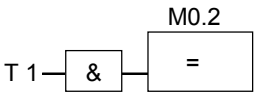
Wenn Sie S7–300 einsetzen, können Sie eine Taktgeberfunktion implementieren, indem Sie die zeitgesteuerte Verarbeitung in speziellen Organisationsbausteinen verwenden.

Funktionsplan zum Generieren eines Taktes (Tastverhältnis 1:1)

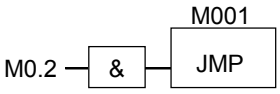
Netzwerk 1: Wenn der Signalzustand der Zeit T1 "0" ist, dann laden Sie den Zeitwert 250 ms in T1 und starten Sie T1 als verlängerten Impuls.



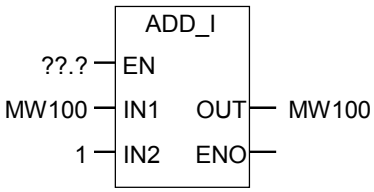
Netzwerk 2: Der Zustand des Timers wird in einem Hilfsmerker zwischengespeichert.



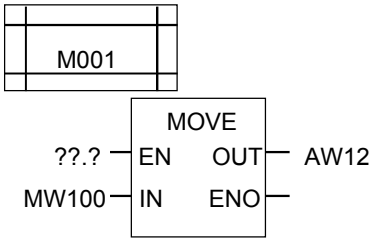
Netzwerk 3: Ist der Signalzustand der Zeit T1 gleich "1", dann springe zur Sprungmarke M001.



Netzwerk 4: Nach jeder abgelaufenen Zeit des Timers T1, wird das Merkerwort 100 um "1" inkrementiert.

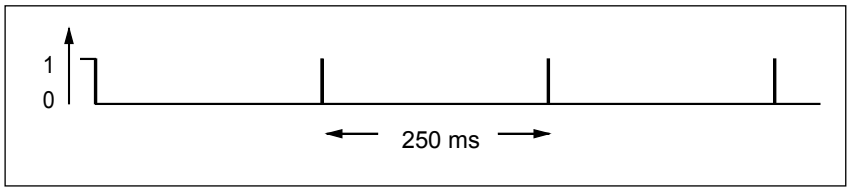


Netzwerk 5: Mit der Operation **MOVE** können Sie sich die unterschiedlichen Taktfrequenzen an den Ausgängen A 12.0 bis A 13.7 anzeigen lassen.

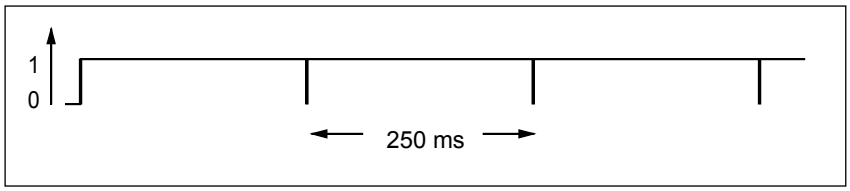


Signalabfrage

Eine Signalabfrage der Zeit T1 liefert für den negierten Eingang der UND-Verknüpfung (M0.2) im Beispiel Taktzeit folgendes Verknüpfungsergebnis:



Sobald die Zeit abgelaufen ist, wird die Zeit erneut gestartet. Daher liefert die Signalabfrage, die vom negierten Eingang der UND-Verknüpfung (M0.2) ausgeführt wird, nur kurz den Signalzustand "1".
Negiertes VKE-Bit:



Alle 250 ms beträgt das VKE–Bit "0". Der Sprung wird ignoriert und der Inhalt des Merkerworts MW100 um "1" inkrementiert.

Erzielen einer bestimmten Frequenz

Mit den Bits der Merkerbytes MB101 und MB100 können Sie folgende Frequenzen erzielen:

