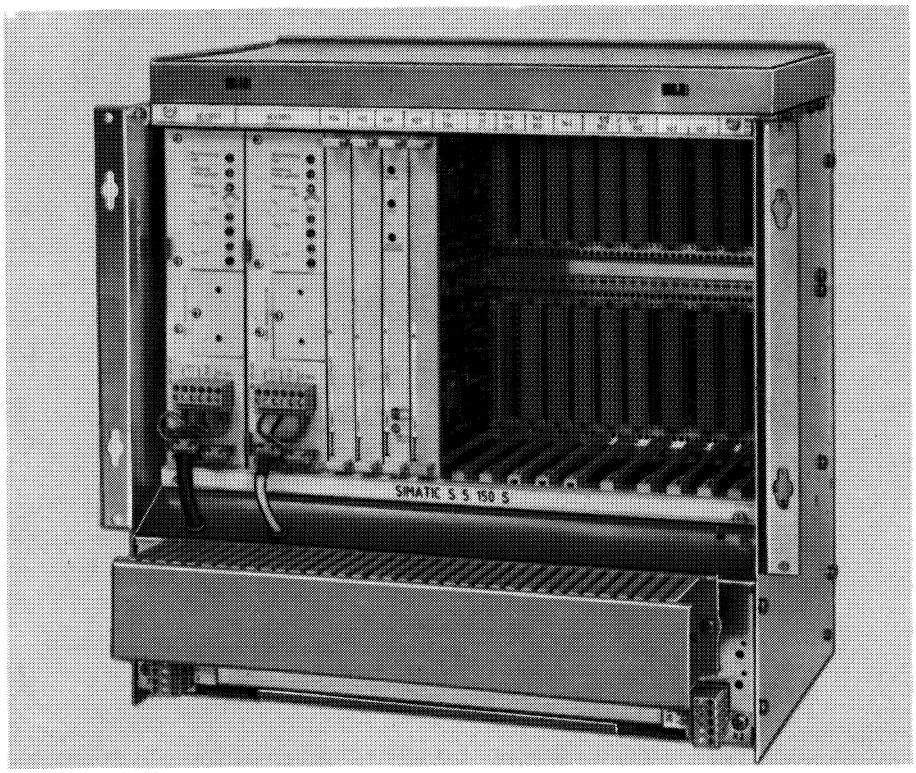
SIEMENS

Automatisierungsgerät S5-150S

Programmieranleitung Bestell-Nr. C79000-B8500-C247-1



Automatisierungsgerät S5-150S

Inhalt

1 Erläuterungen

1.1 Anwendungsbereich
1.2 Programmiersprache STEP 5
1.3 Programmierung

1.1 Anwendungsbereich

Das speicherprogrammierbare Automatisierungsgerät S5-150S ist ein leistungsfähiges Gerät zur Prozeßautomatisierung (Steuern, Melden, Überwachen, Regeln, Protokollieren). Es ist sowohl für den Aufbau einfachster Steuerungen mit binären Signalen als auch zur Lösung umfangreicher Automatisierungsaufgaben einsetzbar. Seine Anwenderprogramme werden mit der Programmiersprache STEP 5 erstellt.

1.2 Programmiersprache STEP 5

Die Operationen der Programmiersprache STEP 5 ermöglichen die Programmierung von einfachen binären Funktionen bis hin zu komplexen digitalen Funktionen und arithmetischen Grundoperationen.

Bei der Programmierung sind die drei Darstellungsarten Funktionsplan (FUP), Kontaktplan (KOP) und Anweisungsliste (AWL) (Bild 1) möglich, so daß die Programmiermethode dem jeweiligen Anwendungsfall angepaßt werden kann. Der von den Programmiergeräten erzeugte Maschinencode ist bei den drei Darstellungsarten identisch. Bei Berücksichtigung bestimmter Programmierregeln (siehe „Hinweise für die Programmierung in Anweisungsliste" Abschnitt 7) kann das Programmiergerät (PG) 670 das Anwenderprogramm von einer Darstellungsart in die jeweils andere übersetzen.

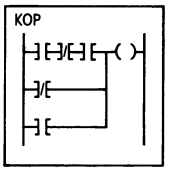
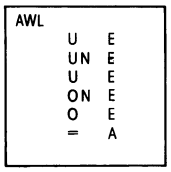
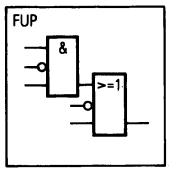
Kontaktplan	Anweisungsliste	Funktionsplan
Programmieren mit grafischen Symbolen wie Stromlaufplan	Programmieren mit mnemotechnischen Abkürzungen der Funktionsbezeichnungen	Programmieren mit grafischen Symbolen
entspricht DIN 19239 (Entwurf)	entspricht DIN 19239 (Entwurf)	entspricht IEC 117-15 DIN 40700 DIN 40719 DIN 19239 (Entwurf)
		

Bild 1
Darstellungsarten der Programmiersprache STEP 5

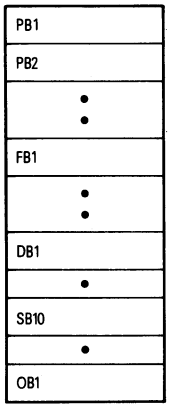


Bild 2
Ablage der Bausteine in beliebiger Reihenfolge in den Programmspeicher

1.3 Programmierung

Programmstruktur

Das Gesamtprogramm eines Automatisierungsgerätes besteht aus dem Systemprogramm und dem Anwenderprogramm. Das Systemprogramm enthält die Gesamtheit aller Anweisungen und Vereinbarungen geräteinterner Betriebsfunktionen (z. B. Sicherstellen von Daten bei Ausfall der Versorgungsspannung, Veranlassung von Anwenderreaktionen für bestimmte Betriebsfälle usw.). Dieses Programm ist ein fester Bestandteil des Automatisierungsgerätes (EPROM) und darf vom Anwender nicht verändert werden.

Das Anwenderprogramm ist die Gesamtheit aller vom Anwender programmierten Anweisungen und Vereinbarungen für die Signalverarbeitung, durch die eine zu steuernde Anlage (Prozeß) gemäß der Steuerungsaufgabe beeinflußt wird.

Das Automatisierungsgerät S5-150S zwingt aufgrund seiner Struktur den Anwender zur strukturierten Programmierung, d. h. zum Aufteilen des Gesamtprogramms in einzelne, in sich abgeschlossene Programmabschnitte (Bausteine). Dieses Verfahren bietet dem Anwender folgende Vorteile:

- einfache und übersichtliche Programmierung auch großer Programme,
- Möglichkeit zum Standardisieren von Programmteilen,
- einfache Programmorganisation,
- leichte Änderungsmöglichkeit,
- einfachen Programmtest,
- einfache Inbetriebnahme.

Für die Gliederung des Anwenderprogramms gibt es verschiedene Bausteintypen, die für unterschiedliche Aufgaben verwendet werden:

Organisationsbausteine (OB)

Sie stellen die Schnittstelle zwischen Systemprogramm und Anwenderprogramm dar.

Programmbausteine (PB)

Sie werden zur Strukturierung des Anwenderprogramms in technologisch orientierte Programmteile eingesetzt.

Funktionsbausteine (FB)

Sie dienen zum Programmieren von häufig wiederkehrenden komplexen Funktionen (z. B. Einzelsteuerung, Melde-, Rechen- und Regelfunktionen).

Schrittbausteine (SB)

Sie sind Sonderformen von Programmbausteinen zur Bearbeitung von Ablaufketten.

Datenbausteine (DB)

Sie dienen zum Abspeichern von Daten oder Texten. Dieser Bausteintyp unterscheidet sich in seiner Funktion grundsätzlich von den restlichen Bausteinen.

Der Anwender kann alle ihm zur Verfügung stehenden Organisationsbausteine programmieren. Diese sind für definierte Betriebsfälle vorgesehen und dienen zum Anstoßen der verschiedenen Möglichkeiten der Programmbearbeitung (siehe Übersichtstabelle Seite 11). Von jedem der Programm-, Funktions-, Schritt- und Datenbausteine können maximal 255 Bausteine programmiert werden. Kein Baustein sollte 256 Anweisungen überschreiten (Bausteinlänge max. 2×2^{10} Anweisungen).

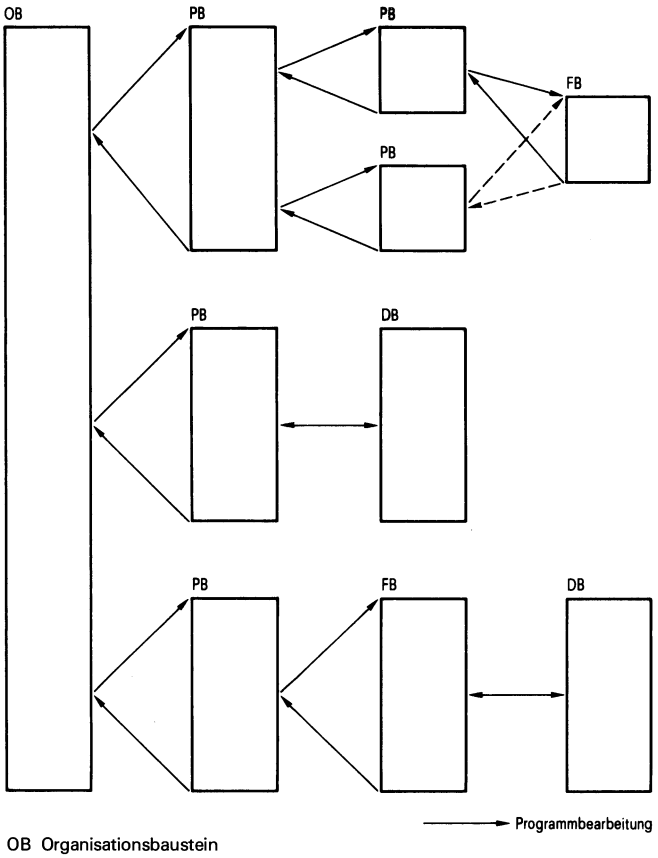
Alle programmierten Bausteine werden vom Programmiergerät in beliebiger Reihenfolge im Programmspeicher hinterlegt (Bild 2).

Programmorganisation

Mit der Programmorganisation wird festgelegt, ob und in welcher Reihenfolge die vom Anwender erstellten Bausteine bearbeitet werden (Bild 3). Dazu werden in Organisationsbausteinen entsprechende Aufrufe (bedingt oder unbedingt) der gewünschten Bausteine programmiert.

Von Organisations-, Programm-, Funktions- und Schrittbausteinen können weitere Programm-, Funktions- und Schrittbausteine in beliebiger Kombination aufgerufen werden.

Ein Richtwert für die maximal zulässige Schachtelungstiefe liegt insgesamt bei 18 Bausteinen. Dieser Richtwert soll als Summe der aus allen drei möglichen Betriebsarten (zyklisch, alarmgesteuert, zeitgesteuert) resultierenden Bausteinschachtelungstiefe verstanden werden.



OB Organisationsbaustein
PB Programmbaustein
FB Funktionsbaustein
DB Datenbaustein

Bild 3
Programmorganisation in der Programmiersprache STEP5

Programmbearbeitung

Das Anwenderprogramm kann auf drei verschiedene Arten bearbeitet werden (Bild 4):

Zyklische Programmbearbeitung

Für eine zyklische Programmbearbeitung des Anwenderprogramms ist der Organisationsbaustein 1 vorgesehen. Dieser Baustein wird zyklisch durchlaufen und ruft dabei die dort programmierten Bausteine auf.

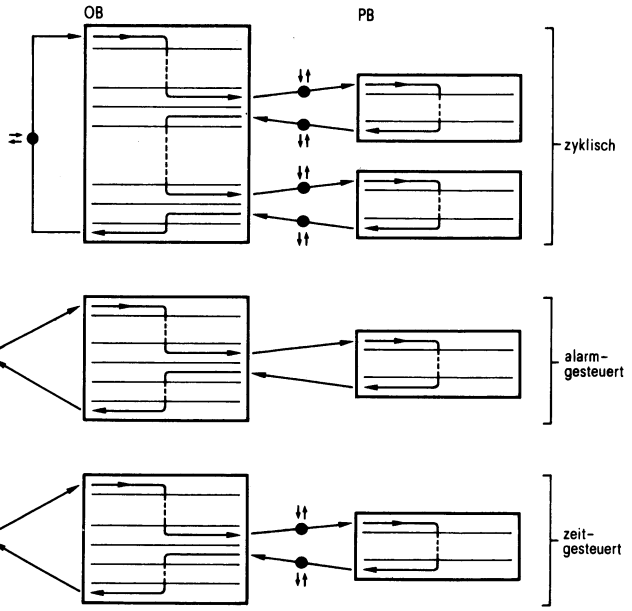
Alarmgesteuerte Programmbearbeitung

Bei dieser Art der Programmbearbeitung wird die zyklische Programmbearbeitung, abhängig von acht Eingangssignalen (Prozeßalarme), und in Abhängigkeit vom Anwenderprogramm oder über eine periphere Initiative (Anforderungsalarme) jeweils bei einem Bausteinwechsel unterbrochen. Zum Aufruf der Alarmprogramme sind die Organisationsbausteine OB 2 bis OB 9 für Prozeßalarme und OB 35 bis OB 39 für Anforderungsalarme vorgesehen.

Zeitgesteuerte Programmbearbeitung

Bei dieser Art der Programmbearbeitung werden bestimmte Programmabschnitte (aufgerufen durch OB 10 bis OB 18) in einem wählbaren Zeitraster automatisch in die zyklische Programmbearbeitung eingeschoben. Durch diese Programmiermöglichkeit kann die mittlere Zykluszeit für ein Anwenderprogramm gesenkt werden.

Die zeitgesteuerte Programmbearbeitung ist erforderlich für die Lösung von Regelungsaufgaben.



⇅ Unterbrechungspunkt, an dem eine alarmgesteuerte oder zeitgesteuerte
● Programmbearbeitung eingeschoben wird.

Bild 4
Arten der Programmbearbeitung

1.4 Allgemeine Hinweise

Werden Standardfunktionsbausteine eingesetzt, so werden die Merkerbytes 200 bis 255 belegt und sind für den Anwender nicht mehr verwendbar.

Ebenso ist die Zeit 0, der Zähler 0 und der Datenbaustein 0 bereits belegt.

Wird das Servicegerät 333 eingesetzt, so ist der Datenbaustein 1 oder 6 belegt. Die Datenwörter 0 der Datenbausteine müssen freigegeben werden.

Für die Regelungsfunktionen sind die Datenbausteine 2, 3 und 4 zur Datenübergabe freizuhalten.

Für den Datum- und Uhrzeitbaustein ist der Datenbaustein 5 freizuhalten.

Standard-Funktionsbausteine belegen die Nummern 1 bis 199. Anwender-Funktionsbausteine sind daher nur mit den Nummern 200 bis 255 zu erstellen.

Programmbausteine können in den drei Darstellungsarten (AWL, KOP, FUP) mit dem ,,Grundoperationsvorrat'' der Programmiersprache STEP 5 programmiert werden.

2.1 Programmierung von Programmbausteinen

Die folgende Beschreibung gilt für die Programmierung der Organisations-, Programm- und Schrittbausteine. Diese drei Bausteintypen unterscheiden sich hinsichtlich ihrer Programmierung nicht (Programmierung von Funktionsbausteinen siehe Seite 8). Sie können in den drei Darstellungsarten (AWL, KOP, FUP) der Programmiersprache STEP 5 programmiert werden. Die Programmierung fängt mit der Eingabe einer Bausteinnummer an:

Programmbausteine	1 bis 255
Schrittbausteine	1 bis 255
Organisationsbausteine	1 bis 39

Danach folgt das eigentliche Steuerungsprogramm, das mit der Anweisung ,,BE" abgeschlossen wird. Verwendet werden darf dabei nur der STEP-5-Grundoperationsvorrat.

Ein Baustein sollte einschließlich ,,BE" aus maximal 256 Anweisungen bestehen (Bild 5). Der Bausteinkopf, den das Programmiergerät automatisch zum Baustein generiert, belegt weitere 5 Wörter im Programmspeicher.

Ein Programmbaustein soll immer ein abgeschlossenes Programm beinhalten. Verknüpfungen über Bausteingrenzen hinweg sind nicht sinnvoll.

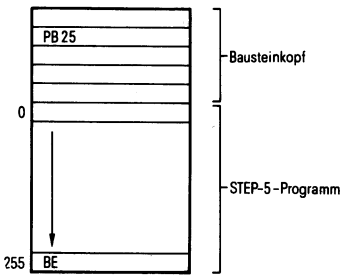


Bild 5
Aufbau eines Organisations-, Programm- und Schrittbausteins

2.2 Aufruf von Programmbausteinen

Durch Bausteinaufrufe werden die Bausteine zum Bearbeiten freigegeben (Bild 6). Diese Bausteinaufrufe können innerhalb eines Organisations-, Schritt-, Programm- oder Funktionsbausteins programmiert werden. (Nur Organisationsbausteine dürfen nicht vom Anwenderprogramm aufgerufen werden). Sie sind vergleichbar mit ,,Sprünge in ein Unterprogramm" und können sowohl unbedingt als auch bedingt ausgeführt werden.

Nach der Anweisung ,,BE" wird in den Baustein zurückgesprungen, in dem der Bausteinaufruf programmiert wurde. Sowohl nach einem Bausteinaufruf als auch nach ,,BE" kann das Verknüpfungsergebnis nicht mehr weiter verknüpft werden. Das Verknüpfungsergebnis wird jedoch in den ,,neuen Baustein" mitgenommen und kann ausgewertet werden.

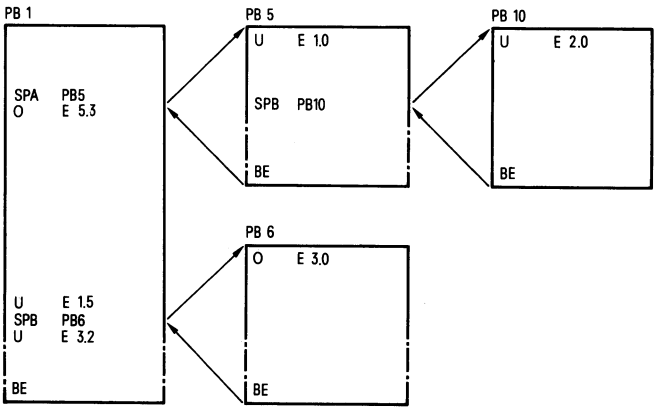


Bild 6
Bausteinaufrufe, die die Bearbeitung eines Programmbausteins freigeben

Unbedingter Aufruf: SPA xx

Der angesprochene Programmbaustein wird unabhängig vom vorherigen Verknüpfungsergebnis bearbeitet.

Bedingter Aufruf: SPB xx

Der angesprochene Programmbaustein wird abhängig vom vorherigen Verknüpfungsergebnis bearbeitet.

3 Datenbausteine

3.1 Programmierung von Datenbausteinen

In Datenbausteinen (DB) werden die Daten abgespeichert, die innerhalb des Anwenderprogramms erforderlich sind. In Datenbausteinen werden keine STEP-5-Operationen ausgeführt.

Daten können sein:

beliebige Bitmuster; z. B. für Anlagenzustände

Zahlen (Hexa, Dual, Dezimal); z. B. für Zeitwerk, Rechenergebnisse

alphanumerische Zeichen; z. B. für Meldetexte.

Das Erstellen eines Datenbausteins beginnt mit der Angabe einer Datenbaustein-Nummer zwischen 1 und 255 (Beispiel: DB 25). Jeder Datenbaustein kann aus bis zu 256 Datenwörtern (16 bit) bestehen (Bild 7). Das Eingeben von Daten muß in aufsteigender Reihenfolge der Datenwörter, beginnend mit Datenwort 0, erfolgen, wobei das Datenwort 0 (DW 0) vom Anwender nicht verwendet werden sollte, da es als Zwischenspeicher für bestimmte Funktionsbausteine vorgesehen ist.

Pro Datenwort wird im Programmspeicher ein Speicherwort belegt. Vom Programmiergerät wird außerdem zu jedem Datenbaustein ein Bausteinkopf generiert, der weitere 5 Wörter im Programmspeicher belegt.

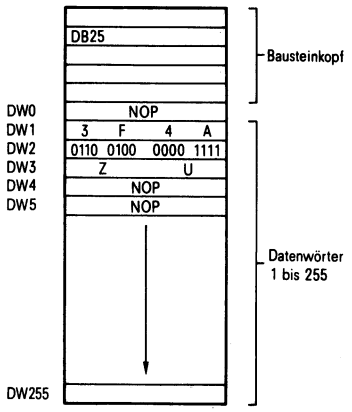


Bild 7
Aufbau eines Datenbausteins

3.2 Aufruf von Datenbausteinen

Datenbausteine (DB) können nur unbedingt aufgerufen werden. Die Anwahl eines Datenbausteines bleibt solange gültig, bis ein neuer Datenbaustein angewählt wird. Ein Datenbaustein kann innerhalb eines OB, PB, SB oder FB mit dem Befehl ,,A DBxxx'' adressiert werden.

Beispiel

Es soll der Inhalt des Datenwortes 1 vom Datenbaustein 10 in das Datenwort 1 des Datenbausteins 20 transferiert werden (Bild 8).

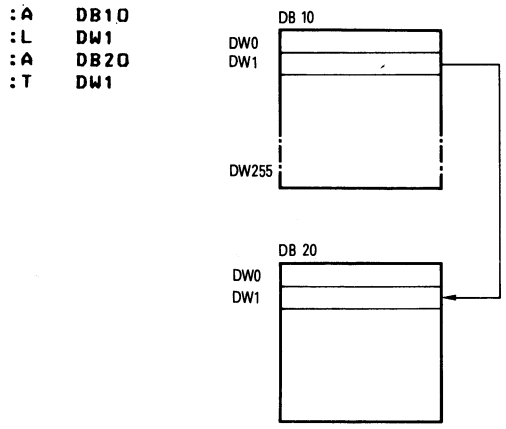


Bild 8
Adressieren eines Datenbausteins

Wird von einem Programmbaustein, in dem bereits ein Datenbaustein adressiert wurde, ein weiterer Programmbaustein aufgerufen und in diesem Baustein ein anderer Datenbaustein adressiert, so ist dieser Datenbaustein nur in dem aufgerufenen Programmbaustein gültig. Nach dem Rücksprung in den aufgerufenen Programmbaustein gilt wieder der alte Datenbaustein (Bild 9).

Beispiel

Im Programmbaustein 7 wird der Datenbaustein 10 angewählt. In der folgenden Bearbeitung werden die Daten dieses Datenbausteins bearbeitet.

Nach dem Aufruf wird der Programmbaustein 20 bearbeitet. Der Datenbaustein 10 ist jedoch nach wie vor gültig. Erst mit dem Aufruf von Datenbaustein 11 wird der Datenbereich gewechselt. Bis zum Ende von Programmbaustein 20 ist nun Datenbaustein 11 gültig.

Nach dem Bausteinwechsel zurück in Programmbaustein 7 ist wieder der Datenbaustein 10 gültig.

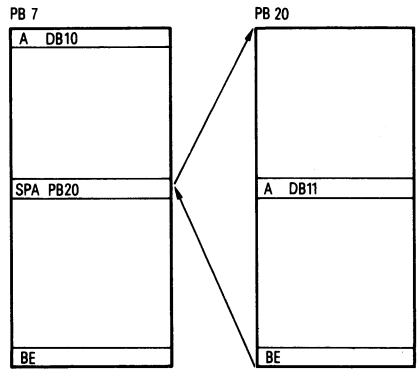


Bild 9
Gültigkeitsbereich eines angewählten Datenbausteins

4.1 Allgemeines

Funktionsbausteine sind ebenso Teile des Anwenderprogramms wie z. B. Programmbausteine. Sie weisen jedoch gegenüber den Organisations-, Programm- und Schrittbausteinen vier wesentliche Unterschiede auf:

- Funktionsbausteine lassen sich parametrieren, d. h. es können die Aktualoperanden, mit denen ein Funktionsbaustein arbeiten soll, durch Formaloperanden variabel gestaltet werden.
- Funktionsbausteine können mit einem gegenüber den Organisations-, Programm- und Schrittbausteinen erweiterten Operationsvorrat programmiert werden. (Seite 44).
- Das Programm eines Funktionsbausteins läßt sich nur als Anweisungsliste erstellen und dokumentieren.
- Der Aufruf eines Funktionsbausteins wird graphisch als ,,schwarzer Kasten" dargestellt.

  Funktionsbausteine stellen innerhalb des Anwenderprogramms eine komplexe, abgeschlossene Funktion dar. Ein Funktionsbaustein kann entweder als Softwareprodukt von Siemens bezogen werden (,,Standard-Funktionsbausteine" auf Mini-Diskette) oder vom Anwender selbst programmiert werden. Die ergänzenden Operationen, die es zusätzlich zu den Grundoperationen (Seite 28) gibt, können nur in Funktionsbausteinen programmiert werden.

4.2 Aufbau von Funktionsbausteinen

Ein Funktionsbaustein besteht aus dem Bausteinkopf und dem Bausteinrumpf (Bild 10).

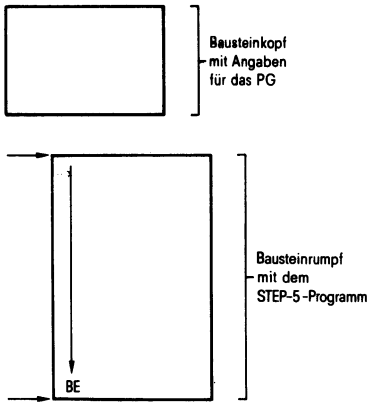


Bild 10
Aufbau eines Funktionsbausteins

Bausteinkopf

Der Bausteinkopf enthält alle Angaben, die das Programmiergerät benötigt, um den Funktionsbaustein graphisch darstellen zu können und um die Operanden bei der Parametrierung des Funktionsbausteins prüfen zu können. Vor der Programmierung des Funktionsbausteins wird dieser Bausteinkopf (mit Unterstützung des Programmiergeräts) vom Anwender eingegeben (siehe ,,Erstellung eines Funktionsbausteins" Seite 8).

Bausteinrumpf

Der Bausteinrumpf enthält das eigentliche Programm des Funktionsbausteins. Im Bausteinrumpf ist die auszuführende Funktion mit der Programmiersprache STEP 5 beschrieben und niedergelegt. Bei einem Aufruf des Funktionsbausteins wird nur der Bausteinrumpf bearbeitet. Zum Programmieren eines Funktionsbausteins ist ein gegenüber den Grundoperationen erweiterter Operationsvorrat vorhanden (siehe ,,Beschreibung der ergänzenden Operationen" Seite 44).