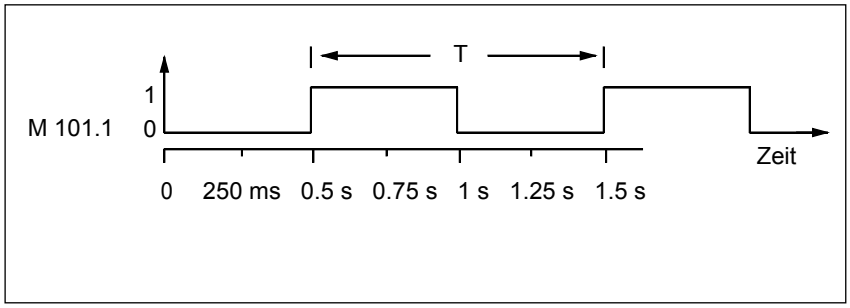
MB101, MB100	Frequenz in Hertz	Dauer	
M 101.0	2.0	0.5 s	(250 ms ein / 250 ms aus)
M 101.1	1.0	1 s	(0.5 s ein / 0.5 s aus)
M 101.2	0.5	2 s	(1 s ein / 1 s aus)
M 101.3	0.25	4 s	(2 s ein / 2 s aus)
M 101.4	0.125	8 s	(4 s ein / 4 s aus)
M 101.5	0.0625	16 s	(8 s ein / 8 s aus)
M 101.6	0.03125	32 s	(16 s ein / 16 s aus)
M 101.7	0.015625	64 s	(32 s ein / 32 s aus)
M 100.0	0.0078125	128 s	(64 s ein / 64 s aus)
M 100.1	0.0039062	256 s	(128 s ein / 128 s aus)
M 100.2	0.0019531	512 s	(256 s ein / 256 s aus)
M 100.3	0.0009765	1024 s	(512 s ein / 512 s aus)
M 100.4	0.0004882	2048 s	(1024 s ein / 1024 s aus)
M 100.5	0.0002441	4096 s	(2048 s ein / 2048 s aus)
M 100.6	0.000122	8192 s	(4096 s ein / 4096 s aus)
M 100.7	0.000061	16384 s	(8192 s ein / 8192 s aus)

Signalzustände der Bits von Merkerbyte MB101

Zyklus	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Zeitwert in ms
0	0	0	0	0	0	0	**0**	0	250
1	0	0	0	0	0	0	**0**	1	250
2	0	0	0	0	0	0	1	0	250
3	0	0	0	0	0	0	1	1	250
4	0	0	0	0	0	1	**0**	0	250
5	0	0	0	0	0	1	**0**	1	250
6	0	0	0	0	0	1	1	0	250
7	0	0	0	0	0	1	1	1	250
8	0	0	0	0	1	0	**0**	0	250
9	0	0	0	0	1	0	**0**	1	250
10	0	0	0	0	1	0	1	0	250
11	0	0	0	0	1	0	1	1	250
12	0	0	0	0	1	1	**0**	0	250