4.3 Aufruf und Parametrierung

Mit Funktionsbausteinen (FB) werden häufig wiederkehrende oder sehr komplexe Funktionen realisiert. Sie stehen nur einmal im Programmspeicher und werden von einem übergeordneten Baustein einmal oder mehrfach aufgerufen, wobei bei jedem Aufruf andere Parameter verwendet werden können.

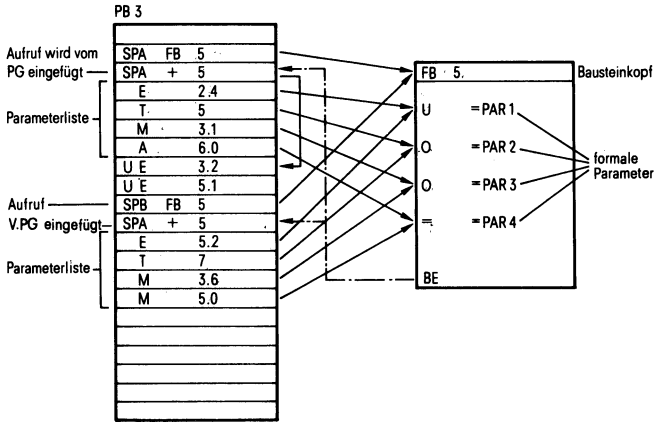


Bild 11
Aufruf eines Funktionsbausteins

Funktionsbausteine werden unter einer bestimmten Bezeichnung (FB 1 bis FB 255) im Programmspeicher hinterlegt. Anwender-Funktionsbausteine sollen von FB 255 abfallend adressiert werden, um nicht mit den Standard-Funktionsbausteinen, die von FB 1 bis FB 199 adressiert sind, zu kollidieren.

Der Aufruf eines Funktionsbausteins kann innerhalb eines Organisationsbausteins, Schrittbausteins, Programmbausteins oder eines anderen Funktionsbausteins programmiert werden. Der Aufruf setzt sich aus der Aufrufanweisung und der Parameterliste zusammen.

Aufrufanweisung

SPA FBn unbedingter Aufruf
SPB FBn bedingter Aufruf

Unbedingter Aufruf:

Der angesprochene Funktionsbaustein wird unabhängig vom vorherigen Verknüpfungsergebnis bearbeitet.

Bedingter Aufruf:

Der angesprochene Funktionsbaustein wird nur dann bearbeitet, wenn das vorherige Verknüpfungsergebnis ,,VKE" = 1 ist.

4.4 Programmierung von Funktionsbausteinen

Parameterliste

Die Parameterliste steht direkt nach der Aufrufanweisung (Bild 11). In ihr werden die Eingangs-, Ausgangsvariablen und Daten definiert (siehe ,,Art der Bausteinparameter" Seite 10). Die Parameterliste kann maximal 40 Variable enthalten.

Bei der Bearbeitung des Funktionsbausteins werden anstelle der formalen Parameter die Variablen aus der Parameterliste verwendet. Die Reihenfolge der Variablen in der Parameterliste wird durch das Programmiergerät überwacht.

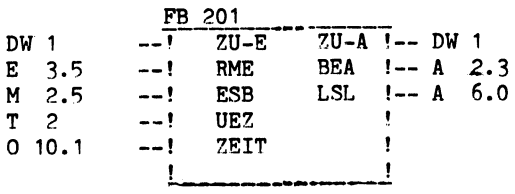
Die Sprunganweisung hinter dem FB-Aufruf wird vom Programmiergerät automatisch eingefügt, beim Auslesen jedoch nicht angezeigt.
Der FB-Aufruf belegt im Programmspeicher zwei Wörter, jeder Parameter ein weiteres Speicherwort.
Die erforderliche Speicherlänge der Standard-Funktionsbausteine sowie die Laufzeit werden im Katalog ST 56 angegeben.

Beispiel für den Aufruf eines Funktionsbausteins und Übergabe von Parametern mit der Programmiermethode Anweisungsliste und Kontakt-/Funktionsplan:

AWL

```
      : SPA FB 201
NAME : E-ANTR
ZU-E :     DW   1
RME  :     E    3.5
ESB  :     M    2.5
UEZ  :     T    2
ZEIT :     KTO  10.1
ZU-A :     DW   1
REA  :     A    2.3
LSL  :     A    6.0
```

KOP / FUP



Die bei der Programmierung am Programmiergerät erscheinenden Bezeichner für die Ein- und Ausgänge des Funktionsbausteins, sowie der Name, sind im Funktionsbaustein selbst abgelegt. Deshalb müssen, bevor mit der Programmierung am Programmiergerät begonnen wird, alle erforderlichen Funktionsbausteine auf die Programmdiskette überspielt bzw. direkt in den Programmspeicher des Automatisierungsgerätes eingegeben werden (Näheres siehe Bedienungsanleitung PG 670).

4.4 Programmierung von Funktionsbausteinen

Entsprechend dem Aufbau eines Funktionsbausteins gliedert sich die Erstellung in zwei Teile:

Eingabe des Bausteinkopfes und Eingabe des Bausteinrumpfes.

Vor der Eingabe des Bausteinrumpfes (STEP-5-Programm) wird der Bausteinkopf eingegeben. Der Bausteinkopf enthält:

die Bibliotheksnummer,

den Namen des Funktionsbausteins,

Formaloperanden (die Namen der Bausteinparameter),

die Art des Bausteinparameters,

den Typ des Bausteinparameters.

Bibliotheksnummern

Es kann eine Nummer von 0 bis 65535 vorgegeben werden. Dem Funktionsbaustein wird diese Nummer zugeordnet, unabhängig von seinem symbolischen oder absoluten Parameter.

Eine Bibliotheksnummer sollte nur einmal vorgegeben werden, um einen bestimmten Funktionsbaustein eindeutig identifizieren zu können. Standard-Funktionsbausteine haben eine Produktnummer.

Name des Funktionsbausteins

Der Name, der den Funktionsbaustein bezeichnet, kann maximal 8 Zeichen lang sein. Er ist nicht mit dem symbolischen Anlagenkennzeichen identisch.

Formaloperand (Name des Bausteinparameters)

Der Formaloperand kann maximal 4 Zeichen lang sein und muß mit einem Buchstaben beginnen.
Es können pro Funktionsbaustein max. 40 Parameter programmiert werden.

Art der Bausteinparameter

Als Art des Bausteinparameters kann ,,E", ,,A", ,,D", ,,B", ,,T" oder ,,Z" eingegeben werden.

E = Eingangsparameter
A = Ausgangsparameter
D = Datum
B = Befehl
T = Zeit (Timer)
Z = Zähler

,,E, D, B, T" oder ,,Z" sind Parameter, die bei graphischer Darstellung auf der linken Seite des Funktionssymbols gezeichnet werden. Mit ,,A" gekennzeichnete Parameter werden bei der graphischen Darstellung auf der rechten Seite des Funktionssymbols gezeichnet.

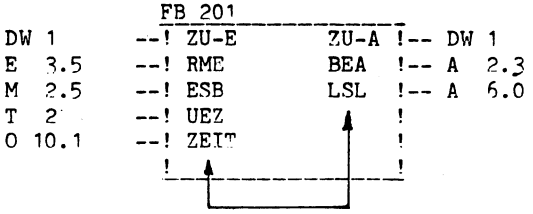
Beispiel für den Aufruf eines Funktionsbausteins

AWL

```
     :SPA FB 201
NAME :E-ANTR ——— Name des Funktionsbausteins
ZU-E :    DW  1
RME  :    E   3.5
ESB  :    M   2.5
UEZ  :    T   2
ZEIT :    KTO10.1
ZU-A :    DW  1
BEA  :    A   2.3
LSL  :    A   6.0
```

↖ Formaloperanden (Name der Bausteinparameter)

KOP / FUP



Formaloperanden (Name der Bausteinparameter)

Parametrierung

Operationen (Substitutionsbefehle), die parametrierbar sein sollen, werden im Funktionsbaustein mit dem Formaloperanden programmiert (formal). Dabei können die Formaloperanden auch mehrmals an verschiedenen Stellen im Funktionsbaustein angesprochen werden.

Beispiel: Aufruf des Funktionsbausteins

AWL

```
     : SPA FB 202
NAME : BEISPIEL
ANNA :    E 13.5
BERT :    M 17.7
HANS :    A 23.0
```

KOP/FUP

```
          FB 202
E 13.5    --!ANNA        HANS!--A 23.0
M 17.7    --!BERT             !
            !                 !
            !_________________!
```

Programm im Funktionsbaustein

```
NAME : BEISPIEL
BEZ  : ANNA     E/A/D/B/T/Z :   E   BI/BY/W/D : BI
BEZ  : BERT     E/A/D/B/T/Z :   E   BI/BY/W/D : BI
BEZ  : HANS     E/A/D/B/T/Z :   A   BI/BY/W/D : BI
     : U  = ANNA
     : U  = BERT
     : =  = HANS
```

Formaloperand — Parameterart — Parametertyp

Ausgeführtes Programm

```
     : U    E 13.5
     : U    M 17.7
     : =    A 23.0
```

Aktualoperand

Beispiel: Standard-Funktionsbaustein FB 16 / RAD: 84 (ST 56 1981)

Mit dem Funktionsbaustein ,,Radizierer: B4'' läßt sich eine Zahl, die im BCD-Code (4 Dekaden) vorliegt, radizieren. Das Ergebnis liegt im BCD-Code (2 Dekaden Wurzel, 4 Dekaden Rest) vor.

Funktion: Y = A
Y = Z 3; Rest = Z 4; A = Z 1

Aufruf FB 16

```
          AWL                          KOP
          :SPA  FB 16     DW 6 --  Z 1      Z 4  --  DL 7
NAME      :RAD: B  4               |        Z 3  --  DW 8
Z 1       :     DW 6               |_____________|
Z 4       :     DL 7
Z 3       :     DW 8
```

Parametername	Benennung	Art	Typ
Z 1	Radikand	E	W
Z 4	Rest	A	W
Z 3	Wurzel	A	BY

Im obigen Beispiel wird die Zahl, die im Datenwort DW 6 im BCD-Code (4 Dekaden) bereitgestellt ist, mit dem Aufruf des Standard-Funktionsbausteins FB 16 radiziert. Das Ergebnis wird im BCD-Code (2 Dekaden) in Datenwort 7, linkes Byte, abgelegt, sowie ein 4 Dekaden-Rest in DW 8.

Allgemeine Hinweise:

Werden Standard-Funktionsbausteine eingesetzt, so werden die Merkerbytes 200 bis 255 belegt und sind für den Anwender nicht mehr verwendbar.
Ebenso ist die Zeit 0, der Zähler 0 und der Datenbaustein 0 bereits belegt.
Wird das Servicegerät 333 eingesetzt, so ist der Datenbaustein 1 belegt. Die Datenwörter 0 der Datenbausteine müssen freigehalten werden.
Für die Regelungsfunktionen sind die Datenbausteine 2, 3 und 4 zur Datenübergabe freizuhalten.

Standard-Funktionsbausteine belegen die Nummern 1 bis 199. Anwender-Funktionsbausteine sind daher nur mit den Nummern 200 bis 255 zu erstellen.

Typ des Bausteinparameters und zugelassener Aktualoperand

Art des Parameters	Typ des Parameters		Zugelassene Aktualoperanden		
E, A	BI	für einen Operanden mit Bit-Byteadresse	E A M	n.m n.m n.m	Eingänge Ausgänge Merker
	BY	für einen Operanden mit Byte-Byteadresse	EB AB MB DL DR PB	n n n n n n	Eingangsbytes Ausgangsbytes Merkerbytes Datenbyte links Datenbyte rechts Peripheriebytes
	W	für einen Operanden mit Wortadresse	EW AW MW DW PW BS* BT* BA BB	n n n n n n n n n	Eingangswörter Ausgangswörter Merkerwörter Datenwörter Peripheriewörter Bereich Sytemdaten Systemdatenerweiterung Bereich Anschaltungsdatum Anschaltungsdatenerweiterung
	D	für einen Operanden mit Doppelwortadresse	ED AD MD DD	n n n n	Eingangs-Doppelwörter Ausgangs-Doppelwörter Merker-Doppelwörter Datum-Doppelwörter
D	KM	für ein Binärmuster (16 Stellen)	Konstanten		
	KY	für zwei byteweise Betragszahlen im Bereich jeweils von 0 bis 255			
	KH	für ein Hexadezimalmuster bis 4 Stellen			
	KC	für ein Zeichen (max. 2 alphanumerische Zeichen)			
	KT	für einen Zeitwert (BCD-codiert) mit Zeitraster 1.0 bis 999.3			
	KZ	für einen Zählwert (BCD-codiert) 0 bis 999			
	KF	für eine Festpunktzahl — 32768 bis + 32767			
	KG	für eine Gleitkommazahl			
B	keine Typangabe zulässig		DB FB PB SB	n n n n	Datenbausteine, ausgeführt wird der Befehl ADBn. Funktionsbausteine (nur ohne Parameter zulässig) werden unbedingt (SPA ..n) aufgerufen Programmbausteine werden unbedingt (SPA ..n) aufgerufen Schrittbausteine werden unbedingt (SPA ..n) aufgerufen
T	keine Typangabe zulässig		T		Zeit; der Zeitwert ist als Datum zu parametrieren oder als Konstante im Funktionsbaustein zu programmieren.
Z	keine Typangabe zulässig		Z		Zähler; der Zählwert ist als Datum zu parametrieren oder als Konstante im Funktionsbaustein zu programmieren.

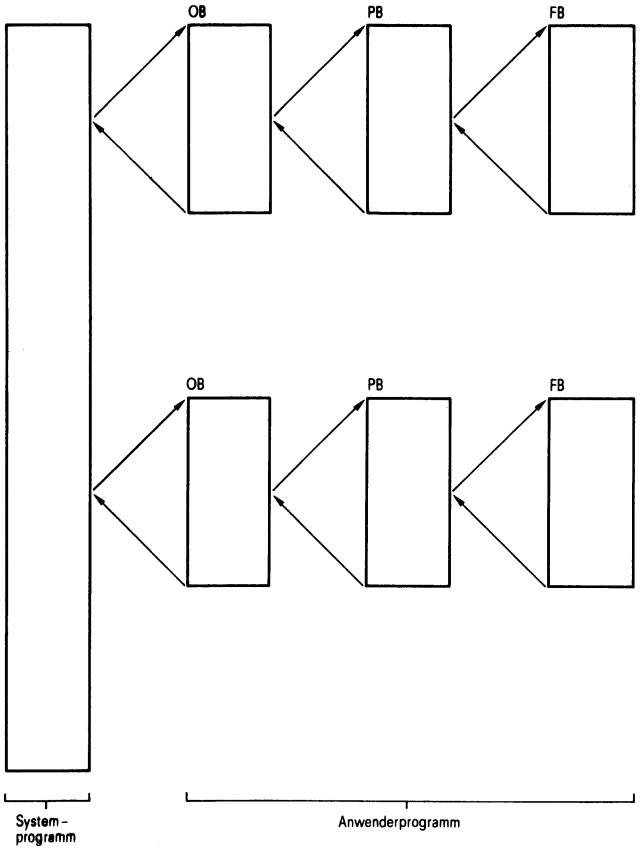
5.1 Allgemeines

Die Schnittstelle zwischen dem Systemprogramm und dem Anwenderprogramm sind die Organisationsbausteine.
Die Organisationsbausteine (OB) sind Teile des Anwenderprogramms genauso wie Programmbausteine, Schrittbausteine oder Funktionsbausteine. Ein Organisationsbaustein wird jedoch nur vom Systemprogramm aufgerufen. Als Anwender kann man einen Organisationsbaustein nicht aufrufen. Man kann ihn jedoch programmieren und somit indirekt Einfluß auf das Systemprogramm nehmen (Bild 12).

Bei entsprechender Programmierung der Organisationsbausteine können folgende Betriebsarten eingestellt werden:

Bearbeitungsmodus des Anwenderprogramms
- zyklische Bearbeitung (siehe ,,Programmierung der zyklischen Bearbeitung" Seite 12)
- alarmgesteuerte Bearbeitung (siehe ,,Programmierung der alarmgesteuerten Bearbeitung" Seite 14)
- zeitgesteuerte Bearbeitung (siehe ,,Programmierung der zeitgesteuerten Bearbeitung" Seite 18)

Neustart- und Anlaufbetrieb
- Neustart manuell (siehe ,,Manueller Neustart" Seite 21)
- Wiederanlauf manuell (siehe ,,Manueller Wiederanlauf" Seite 21)
- Wiederanlauf automatisch (siehe ,,Automatischer Wiederanlauf" Seite 21)



OB Organisationsbaustein
PB Programmbaustein
FB Funktionsbaustein

Bild 12
Gesamtprogramm eines Automatisierungsgeräts

Gerätefehlerbehandlung
- Aufruf eines nichtgeladenen Bausteins (Seite 22)
- Quittungsverzug bei Einzelzugriff auf Peripheriebaugruppen (Seite 22)
- Quittungsverzug beim Aktualisieren des Prozeßabbilds (Seite 22)
- Adressierfehler (Seite 22)
- Zykluszeitüberschreitung (Seite 22)
- Substitutionsfehler (Seite 23)
- Quittungsverzug beim Zugriff auf das erweiterte Peripherievolumen (Seite 22)
- Quittungsverzug beim Zugriff auf Externspeicher (Seite 22)
- Parity-Fehler beim Externspeicher (Seite 23)
- Transfer-Fehler im Datenbaustein (Seite 23)
- Weck-Fehler (Seite 23)

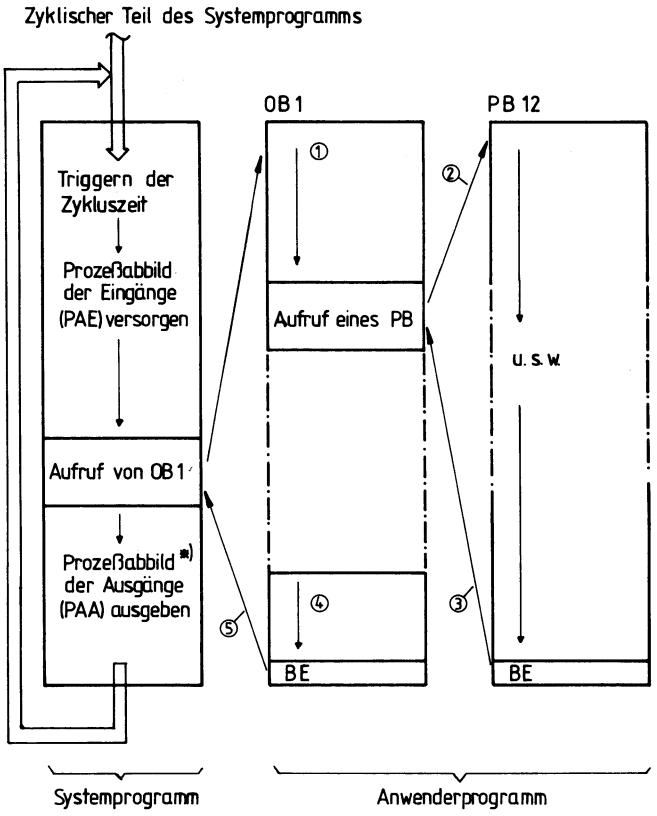
5.2 Übersicht

Absoluter Parameter	Bezeichnung bzw. Bearbeitungsanstoß	Seite
OB für zyklische Bearbeitung		
OB 1	Programmanfang	12
OB für prozeß-alarmgesteuerte Bearbeitung		
	Signalzustandwechsel an	14
OB 2	E 0.0	
OB 3	E 0.1	
OB 4	E 0.2	
OB 5	E 0.3	
OB 6	E 0.4	
OB 7	E 0.5	
OB 8	E 0.6	
OB 9	E 0.7	
OB für anforderungs-alarmgesteuerte Bearbeitung		
	ereignisgesteuert von	15
OB 35	BS 0.8	
OB 36	BS 0.9	
OB 37	BS 0.10	
OB 38	BS 0.11	
OB 39	BS 0.12	
OB für zeitgesteuerte Bearbeitung		
	Zeitraster mit	18
OB 10	0.01 s	
OB 11	0.02 s	
OB 12	0.05 s	
OB 13	0.1 s	
OB 14	0.2 s	
OB 15	0.5 s	
OB 16	1.0 s	
OB 17	2.0 s	
OB 18	5.0 s	
OB für Neustart und Anlaufbetrieb		
OB 20	Neustart manuell	21
OB 21	Wiederanlauf manuell	
OB 22	Wiederanlauf automatisch nach Netzspannungsausfall	
OB 31	Einstellung der Zykluszeit	

5.3 Programmierung der zyklischen Bearbeitung

OB für Gerätefehlerbehandlung

Absoluter Parameter	Bezeichnung bzw. Bearbeitungsanstoß	Reaktion bei nicht programmiertem OB	Seite
OB 19	Aufruf eines nicht geladenen Bausteins	keine	22
OB 23	Quittungsverzug bei Einzelzugriff auf Peripheriebaugruppen	keine	22
OB 24	Quittungsverzug beim Aktualisieren des Prozeßabbildes	keine	22
OB 25	Adressierfehler	Stopp	22
OB 26	Zykluszeitüberschreitung	Stopp	22
OB 27	Substitutionsfehler	Stopp	23
OB 28	Quittungsverzug bei Eingangsbyte 0	Stopp	22
OB 29	Quittungsverzug bei dezentraler Peripherie, erweitertes Adressiervolumen	keine	22
OB 30	Quittungsverzug beim Externspeicher/Parity-Fehler beim Externspeicher	keine	22/23
OB 32	Transfer-Fehler im Datenbaustein/Transfer auf ein Datenwort obwohl noch kein Baustein eröffnet wurde	Stopp	23
OB 33	Weck-Fehler	Stopp	23



① Erste Anweisung des STEP-5-Anwenderprogramms.
② Erster Aufruf eines Programmbausteins. In diesem Baustein können auch weitere Aufrufe stehen (siehe ,,Programmorganisation" Seite 3).
③ Rücksprung vom letzten bearbeiteten Baustein in OB 1.
④ Der Organisationsbaustein wird mit BE abgeschlossen.
⑤ Rücksprung ins Systemprogramm.
* Falls der rechte Zyklus mit Wiederanlauf (siehe Seite 21 fortgesetzt wurde, werden an dieser Stelle PAE und PAA gelöscht.

Bild 13
Zyklische Programmbearbeitung

5.3 Programmierung der zyklischen Bearbeitung

Die zyklische Bearbeitung ist die ,,normale" Bearbeitung bei speicherprogrammierbaren Steuerungen (Bild 13). Der Prozessor beginnt mit der Programmbearbeitung am Anfang des STEP-5-Programms, arbeitet die STEP-5-Anweisungen der Reihe nach bis zum Ende des Programms ab und beginnt dann wieder mit der Bearbeitung am Programmanfang.

Der Organisationsbaustein 1 ist die Schnittstelle zwischen dem Systemprogramm und der zyklischen Bearbeitung des Anwenderprogramms. Die erste STEP-5-Anweisung im Organisationsbaustein 1 ist gleichzeitig die erste Anweisung des Anwenderprogramms, also gleichbedeutend mit dem Programmanfang.

Im Organisationsbaustein 1 werden die Programm-, Schritt- und Funktionsbausteine des zyklischen Programms aufgerufen. In diesen aufgerufenen Bausteinen können wieder Bausteinaufrufe stehen, d. h. die Bausteine können geschachtelt werden (siehe ,,Programmorganisation" Seite 3).

Die Laufzeit des Anwenderprogrammes ergibt sich aus der Summe aller Laufzeiten der aufgerufenen Bausteine. Wird ein Baustein ,,n" mal aufgerufen, so muß seine Laufzeit ,,n" mal bei der Summenbildung berücksichtigt werden.

Grobgliederung des Programms

Im Organisationsbaustein OB 1 steht eine Grobgliederung des Anwenderprogramms. Die Dokumentation dieses Bausteins soll auf den ersten Blick die wesentlichen Programmstrukturen zeigen (Bild 14) bzw. programmtechnisch zusammenhängende Anlagenteile hervorheben (Bild 15).

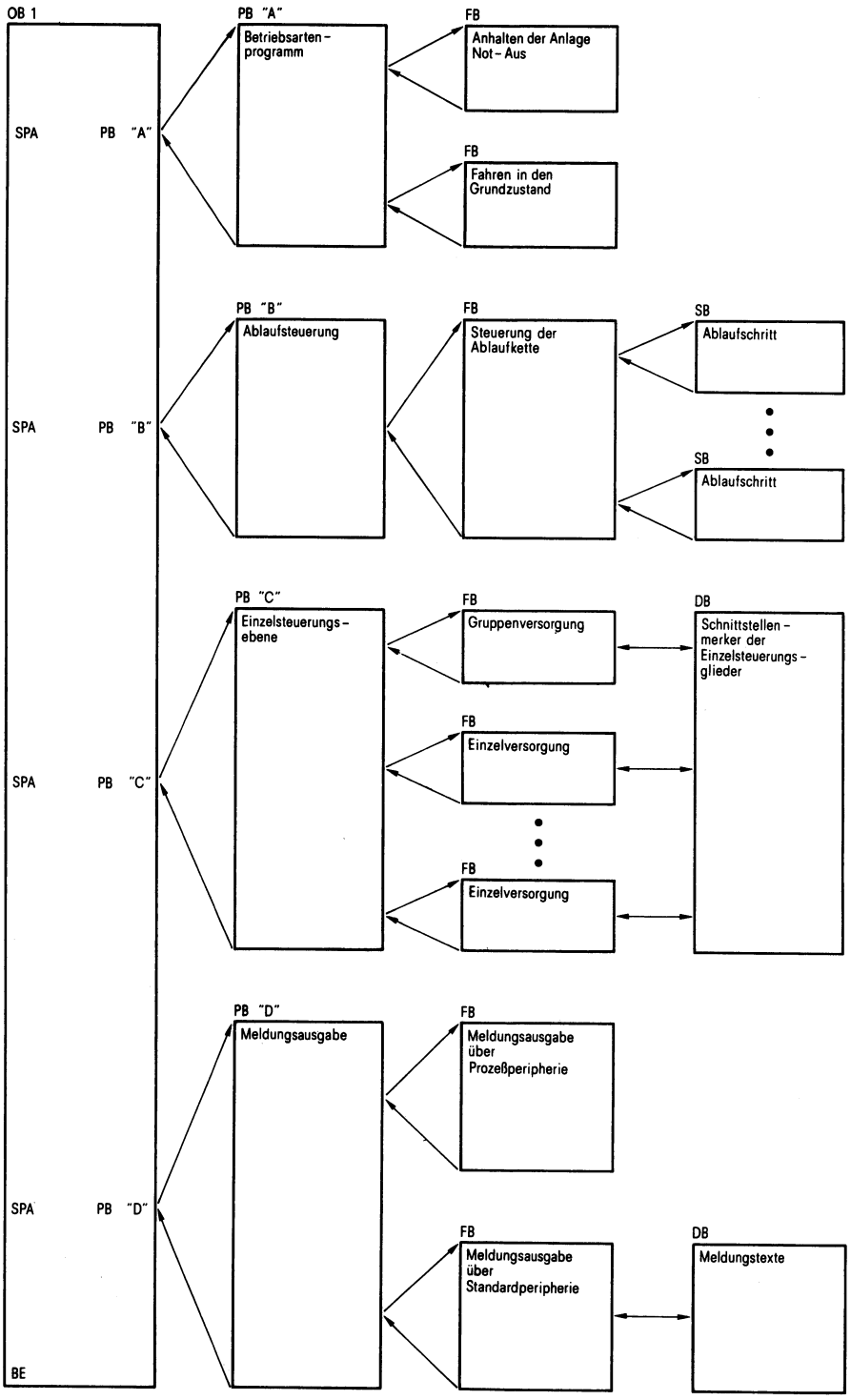


Bild 14
Gliederung des Anwenderprogramms nach Programmstruktur

5.4 Programmierung der alarmgesteuerten Bearbeitung

Mit den Automatisierungsgeräten S5-150S kann eine ,,alarmgesteuerte'' Bearbeitung durchgeführt werden. Bei dieser Betriebsart wird die zyklische Programmbearbeitung unterbrochen und ein spezifisches Programm bearbeitet. Nach der Bearbeitung dieses Programms kehrt der Prozessor zur Unterbrechungsstelle zurück und setzt dort die zyklische Bearbeitung fort (Bilder 17 und 18).

Die alarmgesteuerte Bearbeitung wird auf zwei Arten angestoßen:

1 Signalzustandswechsel eines Bits am Eingabebyte 0 führt zu einem Prozeßalarm (flankengesteuert).
2 Signalzustand ,,1'' eines oder mehrerer Bits 8 bis 12 im Systemdatum (BS) 0 führt zu einem Anforderungsalarm (ereignisgesteuert).

Die Prozeßalarmbearbeitung ermöglicht dem Anwender die unmittelbare Reaktion auf Prozeßsignale, die am Eingangsbyte 0 angeschlossen sind. Damit wird ein Flankenwechsel dieser Signale registriert, bevor das Prozeßabbild aktualisiert wird. Diese Eigenschaft verringert die Reaktionszeit auf zeitkritische Vorgänge des zu steuernden oder zu regelnden Prozesses. Für die Anforderungsalarmbearbeitung kommt der Anstoß — im Gegensatz zu Prozeßalarmen — vom Anwender, der bei geeignetem Ausbaugrad über periphere Geräte und Anschaltungen die Bearbeitung eines bestimmten Programmabschnittes initialisieren kann.

Schnittstelle zwischen Systemprogramm und alarmgesteuerter Bearbeitung

Die Organisationsbausteine 2 bis 9 und 35 bis 39 sind die Schnittstellen zwischen Systemprogramm und alarmgesteuerter Bearbeitung. Die erste Gruppe ist der Prozeßalarm- und die zweite Gruppe der Anforderungsalarmbearbeitung zugeordnet.

Prozeßalarmbearbeitung

Organisationsbaustein	Eingang
OB 2	E 0.0
OB 3	E 0.1
OB 4	E 0.2
OB 5	E 0.3
OB 6	E 0.4
OB 7	E 0.5
OB 8	E 0.6
OB 9	E 0.7

Der Signalzustandswechsel eines Bits von ,,0'' nach ,,1'' (positive Flanke) oder von ,,1'' nach ,,0'' (negative Flanke) am Eingangsbyte 0 veranlaßt die Prozeßalarmbearbeitung; d. h., das Systemprogramm ruft den dem jeweiligen Eingangsbit zugeordneten Organisationsbaustein auf, womit das vom Anwender in diesem Baustein hinterlegte Programm abgearbeitet wird.

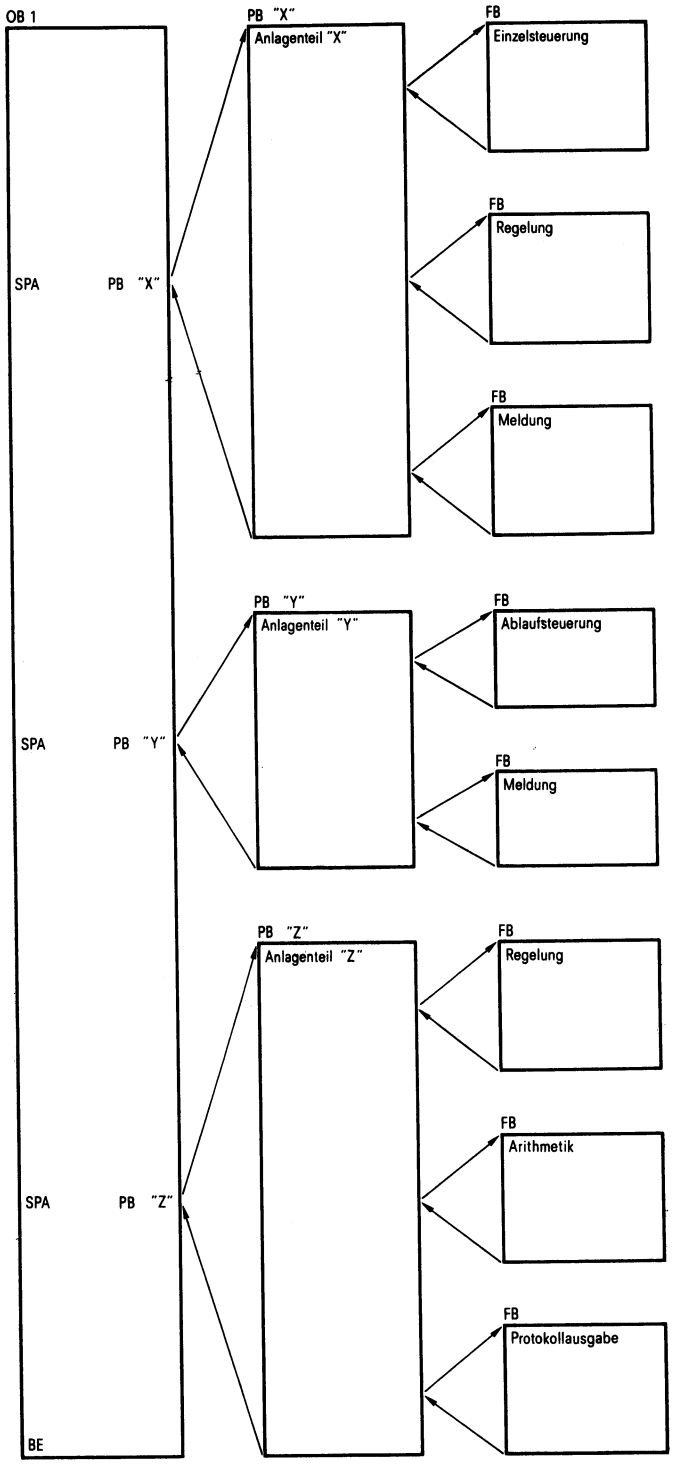


Bild 15
Gliederung des Anwenderprogramms nach Anlagenstruktur

Anforderungsalarmbearbeitung:

Der Signalzustand ,,1" eines Bits (Bit 8, 9, 10, 11, 12) im Systemdatum (BS) 0 veranlaßt das Systemprogramm ebenfalls zum Aufrufen des zugehörigen Organisationsbausteins. Das für diesen Fall vom Anwender hinterlegte Programm wird somit abgearbeitet.

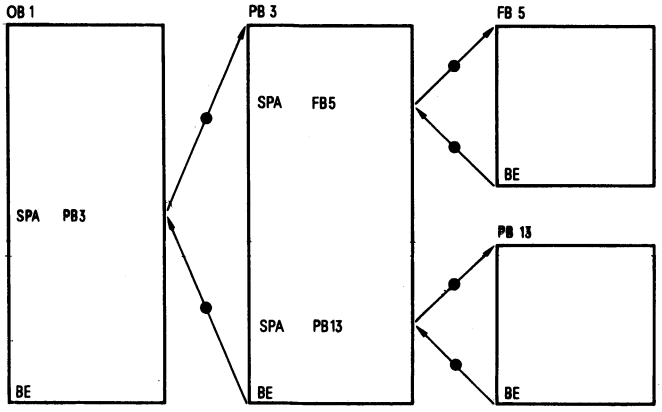
Organisationsbaustein	Sytemdatum (BS) Ø
OB 35	Bit 8
OB 36	Bit 9
OB 37	Bit 10
OB 38	Bit 11
OB 39	Bit 12

Unterbrechungsstellen

Das zyklisch bearbeitete Programm kann nicht an jeder beliebigen Stelle durch eine alarmgesteuerte Bearbeitung unterbrochen werden. Dies ist nur an den Bausteingrenzen möglich (Bild 16). Nur dann, wenn von einem Baustein auf einen anderen gewechselt wird — sei es durch den Aufruf eines neuen Bausteins oder durch die Rückkehr zum übergeordneten Baustein nach einer Bausteinende-Anweisung — kann das Systemprogramm einen Organisationsbaustein für die alarmgesteuerte Bearbeitung aufrufen.

Werden bei Bearbeitung eines Organisationsbausteins für alarmgesteuerte Bearbeitung weitere Bausteine aufgerufen, so ist das Programm an diesen Bausteingrenzen von einer zeitgesteuerten Bearbeitung (siehe ,,Programmierung der zeitgesteuerten Bearbeitung" Seite 17) unterbrechbar (siehe ,,Zusammenfassung der Alarmbearbeitung" Seite 20).

Das Programm kann auch bei allen Fehlern gemäß Seite 21 unterbrochen werden.



Unterbrechungsstellen, an denen eine alarmgesteuerte Bearbeitung ,,eingefügt" werden kann.

Bild 16
Unterbrechungsstellen im zyklisch arbeitenden Programm

Prioritierung der Alarme

Bei gleichzeitiger Ansteuerung von Prozeß- und Anforderungsalarmen wird zuerst der anstehende Prozeßalarm bearbeitet. Danach wird (falls keine Prozeßalarme mehr anstehen) der Anforderungsalarm bearbeitet. Der Prozeßalarm hat somit gegenüber dem Anforderungsalarm die höhere Priorität.
Prozeßalarme und Anforderungsalarme sind jeweils auch in sich prioritiert. Die Reihenfolge der Bearbeitung bei gleichzeitig anstehenden mehreren Prozeßalarmen gibt das untere Übersichtsbild:

E 0.0
E 0.1
E 0.2
E 0.3
E 0.4
E 0.5
E 0.6
E 0.7

fallende Priorität

Erkennt der Prozessor dagegen mehrere Anforderungsalarme gleichzeitig, so wird vom Systemprogramm die folgende Bearbeitungsreihenfolge vorgenommen:

SD 0 Bit 8
SD 0 Bit 9
SD 0 Bit 10
SD 0 Bit 11
SD 0 Bit 12

fallende Priorität

Wenn das zyklische Programm durch einen Alarm unterbrochen wird, werden alle anstehenden Alarme bearbeitet, bevor die zyklische Programmbearbeitung fortgesetzt wird. Dies gilt sowohl für die Alarme, die zur Unterbrechung des zyklischen Betriebs geführt haben, als auch für alle, die während einer Alarmbearbeitung entstehen.

Dabei wird nach Beendigung eines jeden Alarmbearbeitungsprogramms eine neue Prioritierung der noch anstehenden Alarme nach den oben angegebenen Regeln vorgenommen, um den nächsten höchstprioren Alarm zu ermitteln und dessen Bearbeitung einzuleiten.

Reaktionszeit

Während der Bearbeitung eines Bausteins kann keine alarmgesteuerte Bearbeitung stattfinden. Ein auftretender Alarm wird erst bei einem Bausteinwechsel bearbeitet, also wenn ein Baustein aufgerufen oder beendet wird. Die maximale Reaktionszeit zwischen dem Auftreten eines Alarms und seiner Bearbeitung entspricht daher der Bearbeitungszeit eines Bausteins. Treten zwei Alarme gleichzeitig auf, vergrößert sich die Reaktionszeit für den Alarm mit der niedrigeren Priorität. Zuerst wird das zyklische Programm bis zum nächsten Bausteinwechsel bearbeitet, und danach zieht der Prozessor die Bearbeitung des höher prioren ,,Alarmprogramms" vor. Ist das höher priore ,,Alarmprogramm" beendet, wird mit der Bearbeitung des niedriger prioren ,,Alarmprogramms" begonnen. Die Reaktionszeit dieses niedriger prioren Programms ist somit um die Bearbeitungszeit des höher prioren Programms verlängert worden.

Wenn mehrere Alarme gleichzeitig auftreten, dann wird die Bearbeitung des Alarms mit der niedrigsten Priorität erst dann aufgenommen, wenn alle höher prioren Alarme bearbeitet sind.

Die Prioritäten einer alarmgesteuerten Bearbeitung können sich verschieben, wenn während der Bearbeitung eines alarmgesteuerten Programms erneut ein Alarm auftritt. Nach Abschluß der Bearbeitung des höher prioren Alarms wird neu prioritiert, so daß sich die Reaktionszeit für den am niedrigsten prioritierten Alarm weiter vergrößern kann.

Legende zu den Bildern 17 und 18

① Beginn der zyklischen Bearbeitung. Das Systemprogramm ruft den Organisationsbaustein OB 1 auf.

② Durch einen Signalwechsel von ,,0'' auf ,,1'' am Eingang 0.1 tritt ein Prozeßalarm auf.

③ Bausteinwechsel. Der Signalzustandswechsel am Eingang 0.1 wird registriert. Die Bearbeitung des zyklischen Programms wird unterbrochen.

④ Das Systemprogramm ruft OB 3 auf, der dem Eingang 0.1 zugeordnet ist. OB 3 wird bearbeitet.

⑤ Da kein weiterer Alarm vorliegt, wird die Bearbeitung des zyklischen Programms an der unterbrochenen Stelle fortgesetzt.

⑥ Bit 10 im BA 0 steht auf Signalzustand ,,1'', d. h., ein Anforderungsalarm tritt auf.

⑦ Bausteinwechsel: Der Anforderungsalarm wird registriert. Die Bearbeitung des zyklischen Programms wird unterbrochen.

⑧ Das Systemprogramm ruft OB 37 auf, der dem Bit 10 im BA 0 zugeordnet ist. OB 37 wird bearbeitet.

⑨ Da kein weiterer Alarm vorliegt, wird die Bearbeitung des zyklischen Programms an der unterbrochenen Stelle fortgesetzt.

⑩ Bit 8 in BA 0 steht auf Signalzustand ,,1'', d. h. ein Anforderungsalarm tritt auf.

⑪ Durch den Signalzustandswechsel am Eingang 0.1 von ,,1'' auf ,,0'' tritt ein Prozeßalarm auf.

⑫ Bit 11 in BA 0 steht auf Signalzustand ,,1'', d. h. ein weiterer Anforderungsalarm tritt auf.

⑬ Bausteinwechsel: Anstehende Alarme werden registriert. Die Bearbeitung des zyklischen Programms wird unterbrochen.

⑭ Da ein Prozeßalarm gegenüber einem Anforderungsalarm die höhere Bearbeitungspriorität hat, wird zunächst der Prozeßalarm bearbeitet. Das Systemprogramm ruft den dem Eingang 0.1 zugeordneten Organisationsbaustein OB 3 auf. OB 3 wird bearbeitet. Obwohl während der Bearbeitung des OB 3 Bausteinwechsel auftreten, werden ,,alte'' (bereits bestehende) Anforderungsalarme nicht beachtet.

⑮ Während der Bearbeitung des Prozeßalarms tritt durch den Signalbestandswechsel von ,,1'' auf ,,0'' am Eingang 0.2 ein Prozeßalarm auf.

⑯ Da noch Alarme vorliegen, wird das zyklische Programm nicht fortgesetzt. Die anstehenden Alarme werden aufs neue prioritiert. Die Abarbeitung der Anforderungsalarme wird weiterhin zurückgestellt, da in der Zwischenzeit ein neuer Prozeßalarm aufgetreten ist.

⑰ Das Systemprogramm ruft den dem Eingang 0.2 zugeordneten Organisationsbaustein OB 4 auf. OB 4 wird zu Ende bearbeitet.

⑱ Keine Rückkehr in das zyklische Programm, da noch zwei Alarme anstehen. Da kein weiterer Prozeßalarm mehr vorliegt, kann die Bearbeitung der Anforderungsalarme gestartet werden. Weil hier der Alarm, der mit Bit 8 im BA 0 ausgelöst wird, die höhere Priorität besitzt, wird der zugehörige Organisationsbaustein OB 35 vom Systemprogramm aufgerufen und zu Ende bearbeitet.

⑲ Der Rücksprung in das zyklische Programm findet noch nicht statt, da noch ein Anforderungsalarm ansteht. Der zugehörige Organisationsbaustein OB 38 wird aufgerufen.

⑳ Nach vollständiger Bearbeitung des OB 38 wird die zyklische Bearbeitung an der unterbrochenen Stelle fortgesetzt, weil keine weiteren Alarme mehr anstehen.

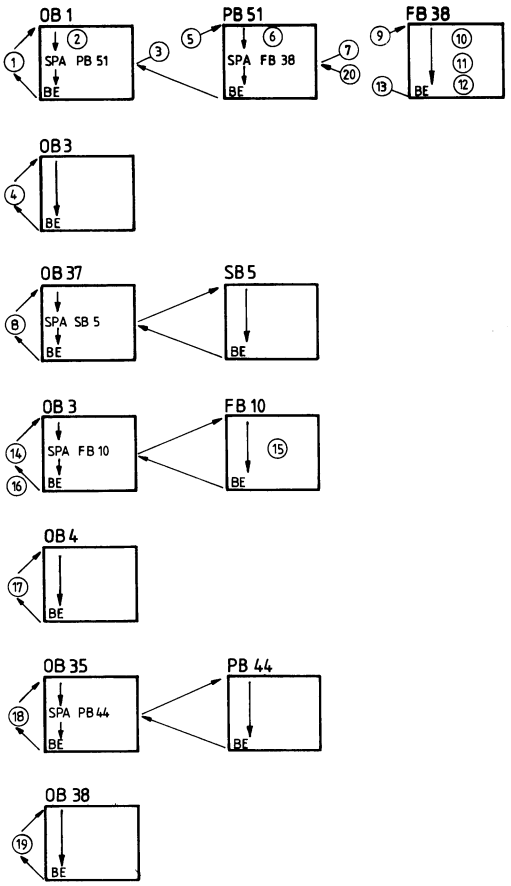


Bild 17
Alarmgesteuerte Bearbeitung bei Auftritt mehrerer Alarme

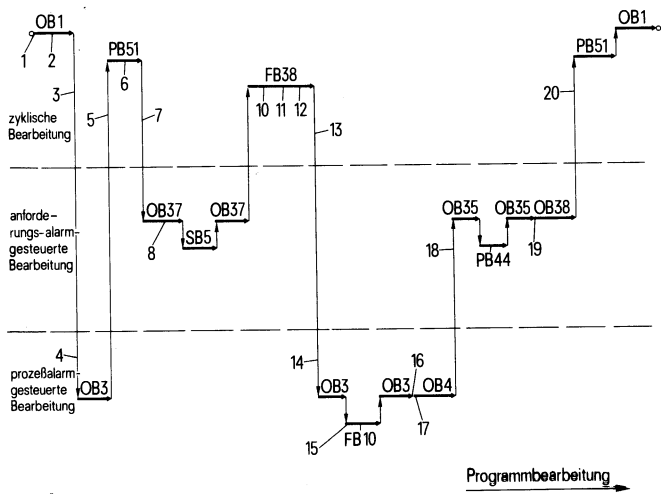


Bild 18
Darstellung der Bausteine des vorhergehenden Beispiels in der Reihenfolge ihrer Bearbeitung in einem Diagramm

Sperren der alarmgesteuerten Bearbeitung

Ein alarmgesteuertes Programm wird an einer Bausteingrenze in das zyklische Programm ,,eingeschoben". An dieser Stelle wird das zyklische Programm unterbrochen. Diese Unterbrechung kann sich negativ auswirken, wenn ein zyklischer Programmteil in einer bestimmten Zeit bearbeitet werden muß, um z. B. eine bestimmte Reaktionszeit zu erreichen.

Wenn ein Programmteil durch eine alarmgesteuerte Bearbeitung nicht unterbrochen werden darf, kommen folgende Programmiermöglichkeiten in Frage:
Das Programm enthält keinen Bausteinwechsel. Dadurch kann es auch nicht unterbrochen werden.
Das Programm steht selbst in einem alarmgesteuerten Programm. Hier kann es auch bei einem Bausteinwechsel nicht von einem weiteren Alarm unterbrochen werden.
Man programmiert die Operation ,,Prozeßalarme sperren" AS oder ,,Anforderungsalarme sperren" ASS und hebt die sperrende Wirkung mit der Operation ,,Prozeßalarme freigeben" AF oder ,,Anforderungsalarme freigeben" AAF wieder auf (nur in Funktionsbausteinen möglich). Zwischen den Operationen AS und AF wird keine prozeßalarmgesteuerte Programmbearbeitung, zwischen den Operationen AAS und AAF keine anforderungsalarmgesteuerte Programmbearbeitung durchgeführt.

Starten der alarmgesteuerten Bearbeitung

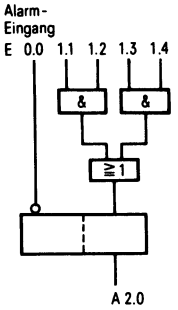
Die Prozeßalarmsignale werden über eine normale Eingabebaugruppe eingegeben. Dabei muß die Codierung der Eingangsadressen mit Byte ,,0" beginnen. Die Eingänge E0.0 bis E0.7 lösen die Alarmbearbeitung aus. Die weiteren auf der Baugruppe vorhandenen Eingänge können als ,,normale" Eingänge verwendet werden.

Beispiel für eine Alarmbearbeitung

Genaues Positionieren mit einem Endschalter

Funktionsbeschreibung:
Ein Ausgang wird über eine Verriegelungsbedingung eingeschaltet und soll nach Ansprechen eines Endschalters mit möglichst kurzer und gleichbleibender Verzögerungszeit wieder ausgeschaltet werden (Bild 19).

Programmierung:
Die Setzbedingung für den Ausgang ist in dem Programmbaustein 2 programmiert.
Der Rücksetzeingang ist auf den Alarmeingang E0.0 gelegt, so daß der Organisationsbaustein 2 sowohl bei einem Wechsel des Eingangs E0.0 von ,,0" nach ,,1" als auch von ,,1" nach ,,0" bearbeitet wird. OB 2 ist jedoch so programmiert, daß nur bei einem Flankenwechsel von ,,1" nach ,,0" am Eingang E0.0 der Ausgang ausgeschaltet wird:
Um den aktuellen Zustand des Eingangs E0.0 abfragen zu können, wird zunächst das Prozeßabbild des Eingangsbytes ,,0" mit den Operationen L PB und T EB aktualisiert. Bei einer fallenden Flanke am Alarmeingang E0.0 wird das Prozeßabbild des Ausgangs A 2.0 zurückgesetzt und mit den Operationen L AB und T PB direkt zur Ausgangsbaugruppe transferiert. Beim Transferieren zu den Peripheriebytes 0 bis 127 wird das Ausgangsprozeßabbild automatisch mit nachgeführt. Der Organisationsbaustein muß mit der Anweisung BE abgeschlossen werden.




```
OB2                 PB2
:L    PBO           :U   E 1.1
:T    EBO           :U   E 1.2
:UN   E 0.0         :O
:SPB  PB2           :U   E 1.3
:L    AB2           :U   E 1.4
:T    AB2           :S   A 2.0
:BE                 :UN  E 0.0
                    :R   A 2.0
                    :BE
```

Bild 19
Beispiel für eine Alarmbearbeitung

5.5 Programmierung der zeitgesteuerten Bearbeitung (Bearbeitung eines Zeitalarms)

Der Prozessor des Automatisierungsgeräts S5-150S führt auch eine zeitgesteuerte Bearbeitung durch. Eine zeitgesteuerte Bearbeitung liegt vor, wenn ein von einer „inneren Uhr" kommendes Signal den Prozessor im Automatisierungsgerät veranlaßt, die „normale" zyklische Bearbeitung zu unterbrechen und ein spezifisches Programm zu bearbeiten.

Nach der Bearbeitung dieses Programms kehrt der Prozessor zur Unterbrechungsstelle im zyklischen Programm zurück und setzt dort seine Bearbeitung fort (Bilder 20 und 21).

Schnittstelle zwischen Systemprogramm und zeitgesteuerter Bearbeitung

Die Organisationsbausteine 10 bis 18 sind die Schnittstelle zwischen Systemprogramm und zeitgesteuerter Bearbeitung. Jeder dieser Organisationsbausteine wird in einem bestimmten Zeitraster vom Systemprogramm aufgerufen.

Einstellen über Brücken auf der CPU 926

Organisationsbaustein	Zeitraster	Weckimpulse (ms) 10	100
OB 10	0,01 s	X	—
OB 11	0,02 s	X	—
OB 12	0,05 s	X	—
OB 13	0,1 s	X	X
OB 14	0,2 s	X	X
OB 15	0,5 s	X	X
OB 16	1 s	X	X
OB 17	2 s	X	X
OB 18	5 s	X	X

A – B	B – C	
ein	offen	⇒ 100ms
offen	ein	⇒ 10 ms
offen	offen	⇒ kein Weckalarm

Unterbrechungsstellen

Das zyklisch bearbeitete Programm kann nur an den Bausteingrenzen durch eine zeitgesteuerte Bearbeitung unterbrochen werden. Das Programm kann auch bei Fehlern gemäß Seite 21 unterbrochen werden. In diesen Fällen erfolgt die Unterbrechung zu beliebigen Zeitpunkten.