Signalzustand des Merkerbits M 101.1

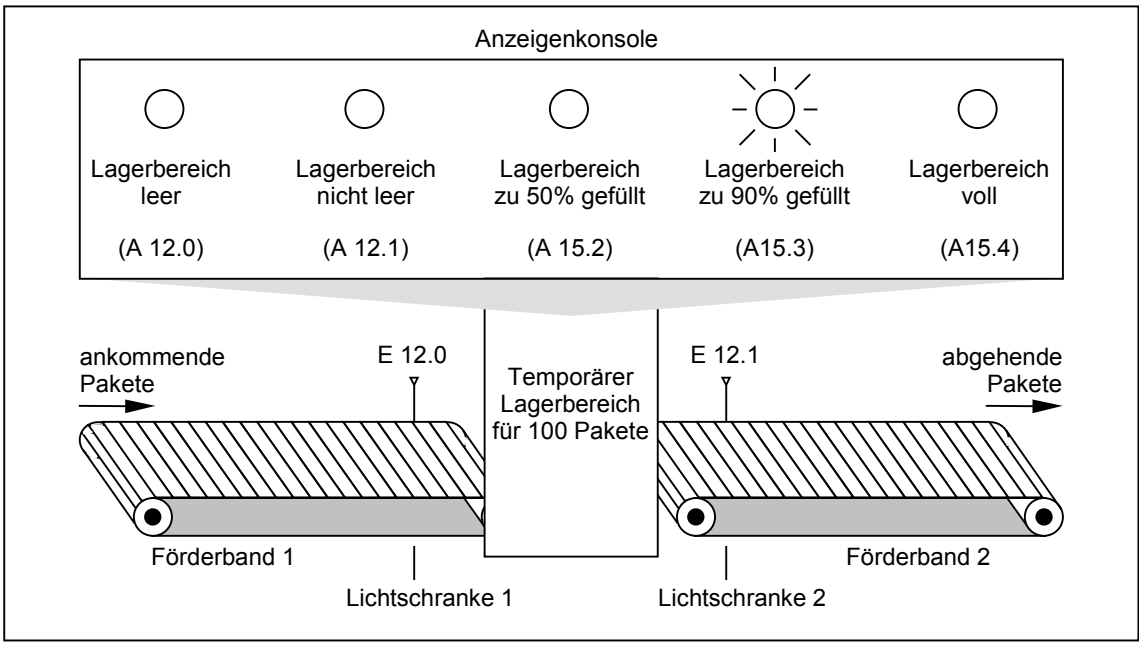
Frequenz = 1/T = 1/1 s = 1 Hz



B.4 Zähl- und Vergleichsoperationen Beispiel

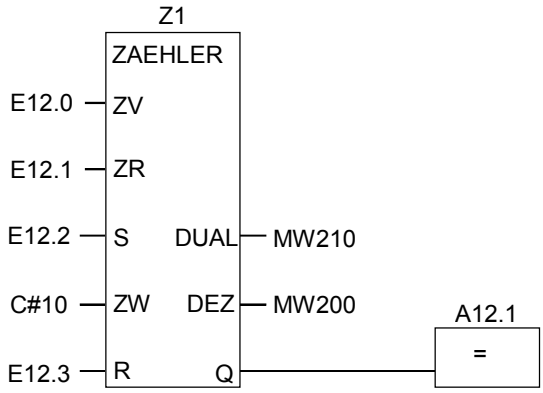
Lagerbereich mit Zähler und Vergleicher

Das folgende Bild zeigt ein System mit zwei Förderbändern und einem temporären Lagerbereich dazwischen. Förderband 1 transportiert die Pakete zum Lagerbereich. Eine Lichtschranke am Ende des Förderbandes 1 neben dem Lagerbereich ermittelt, wie viele Pakete in den Lagerbereich transportiert werden. Förderband 2 transportiert Pakete von diesem temporären Lagerbereich zu einer Laderampe, wo sie zur Auslieferung beim Kunden auf LKW verladen werden. Eine Lichtschranke am Ende des Förderbandes 2 neben dem Lagerbereich ermittelt, wie viele Pakete aus dem Lagerbereich heraus zur Laderampe transportiert werden. Fünf Anzeigeleuchten zeigen an, wie weit der temporäre Lagerbereich gefüllt ist.

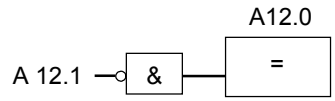


Funktionsplan, der die Anzeigeleuchten aktiviert

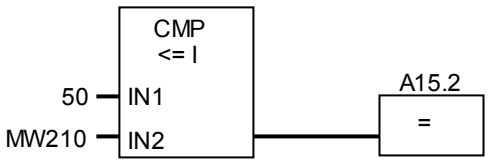
Netzwerk 1: Zähler Z1 zählt vorwärts bei einer Signalflanke von "0" auf "1" an Eingang ZV und zählt rückwärts bei einer Signalflanke von "0" auf "1" an Eingang ZR. Mit einer Signalflanke von "0" auf "1" an Eingang S wird der Zählwert auf den Wert von ZW gesetzt. Mit einer Signalflanke von "0" an Eingang R wird der Zählwert auf "0" gesetzt. Im MW200 ist stets der aktuelle Zählwert von Z1 hinterlegt. A 12.1 zeigt an "Lagerbereich nicht leer".



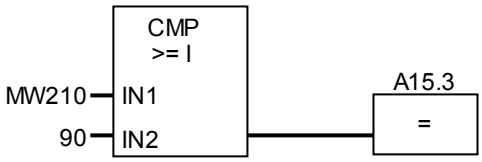
Netzwerk 2: A 12.0 zeigt an "Lagerbereich leer".



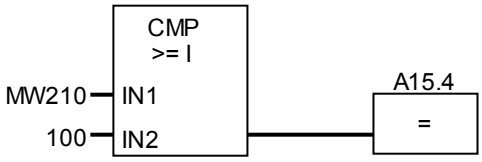
Netzwerk 3: Ist 50 kleiner oder gleich Zählwert (bzw. ist der aktuelle Zählerstand größer oder gleich 50), schaltet sich die Anzeigeleuchte für die Meldung "Lagerbereich zu 50 % voll" ein.



Netzwerk 4: Ist der Zählwert größer oder gleich 90, dann schaltet sich die Anzeigeleuchte für die Meldung "Lagerbereich zu 90 % voll" ein.