Ist die Bearbeitungszeit eines zeitgesteuerten Programms größer als das kleinste Zeitraster (10 oder 100 ms), so daß sich dieses zeitgesteuerte Programm selbst unterbrechen müßte, erkennt das Systemprogramm einen „kritischen" Zustand und ruft den Anwenderorganisationsbaustein OB 33 auf. In diesem Baustein kann der Anwender die erwünschte Reaktion für diesen Betriebsfall vornehmen. Nach der Bearbeitung des OB 33 wird die Bearbeitung des zeitgesteuerten Programms an der unterbrochenen Stelle fortgesetzt. Ist OB 33 vom Anwender nicht programmiert, löst das Systemprogramm den Stopp-Zustand im Automatisierungsgerät aus.

Treten während einer zeitgesteuerten Bearbeitung Bausteinwechsel auf, so kann an diesen Stellen das Systemprogramm eine alarmgesteuerte Bearbeitung einschieben. Ist dies nicht erwünscht, muß man während der zeitgesteuerten Bearbeitung die alarmgesteuerte Bearbeitung sperren (siehe „Sperren der alarmgesteuerten Bearbeitung" Seite 17).

Wenn der Anstoß für eine prozeß-/anforderungsalarmgesteuerte und eine zeitgesteuerte Programmbearbeitung gleichzeitig registriert wird, so wird zuerst das prozeß-/anforderungsalarmgesteuerte Programm bearbeitet. Die zeitgesteuerte Programmbearbeitung hat somit in diesem Fall die niedrigere Priorität.

Legende zu den Bildern 20 und 21

① Beginn der zyklischen Bearbeitung. Das Systemprogramm ruft den Organisationsbaustein OB 1 auf.

② Auftreten eines Zeitimpulses der „inneren Uhr".

③ Bausteinwechsel: Die Anforderung für das zeitgesteuerte Programm wird registriert. Die zyklische Programmbearbeitung wird unterbrochen.

④ Systemprogramm ruft den Anwenderorganisationsbaustein OB 10 auf, der im 10 ms-Raster bearbeitet werden soll.

⑤ Nach der Bearbeitung des zeitgesteuerten Programms wird die zyklische Programmbearbeitung fortgesetzt.

⑥ Auftreten eines Zeitimpulses der „inneren Uhr".

⑦ Auftreten eines Anforderungsalarms: Signalzustand „1" des Bit 10 im Systemdatum BS 0.

⑧ Bausteinwechsel: Die Anforderung für die zeitgesteuerte und alarmgesteuerte Programmbearbeitung wird registriert. Die zyklische Programmbearbeitung wird unterbrochen.

⑨ Da die alarmgesteuerte Bearbeitung des Anwenderprogramms gegenüber der zeitgesteuerten Bearbeitung höher prior ist, wird zuerst die alarmgesteuerte Bearbeitung durchgeführt. Das Systemprogramm ruft den Organisationsbaustein OB 37 auf.

⑩ Nach der vollständigen Bearbeitung des Anforderungsalarms wird nicht die zyklische Programmbearbeitung aufgenommen, da noch eine zeitgesteuerte Programmbearbeitung ansteht. Das Systemprogramm ruft den Organisationsbaustein OB 10 auf, der im 10 ms-Raster bearbeitet wird.

⑪ Im Rahmen der zeitgesteuerten Programmbearbeitung werden zusätzlich alle zeitgesteuerten Anwender-Organisationsbausteine wie folgt aufgerufen:

OB 11 nach jedem 2.	Eintreffen des Zeitimpulses der „inneren Uhr"
OB 12 nach jedem 5.	
OB 13 nach jedem 10.	
OB 14 nach jedem 20.	
OB 15 nach jedem 50.	
OB 16 nach jedem 100.	
OB 17 nach jedem 200.	
OB 18 nach jedem 500.	

In diesem Beispiel wird zur Veranschaulichung dieser Tatsache angenommen, daß das Systemprogramm noch den Organisationsbaustein OB 11 aufruft (20 ms-Raster).

⑫ Nach der vollständigen Bearbeitung des zeitgesteuerten Programms wird die zyklische Programmbearbeitung aufgenommen, weil kein Alarm mehr ansteht.

⑬ Auftreten eines Prozeßalarms: Signalzustandswechsel von „1" auf „0" am Eingang 0.3.

⑭ Bausteinwechsel: Prozeßalarm wird registriert; die zyklische Programmbearbeitung wird unterbrochen.

⑮ Systemprogramm ruft den dem Eingang 0.3 zugeordneten Organisationsbaustein OB 5 auf.

⑯ Eintreffen eines Zeitimpulses der „inneren Uhr" stellt eine Anforderung für die zeitgesteuerte Programmbearbeitung, während der Bearbeitung des Prozeßalarms.

⑰ Bausteinwechsel: Die Anforderung für die zeitgesteuerte Programmbearbeitung wird registriert. Die prozeßalarmgesteuerte Programmbearbeitung wird unterbrochen.

⑱ Das Systemprogramm ruft den Organisationsbaustein OB 10 auf und initialisiert damit die zeitgesteuerte Programmbearbeitung.

⑲ Nach vollständiger Bearbeitung des zeitgesteuerten Programms wird die unterbrochene prozeßalarmgesteuerte Programmbearbeitung zu Ende geführt.

⑳ Da kein weiterer Alarm mehr anliegt, wird die zyklische Programmbearbeitung an der unterbrochenen Stelle fortgesetzt.

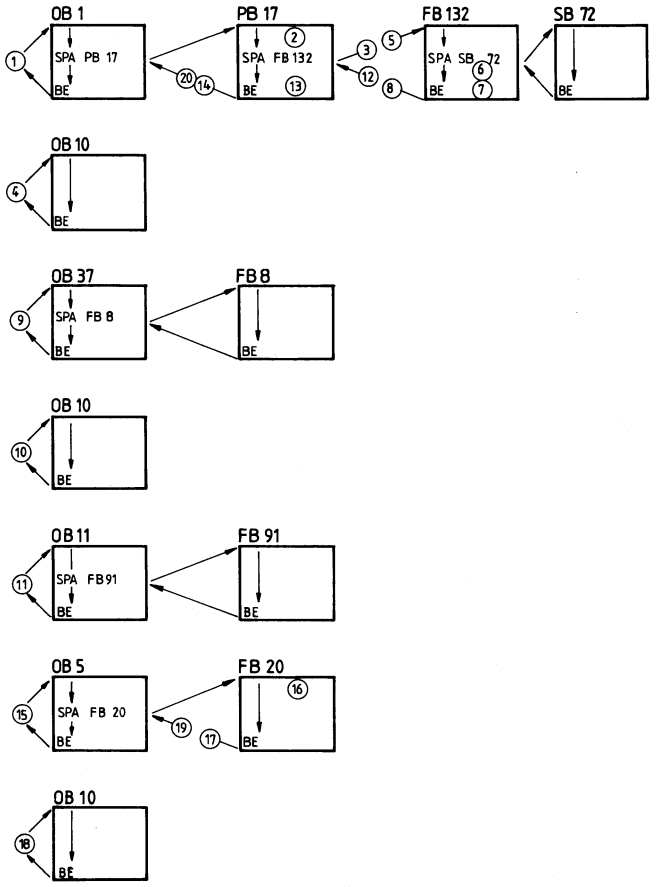


Bild 20
Alarmgesteuerte und zeitgesteuerte Bearbeitung bei Auftritt mehrerer Alarme

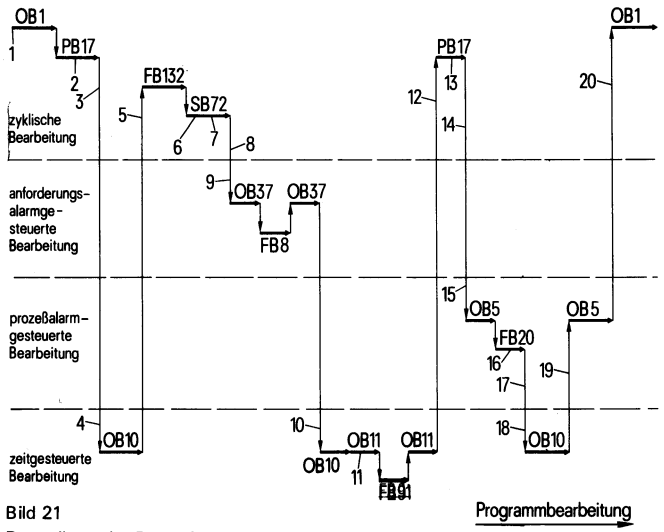


Bild 21
Darstellung der Bausteine des vorhergehenden Beispiels in der Reihenfolge ihrer Bearbeitung in einem Diagramm

Reaktionszeit

Während der Bearbeitung eines Bausteins kann keine zeitgesteuerte Bearbeitung stattfinden. Deshalb erfolgt der Aufruf eines zeitgesteuerten Programms erst dann, wenn ein Baustein aufgerufen oder beendet wird. Die maximale Reaktionszeit zwischen dem Auftreten und der Bearbeitung entspricht daher der Bearbeitungszeit eines Bausteins. Stehen zu diesem Zeitpunkt noch Prozeß- oder Anforderungsalarme an, so wird das zeitgesteuerte Programm erst dann bearbeitet, nachdem alle anstehenden (und während ihrer Bearbeitung neu hinzukommenden) Prozeß- und Anforderungsalarme vollständig abgearbeitet sind. Die maximale Reaktionszeit zwischen dem Auftreten und der Bearbeitung des zeitgesteuerten Programms wächst in diesem Fall um die Bearbeitungszeit der Prozeß- bzw. Anforderungsalarme. Überschreitet jedoch diese Zeit das kleinste Zeitraster (10 oder 100 ms), führt dies zu einem ,,Weckfehler'' (siehe Gerätefehler).

Beispiel:

Anwendung einer zeitgesteuerten Bearbeitung:

Im Programmbaustein PB 73 steht ein Programmteil, an den keine hohen Anforderungen bezüglich Reaktionszeit gestellt werden. Es genügt, wenn die dort programmierten Eingänge und Ausgänge alle 2 Sekunden bearbeitet werden. (Durch diese Programmiermethode kann die durchschnittliche Zykluszeit des Automatisierungsgerätes gesenkt werden).

Programmierung:
OB17 : SPA PB 73
: BE

Der Aufruf des Programmbausteins PB 73 wird in den Organisationsbaustein OB 17 programmiert, der alle 2 Sekunden bearbeitet wird. Dabei kann der Aufruf unbedingt oder auch bedingt abhängig von zuvor programmierten Verknüpfungen sein. Der Organisationsbaustein wird mit der Anweisung BE abgeschlossen.

5.6 Zusammenfassung der Alarmbearbeitung

Auftreten der Alarme	Arbeitsweise des Automatisierungsgerätes
Auftritt eines Prozeß-/Anforderungsalarms bei Bearbeitung des zyklischen Programms	Unterbrechung der Bearbeitung des zyklischen Programms an der nächsten Bausteingrenze, vollständige Bearbeitung des Prozeß-/Anforderungsalarms, Rücksprung in das unterbrochene zyklische Programm.
Auftritt eines Zeitalarms bei Bearbeitung des zyklischen Programms	Unterbrechung der Bearbeitung des zyklischen Programms an der nächsten Bausteingrenze, vollständige Bearbeitung des zeitgesteuerten Programms, Rücksprung in das unterbrochene zyklische Programm.
Gleichzeitiger Auftritt von Zeitalarm und Prozeß-/Anforderungsalarm bei Bearbeitung des zyklischen Programms	Unterbrechung der Bearbeitung des zyklischen Programms an der nächsten Bausteingrenze, vollständige Bearbeitung des prozeß-/anforderungsalarmgesteuerten Programms, vollständige Bearbeitung des zeitgesteuerten Programms, Rücksprung in das unterbrochene Programm der zyklischen Bearbeitung.
Auftritt weiterer Prozeß-/Anforderungsalarme bei Bearbeitung eines Prozeß-/Anforderungsalarmes	Vollständige Bearbeitung des 1. Prozeß-/Anforderungsalarmes, vollständige Bearbeitung der weiteren Prozeß-/Anforderungsalarme, nachdem jeder Bearbeitung eine neue Prioritätsanalyse vorausgeht. (Keine Schachtelung möglich).
Auftritt eines weiteren Zeitalarms bei der Bearbeitung eines zeitgesteuerten Programms	Aufruf des Organisationsbausteins OB33 (wurde OB 33 nicht programmiert, wird die Kennung WECKFEHL im Systemdatum BS7 Bit 11 gesetzt und das Automatisierungsgerät geht in den STOPP-Zustand). Rücksprung in die unterbrochene Zeitalarmbearbeitung.
Auftritt eines Prozeß-/Anforderungsalarms bei Bearbeitung eines Zeitalarms	Unterbrechung der zeitgesteuerten Programmbearbeitung an der nächsten Bausteingrenze, vollständige Abarbeitung des Prozeß-/Anforderungsalarms, Rücksprung in das unterbrochene zeitgesteuerte Programm.
Auftritt eines Zeitalarms bei Bearbeitung eines Prozeß-/Anforderungsalarms	Unterbrechung der prozeß-/anforderungsalarmsgesteuerten Programmbearbeitung an der nächsten Bausteingrenze, vollständige Bearbeitung des zeitgesteuerten Programms, Rücksprung in das unterbrochene Prozeß-/Anforderungsalarmprogramm.
Auftritt eines Prozeß-/Anforderungsalarms und eines weiteren Zeitalarms bei der Bearbeitung des Zeitalarms	Unterbrechung der zeitgesteuerten Bearbeitung an der nächsten Bausteingrenze, vollständige Bearbeitung des Prozeß-/Anforderungsalarms, Aufruf des Organisationsbausteins OB33 (wurde OB33 nicht programmiert wird die Kennung WECKFEHL im Systemdatum BS7 Bit 11 gesetzt und das Automatisierungsgerät geht in den STOPP-Zustand). Rücksprung in die unterbrochene Zeitalarmbearbeitung.
Auftritt eines Zeitalarms und eines weiteren Prozeß-/Anforderungsalarms bei Bearbeitung eines Prozeß-/Anforderungsalarms	Unterbrechung der Bearbeitung des Prozeß-/Anforderungsalarms an der nächsten Bausteingrenze, vollständige Bearbeitung des zeitgesteuerten Programms, Rücksprung in das unterbrochene Prozeß-/Anforderungsalarmprogramm und die vollständige Bearbeitung, Bearbeitung des noch anstehenden Prozeß-/Anforderungsalarms.

5.7 Programmierung des Anlaufverhaltens

Das Systemprogramm unterscheidet drei verschiedene Anlaufarten des Automatisierungsgeräts:

manueller Neustart

manueller Wiederanlauf

automatischer Wiederanlauf

Für jede Anlaufart ruft das Systemprogramm einen Organisationsbaustein auf, den der Anwender für ein bestimmtes Anlaufverhalten programmieren kann. Ist dies nicht erforderlich, kann die Programmierung dieser Organisationsbausteine unterbleiben.

Manueller Neustart

Ein Neustart wird durch Handbedienung ausgelöst, indem der Stopp-Schalter auf der Zentralbaugruppe von der Stellung ,,STOP'' in die Stellung ,,Betrieb'' gebracht wird.

Dabei führt das Systemprogramm folgende Tätigkeiten durch:

Löschen aller aktuellen Zeitwerte

Löschen aller aktuellen Zählwerte

Rücksetzen aller Merker

Laden des Eingangs-Prozeßabbilds mit Generierung der Kontrollspur (für Erfassung aller gesteckten und nicht defekten Eingabebaugruppen)

Löschen des Ausgangs-Prozeßabbilds mit Generierung der Kontrollspur und Rücksetzen aller Ausgänge

Aufbau der Adreßliste (Adressen aller im Anwenderspeicher programmierten Bausteine)

Aufruf des Organisationsbausteins OB 31

Im Organisationsbaustein OB 31 kann der Anwender die Zykluszeit vorgeben, die dann vom Systemprogramm gestartet wird. Die Vorgabe durch den Anwender erfolgt durch die Versorgung des Akku 1 mit einem Wert 0 — 255. Die eingestellte Zykluszeit ist (Akku 1-Inhalt) * 10 ms. Wird dieser Baustein vom Anwender nicht programmiert, so stellt das Systemprogramm die Zykluszeit von 200 ms ein.

Aufruf des Organisationsbausteins OB 20

Im Organisationsbaustein OB 20 kann der Anwender ein Programm hinterlegen, das vor Beginn der zyklischen Programmbearbeitung bestimmte Tätigkeiten ausführt; z. B. Merker setzt, Zeiten startet, Ausgänge setzt und bei geeignetem Ausbaugrad den Datenverkehr des Automatisierungsgeräts mit periphären Geräten vorbereitet. Der Organisationsbaustein muß mit BE abgeschlossen werden.
Nach der Bearbeitung des OB 20 beginnt die zyklische Bearbeitung durch Aufrufen des OB 1.

Manueller Wiederanlauf

Beim Wiederanlauf werden die vor dem ,,Stopp'' des Automatisierungsgeräts erfaßten Ergebnisse und Betriebszustände berücksichtigt.
Das Automatisierungsgerät läuft dann wieder an, wenn der Stopp-Schalter von der Stellung ,,Stopp'' in die Stellung ,,Betrieb'' geschaltet und gleichzeitig die Wiederanlauftaste betätigt wird. Dabei ruft das Systemprogramm zunächst den Organisationsbaustein 21 auf, in den der Anwender ein Programm zum Voreinstellen bestimmter Zustände programmieren kann.

Der OB 21 muß mit BE abgeschlossen werden. Nach der Bearbeitung des OB 21 wird das zyklische Programm an der unterbrochenen Stelle fortgesetzt. Achtung: Zeiten sind während der Stopp-Phase weitergelaufen. Merker, Zeiten, Zähler und Prozeßabbild werden während der Anlaufphase vom Systemprogramm nicht verändert und weiterhin nur vom Anwenderprogramm beeinflußt. Am Ende des festgesetzten Zyklus werden dann die Ausgänge sowie das Ausgangsprozeßabbild gelöscht und die Sperre der Befehlsausgabe aufgehoben, so daß alle Ausgänge Nullsignal ausgeben. Anschließend wird das Prozeßabbild der Eingänge geladen und die zyklische Programmbearbeitung durch Aufruf des OB 1 weiter fortgesetzt.

Automatischer Wiederanlauf

Bei Netzspannungsausfall und anschließender Spannungswiederkehr versucht das Automatisierungsgerät automatisch einen Wiederanlauf durchzuführen. Dabei wird zuerst vom Systemprogramm der Organisationsbaustein OB 22 aufgerufen, in den der Anwender ein Voreinstellen bestimmter Zustände programmieren kann. Die Funktion des automatischen Wiederanlaufs ist mit der des manuellen Wiederanlaufs identisch. Soll das Automatisierungsgerät keinen automatischen Wiederanlauf durchführen, dann muß in den OB 22 die Anweisung ,,Stopp'' programmiert werden.

```
OB 22 : STP (Stopp)
      : BE  (Bausteinende)
```

5.8 Auswertung eines Gerätefehlers

Das Systemprogramm kann fehlerhaftes Arbeiten des Zentralprozessors, Fehler im Systemprogramm oder Auswirkungen einer fehlerhaften Programmierung durch den Anwender feststellen. Bei einigen dieser Fehler arbeitet der Zentralprozessor nicht mehr einwandfrei. Das Automatisierungsgerät geht dann in den Stopp-Zustand. Beeinträchtigt jedoch dieser Fehler die Arbeitsweise des Zentralprozessors nicht, so ermöglicht das Systemprogramm dem Anwender durch Aufruf entsprechender Organisationsbausteine das weitere Verhalten des Automatisierungsgeräts bei den folgend aufgeführten Fehlern selbst zu bestimmen.

- Aufruf eines nicht geladenen Bausteins
- Quittungsverzug bei Einzelzugriff auf Peripheriebaugruppen
- Quittungsverzug beim Aktualisieren des Prozeßabbilds
- Adressierfehler
- Zykluszeitüberschreitung
- Substitutionsfehler
- Quittungsverzug bei Eingangsbyte 0
- Quittungsverzug beim Zugriff auf das erweiterte Peripherievolumen
- Quittungsverzug beim Zugriff auf Externspeicher
- Parity-Fehler beim Externspeicher
- Transfer-Fehler in Datenbaustein
- Weck-Fehler

Der Anwender kann beim Auftreten einer dieser Fehler das Automatisierungsgerät weiterlaufen lassen, in den Stopp-Zustand bringen oder ein spezielles Programm bearbeiten lassen (z. B. Setzen von Anzeigen, die nicht durch das Signal ,,Befehlsausgabe sperren'' BASP abgeschaltet werden mit anschließendem Stopp).

5.8 Auswertung eines Gerätefehlers

Aufruf eines nicht geladenen Bausteins

Vom Systemprogramm wird erkannt, wenn im Anwenderprogramm ein Baustein aufgerufen wird, der nicht geladen bzw. durch das Programmiergerät 670 für ,,ungültig" erklärt worden ist. Dies gilt für Programmbausteine, Funktionsbausteine und Schrittbausteine, die sowohl unbedingt oder auch bedingt aufgerufen werden.
Wenn der Aufruf eines nicht geladenen Bausteins erkannt wird, ruft das Systemprogramm den Organisationsbaustein OB 19 auf. In diesem Organisationsbaustein kann das weitere Verhalten des Automatisierungsgeräts bestimmt werden. Wird der Organisationsbaustein OB 19 nicht belegt, dann wird vom Prozessor der Aufruf des nicht geladenen Bausteins wie eine Nulloperation (NOP) behandelt. Die Bearbeitung des STEP-5-Programms wird fortgesetzt.

Quittungsverzug

Ein Quittungsverzug tritt auf, wenn sich eine Eingabe- oder Ausgabebaugruppe nach einer Adressierung innerhalb einer bestimmten Zeit nicht mit einem ,,Ready-Signal" zurückmeldet. Voraussetzung dafür ist, daß diese Peripheriebaugruppe beim Neustart des Automatisierungsgeräts vorhanden und nicht defekt war.
Die Ursache des Quittungsverzugs kann ein Defekt auf der Baugruppe sein oder das Entfernen der Baugruppe während des Betriebs.

Folgende ,,Quittungsverzugs-Fehler" unterbrechen die zyklische Bearbeitung und rufen einen entsprechenden Organisationsbaustein auf:

Quittungsverzug bei Einzelzugriff auf eine Peripheriebaugruppe (Lade- oder Transferoperationen). Das Systemprogramm ruft den OB 23 auf.

Beim Aktualisieren des Prozeßabbilds. Das Systemprogramm ruft den OB 24 auf.

Quittungsverzug bei Einzelzugriff auf eine dezentrale Peripheriebaugruppe im erweiterten Adressiervolumen. Das Systemprogramm ruft OB 29 auf.
Sind die aufgerufenen Organisationsbausteine nicht programmiert, wird die Bearbeitung des Anwenderprogramms fortgesetzt.
Bei Anwahl des Eingangsbytes 0 (Alarmeingänge). Das Systemprogramm ruft den OB 28 auf.
Wird der OB 28 nicht programmiert, geht das AG beim Auftreten der Fehler in den Stopp-Zustand.

Die Ursache eines Quittungsverzugs kann auch ein defekter oder nicht vorhandener Speicher sein. Meldet sich eine angesprochene Adresse im Externspeicher nicht mit einem ,,Ready-Signal" zurück, so wird das vom Systemprogramm erkannt und OB 30 wird aufgerufen. Wurde dieser Baustein nicht programmiert, wird die Bearbeitung des Anwenderprogramms fortgesetzt.

ACHTUNG:
Tritt bei der Ausführung eines Lesevorgangs ein Quittungsverzug auf (zu lesende Speicheradresse quittiert nicht), so erscheint als Ergebnis des Lesevorgangs fälschlich Signalzustand ,,1".

Adressierfehler

Ein Adressierfehler tritt auf, wenn mit einer STEP-5-Operation ein Eingang oder Ausgang im Prozeßbild angesprochen wird, zu dem zum Zeitpunkt des letzten Neustarts keine Peripheriebaugruppe zugeordnet war.
Das Systemprogramm unterbricht nun die Bearbeitung des STEP-5-Programms und ruft den Organisationsbaustein OB 25 auf. In diesem Organisationsbaustein kann das weitere Verhalten des Automatisierungsgeräts bestimmt werden.

ACHTUNG:
Wird der Organisationsbaustein OB 25 nicht programmiert, geht der Prozessor beim Auftreten eines Adressierfehlers in den Stopp-Zustand.

Soll bei einem erkannten Adressierfehler kein spezielles Programm bearbeitet werden, genügt es, wenn eine Bausteinende-Anweisung im Operationsbaustein 25 programmiert wird:
OB 25: BE (Bausteinende)

Wurde vor dem Auftreten eines Adressierfehlers die Anweisung ,,AFS" programmiert, so wird der Organisationsbaustein OB 25 nicht aufgerufen. Der Befehl ,,AFF" hebt diese sperrende Wirkung wieder auf.

Zykluszeitüberschreitung

Die Zykluszeit umfaßt die gesamte Zeitdauer einer Bearbeitung des zyklischen Programms. Darin enthalten sind der Aufruf und die Bearbeitung des Organisationsbausteins OB 1 und die in diesem Organisationsbaustein aufgerufenen Programm- und Funktionsbausteine mit den Schachtelungen sowie alle in diesem Zyklus bearbeiteten zeit- und alarmgesteuerten Programmteile. Das zyklische Programm endet mit einer Bausteinende-Anweisung im Organisationsbaustein OB 1. Überschreitet die Bearbeitungszeit eine bestimmte Zeitdauer (die im Prozessor eingestellte ,,Zykluszeit"), erkennt das Systemprogramm den Fehler ,,Zykluszeitüberschreitung".

Die Zykluszeitüberschreitung kann z. B. durch fehlerhafte Programmierung ausgelöst werden, wenn bei einem bestimmten Prozeßstand der Prozessor in einer Programmschleife läuft oder durch Ausfall des Taktgenerators.

Tritt eine Zykluszeitüberschreitung auf, unterbricht das Systemprogramm die Bearbeitung des STEP-5-Programms und ruft den Organisationsbaustein OB 26 auf. In diesem Organisationsbaustein kann das weitere Verhalten des Automatisierungsgeräts bestimmt werden.

ACHTUNG:
Wird der Organisationsbaustein OB 26 nicht belegt, geht der Prozessor beim Auftreten einer Zykluszeitüberschreitung in den Stopp-Zustand.