Netzwerk 5: Ist der Zählwert größer oder gleich 100, dann schaltet sich die Anzeigeleuchte für die Meldung "Lagerbereich voll" ein.



B.5 Arithmetische Operationen mit Ganzzahlen Beispiel

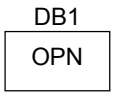
Berechnen einer Gleichung

Das folgende Programmbeispiel zeigt, wie Sie mit drei arithmetischen Operationen für Ganzzahlen das gleiche Ergebnis erzielen, wie die folgende Gleichung:

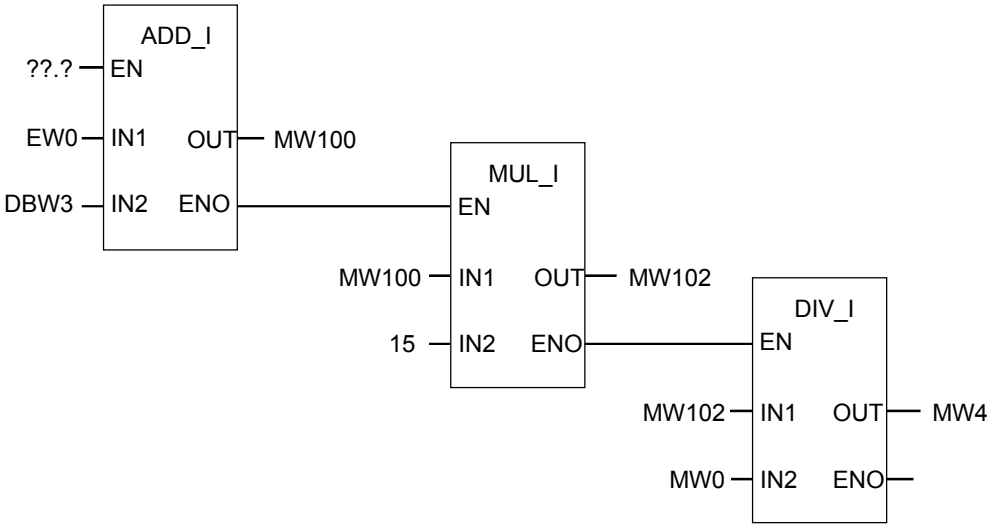
MW4 = ((EW0 + DBW3) x 15) / MW0

Funktionsplan

Netzwerk 1: Datenbaustein DB1 öffnen



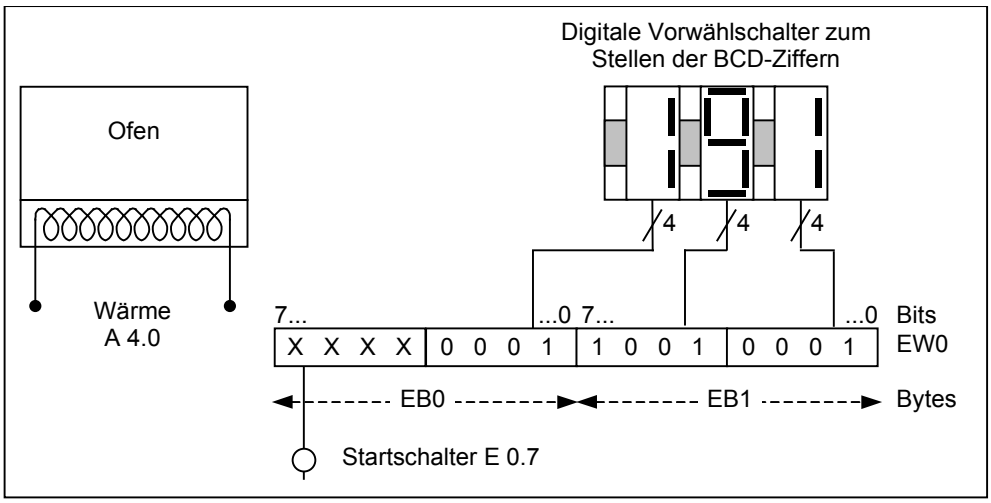
Netzwerk 2: Addiere Eingangswort EW0 und Globaldatenwort DBW3 (der Datenbaustein muß definiert und geöffnet sein). Die Summe wird in MW100 gespeichert. Multipliziere MW100 mit 15. Das Ergebnis wird im Merkerwort MW102 gespeichert. Dividiere MW102 durch MW0.
Das Ergebnis wird in MW4 gespeichert.



B.6 Wortverknüpfungsoperationen Beispiel

Heizen eines Ofens

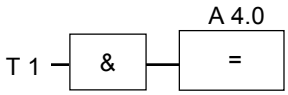
Der Bediener startet das Heizen des Ofens, indem er den Startschalter drückt. Mit den digitalen Vorwählschaltern kann er die Dauer der Heizzeit festlegen. Der Wert, den er setzt, gibt die Sekunden im binär–codierten Dezimalformat (BCD) an.



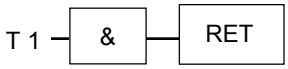
Systemkomponente	Absolute Adresse
Startschalter	E 0.7
Digitale Vorwählschalter für Einer	E 1.0 bis E 1.3
Digitale Vorwählschalter für Zehner	E 1.4 bis E 1.7
Digitale Vorwählschalter für Hunderter	E 0.0 bis E 0.3
Beginn Heizvorgang	A 4.0

Funktionsplan für das Heizen eines Ofens

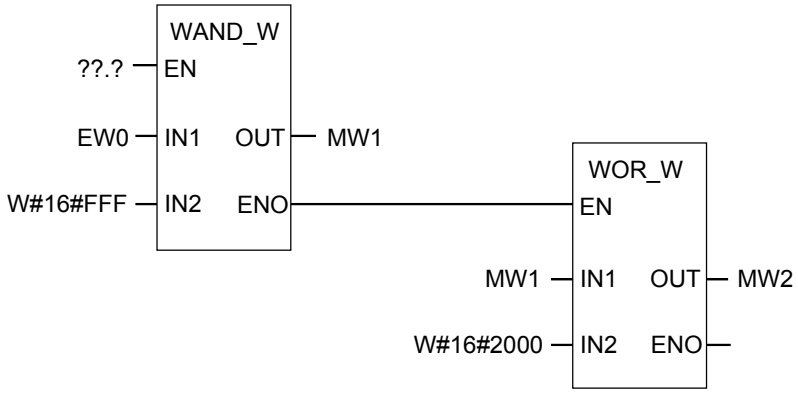
Netzwerk 1: Wenn die Zeit läuft, dann beginne den Heizvorgang.



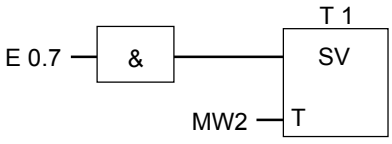
Netzwerk 2: Wenn die Zeit läuft, dann beendet die Operation **Springe zurück** die Bearbeitung hier.



Netzwerk 3: Maskiere die Eingangsbits E 0.4 bis E 0.7 (d. h. setze sie auf "0" zurück). Diese Bits der Vorwählschalter-Eingänge werden nicht verwendet. Die 16 Bits der Vorwählschalter-Eingänge werden mit W#16#0FFF nach der Operation **16 Bit UND verknüpfen** verknüpft. Das Ergebnis wird in das Merkerwort MW1 geladen. Um den Zeitwert in Sekunden einzustellen, wird die Voreinstellung mit W#16#2000 nach der Operation **16 Bit ODER verknüpfen** verknüpft. Bit 13 wird auf "1" gesetzt, Bit 12 wird auf "0" zurückgesetzt.



Netzwerk 4: Starte die Zeit T1 als verlängerten Impuls, wenn der Schalter gedrückt wird. Merkerwort MW2 wird als Voreinstellung geladen (abgeleitet von der Verknüpfungsoperation von oben).



C Arbeiten mit FUP

C.1 EN-/ENO-Mechanismus

Die Freigabe (EN) und der Freigabeausgang (ENO) der FUP/KOP-Boxen wird mittels des BIE-Bits realisiert.

Wenn EN und ENO beschaltet sind, dann gilt:

ENO = EN AND NOT (Fehler der Box)

Wenn kein Fehler auftritt (Fehler der Box = 0), ist somit ENO = EN.

Der EN-/ENO-Mechanismus wird verwendet für:

- arithmetische Operationen,
- Übertragungs- und Umwandlungsoperationen,
- Schiebe und Rotieroperationen,
- Bausteinaufrufe.

Dieser Mechanismus wird **nicht** verwendet für:

- Vergleicher,
- Zähler,
- Timer.