Soll bei einer erkannten Zykluszeitüberschreitung kein spezielles Programm bearbeitet werden, genügt es, wenn eine Bausteinende-Anweisung im Operationsbaustein 26 programmiert wird:
OB 26: BE (Bausteinende)

Substitutionsfehler

Der Prozessor führt bei der Bearbeitung des STEP-5-Programms innerhalb eines Funktionsbausteins eine ,,Substitution'' durch, wenn er eine Operation mit dem Operanden ,,Bausteinparameter'' X ausführt. Der Operand X wird dabei durch den im Aufruf des Funktionsbausteins stehenden Operanden ersetzt (,,substituiert'', siehe ,,Aufbau von Funktionsbausteinen'' Seite 7).

Eine nicht zulässige Substitution (siehe ,,Erstellung eines Funktionsbausteins'' Seite 7) wird vom Prozessor erkannt. Das Systemprogramm unterbricht dann die Bearbeitung des STEP-5-Programms und ruft den Organisationsbaustein OB 27 auf. In diesem Organisationsbaustein kann das weitere Verhalten des Automatisierungsgeräts bestimmt werden.

ACHTUNG:
Wird der Organisationsbaustein OB 27 nicht belegt, geht der Prozessor beim Auftreten eines Substitutionsfehlers in den Stopp-Zustand.

Soll bei einem erkannten Substitutionsfehler kein spezielles Programm bearbeitet werden, genügt es, wenn eine Bausteinende-Anweisung im Operationsbaustein 27 programmiert wird:
OB 27: BE (Bausteinende)

Parity-Fehler im Externspeicher

Wenn im Gesamtaufbau ein Externspeicher und eine überwachende Parity-Baugruppe vorhanden ist, werden eventuelle Unstimmigkeiten zwischen einem Lese- und Schreibvorgang im Externspeicher dem Systemprogramm als Parity-Fehler mitgeteilt. Das Systemprogramm ruft dann OB 30 auf, so daß durch das Programmieren dieses Bausteins die Möglichkeit besteht, den gemeldeten Fehler zu behandeln. Sonst erfolgt keine Reaktion.

Transfer-Fehler

Wird beim Transferieren von Daten in Datenbausteine (DB) durch den angegebenen Parameter des Transferbefehls die Datenbausteinlänge überschritten, so wird dies als Transfer-Fehler erkannt. Dies schützt vor irrtümlichen Überschreiben von Daten im Speicher. Beim erfaßten Transfer-Fehler ruft das Systemprogramm den Organisationsbaustein OB 32 auf. Wurde dieser Baustein programmiert, so kann die Fehlermeldung entsprechend behandelt werden, sonst geht das Automatisierungsgerät in Stopp. OB 32 wird auch dann aufgerufen, wenn ein Transfer auf ein Datenwort stattfindet, obwohl noch kein Datenbaustein eröffnet wurde (mit Befehl ADBxx.).

Weck-Fehler

Tritt bei der Bearbeitung eines Zeitalarms ein weiterer Zeitalarm auf, wird vom Systemprogramm OB 33 aufgerufen. Durch das Programmieren dieses Bausteins kann der Anwender auf diesen Betriebsfall gezielt reagieren. Wurde jedoch OB 33 nicht programmiert, so führt dies zum Stopp des Automatisierungsgerätes!

Das Automatisierungsgerät AG 150S verwaltet intern Kennungen, die den jeweils aktuellen Betriebszustand wiedergeben. Bei einer Unterbrechung der Tätigkeit des Prozessors durch einen unzulässigen Betriebsfall werden die Zustandskennungen im Unterbrechungs-Stack gerettet. Auf dem Programmiergerät können diese Kennungen sichtbar werden. Die Ausgabe und ihre Bedeutung sind folgende:

5.8 Auswertung eines Gerätefehlers

```
S T E U E R B I T S

ENDSCH  PBSSCH  BSTSCH  SCHTAE  ADRBAU  SPABBR  NAUAS   QUITT

EXSPVH  NSTPAN    NB      NB    PFEANW  PFESYS  PBEXSP  PBHSP
          X

STOZUS  STOANZ  NEUSTA  WIEDAN  BATPUF  DATEIN   BARB   BARBEND
          X                       X       X
  NB    UAFEHL  MAFEHL   EOVH   WANAU   WWAKT   OBWIED  OBNAU
                                          X

TESBST  QVZNIO  KOPFNI  PROEND  WECKFE  PADRFE  ASPLUE  RAMADFE

  NB    SYNFEH  NINEU   NIWIED  RUFBST  QVZNIN   SUMF    URLAD


 STPA   TBBUNT    NB      NB    TBWFEH  LIRTIR  WASTOP  WIEEND
          X
LUECK     NB    DATANF   UEBE   UESYS   WECKAK  PROMEI  QVZTES


*KOMMANDO* HARDCOPY
AUFRUF  !STACK!



U N T E R B R E C H U N G S - S T A C K


TIEFE:   01

BEF-REG:  0300     SAZ:     0E1C     DB-ADR:   0000     BA-ADR:   05EB
BST-STP:  EBA7     PB-NR.:    43     DB-NR.:           OB-NR.:
                   REL-SAZ:  003E     DBL-REG:  0000
VEK-ADR:  0040     UAMK:     0055     UALW:     FFFF

AKKU1: EA0F EEFF   AKKU2: EA17 EFFF   AKKU3: 0001 0E1C   AKKU4: 0000 0000


ERGEBNISANZEIGE:   ANZ1 ANZ0 OVFL CARRY ODER STATUS VKE ERAB
                                          X     X

STOERUNGSURSACHE:  STOPS STUEB NAU QVZ ZYK BAU SUF STUEU ADF PARI TRAF
                     X

*KOMMANDO* HARDCOPY
AUFRUF  !STACK!
```

ENDSCH: Baustein zu Ende schieben
PBSSCH: Bausteinschieben vor PROM-Einsatz
SCHTAE: Schieben tätig
SPABBR: Speicherschieben-Abbruch
EXSPVH: Externspeicher vorhanden
PFEANW: Parity-Fehler im Anwenderspeicher
PFESYS: Parity-Fehler im Systemprogramm
PBEXSP: Parity-Baugruppe Externspeicher
PBHSP: Parity-Baugruppe Hauptspeicher
STOZUS: AG im Stoppzustand
NEUSTA: AG geht durch einen Neustart in den Zyklus
WIEDAN: AG geht durch einen Wiederanlauf in den zyklischen Betrieb
BATPUF: Netzgerät hat eine Batteriepufferung (Betrieb ohne Batteriepufferung nicht zulässig)
DATEIN: Inhalt der Datum- und Uhrzeitzellen für Weckalarm nicht gültig
BARB: Bearbeitungskontrolle
MAFEHL: Kennung, daß im Maschinenfehlerwort ein Eintrag gemacht wurde
EOVH: Signalformer EO vorhanden
WANAU: Wiederanlauf nach Netzspannungsausfall
OBWIED: Anwender OB21 wird bearbeitet bzw. noch nicht beendet
OBNAU: Anwender OB22 wird bearbeitet bzw. noch nicht beendet
TESBST: Testbaustein nicht in Ordnung
KOPFNI: Bausteinkopf ist nicht interpretierbar
PROEND: Schieben vor Promeinsatz beendet
WECKFE: Weckalarmbearbeitungsfehler (WECK-Alarm hat sich selbst unterbrochen)
PADRFE: Adressierfehler im Anwender-PROM-Speicher
ASPLUE: Anwenderspeicher ist lückenhaft adressiert
RAMADFE: Adressierfehler im Anwender-RAM-Speicher
SYNFE: Synchronisationsfehler im Anwenderspeicher (unzulässiger Code)
NINEU: Neustart nicht möglich (Urladen notwendig)
NIWIED: Wiederanlauf nicht möglich (Neustart notwendig)
RUFBST: Aufruf eines nicht vorhandenen Bausteins
SUMF: Summenfehler (Auswechseln des Systemprogrammspeichers)
URLAD: Urladen d. h. Urlöschen notwendig
WASTOP: Wiederanlauf, wenn auch der Stopp-Schalter eingelegt war
LUECK: Lücke zwischen den Bausteinen im RAM-Speicher
DATANF: Aufforderung zur Eingabe des aktuellen Datums und Uhrzeit
WECKAK: Weckalarmbearbeitung nicht aktiv
BR-REG: Befehlsregister
BST-STP: Bausteinstackpointer
VEK-ADR: Vektoradresse für Externspeicher
SAZ: Step-Adreß-Zähler-Stand
PB-NR: Aktueller Baustein, dessen Bearbeitung die Unterbrechung verursacht
REL-SAZ: Relativer Step-Adreß-Zähler-Stand im aktuellen Baustein
UAMK: Unterbrechungsanzeigenmaske
DB-ADR: Datenbausteinanfangsadresse
DB-NR: Aktueller Datenbaustein
DBL-REG: Register, in dem Datenbausteinlänge geführt wird
UALW: Unterbrechungsanzeigenlöschwert
BA-ADR: Bausteinanfangsadresse
OB-NR: Baustein, von dem aus der aktuelle Baustein aufgerufen wird
STOPS: Stopschalter
STUEB: Bausteinstacküberlauf
NAU: Netzspannungsausfall
QVZ: Quittungsverzug
ZYK: Zykluszeitüberschreitung
BAU: Batterieausfall
SUF: Substitutionsfehler
STUEU: Überlauf U-Stack
ADF: Adressierfehler
PARI: Parity-Fehler
TRAF: Transfer-Fehler

6.1 Allgemeine Hinweise

Ein überwiegender Teil der STEP-5-Operationen verwendet als Quelle für die Operanden und als Ziel für die Ergebnisse zwei Register (je 32 bit breit): Akkumulator 1 (Akku 1) und Akkumulator 2 (Akku 2). Da diese Register nicht immer in ihrer vollen Breite verwendet bzw. beeinflußt werden, werden diese für die folgenden Beschreibungen wie unten angegeben in kleinere Einheiten eingeteilt:

Bitwertigkeit $2^{31}...2^{16}$ $2^{15}...2^{0}$

Akku-H Akku-L

Lade- und Transferbefehle verwenden je nach Adressierung (byte-, wort- oder doppelwortweise) wie folgt den Inhalt von Akku 1:

byteweise: Akku 1-L, Bit 2^7 bis 2^0 ⇄ (Transferieren / Laden) adressiertes Byte

wortweise: Akku 1-L, ⇄ adressiertes Wort

doppelwortweise: Akku 1 ⇄ adressiertes Doppelwort

Bei Ladeoperationen werden die nicht beteiligten Bitstellen des Akku 1 stets mit Nullen gefüllt. Allen Ladebefehlen ist es gemeinsam, daß zuerst der Inhalt von Akku 1 von Akku 2 übernommen wird, bevor der Inhalt der angesprochenen Adresse in Akku 1 geladen wird. Bei Transferbefehlen bleiben Akku 1 und Akku 2 unverändert.

Zahlendarstellungen

Als Operanden für die STEP-5-Befehle, die Inhalte von Akku 1 und Akku 2 verknüpfen, verändern oder vergleichen, sind Zahlen in verschiedenen Darstellungen zulässig. Je nach durchzuführender Operation wird der Inhalt von Akku 1 bzw. Akku 2 als eine der folgenden Darstellungen interpretiert:

I) Festpunktzahl:
Steht in Akku-L und wird als eine 16-bit-Dualzahl in 2-er-Komplement-Darstellung aufgefaßt.

II) Festpunkt-Doppelwort:
Steht in Akku und wird als eine 32-bit-Dualzahl in 2-er-Komplement-Darstellung aufgefaßt.

III) Gleitkommazahl: $m \cdot 2^{exp}$
m = Mantisse
exp = Exponent

Wird in Akku wie folgt dargestellt:

Bitwertigkeit: $2^{31}...2^{24}$ $2^{23}...2^{0}$

exp m

Der Exponent ist eine 8-bit-Dualzahl in 2-er-Komplement-Darstellung:
$-128 \leqq exp < 127$

Die Mantisse ist 24 bit breit und normiert:
$0{,}5 \leqq$ positive Mantisse < 1;
$-1 <$ negative Mantisse $\leqq 0{,}5$

IV) BCD-codierte Zahl mit Vorzeichen + 3 Ziffern:
Belegung im Akku-L

Bitwertigkeit: $\underbrace{2^{15}-2^{12}}_{\text{Vorzeichen}}$ $\underbrace{2^{11}-2^{8}}_{10^2}$ $\underbrace{2^{7}-2^{4}}_{10^1}$ $\underbrace{2^{3}-2^{0}}_{10^0}$

Die einzelnen Ziffern sind positive 4-bit-Dualzahlen in 2-er-Komplement-Darstellung.

Vorzeichen: 0000 wenn die Zahl positiv ist
1111 wenn die Zahl negativ ist

V) BCD-codierte Zahl mit Vorzeichen + 7 Ziffern:
Belegung im Akku

Bitwertigkeit: $\underbrace{2^{31}-2^{28}}_{\text{Vorzeichen}}$ $\underbrace{2^{27}-2^{24}}_{10^6}$ $\underbrace{2^{23}-2^{20}}_{10^5}$

$\underbrace{2^{19}-2^{16}}_{10^4}$ $\underbrace{2^{15}-2^{12}}_{10^3}$ $\underbrace{2^{11}-2^{8}}_{10^2}$ $\underbrace{2^{7}-2^{4}}_{10^1}$ $\underbrace{2^{3}-2^{0}}_{10^0}$

Hinweis:
Diese interne Darstellung muß nicht dem Format entsprechen in dem die Zahlen beim Erstellen eines Programms über das Programmiergerät eingegeben werden. Das Programmiergerät erzeugt die oben aufgeführten Darstellungen.

Ergebnisanzeigen des AG 150S

Es gibt Befehle für die Verarbeitung einzelner Bit-Informationen und es gibt Befehle für die Verarbeitung von Wort-Informationen (8, 16 oder 32 bit).

In beiden Gruppen gibt es anzeigen-setzende Befehle und anzeigen-auswertende Befehle.
Entsprechend den beiden Befehlsgruppen gibt es ,,Bit-Anzeigen" und ,,Wort-Anzeigen". Das Anzeigenbyte bei AG 150S sieht folgendermaßen aus

2^7	Wort-Anz.			Bit-Anz.			2^0
Anz. 1	Anz. 0	OV	OS	OR	STA	VKE	ER

Zu den Bit-Anzeigen:

$\overline{ER}$ $\overline{ERAB}$ ist die Abkürzung von Erstabfrage. Mit ihr beginnt eine logische Verknüpfung. Am Ende einer log. Verknüpfungskette (Speicheroperationen) wird $\overline{ERAB}$ gesetzt.

VKE: Verknüpfungsergebnis; Ergebnis bit-breiter Verknüpfungen.
Wahrheitsaussage bei den Vergleichsbefehlen.

STA: STATUS gibt bei Bit-Befehlen den log. Zustand des gerade abgefragten oder gesetzten Bits an. Status wird bei binären Verknüpfungsoperationen (ausgenommen U(, O(,),O) und bei Speicheroperationen aktualisiert.

6 STEP-5-Befehlsvorrat mit Programmierbeispielen

6.1 Allgemeine Hinweise

OR: ODER; sagt dem Prozessor, daß folgende UND-Verknüpfungen vor einer ODER Verknüpfung (UND vor ODER) behandelt werden müssen.

Zu den Wort-Anzeigen:

OV: OVER, gibt an, ob bei der eben abgeschlossenen arithmetischen Operation der zulässige Zahlenbereich überschritten worden ist.

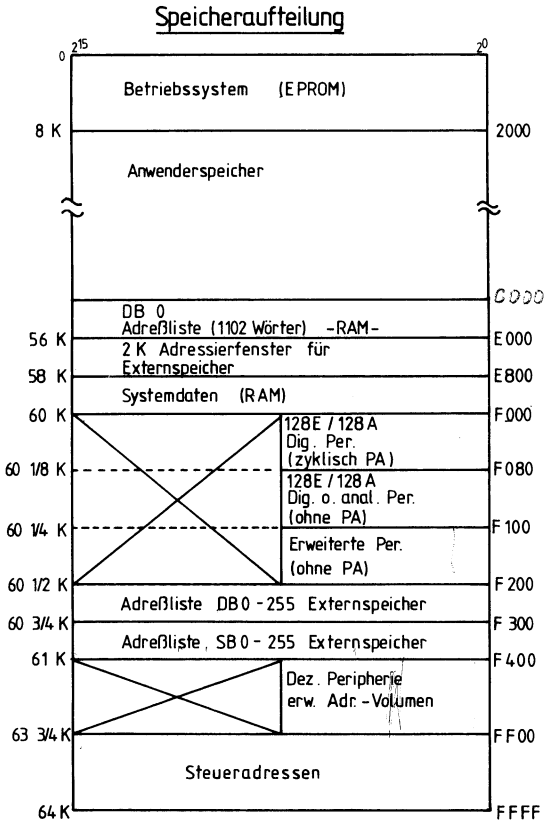
OS: OVER SPEICHERND; das Over-Bit ist gespeichert; dient dazu im Verlaufe mehrerer arithmetischer Operationen zu erkennen, ob irgendwann ein Fehler durch Überlauf (OVER) aufgetreten ist.

Anz 1, Anz 0 sind codierte Ergebnis-Anzeigen, deren Interpretation aus der folgenden Tabelle ersichtlich wird.

Wort-Ergebnis-Anzeigen			Festpunktrechnung Ergebnis	Bool Ergebnis	Vergleich Inhalte von Akku 1 + Akku 2	Schieben geschobenes Bit	Gleitpunktrechnung Ergebnis
ANZ 1	ANZ 2	OVER					
0	0	0	Erg. = 0	= 0	Akku 2 = Akku 1	0	Mantisse = 0; Expo. zulässig
0	1	0	Erg. < 0	–	Akku 2 < Akku 1	–	Mantisse < 0; Expo. zulässig
1	0	0	Erg. > 0	≠0	Akku 2 > Akku 1	1	Mantisse > 0; Expo. zulässig
0	0	1	,,Over-Null''*)	–	–	–	Mantisse ≠0; Expo. – 128
0	1	1	0 aus pos. Bereich	–	–	–	Mantisse < 0; Expo. + 127
1	0	1	0 aus neg. Bereich	–	–	–	Mantisse > 0; Expo. + 127
1	1	1	Division durch Null	–	–	–	Division durch Null

*) Sonderfall: Addition der betragsgrößten negativen Zahl zu sich selbst

Zur unmittelbaren Auswertung der Anzeigen stehen **Sprungoperationen** zur Verfügung (siehe ,,ergänzende Operationen'').



Bereich		wird angesprochen mit	Parameter	zugehöriges Prozeßabbild (PA)
0 … 127	PAE Prozeßabbild der Eingänge	L EB / T EB L EW / T AB L ED / T ED	0–127 0–126 0–124	——
0 … 127	PAA Prozeßabbild der Ausgänge	L AB / T AB L AW / T AW L AD / T AD	0–127 0–126 0–124	——
0 … 127	digitale Peripherie Eingänge	L PB L PW	0–127 0–126	PAE
0 … 127	digitale Peripherie Ausgänge	T BP T PW	0–127 0–126	PAA
128 … 255	digitale oder analoge Peripherie Eingänge	L PB L PW	128–255 128–254	kein PA
128 … 255	digitale oder analoge Peripherie Ausgänge	L PB L PW	128–255 128–254	kein PA
0 … 255	erweiterte Peripherie Eingänge	L QB L QW	0–255 0–254	kein PA
0 … 255	erweiterte Peripherie Ausgänge	T QB L QW	0–255 0–254	kein PA
	Peripherie mit erweitertem Adressiervolumen	Adressierung: Siehe Beschreibung „Serielle Kopplung" Siehe Beschreibung „Standard-FB"		kein PA

6.2 Grundoperationsvorrat

Verknüpfungsoperationen, binär

Operation	Parameter	Funktion
)		Klammer zu
U (		UND-Verknüpfung von Klammerausdrücken
O (		ODER-Verknüpfung von Klammerausdrücken
O		ODER-Verknüpfung von UND-Funktionen
U □□		UND-Verknüpfung mit
O □□		ODER-Verknüpfung mit
E	0.0 bis 127.7	Abfrage eines Eingangs auf Signalzustand ,,1"
A	0.0 bis 127.7	Abfrage eines Ausgangs auf Signalzustand ,,1"
M	0.0 bis 255.7	Abfrage eines Merkers auf Signalzustand ,,1"
D	0.0 bis 255.15	Abfrage eines Datenwortes auf Signalzustand ,,1"
N E	0.0 bis 127.7	Abfrage eines Eingangs auf Signalzustand ,,0"
N A	0.0 bis 127.7	Abfrage eines Ausgangs auf Signalzustand ,,0"
N M	0.0 bis 255.7	Abfrage eines Merkers auf Signalzustand ,,0"
N D	0.0 bis 255.15	Abfrage eines Datenwortes auf ,,Signalzustand ,,0"
T	0 bis 255	Abfrage einer Zeit auf Signalzustand ,,1"
N T	0 bis 255	Abfrage einer Zeit auf Signalzustand ,,0"
Z	0 bis 255	Abfrage eines Zählers auf Signalzustand ,,1"
N Z	0 bis 255	Abfrage eines Zählers auf Signalzustand ,,0"

Die binären Verknüpfungsoperationen erzeugen als Ergebnis das ,,VKE" (Verknüpfungsergebnis).

Am Anfang einer Verknüpfungskette hängt die Ergebnisbildung nur von der Verknüpfungsart (U ≙ UND, UN ≙ UND-NICHT, 0 ≙ ODER, ON ≙ ODER-NICHT) und dem abgefragten Signalzustand ab.
Innerhalb einer Verknüpfungskette wird das VKE aus Verknüpfungsart, bisherigem VKE und dem abgefragten Signalzustand gebildet. Eine Verknüpfungskette wird durch einen schrittbegrenzten Befehl (z. B. Speicheroperationen) abgeschlossen.

Zeit- und Zähloperationen

Um eine Zeit bzw. einen Zähler durch einen Setzbefehl zu laden, muß der Wert vorher in den Akkumulator geladen werden.

Sinnvoll sind folgende Ladeoperationen: [1]
für Zeiten: L KT, L EW, L AW, L MW, L DW
für Zähler: L KZ, L EW, L AW, L MW, L DW

Operation	Parameter	Funktion
S I T	0 bis 255	Starten einer Zeit als Impuls
S V T	0 bis 255	Starten einer Zeit als verlängerter Impuls
S E T	0 bis 255	Starten einer Zeit als Einschaltverzögerung
S S T	0 bis 255	Starten einer Zeit als speichernde Einschaltverzögerung
S A T	0 bis 255	Starten einer Zeit als Ausschaltverzögerung
R T	0 bis 255	Rücksetzen einer Zeit
S Z	0 bis 255	Setzen eines Zählers
R T	0 bis 255	Rücksetzen eines Zählers
Z V Z	0 bis 255	Vorwärtszählen eines Zählers
Z R T	0 bis 255	Rückwärtszählen eines Zählers

Lade-, Transfer- und Vergleichsfunktionen

Operation	Parameter	Funktionen
L □□		Laden
T □□		Transferieren
E B	0 bis 127	eines Eingabebytes vom PAE[3]
E W	0 bis 126	eines Eingabewortes vom PAE
E D	0 bis 124	eines Eingabe-Doppelwortes vom PAE
A B	0 bis 127	eines Ausgabebytes vom PAA[4]
A W	0 bis 126	eines Ausgangswortes vom PAA
A D	0 bis 124	eines Ausgangs-Doppelwortes vom PAA
M B	0 bis 225	eines Merkerbytes
M W	0 bis 254	eines Merkerwortes
M D	0 bis 252	eines Merker-Doppelwortes
D R	0 bis 255	eines Datums (rechtes Byte)
D L	0 bis 255	eines Datums (linkes Byte)
D W	0 bis 255	eines Datenwortes
O D	0 bis 254	eines Datum-Doppelwortes
T [2]	0 bis 255	eines Zeitwertes
Z [2]	0 bis 255	eines Zählwertes
P B	0 bis 255	eines Peripheriebytes der Digital-Eingaben bzw. -Ausgaben
	128 bis 255	eines Peripheriebytes der Analog-Eingaben bzw. -Ausgaben
Q B	0 bis 255	eines Bytes der erweiterten Peripherie
P W	0 bis 126	eines Peripheriewortes der Digital-Eingaben bzw. -Ausgaben
	128 bis 254	eines Peripheriewortes der Analog-Eingaben bzw. -Ausgaben
Q W	0 bis 254	eines Wortes der erweiterten Peripherie
K M	16-Bit-Muster	einer Konstanten als Bitmuster
K H	0 bis FFFFH	einer Konstanten im Hexa-Code
K F	0 bis + ($2^{16}-1$)	einer Konstanten als Festpunktzahl
K Y	0 bis 255 für jedes Byte	einer Konstanten, 2 byte
K B	0 bis 255	einer Konstanten, 1 byte
K C	2 alphanumerische Zeichen	einer Konstanten, 2 ASCII-Zeichen
K G	$\pm 0{,}1469368 \cdot 10^{-38}$ bis $\pm 0{,}1701412 \cdot 10^{+39}$	einer Konstanten als Gleitpunktzahl
K T [2]	0.0 bis 999.3	eines Zeitwertes (Konstante)
K Z [2]	0 bis 999	eines Zählwertes (Konstante)
! = □		Vergleich auf gleich
>< □		Vergleich auf ungleich
> □		Vergleich auf größer
>= □		Vergleich auf größer – gleich
< □		Vergleich auf kleiner
<= □		Vergleich auf kleiner – gleich
F		zweier Festpunkt-Zahlen
G		zweier Gleitkomma-Zahlen
D		zweier Festpunkt-Doppelwort-Zahlen

Die Lade- und Transferoperationen beeinflussen die Anzeigen nicht. Die Vergleichsbefehle erzeugen als Ergebnis das VKE und die Wort-Anzeigen ANZ 1, ANZ 0 (siehe Seite 26).

1) Zeit- oder Zähloperationen verändern den Inhalt von Akku 1 nicht.
2) nicht bei Transferieren
3) PAE Prozeßabbild für Eingänge
4) PAA Prozeßabbild für Ausgänge

Bausteinaufrufe

Operation	Parameter	Funktion
S P A □□		Sprung unbedingt
S P B □□		Sprung unbedingt (abhängig von VKE)
↑↑		
P B	1 bis 255	zu einem Programmbaustein
F B	1 bis 255	zu einem Funktionsbaustein
S B	1 bis 255	zu einem Schrittbaustein
A D B	1 bis 255	Aufruf eines Datenbausteins
B E		Bausteinende
B E B		Bausteinende bedingt (abhängig vom VKE)

Speicheroperationen

Operation	Parameter	Funktion
S □		Setzen
R □		Rücksetzen
= □		Zuweisen
↑		
E	0.0 bis 127.7	eines Eingangs
A	0.0 bis 127.7	eines Ausgangs
M	0.0 bis 255.7	eines Merkers
D	0.0 bis 255.15	eines Datenwortes

Nulloperationen

Operation	Parameter	Funktion
N O P 0		Nulloperation (alle Bits gelöscht)
N O P 1		Nulloperation (alle Bits gesetzt)
B L D	*)	Bildaufbauanweisung für das Programmiergerät: wird vom AG wie eine Nulloperation behandelt.