Um die eigentlichen Befehle der Box werden für den EN-/ENO-Mechanismus zusätzliche AWL-Befehle in Abhängigkeit von vorhandenen Vor- und Nachverknüpfungen generiert. Die vier möglichen Fälle werden am Beispiel eines Addierers gezeigt:

- Addierer mit EN- und mit ENO-Beschaltung
- Addierer mit EN- und ohne ENO-Beschaltung
- Addierer ohne EN- und mit ENO-Beschaltung
- Addierer ohne EN- und ohne ENO-Beschaltung

Hinweis zum Erstellen eigener Bausteine

Wenn Sie Bausteine schreiben wollen, die Sie in FUP/KOP aufrufen wollen, so müssen Sie dafür sorgen, daß beim Verlassen des Bausteins das BIE gesetzt ist. Das vierte Beispiel zeigt, daß dies nicht automatisch der Fall ist. Sie können das BIE nicht als Merker verwenden, da es ständig vom EN/ENO-Mechanismus überschrieben wird. Benutzen Sie statt dessen eine temporäre Variable, in der Sie aufgetretene Fehler speichern. Initialisieren Sie diese Variable (im Beispiel unten: fehler) mit 0. An jeder Stelle im Baustein, an der Sie meinen, daß eine mißlungene Operation einen Fehler für den gesamten Baustein darstellt, setzen Sie unter Zuhilfenahme des EN-/ENO-Mechanismus diese Variable. Dazu reicht ein NOT und eine Setze-Spule. Am Ende des Bausteins programmieren Sie ein Netzwerk:

```
ende:  UN fehler
       SAVE
```

Achten Sie darauf, daß dieses Netzwerk in jedem Fall durchlaufen wird, d.h. Sie dürfen innerhalb des Bausteins keinen BEB verwenden und dieses Netzwerk nicht überspringen.

C.1.1 Addierer mit EN- und mit ENO-Beschaltung

Hat der Addierer sowohl EN-Beschaltung als auch ENO-Beschaltung, so werden folgende AWL-Befehle abgesetzt:

```
1   U   E   0.0          // EN-Beschaltung
2   SPBNB _001           // VKE ins BIE schieben und springen wenn VKE == 0
3   L     in1            // Box-Parameter
4   L     in2            // Box-Parameter
5   +I                   // Eigentliche Addition
6   T     out            // Box-Parameter
7   UN    OV             // Fehlererkennung
8   SAVE                 // Fehler im BIE speichern
9   CLR                  // Erstabfrage
10  _001:   U     BIE    // BIE ins VKE schieben
11  =     A   4.0
```

Nach Zeile1 enthält das VKE das Ergebnis der Vorverknüpfung. Der SPBNB-Befehlt kopiert das VKE in das BIE und setzt das Erstabfragebit.

- Ist das VKE 0, dann wird in Zeile 10 gesprungen und mit U BIE fortgesetzt. Die Addition wird nicht ausgeführt. In Zeile 10 wird das BIE wieder ins VKE kopiert und dem Ausgang somit 0 zugewiesen.
- Ist das VKE 1, wird nicht gesprungen, d.h. die Addition wird ausgeführt. In Zeile 7 wird ermittelt, ob bei der Addition ein Fehler auftrat, dies wird in Zeile 8 im BIE gespeichert. Zeile 9 setzt das Erstabfragebit. Nun wird in Zeile 10 das BIE-Bit wieder zurück ins VKE kopiert und somit im Ausgang angezeigt, ob die Addition erfolgreich war.

Das BIE Bit wird durch die Zeilen 10 und 11 nicht mehr verändert, es zeigt also ebenso an, ob die Addition erfolgreich war.

C.1.2 Addierer mit EN- und ohne ENO-Beschaltung

Hat der Addierer eine EN-Beschaltung aber keine ENO-Beschaltung, so werden folgende AWL-Befehle abgesetzt:

```
1          U    E    0.0        // EN-Beschaltung
2          SPBNB _001           // VKE ins BIE schieben und springen wenn VKE == 0
3          L    in1             // Box-Parameter
4          L    in2             // Box-Parameter
5          +I                   // Eigentliche Addition
6          T    out             // Box-Parameter
7  _001:   NOP  0
```

Nach Zeile1 enthält das VKE das Ergebnis der Vorverknüpfung. Der SPBNB-Befehlt kopiert das VKE in das BIE und setzt das Erstabfragebit.

- Ist das VKE 0, dann wird in Zeile 7 gesprungen, die Addition wird nicht ausgeführt, VKE und BIE sind 0.
- War das VKE 1, wird nicht gesprungen, d.h. die Addition wird ausgeführt. Es wird nicht ermittelt, ob bei der Addition ein Fehler auftrat. Das VKE und das BIE sind 1.

C.1.3 Addierer ohne EN- und mit ENO-Beschaltung

Hat der Addierer keine EN-Beschaltung aber ENO-Beschaltung, so werden folgende AWL-Befehle abgesetzt:

```
1   L     in1   // Box-Parameter
2   L     in2   // Box-Parameter
3   +I          // Eigentliche Addition
4   T     out   // Box-Parameter
5   UN    OV    // Fehlererkennung
6   SAVE        // Fehler im BIE speichern
7   CLR         // Erstabfrage
8   U     BIE   // BIE ins VKE schieben
9   =     A     4.0
```

Die Addition wird auf jeden Fall ausgeführt. In Zeile 5 wird ermittelt ob bei der Addition ein Fehler auftrat, dies wird in Zeile 6 im BIE gespeichert. Zeile 7 setzt das Erstabfragebit. Nun wird in Zeile 8 das BIE-Bit wieder zurück ins VKE kopiert und somit im Ausgang angezeigt, ob die Addition erfolgreich war.

Das BIE-Bit wird durch die Zeilen 8 und 9 nicht mehr verändert, es zeigt also ebenso an, ob die Addition erfolgreich war.

C.1.4 Addierer ohne EN- und ohne ENO-Beschaltung

Hat der Addierer weder EN-Beschaltung noch ENO-Beschaltung, so werden folgende AWL-Befehle abgesetzt:

```
1    L     in1    // Box-Parameter
2    L     in2    // Box-Parameter
3    +I           // Eigentliche Addition
4    T     out    // Box-Parameter
5    NOP 0
```

Die Addition wird ausgeführt. Das VKE und das BIE-Bit bleiben unverändert.

C.2 Parameterübergabe

Die Parameter eines Bausteins werden als Wert übergeben. Bei Funktionsbausteinen wird innerhalb des aufgerufenen Bausteins eine Kopie des Aktualparameterwertes im Instanz-DB verwendet. Bei Funktionen liegt eine Kopie des Aktualwertes im Lokaldatenstack. Zeiger werden nicht kopiert. Vor dem Aufruf werden die INPUT-Werte in den Instanz-DB bzw auf den L-Stack kopiert. Nach dem Aufruf werden die OUTPUT-Werte zurück in die Variablen kopiert. Innerhalb des aufgerufenen Baustein arbeitet man nur auf einer Kopie. Die dafür notwendigen AWL-Befehle befinden sich im aufrufenden Baustein und bleiben dem Anwender verborgen.

Hinweis

Wenn Merker, Eingänge, Ausgänge, Peripherieeingänge oder Peripherieausgänge als Aktualoperanden an einer Funktion verwendet werden, werden diese anders behandelt als die anderen Operanden. Die Aktualisierung erfolgt hier nicht über den L-Stack, sondern direkt.

Ausnahme:

Ist der zugehörige Formalparameter ein Eingangsparameter vom Datentyp BOOL, erfolgt die Aktualisierung der Aktualparameter über den L-Stack.

Warnung

Sorgen Sie bei der Programmierung des aufgerufenen Bausteins dafür, daß die als OUTPUT deklarierten Parameter auch beschrieben werden. Sonst sind die ausgegebenen Werte zufällig! Bei Funktionsbausteinen bekommt man den vom letzen Aufruf gemerkten Wert aus dem Instanz-DB, bei Funktionen den zufällig auf dem L-Stack liegenden Wert.

Beachten Sie folgende Punkte:

- Initialisieren Sie wenn möglich alle OUTPUT Parameter.
- Verwenden Sie möglichst keine Setze- und Rücksetze-Befehle. Diese Befehle sind VKE-abhängig. Wenn das VKE den Wert 0 hat, bleibt der zufällige Wert erhalten!
- Wenn Sie innerhalb des Bausteins springen, so achten Sie darauf, daß Sie keine Stellen überspringen, in denen OUTPUT-Parameter beschrieben werden. Denken Sie dabei auch an BEB und die Wirkung der MCR-Befehle.

Index

D

E

F

G

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z