*) siehe Programmiergerätebeschreibung

Stop-Anweisung

Operation	Parameter	Funktion
S T P		Stop

Codeoperationen

Operation	Parameter	Funktion
L C □		Lade codiert
↑		
T	0 bis 255	von Zeitwerten
Z	0 bis 255	von Zählwerten

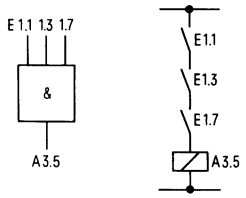
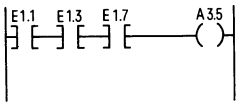
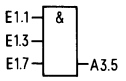
Arithmetische Operationen

Operation	Parameter	Funktion
+ □		Addition
– □		Subtraktion
* □		Multiplikation
/ □		Division
↑		
F		zweier Festpunktzahlen
G		zweier Gleitpunktzahlen

Programmierbeispiele für Verknüpfungs-, Speicher-, Zeit-, Zähl- und Vergleichsfunktionen

Verknüpfungsfunktionen

UND-Verknüpfung

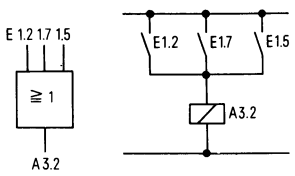
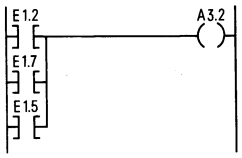
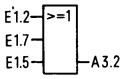
Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 1.1 U E 1.3 U E 1.7 = A 3.5		

Am Ausgang A 3.5 erscheint Signalzustand ,,1'', wenn alle Eingänge gleichzeitig den Signalzustand ,,1'' aufweisen.

Am Ausgang A 3.5 erscheint Signalzustand ,,0'', wenn mindestens einer der Eingänge den Signalzustand ,,0'' aufweist.

Die Anzahl der Abfragen und die Reihenfolge der Programmierung ist beliebig.

ODER-Verknüpfung

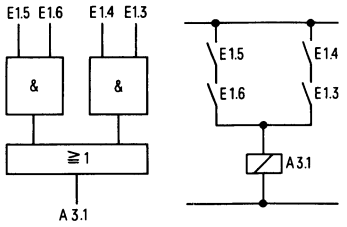
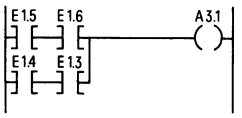
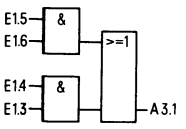
Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	O E 1.2 O E 1.7 O E 1.5 = A 3.2		

Am Ausgang A 3.2 erscheint Signalzustand ,,1'', wenn mindestens einer der Eingänge den Signalzustand ,,1'' aufweist.

Am Ausgang A 3.2 erscheint Signalzustand ,,0'', wenn alle Eingänge gleichzeitig den Signalzustand ,,0'' aufweisen.

Die Anzahl der Abfragen und die Reihenfolge der Programmierung ist beliebig.

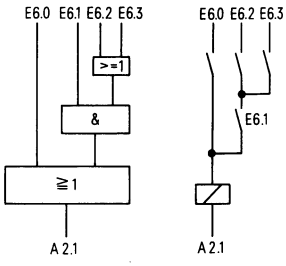
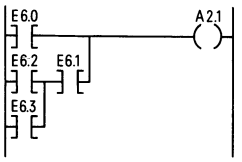
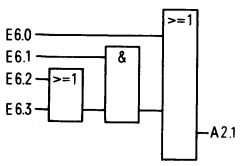
UND-vor-ODER-Verknüpfung

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 1.5 U E 1.6 O U E 1.4 U E 1.3 = A 3.1		

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

Am Ausgang A 3.1 erscheint Signalzustand ,,0'', wenn keine UND-Verknüpfung erfüllt ist.

ODER-vor-UND-Verknüpfung

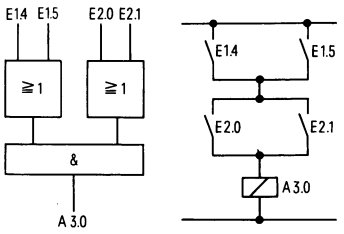
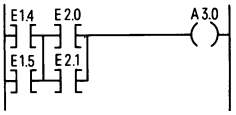
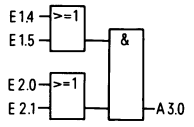
Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	O E 6.0 O U E 6.1 U(O E 6.2 O E 6.3) = A 2.1		

Am Ausgang A 2.1 erscheint Signalzustand ,,1'', wenn Eingang E 6.0 oder Eingang E 6.1 und einer der Eingänge E 6.2 bzw. E 6.3 Signal ,,1'' führen.

Am Ausgang A 2.1 erscheint Signalzustand ,,0'' wenn, Eingang E 6.0 Signal ,,0'' führt und die UND-Verknüpfung nicht erfüllt ist.

Programmierbeispiele (Fortsetzung)

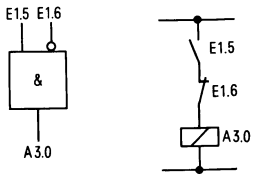
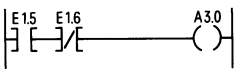
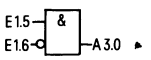
ODER-vor-UND-Verknüpfung

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U(O E 1.4 O E 1.5) U(O E 2.0 O E 2.1) = A 3.0		

Am Ausgang A 3.0 erscheint Signalzustand ,,1'', wenn beide ODER-Verknüpfungen erfüllt sind.

Am Ausgang A 3.0 erscheint Signalzustand ,,0'', wenn mindestens eine ODER-Verknüpfung nicht erfüllt ist.

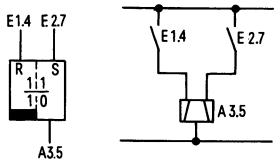
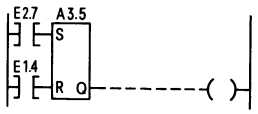
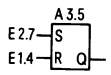
Abfrage auf Signalzustand ,,0''

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 1.5 UN E 1.6 = A 3.0		

Am Ausgang A 3.0 erscheint Signalzustand ,,1'' nur dann, wenn der Eingang E 1.5 den Signalzustand ,,1'' (Schließer betätigt) und der Eingang E 1.6 den Signalzustand ,,0'' (Öffner betätigt) führt.

Speicherfunktionen

RS-Speicherglied für speichernde Signalausgabe

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 2.7 S A 3.5 U E 1.4 R A 3.5		

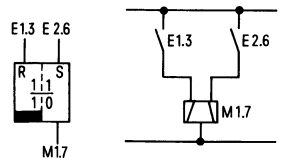
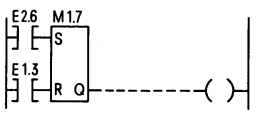
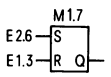
Signalzustand ,,1" am Eingang E 2.7 bewirkt das Setzen des Speicherglieds.
Wechselt der Signalzustand am Eingang E 2.7 nach ,,0", so bleibt dieser Zustand erhalten, d. h. das Signal wird gespeichert.

Signalzustand ,,1" am Eingang E 1.4 bewirkt das Rücksetzen des Speicherglieds.

Wechselt der Signalzustand am Eingang E 1.4 nach ,,0", so bleibt dieser Zustand erhalten.

Bei gleichzeitigem Anliegen des Setzsignals (Eingang E 2.7) und des Rücksetzsignals (Eingang E 1.4) ist die zuletzt programmierte Abfrage (hier U E 1.4) während der Bearbeitung des übrigen Programms wirksam.

RS-Speicherglied mit Merkern

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 2.6 S M1.7 U E 1.3 R M1.7		

Signalzustand ,,1" am Eingang E 2.6 bewirkt das Setzen des Speichergliedes.
Wechselt der Signalzustand am Eingang E 2.6 nach ,,0", so bleibt dieser Zustand erhalten, d. h. das Signal wird gespeichert.

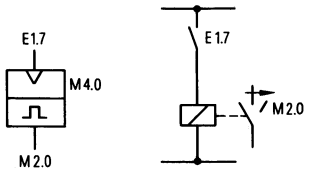
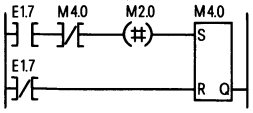
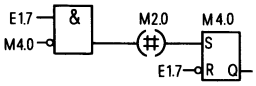
Signalzustand ,,1" am Eingang E 1.3 bewirkt das Rücksetzen des Speichergliedes.

Wechselt der Signalzustand am Eingang E 1.3 nach ,,0", so bleibt dieser Zustand erhalten.

Bei gleichzeitigem Anliegen des Setzsignals (Eingang E 2.6) und des Rücksetzsignals (Eingang E 1.3) ist die zuletzt programmierte Abfrage (hier U E 1.3) während der Bearbeitung des übrigen Programms wirksam.

Programmierbeispiele (Fortsetzung)

Nachbildung eines Wischrelais

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 1.7 UN M4.0 = M2.0 U M2.0 S M4.0 UN E 1.7 R M4.0		

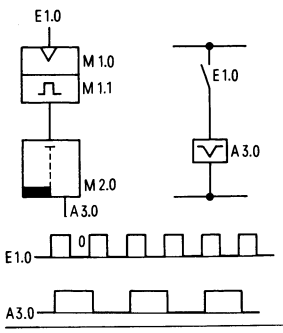
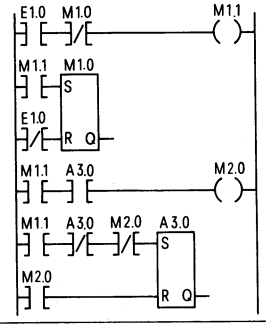
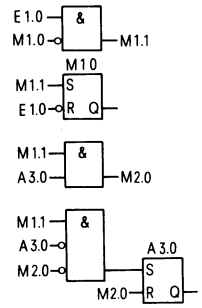
Bei jeder ansteigenden Flanke des Eingangs E 1.7 ist die UND-Verknüpfung (U E 1.7 und UN M 4.0) erfüllt und mit VKE = ,,1'' werden die Merker M 4.0 (,,Flankenmerker'') und M 2.0 gesetzt.

Beim nächsten Bearbeitungszyklus ist die UND-Verknüpfung U E 1.7 und UN M 4.0 nicht erfüllt, da der Merker M 4.0 gesetzt worden ist.

Der Merker M 2.0 wird rückgesetzt.

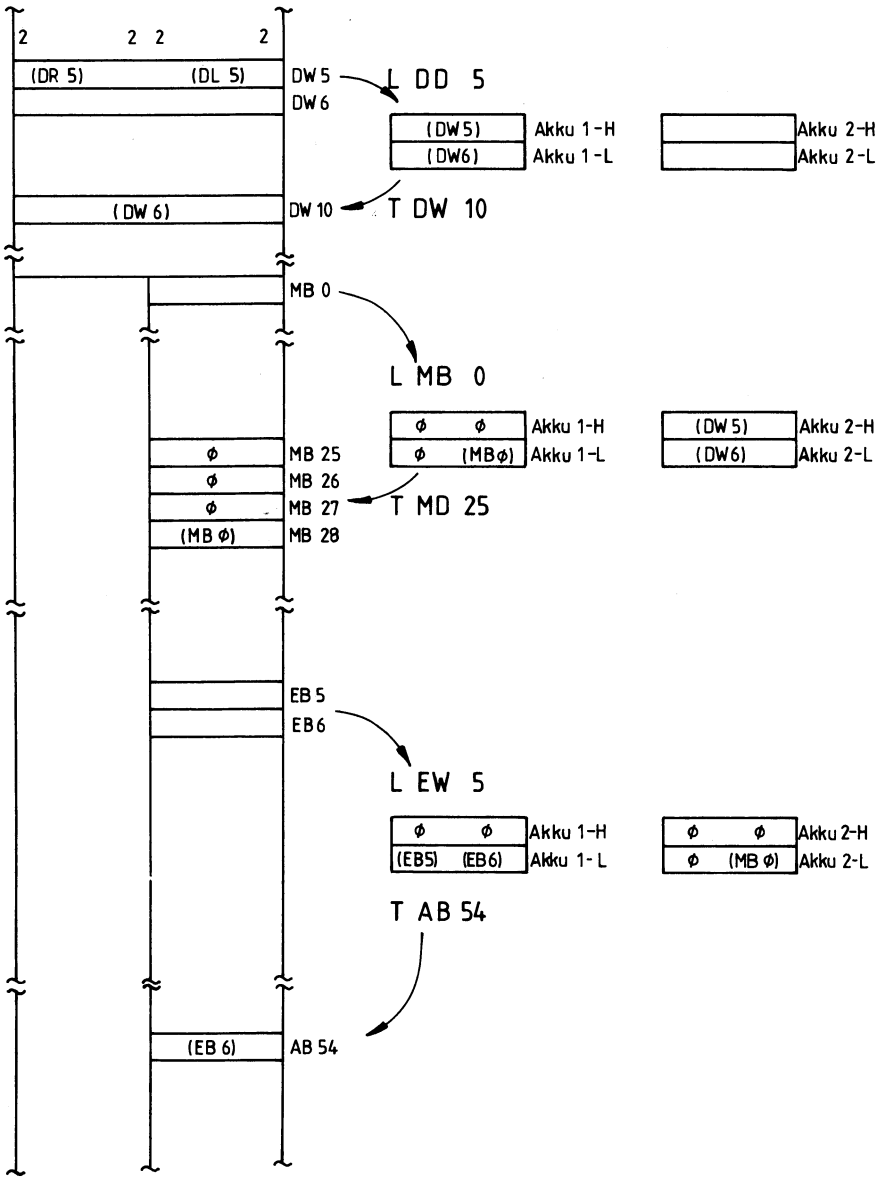
Der Merker M 2.0 führt also während eines einzigen Programmdurchlaufs Signalzustand ,,1''.

Binäruntersetzer (T-Kippglied)

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 1.0 UN M1.0 = M1.1 U M1.1 S M1.0 UN E 1.0 R M1.0 U M1.1 U A 3.0 = M2.0 U M1.1 UN A 3.0 UN M2.0 S A 3.0 U M2.0 R A 3.0		

Der Binäruntersetzer (Ausgang A 3.0) wechselt bei jedem Signalzustandswechsel von ,,0'' nach ,,1'' (ansteigende Flanke) des Einganges E 1.0 seinen Zustand. Am Ausgang des Speicherglieds erscheint deshalb die halbe Eingangsfrequenz.

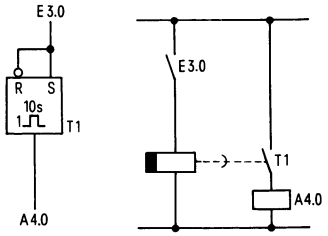
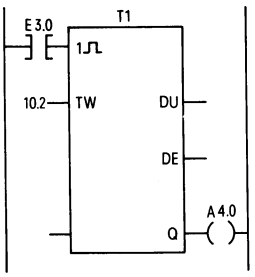
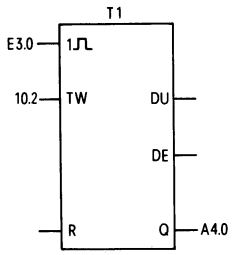
Beispiel: Lade- und Transferfunktion



Programmierbeispiele (Fortsetzung)

Zeitfunktionen

Impuls

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 3.0 L KT 10.2 SI T 1 U T 1 = A 4.0		

Bei Verknüpfungsergebnis ,,1'' und erstmaliger Bearbeitung wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis ,,1'' bleibt das Zeitglied unbeeinflußt.

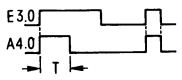
Bei Verknüpfungsergebnis ,,0'' wird das Zeitglied auf Null gesetzt (gelöscht).

Die Abfragen U T bzw. O T liefern Signalzustand ,,1'', solange die Zeit läuft.

KT 10.2:

Das Zeitglied wird mit dem angegebenen Wert (10) geladen. Die Zahl rechts vom Punkt gibt das Zeitraster an:

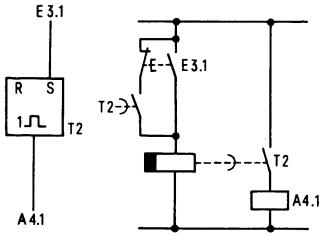
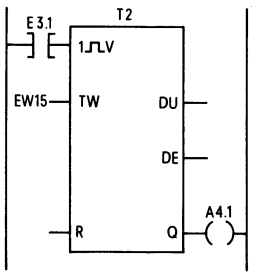
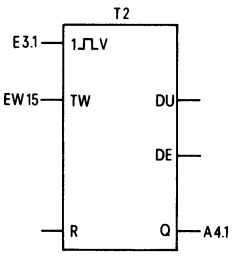
0 = 0.01s 2 = 1s
1 = 0.1s 3 = 10s



DU und DE sind digitale Ausgänge der Zeitzelle. Am Ausgang DU steht der Zeitwert dualcodiert, am Ausgang DE BCD-codiert mit Zeitraster an.

Programmierbeispiele (Fortsetzung)

Verlängerter Impuls

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 3.1 L EW15 SV T 2 U T 2 = A 4.1		

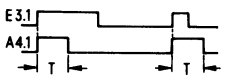
Bei Verknüpfungsergebnis ,,1'' und erstmaliger Bearbeitung wird das Zeitglied gestartet.

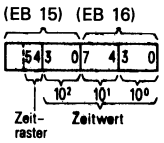
Bei Verknüpfungsergebnis ,,0'' bleibt das Zeitglied unbeeinflußt.

Die Abfragen U T oder O T liefern Signalzustand ,,1'', solange die Zeit läuft.

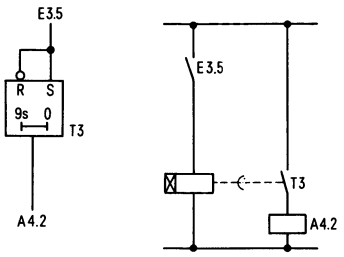
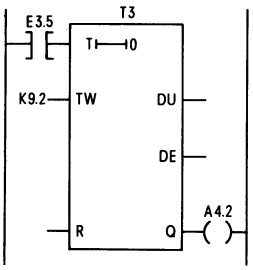
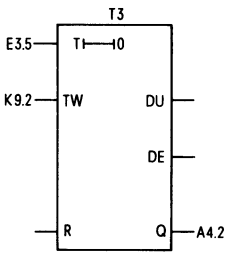
EW 15:

Setzen des Zeitwerts mit dem im BCD-Code vorliegenden Wert der Operanden E, A, M oder D (im Beispiel Eingangswort 15)





Einschaltverzögerung

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 3.5 L KT9.2 SE T 3 U T 3 = A 4.2		

Bei Verknüpfungsergebnis ,,1'' und erstmaliger Bearbeitung wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis ,,1'' bleibt das Zeitglied unbeeinflußt.

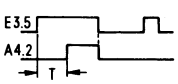
Bei Verknüpfungsergebnis ,,0'' wird das Zeitglied auf Null gesetzt (gelöscht).

Die Abfragen U T bzw. O T liefern Signalzustand ,,1'', wenn die Zeit abgelaufen und das Verknüpfungsergebnis am Eingang noch ansteht.

KT 9.2:

Das Zeitglied wird mit dem angegebenen Wert (9) geladen. Die Zahl rechts vom Punkt gibt das Zeitraster an:

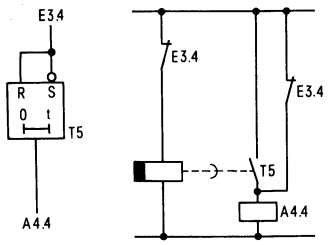
0 = 0.01s 2 = 1s
1 = 0.1s 3 = 10s



Programmierbeispiele (Fortsetzung)

Ausschaltverzögerung

Vorlage

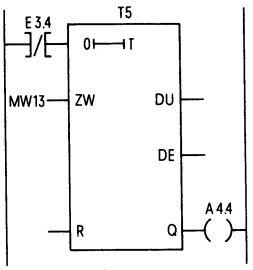


STEP-5-Darstellung

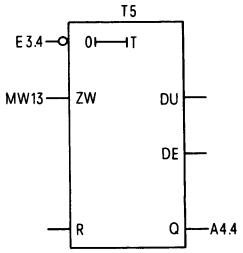
Anweisungsliste

```
UN E  3.4
L  MW13
SA T  5
U  T  5
=  A  4.4
```

Kontaktplan



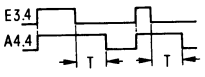
Funktionsplan

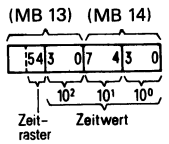


Bei Verknüpfungsergebnis ,,0'' und erstmaliger Bearbeitung wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis ,,0'' bleibt das Zeitglied unbeeinflußt.

Bei Verknüpfungsergebnis ,,1'' wird das Zeitglied auf Null gesetzt (gelöscht).

Die Abfragen U T bzw. O T liefern Signalzustand ,,1'', wenn die Zeit läuft oder das Verknüpfungsergebnis am Eingang noch ansteht.



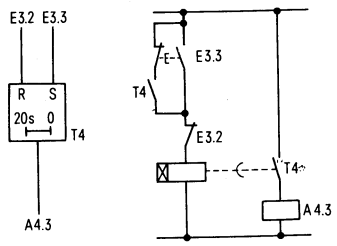


MW 13:

Setzen des Zeitwertes mit dem im BCD-Code vorliegenden Wert der Operanden E, A, M oder D (im Beispiel Merkerwort 13)

Speichernde Einschaltverzögerung

Vorlage

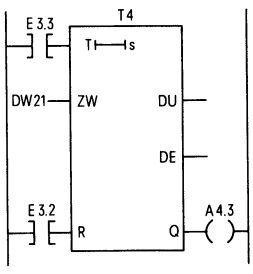


STEP-5-Darstellung

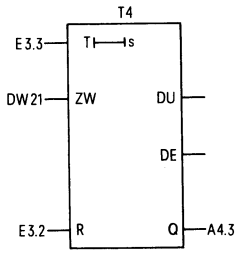
Anweisungsliste

```
U  E  3.3
L  DW21
SS T  4
U  E  3.2
R  T  4
U  T  4
=  A  4.3
```

Kontaktplan



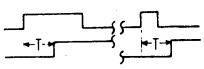
Funktionsplan

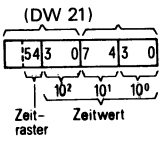


Bei Verknüpfungsergebnis ,,1'' und erstmaliger Bearbeitung wird das Zeitglied gestartet.

Bei Verknüpfungsergebnis ,,0'' bleibt das Zeitglied unbeeinflußt.

Die Abfragen U T bzw. O T liefern Signalzustand ,,1'', wenn die Zeit abgelaufen ist. Der Signalzustand wird erst dann ,,0'', wenn das Zeitglied mit der Funktion R T zurückgesetzt wurde.



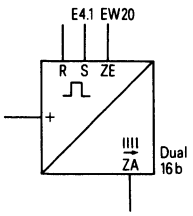
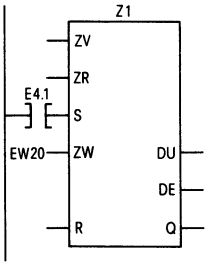
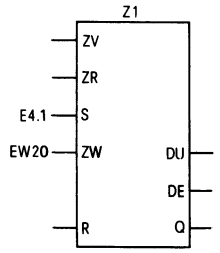


DW 21:

Setzen des Zeitwertes mit dem im BCD-Code vorliegenden Wert der Operanden E, A, M oder D (im Beispiel Datenwort 21)

Zählfunktionen

Zähler setzen

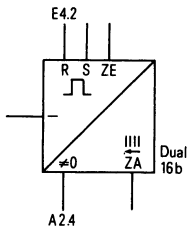
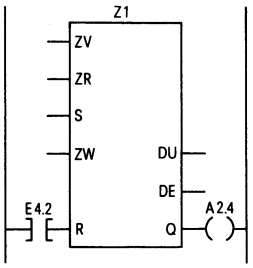
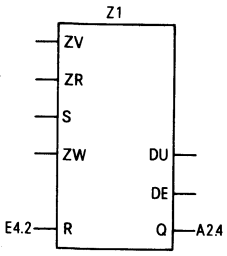
Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 4.1 L EW20 S Z 1		

Bei Verknüpfungsergebnis „1" und erstmaliger Bearbeitung wird der Zähler gesetzt. Bei wiederholter Bearbeitung bleibt der Zähler unbeeinflußt (unabhängig davon, ob das Verknüpfungsergebnis „1" oder „0" ist). Bei erneuter erstmaliger Bearbeitung mit Verknüpfungsergebnis „1" wird der Zähler wieder gesetzt (Flankenauswertung).

Der für die Flankenauswertung des Setzeingangs erforderliche Merker ist im Zählwort mitgeführt.

DU und DE sind digitale Ausgänge der Zählerzelle. Am Ausgang DU steht der Zählwort dualcodiert, am Ausgang DE BCD-codiert an.

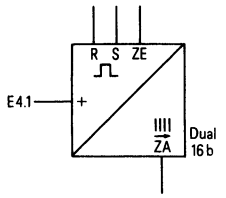
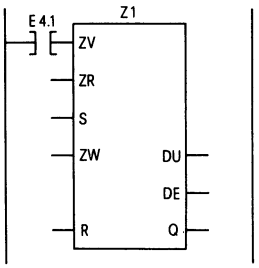
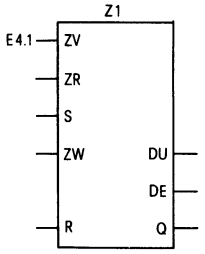
Zähler zurücksetzen

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 4.2 R Z 1 U Z 1 = A 2.4		

Bei Verknüpfungsergebnis „1" wird der Zähler auf Null gesetzt (rückgesetzt).
Bei Verknüpfungsergebnis „0" bleibt der Zähler unbeeinflußt.

Programmierbeispiele (Fortsetzung)

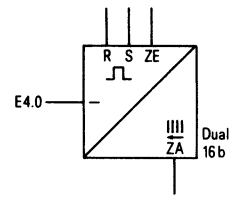
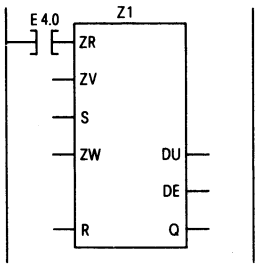
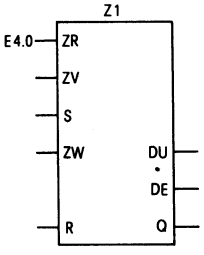
Vorwärts zählen

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 4.1 ZV Z 1		

Der Wert des adressierten Zählers wird um 1 erhöht. Die Funktion ZV wird nur bei einer positiven Flanke (von ,,0'' nach ,,1'') der vor ZV programmierten Verknüpfung ausgeführt. Die für die Flankenauswertung der Zähleingänge erforderlichen Merker sind im Zählwort mitgeführt.

Durch die zwei getrennten Flankenmerker für ZV und ZR kann ein Zähler mit zwei verschiedenen Eingängen als Vorwärts-/Rückwärtszähler verwendet werden.

Rückwärts zählen

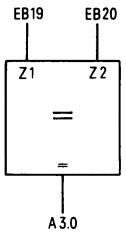
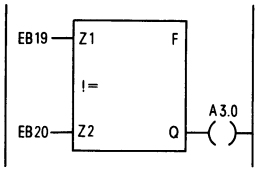
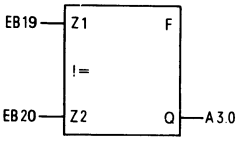
Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	U E 4.0 ZR Z 1		

Der Wert des adressierten Zählers wird um 1 erniedrigt. Die Funktion wird nur bei einer positiven Flanke (von ,,0'' nach ,,1'') der vor ZR programmierten Verknüpfung wirksam. Die für die Flankenauswertung der Zähleingänge erforderlichen Merker sind im Zählwort mitgeführt.

Durch die zwei getrennten Flankenmerker für ZV und ZR kann ein Zähler mit zwei verschiedenen Eingängen als Vorwärts-/Rückwärtszähler verwendet werden.

Vergleichsfunktionen

Vergleich auf gleich

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	L EB19 L EB20 ! = F = A 3.0		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Der Vergleich ergibt ein binäres Verknüpfungsergebnis.

VKE = ,,1'': Vergleich ist erfüllt, wenn
Akku 1-L = Akku 2-L

VKE = ,,0'': Vergleich ist nicht erfüllt, wenn
Akku 1-L ≠ Akku 2-L

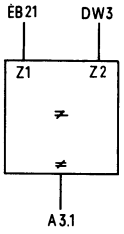
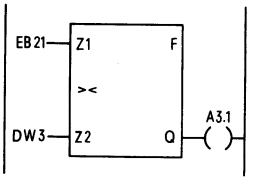
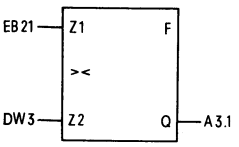
Anzeigen ANZ 1, ANZ 0 werden gemäß auf Tabelle Seite 26 gesetzt.

Akku 2-H und Akku 1-H bleiben beim Festpunktvergleich an der Operation unbeteiligt!

O	EB 19	Akku 2-L
O	EB 20	Akku 1-L

Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d. h. Inhalt von Akku 1-L und Akku 2-L wird als Festpunktzahl interpretiert.

Vergleich auf ungleich

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	L EB21 L EB >< F = A 3.1		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Der Vergleich ergibt ein binäres Verknüpfungsergebnis.

VKE = ,,1'': Vergleich ist erfüllt, wenn
Akku 1-L ≠ Akku 2-L

VKE = ,,0'': Vergleich ist nicht erfüllt, wenn
Akku 1-L = Akku 2-L

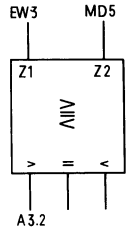
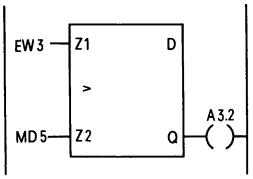
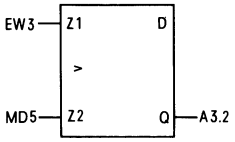
Anzeigen ANZ 1, ANZ 0 werden gemäß Tabelle auf Seite 26 gesetzt.

Akku 2-H und Akku 1-H bleiben beim Festpunktvergleich an der Operation unbeteiligt!

O	EB 19	Akku 2-L
DW 3		Akku 1-L

Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d. h. Inhalt von Akku 1-L und Akku 2-L wird als Festpunktzahl interpretiert.

Vergleich auf größer

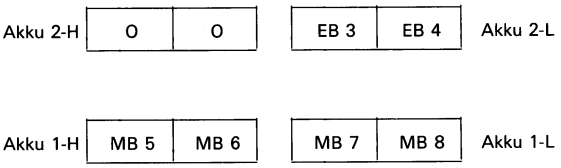
Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	L EW3 L MD5 > D = A 3.2		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen (Z1 < Z2).
Der Vergleich ergibt ein binäres Verknüpfungsergebnis.

VKE = ,,1'': Vergleich ist erfüllt, wenn
Akku 2 > Akku 1

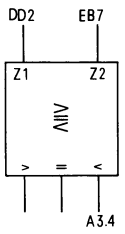
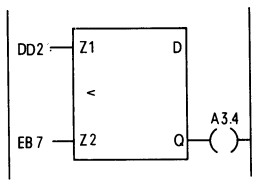
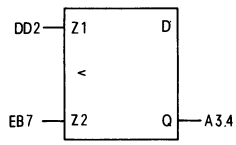
VKE = ,,0'': Vergleich ist nicht erfüllt, wenn
Akku 2 ≦ Akku 1

Anzeigen ANZ 1, ANZ 0 werden gemäß Tabelle auf Seite 26 gesetzt.



Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d. h. Inhalt von Akku 1 und Akku 2 wird als doppelwortbreite Festpunktzahl interpretiert.

Vergleich auf kleiner

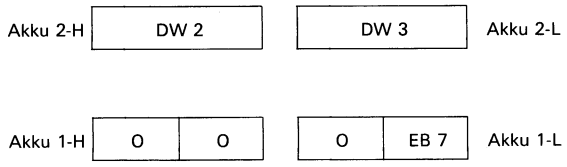
Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	L DD2 L EB7 < D = A 3.4		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen (Z1 < Z2).
Der Vergleich ergibt ein binäres Verknüpfungsergebnis.

VKE = ,,1'': Vergleich ist erfüllt, wenn
Akku 2 < Akku 1

VKE = ,,0'': Vergleich ist nicht erfüllt, wenn
Akku 2 ≧ Akku 1

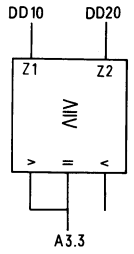
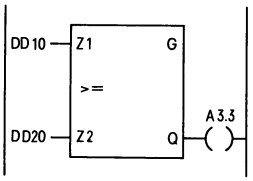
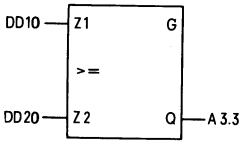
Anzeigen ANZ 1, ANZ 0 werden gemäß Tabelle auf Seite 26 gesetzt.



Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d. h. Inhalt von Akku 1 und Akku 2 wird als doppelwortbreite Festpunktzahl interpretiert.

Programmierbeispiele (Fortsetzung)

Vergleich auf größer-gleich

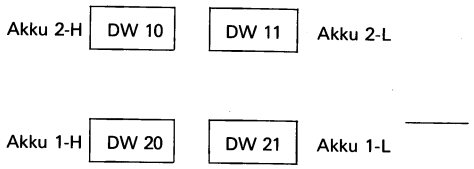
Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	L DD10 L DD20 >=G =		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen (Z1 ≧ Z2).
Der Vergleich ergibt ein binäres Verknüpfungsergebnis.

VKE = ,,1'': Vergleich ist erfüllt, wenn
Akku 2 ≧ Akku 1

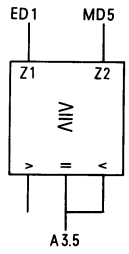
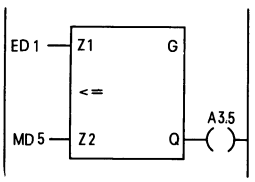
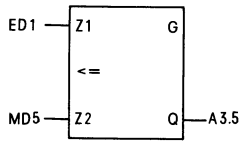
VKE = ,,0'': Vergleich ist nicht erfüllt, wenn
Akku 2 < Akku 1

Anzeigen ANZ 1, ANZ 0 werden gemäß Tabelle auf Seite 26 gesetzt.



Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d. h. Inhalt von Akku 1 und Akku 2 wird als Gleitpunktzahl interpretiert.

Vergleich auf kleiner-gleich

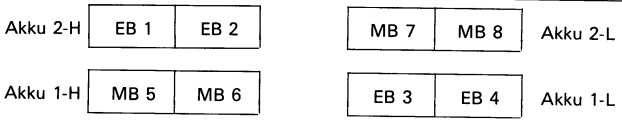
Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	L =ED1 L MD5 <=G = A 3.5		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen (Z1 < Z2).
Der Vergleich ergibt ein binäres Verknüpfungsergebnis.

VKE = ,,1'': Vergleich ist erfüllt, wenn
Akku 2 ≦ Akku 1

VKE = ,,0'': Vergleich ist nicht erfüllt, wenn
Akku 2 > Akku 1

Anzeigen ANZ 1, ANZ 0 werden gemäß Tabelle auf Seite 26 gesetzt.



Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d. h. Inhalt von Akku 1 und Akku 2 wird als Gleitpunktzahl interpretiert.

6.3 Ergänzender Operationsvorrat

Funktionsbausteine können gegenüber den restlichen Bausteinen mit einem erweitertem Operationsvorrat programmiert werden. Der Gesamtoperationsvorrat für Funktionsbausteine besteht aus den Grundoperationen und den ergänzenden Operationen.

Bei den Funktionsbausteinen werden die Operationen nur in Anweisungsliste dargestellt. Die Programme der Funktionsbausteine können also nicht in graphischer Form (FUP oder KOP) programmiert werden.

Im folgenden werden die ergänzenden Operationen beschrieben, die nur bei Funktionsbausteinen verwendet werden können. Zusätzlich sind die Kombinationsmöglichkeiten der Substitionsbefehle mit den Aktualoperanden angegeben.

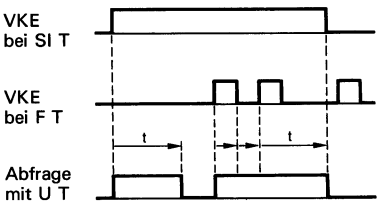
Binäre Verknüpfungen

Operation	Beschreibung
U = ☐	UND Funktion, Abfrage eines Formaloperanden auf Signalzustand „1".
UN = ☐	UND-Funktion, Abfrage eines Formaloperanden auf Signalzustand „0".
O = ☐	ODER-Funktion, Abfrage eines Formaloperanden auf Signalzustand „1".
ON = ☐	ODER-Funktion, Abfrage eines Formaloperanden auf Signalzustand „0".
↑	Formaloperand einsetzen.
	Als Aktualoperand sind binär adressierte Eingänge, Ausgänge und Merker (Parameterart: E, A; Parametertyp: BI) sowie Zeiten und Zähler (Parmeterart: T, Z) zugelassen.

Speicherfunktionen

Operation	Beschreibung
S = ☐	Setzen (binär) eines Formaloperanden.
RB = ☐	Rücksetzen (binär) eines Formaloperanden.
= = ☐	Zuweisen des Verknüpfungsergebnisses an einen Formaloperanden.
↑	Formaloperand einsetzen.
	Als Aktualoperand sind binär adressierte Eingänge, Ausgänge und Merker zugelassen (Parameterart: E, A; Parametertyp: BI).

Zeit- und Zählfunktionen

Operation	Beschreibung
F T 0 bis 255	Freigabe einer Zeit für Neustart Die Operation wird nur bei steigender Flanke des Verknüpfungsergebnisses ausgeführt. Sie bewirkt einen Neustart der Zeit, wenn bei der Startoperation Verknüpfungsergebnis „1" anliegt. 
F Z 0 bis 255	Freigabe eines Zählers Die Operation wird nur bei steigender Flanke des Verknüpfungsergebnisses ausgeführt. Sie bewirkt ein Setzen, Vorwärts- oder Rückwärtszählen des Zählers, wenn an der entsprechenden Operation Verknüpfungsergebnis „1" anliegt.
F = ☐	Freigabe eines Formaloperanden für Neustart (Beschreibung siehe F T bzw. F Z, je nach Formaloperand; Parameterart: T, Z).
RD = ☐	Rücksetzen (digital) eines Formaloperanden (Parameterart: T, Z).
SI = ☐	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als Impuls (Parameterart: T)
SE = ☐	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als verlängerter Impuls bzw. Setzen eines als Formaloperand vorgegebenen Zählers mit dem nachfolgend angegebenen Zählwert (Parameterart: T, Z).
SVZ = ☐	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als speichernde Einschaltverzögerung bzw. Vorwärtszählen eines als Formaloperand vorgegebenen Zählers (Parameterart: T, Z).
SSV = ☐	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als speichernde Einschaltverzögerung bzw. Vorwärtszählen eines als Formaloperand vorgegebenen Zählers (Parameterart: T, Z).
SAR = ☐	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als Ausschaltverzögerung bzw. Rückwärtszählen eines als Formaloperand vorgegebenen Zählers (Parameterart: T, Z).
↑	Formaloperand einsetzen.
	Als Aktualoperand sind Zeiten und Zähler zugelassen; Ausnahme: Bei SI und SE nur Zeiten. Der Zeit- bzw. Zählerwert kann wie bei den Grundoperationen oder als Formaloperand wie folgt vorgegeben werden: Setzen des Zeit- bzw. Zählwerts mit dem im BCD-Code vorliegenden Wert des als Formaloperanden vorgegebenen Operanden EW, AW, MW, DW (Parameterart: E; Parametertyp: W) bzw. als Konstante (Parameterart: D; Parametertyp: KT, KZ).

Beispiele

Funktionsbausteinaufruf	Programm im Funktionsbaustein	ausgeführtes Programm
:SPA FB203 NAME :BEISPIEL ANNA : E 10.3 BERT : T 17 HANS : A 18.4	:U =ANNA :L KT 010.2 :SSV =BERT :U =BERT := =HANS	:U E 10.3 :L KT 010.2 :SS T 17 :U T 17 := A 18.4
:SPA FB204 NAME :BEISPIEL MAXI : E 10.5 IRMA : E 10.6 EVA : E 10.7 DORA : Z 15 EMMA : M 58.3	:U =MAXI :SSV =DORA :U =IRMA :SAR =DORA :U =EVA :L KZ100 :SVZ =DORA :UN =DORA := =EMMA	:U E 10.5 :ZV Z 15 :U E 10.6 :ZR Z 15 :U E 10.7 :L KZ 100 :S Z 15 :UN Z 15 := M 58.3
:SPA FB205 NAME :BEISPIEL KURT : E 10.4 CARL : T 18 EGON : EW20 MAUS : M 100.7	:U =KURT :L =EGON :SVZ =CARL :U =CARL := =MAUS	:U E 10.4 :L EW20 :SV T 18 :U T 18 := M 100.7

Bit-Test-Funktionen

Operation		Beschreibung
P		Prüfe Bit auf Signalzustand ,,1''
PN		Prüfe Bit auf Signalzustand ,,0''
SU		Setze Bit unbedingt
RU		Rücksetze Bit unbedingt
	Z 0.0 bis 255.15	eines Zeitwortes
	T 0.0 bis 255.15	eines Zählwortes
	DW0.0 bis 255.15	eines Datenwortes
	E 0.0 bis 127.7	eines Eingangs im PAE
	A 0.0 bis 127.7	eines Ausgangs im PAA
	M 0.0 bis 255.7	eines Merkers
	BS 0.0 bis 255.15	Bereichssystem 1)
	BT 0.0 bis 255.15	Bereichssystemerweiterung 1)
	BA 0.0 bis 255.15	Bereichsanschaltung
	BB 0.0 bis 255.15	Bereichs-Anschaltungserweiterung

Die Operationen ,,P'' und ,,PN'' sind Abfragen. Sie fragen ein Bit des nachfolgend angegebenen Operanden ab und setzen dann das Verknüpfungsergebnis ein, unabhängig von vorherigen Abfragen und vom vorherigen Stand.

Operation	Signalzustand des Bits im angegebenen Operanden	Verknüpfungsergebnis
P	0	0
	1	1
PN	0	1
	1	0

Das auf diese Weise gebildete Verknüpfungsergebnis kann weiterverknüpft werden. Eine Bit-Test-Operation muß jedoch immer am Beginn einer Verknüpfung stehen.

Beispiel

Der Signalzustand des 10. Bits des Datenworts 205 wird mit dem Signalzustand des Eingangs E 13.7 nach UND verknüpft.

```
P  DW  205.10
U  E    13.7
=  M   210.3
```

Die Operationen ,,SU'' und ,,RU'' werden unabhängig vom Verknüpfungsergebnis ausgeführt. Nach der Bearbeitung dieser Operation ist das adressierte Bit im angegebenen Operanden auf Signalzustand ,,1'' (bei SU) oder auf Signalzustand ,,0'' (bei RU) gesetzt.

Lade- und Transferfunktionen

Operation	Beschreibung
L = ☐	Laden eines Formaloperanden Der Wert des als Formaloperanden vorgegebenen Operanden wird in den Akku geladen (Parameterart: E, A; Parametertyp: BY, W, D).
LC = ☐	Laden codiert eines Formaloperanden Der Wert der als Formaloperand vorgegebenen Zeit- oder Zählzelle wird BCD-codiert in den Akku geladen (Parameter: T, Z).
LW = ☐	Laden des Bitmusters eines Formaloperanden Das Bitmuster des Formaloperanden wird in den Akku geladen (Parameterart: D; Parametertyp: KF, KH, KM, KY, KC, KT, KZ).
LD = ☐	Laden des Bitmusters eines Formaloperanden Das Bitmuster des Formaloperanden wird in den Akku geladen (Parameterart: D; Parametertyp: KG).
T = ☐	Transferieren zu einem Formaloperanden Der Akkumulatorinhalt wird zu dem als Formaloperand vorgegebenen Operanden transferiert (Parameterart: E, A; Parametertyp: BY, W, D).
	Formaloperanden einsetzen.
	Als Aktualoperand sind die den Grundoperationen entsprechenden Operanden zugelassen. Bei LW ist ein Datum in Form eines Binärmusters, Hexamusters, 2 byteweise Betragszahlen, Zeichen, Festpunktzahl, Zeitwerte und Zählwerte zugelassen. Bei LD ist eine Gleitkommazahl als Datum zugelassen
L ☐	Laden eines Wortes in den Akku 1-L aus dem Bereich
B S 0 bis 255	Systemdaten
B T 0 bis 255	Systemdatenerweiterung
B A 0 bis 255	Anschaltungsdatum
B B 0 bis 255	Anschaltungsdatenerweiterung
T ☐	Transferieren des Akku 1-L zu einem Wort des Bereichs
B A 0 bis 255	Anschaltungsdaten
B B 0 bis 255	Anschaltungsdatenerweiterung

Beispiel

Funktionsbausteinaufruf	Programm im Funktionsbaustein	ausgeführtes Programm
:SPA FB206 NAME :VERGL. Z1 : KH7F0A Z2 : DW20	:LW =Z1 :L =Z2 :!=F	:L KH7F0A :L DW20 :!=F

6 STEP-5-Befehlsvorrat mit Programmierbeispielen

6.3 Ergänzender Operationsvorrat

Rechenfunktionen

Operation	Beschreibung
ENT	Eintrag von Daten in die Arithmetikspeicher. Der Befehl ENT bewirkt das Laden der Akkumulatoren Akku 3 und Akku 4, die bei arithmetischen Operationen mitverwendet werden: Akku 4: = Akku 3 Akku 3: = Akku 2

Beispiel
Folgender Bruch soll ausgerechnet werden: (30 + 3 x 4)/6 = 7

	Akku 1	Akku 2	Akku 3	Akku 4
Vorbelegung der Akkus vor der arithmetischen Operationskette	a	b	c	d
L KF 30	30	a	c	d
L KF 3	3	30	c	d
ENT	3	30	30	c
L KF 4	4	3	30	c
* F	12	30	c	c
+ F	42	c	c	c
L KF 6	6	42	c	c
/ F	7	c	c	c

Digitalverknüpfungen

Operation	Beschreibung
UW ☐	UND-Verknüpfung digital von Akku 1-L und Akku 2-L
OW ☐	ODER-Verknüpfung digital von Akku 1-L und Akku 2-L
XOW ☐	Exlusiv-ODER-Verknüpfung digital von Akku 1-L und Akku 2-L
	Durch zwei Ladeoperationen (s. auch Seite 28) können der Akku 1-L und Akku 2-L entsprechend den Operanden der Ladeoperationen geladen werden. Anschließend lassen sich die Inhalte beider Akkus digital verknüpfen. Beispiel: L EW 1 — Akku 1 → Akku 2; EW 1 → EW 1 L EW 2 — EW 2 → EW 1; EW 2 → EW 1 UW — UND-Verknüpfung von EW 2 und EW 1 → Akku 1-L

Organisatorische Funktionen

Sprungfunktionen

Das Sprungziel für unbedingte und bedingte Sprünge wird symbolisch angegeben (maximal 4 Zeichen). Dabei ist der Symbolparameter des Sprungbefehls identisch mit der Symboladresse der anzuspringenden Anweisung. Bei der Programmierung muß berücksichtigt werden, daß die absolute Sprungdistanz nicht mehr als ± 127 Wörter umfaßt und eine STEP-5-Anweisung aus mehr als einem Wort bestehen kann. Sprünge dürfen nur innerhalb eines Bausteins durchgeführt werden; Sprünge über Segmente hinweg sind nicht zulässig.

Sprungfunktionen

Operation	Beschreibung
SPA = ☐	Sprung unbedingt Der unbedingte Sprung wird unabhängig von Bedingungen ausgeführt.
SPB = ☐	Sprung bedingt Der bedingte Sprung wird ausgeführt, wenn das Verknüpfungsergebnis „1" ist. Bei Verknüpfungsergebnis „0" wird die Anweisung nicht ausgeführt und das Verknüpfungsergebnis auf „1" gesetzt.
SPZ = ☐	Sprungbedingung: ANZ1, ANZ0 Der Sprung wird ausgeführt, wenn ANZ1 = 0 ANZ0 = ist. Sonst wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.
SPN = ☐	Sprungbedingung: ANZ1, ANZ0 Der Sprung wird ausgeführt, wenn ANZ1 ≈ ANZ0 ist. Sonst wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.
SPP = ☐	Sprungbedingung: ANZ1, ANZ0 Der Sprung wird ausgeführt, wenn ANZ1 = 1 und ANZ0 = 0 ist. Sonst wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.
SPM = ☐	Der Sprung wird nicht ausgeführt, wenn ANZ1 = 0 und ANZ0 = 1 ist. Sonst wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.
SPO = ☐	Sprung bei Überlauf (Overflow) Der Sprung wird ausgeführt, wenn OVER = 1 ist. Liegt kein Überlauf vor (OVER = 0), wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert. Ein Überlauf entsteht, wenn bei gegebener Zahlendarstellung der zulässige Bereich durch eine arithmetische Operation überschritten wird.
SPS = ☐	Der Sprung wird ausgeführt, wenn „Überlauf gespeichert" gesetzt ist (OS = 1). Sonst (OS = 0) wird der Sprung nicht ausgeführt. „Überlauf gespeichert" wird bei arithmetischen Operationen im Falle eines Überlaufes gesetzt und bleibt solange gespeichert bis die arithmetische Operationskette unterbrochen wird. Symboladresse einsetzen (max. 4 Zeichen).

Hinweis:
Sprunganweisung und Sprungziel müssen in einem Segment liegen. Pro Segment ist nur eine Symboladresse für Sprungziele zugelassen.

Beispiel

```
      :SPA =FORT
      :
      :
      :
FORT  :U   -STOP
      :U   -END
      :
      :
ZIEL  :O   A 7.3
      :O   M 16.6
      :
      :
      :O   -BETR
      :SPB =ZIEL
```

Schiebefunktionen

Operation	Beschreibung
SLW 0 bis 15	Schieben links (von rechts werden Nullen nachgezogen)
SRW 0 bis 15	Schieben rechts (von links werden Nullen nachgezogen)
SVW 0 bis 15	Schieben rechts mit Vorzeichen (von links wird das Vorzeichen nachgezogen)
SLD 0 bis 32	Schieben links Doppelwort (von rechts werden Nullen nachgezogen)
SVD 0 bis 32	Schieben rechts mit Vorzeichen Doppelwort (von links wird das Vorzeichen nachgezogen)
RLD 0 bis 32	Rotieren links
RRD 0 bis 32	Rotieren rechts
	Bei Befehlen SLW, SRW SVW ist Akku 1-L und bei den restlichen Befehlen Akku 1 an der Ausführung beteiligt. Parameterteil dieser Befehle gibt an, um wieviel Stellen der Akku-Inhalt geschoben bzw. rotiert wird.

Die Schiebefunktionen werden unabhängig von Bedingungen ausgeführt. Das zuletzt hinausgeschobene Bit kann mit Sprungfunktionen abgefragt werden.
Mit SPZ kann gesprungen werden, wenn das Bit 0 ist und mit SPN oder SPB, wenn das Bit 1 ist.

Beispiel

STEP-5-Programm: Inhalt der Datenwörter

```
:L   DW52        H = 14AF
:SL     4
:I   DW53        H = 4AF0
```

Beispiel

```
L KHA 956  Akku 2:   = Akku 1
           Akku 1-H: = 0
           Akku 1-L: = A956H
RLD 8      Akku 1-H: = 00A94
           Akku 1-L: = 5600H
```

Umwandlungsfunktionen

Operation	Bedeutung
KEW	1-er Komplement von Akku 1-L
KZW	2-er Komplement von Akku 1-L
KZD	2-er Komplement von Akku 1
DEF	Dezimal- Festpunkt Wandlung
DUF	Festpunkt- Dezimal Wandlung
DED	Dezimal- Festpunkt Doppelwort Wandlung
DUD	Festpunkt Doppelwort- Dezimal Wandlung
FDG	Festpunkt Doppelwort- Gleitpunkt Wandlung
GFD	Gleitpunkt- Festpunkt Doppelwort Wandlung

Beispiele

Der Inhalt des Datenworts 64 soll Bit für Bit invertiert werden und in Datenwort 78 abgelegt werden.

STEP-5-Programm: Belegung der Datenwörter:

```
:L   DW64        BM = 0011111001011011
:KEW
:T   DW78        BM = 1100000110100100
```

Der Inhalt des Datenworts 207 ist als Festpunktzahl zu interpretieren und mit umgekehrtem Vorzeichen im Datenwort 51 abzulegen.

STEP-5-Programm: Belegung der Datenwörter:

```
:L   DW207       F: + 51
:KZW
:T   DW51        F: – 51
```

Dekrementieren/Inkrementieren

Operation	Beschreibung
D 1 bis 255	Dekrementieren
I 1 bis 255	Inkrementieren
	Der Akkumulatorinhalt 1 wird um die im Parameter angegebene Zahl dekrementiert bzw. inkrementiert. Die Operationsausführung ist unabhängig von Bedienungen. Sie beschränkt sich auf das rechte Byte (ohne Übertrag).

Beispiel

STEP-5-Programm: Belegung der Datenwörter:

```
:L   DW7         H = 1010
:I     16
:T   DW8         H = 1020
:D     33
:T   DW9         H = 10FF
```

Bearbeitungsfunktionen

Operation	Bedeutung
B = ☐	Bearbeite Formaloperand (Parameterart: B)
	Formaloperand einsetzen
	Nur die Operationen A DB SPA P B SPA S B SPA F B können substituiert werden (siehe „Typ des Bausteinparameters und zugelassener Aktualoperand" Seite 10)
	Bearbeite Datenwort Die nachfolgend angegebene Operation wird mit dem im Datenwort angegebenen Parameter kombiniert und ausgeführt.
B DW0 bis 254 (Operation)	Bearbeite Datenwort Die nachfolgend angegebene Operation wird mit dem im Datenwort angegebenen Parameter kombiniert und ausgeführt.
B MW0 bis 254 (Operation)	Bearbeite Merkerwort Die nachfolgend angegebene Operation wird mit dem im Merkerwort angegebenen Parameter kombiniert und ausgeführt.

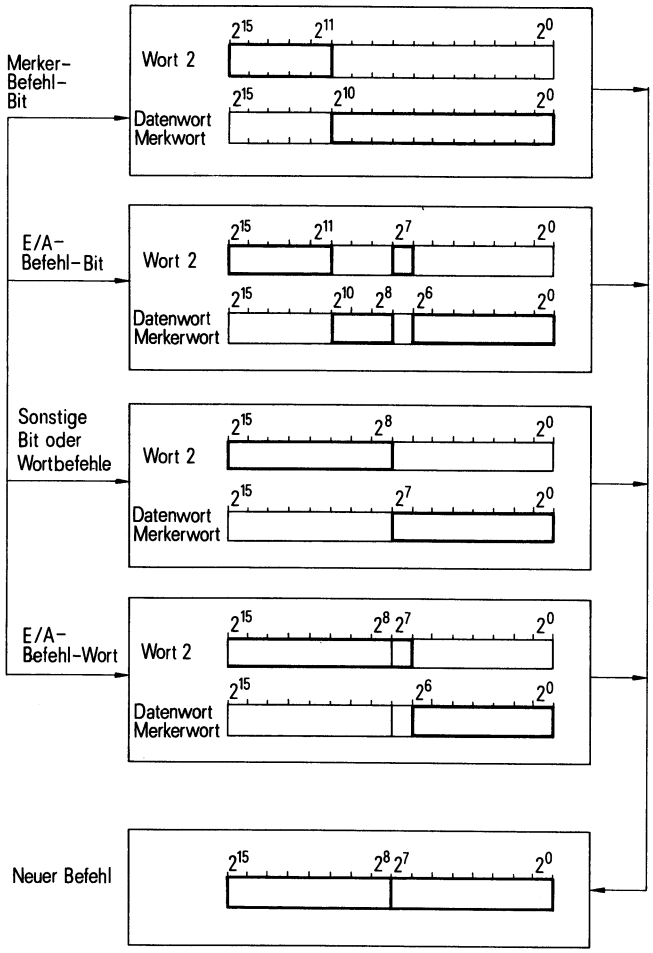
Mit BDW bzw. BMW dürfen alle Operationen kombiniert werden, außer den unten angegebenen:

- A DBO
- alle Zweiwort- bzw. Dreiwort-Befehle
- Operationen mit Formaloperanden in Funktionsbausteinen
- SPA OBxx, SPB OBxx

Das Programmiergerät 670 prüft die Zulässigkeit der Kombinationen nicht ab.

6.3 Ergänzender Operationsvorrat

Im Datenwort bzw. Merkerwort steht der Parameterteil des auszuführenden Befehls. Die dem BDW/BMW-Befehl unmittelbar folgende Operation (Wort 2) wird mit den Daten Merker-Wort wie unten angegeben kombiniert; woraus der auszuführende (neue) Befehl resultiert:



Beispiel: Bearbeite Formaloperand

Funktionsbausteinaufruf	Programm im Funktionsbaustein	ausgeführtes Programm
:SPA FB207		
NAME :KONTROLL		
DB-Q : DB1	:B =DB-Q	:A DB1
QUEL : DW100	:L =QUEL	:L DW100
DB-Z : DB33	:B =DB-Z	:A DB33
ZIEL : DW107	:T =ZIEL	:T DW107

Beispiel: Bearbeite Datenwort

Es sollen die Inhalte der Datenwörter DW 20 bis DW 100 auf Signalzustand „0" gesetzt werden. Das „Indexregister" für den Parameter der Datenwörter ist DW 0.

```
      :L   KF 20      Versorgung des „Indexregisters"
      :T   DW1
M1    :L   KF 0       Rücksetzen
      :B   DW1
      :T   DW0
      :L   DW1        Erhöhen des Indexregisters
      :L   KF 1
      :+F
      :T   DW1
      :L   KF 100
      :<=F
      :SPB =M1        Sprung, wenn Index im Bereich liegt
      ...             weiteres STEP-5-Programm
```

Befehlsausgabe sperren/freigeben

Operation	Beschreibung
BAS	Befehlsausgabe sperren
BAF	Befehlsausgabe freigeben
	Die Operationsausführung ist abhängig vom Verknüpfungsergebnis. Nach Ausführung von BAS wird das Prozeßabbild der Ausgänge nicht mehr beeinflußt. BAF hebt die Wirkung von BAS wieder auf.

„Befehlsausgabe sperren/freigeben" kann z. B. eingesetzt werden, um eine Ablaufkette auf einen bestimmten Schritt nachzuführen, ohne die Ausgänge der durchlaufenen Schritte zu setzen oder rückzusetzen.

Befehlsausgabe sperren/freigeben

Operation	Bearbeitung
AS	Prozeßalarmbearbeitung sperren
AF	Prozeßalarmbearbeitung freigeben
AAS	Anforderungsalarmbearbeitung sperren
AAF	Anforderungsalarmbearbeitung freigeben
AFS	Adressierfehler-Bearbeitung sperren
AFF	Adressierfehler-Bearbeitung freigeben

„Alarme sperren/freigeben" kann z. B. angewendet werden, wenn bei einer zeitgesteuerten Bearbeitung die alarmgesteuerte Bearbeitung unterdrückt werden soll (siehe „Programmierung der alarmgesteuerten Bearbeitung" Seite 14).

7.1 Allgemeines

Jede Darstellungsart der Programmiersprache STEP 5 beinhaltet Grenzen. Daraus ergibt sich, daß ein in AWL geschriebener Programmbaustein nicht ohne weiteres in KOP oder FUP ausgegeben werden kann, und daß darüber hinaus die graphischen Darstellungsarten KOP und FUP gegebenenfalls nicht vollständig kompatibel sein können.

Wurde das Programm in KOP oder FUP eingegeben, so ist es grundsätzlich in AWL rückübersetzbar.

Ziel dieser Abschnitte ist die Aufstellung einiger Regeln. deren Einhaltung eine vollständige Kompatibilität der drei Darstellungsarten gewährleistet.

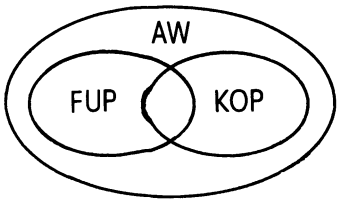


Bild 23
Umfang bzw. Begrenzungen der Darstellungsarten der Programmiersprache STEP 5.

Diese Regeln gliedern sich wie folgt:

- Kompatibilitätsregeln bei graphischer Programmeingabe (KOP, FUP).

Die Einhaltung dieser Regeln ermöglicht, bei Eingabe in einer **graphischen** Darstellungsart, die entsprechende Ausgabe in den übrigen Darstellungsarten.

- Kompatibilitätsregeln bei Programmeingabe in Anweisungsliste.

Die Einhaltung dieser Regeln gewährleistet bei Eingabe in Form der **Anweisungsliste** die entsprechende Ausgabe in den übrigen Darstellungsarten.

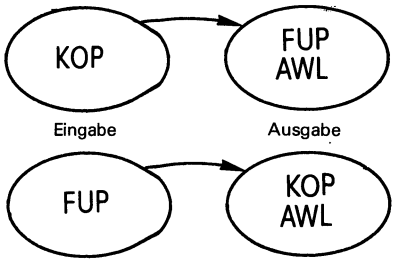


Bild 24
Graphische Eingabe

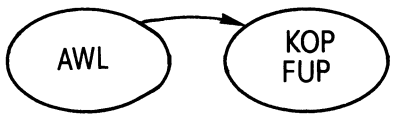
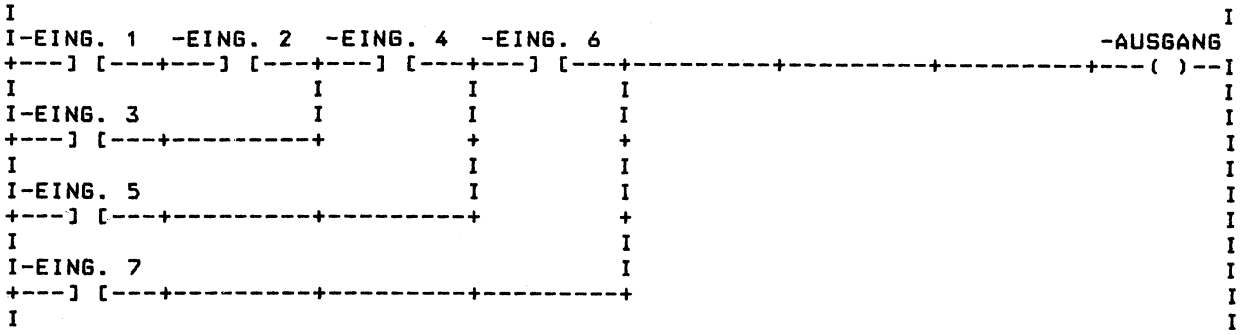


Bild 25
Eingabe in Anweisungsliste

7.2 Kompatibilitätsregeln bei graphischer Programmeingabe (KOP, FUP)

Eine zu große Schachtelung kann im FUP zum Überschreiten der Bildgrenzen (8 Ebenen) führen.

KOP



FUP

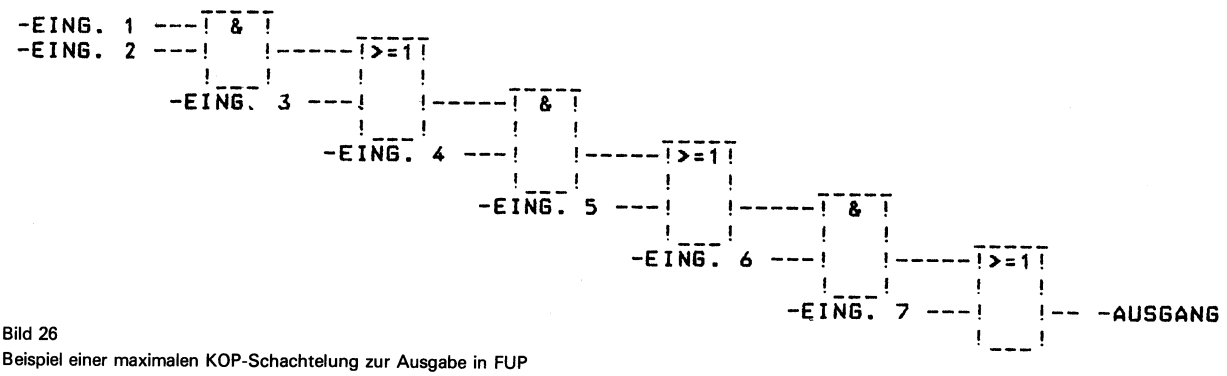
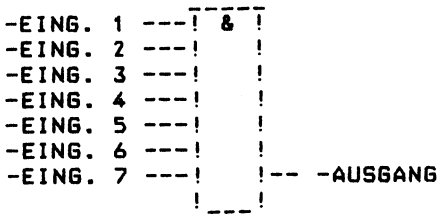


Bild 26
Beispiel einer maximalen KOP-Schachtelung zur Ausgabe in FUP

Eingabe in FUP: Ausgabe in KOP und AWL

Regel 1: Bildgrenzen für KOP nicht überschreiten:
Eine zu große Anzahl von Eingaben an einem FUP-Kasten führt zum Überschreiten der KOP-Bildgrenze.

FUP



KOP

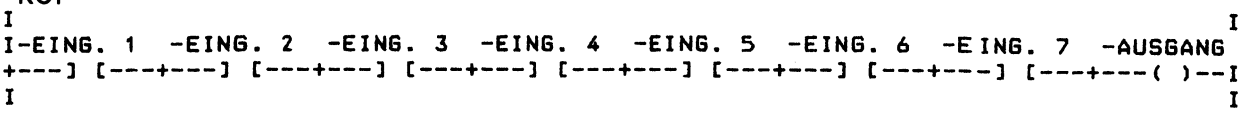


Bild 27
Beispiel eines maximalen UND-Kastenausbaus zur Ausgabe in KOP

Regel 2: Der Ausgang eines komplexen Gliedes (Speicher-, Vergleicher-, Zeit- oder Zählglied) darf nicht mit ODER weiterverknüpft werden!

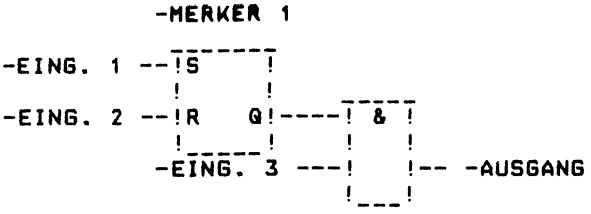


Bild 28
Nur UND-Kasten sind nach einem komplexen Glied zulässig.

Regel 3: Konnektoren

- Konnektoren sind beim ODER-Kasten immer erlaubt
- Konnektoren sind beim UND-Kasten nur am ersten Eingang erlaubt.

(Konnektoren sind Zwischenmerker, um immer wiederkehrende Verknüpfungen einzusparen)

\# Konnektor erlaubt

X Konnektor nicht erlaubt

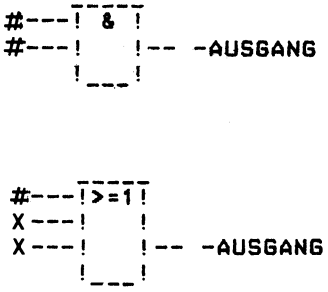


Bild 29
Beispiel für die Zulässigkeit der Konnektoren bei ODER- und UND-Kästen

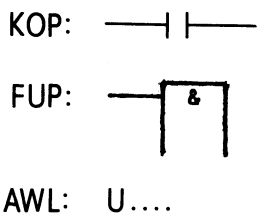
7.3 Kompatibilitätsregeln bei Programmeingabe in Anweisungsliste

Regel 1: UND-Verknüpfung:
(Abfrage des Signalzustandes und der Verknüpfung nach UND).

KOP: Kontakt in Reihe

FUP: Eingang eines UND-Kastens

AWL: Anweisung U . . .



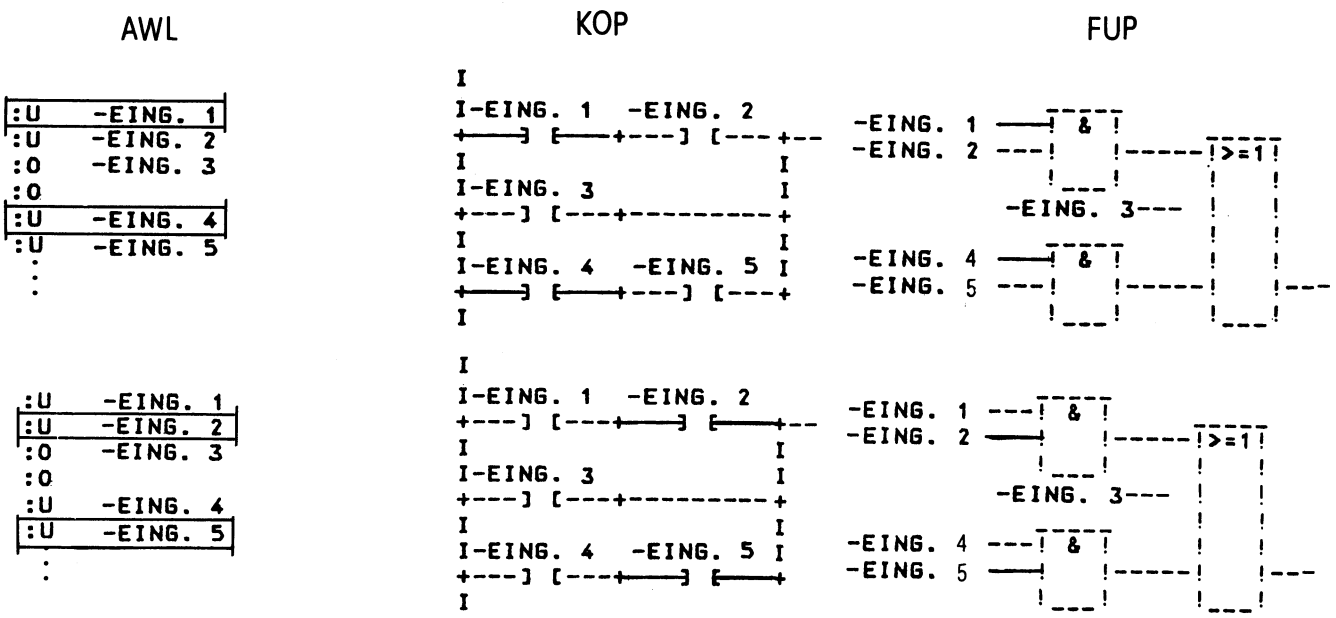


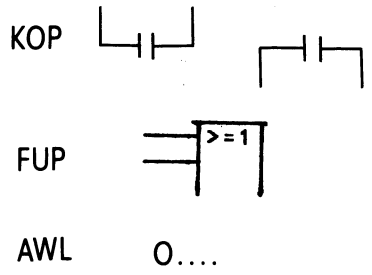
Bild 30
Erläuterungen zur Regel UND-Verknüpfung

Regel 2: ODER-Verknüpfung
(Abfrage des Signalzustandes und der Verknüpfung nach ODER).

KOP: nur ein Kontakt in einem Parallelzweig

FUP: Eingang eines ODER-Kastens

AWL: Anweisung O . . .



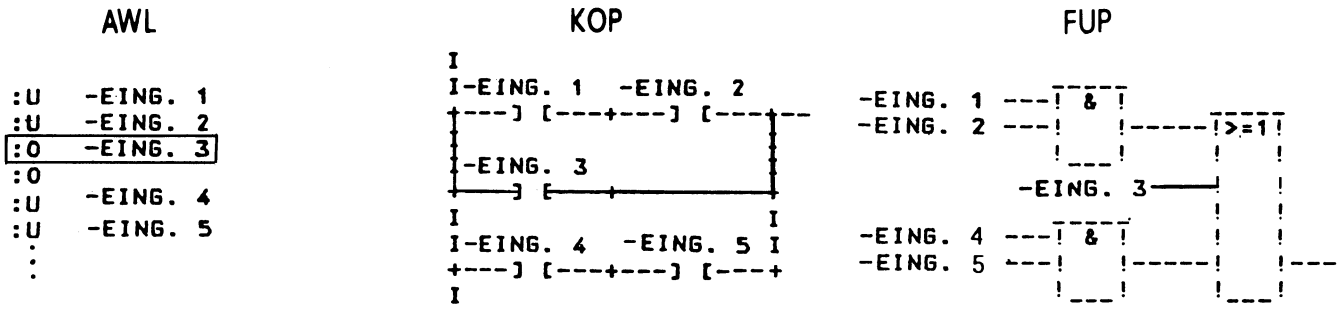


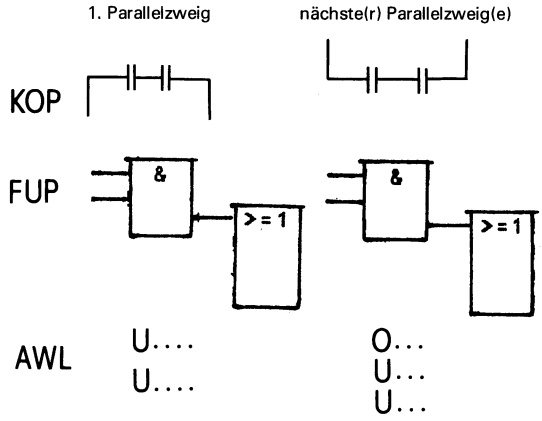
Bild 31
Erläuterungen zur Regel ODER-Verknüpfung

Regel 3: UND-vor-ODER-Verknüpfung
(ODER-Verknüpfung vor UND-Funktionen)

KOP: mehrere Kontakte in einem Parallelzweig

FUP: UND-Kasten vor ODER-Kasten

AWL: Anweisungen O . . .
U . . .
U . . .



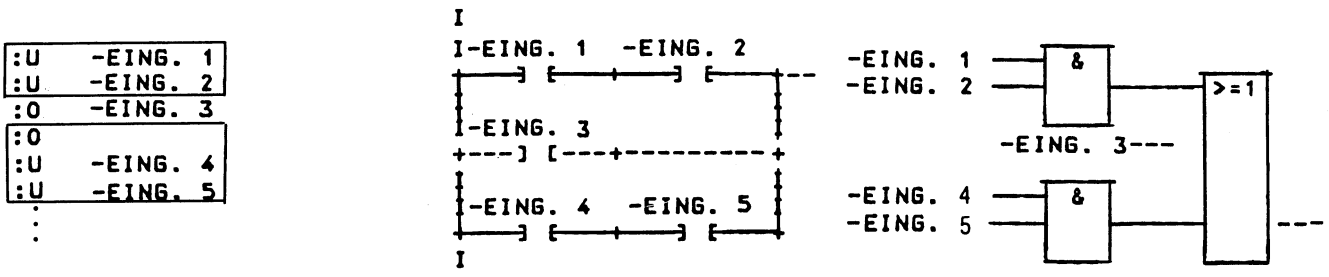
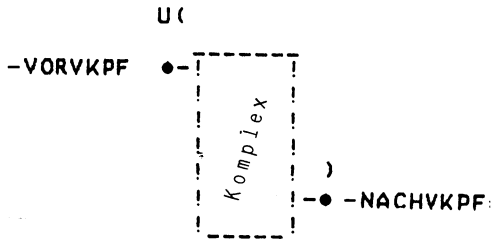


Bild 32
Erläuterungen zur Regel UND-vor-ODER-Verknüpfung

7.3 Kompatibilitätsregeln bei Programmiereingabe in Anweisungsliste

Regel 4: Klammerung

In dieser Regel wird die Klammerung von komplexen, in sich abgeschlossenen, binären Verknüpfungen oder von komplexen Gliedern mit Vor- oder Nachverknüpfungen behandelt.



a) Komplexe binäre Verknüpfung

Zu dieser Verknüpfungs-Klasse gehört die ODER-vor-UND-Verknüpfung, deren Regeln wie folgt lauten:

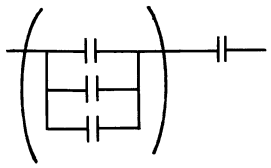
- UND-Verknüpfung vor ODER-Funktionen
  KOP: Parallel-Kontakte in Serie weiterschalten

FUP: ODER-Kasten vor UND-Kasten

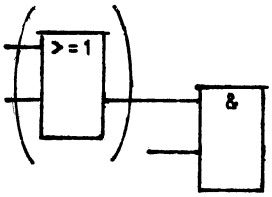
AWL: Anweisungen U(
ODER-VKPF
)
.
.

Die ODER-vor-UND-Verknüpfungen stellen eine Untermenge der komplexen binären Verknüpfungen dar, wobei zwei parallele Kontakte die einfachste Verknüpfung bilden.

KOP



FUP



AWL

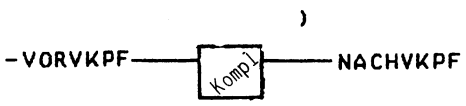
U(
O...
O....
O...
)
U...

Bild 33
Erläuterungen zur Regel ODER-vor-UND-Verknüpfung

AWL

:U -VORVKPF
:U(
:
: Komplex
:)
:U -NACHVKPF

KOP



FUP

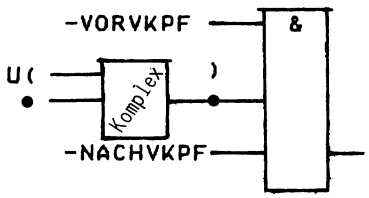


Bild 34
Erläuterungen zur Klammerung von komplexen binären Funktionen

b) Komplexe Glieder (Speicher-, Zeit-, Vergleichs- und Zählfunktionen).

Für komplexe Glieder müssen folgende Regeln eingehalten werden:

- Keine Nachverknüpfung vorhanden: keine Klammerung
- Nachverknüpfung UND: U (. . .) . . .
- Nachverknüpfung ODER: O (. . .) . . .
  (nur für FUP, bei KOP nicht erlaubt)
- Ein komplexes Glied kann keine Vorverknüpfung haben.

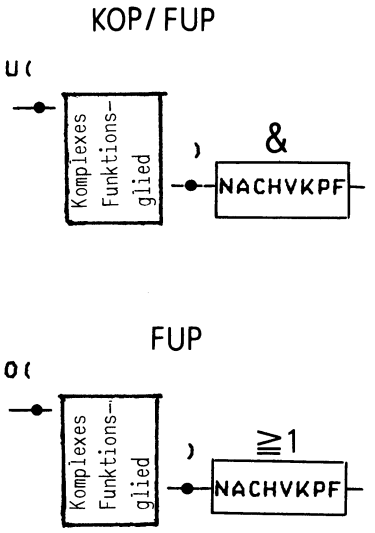


Bild 35
Erläuterungen zur Klammerung von komplexen Gliedern

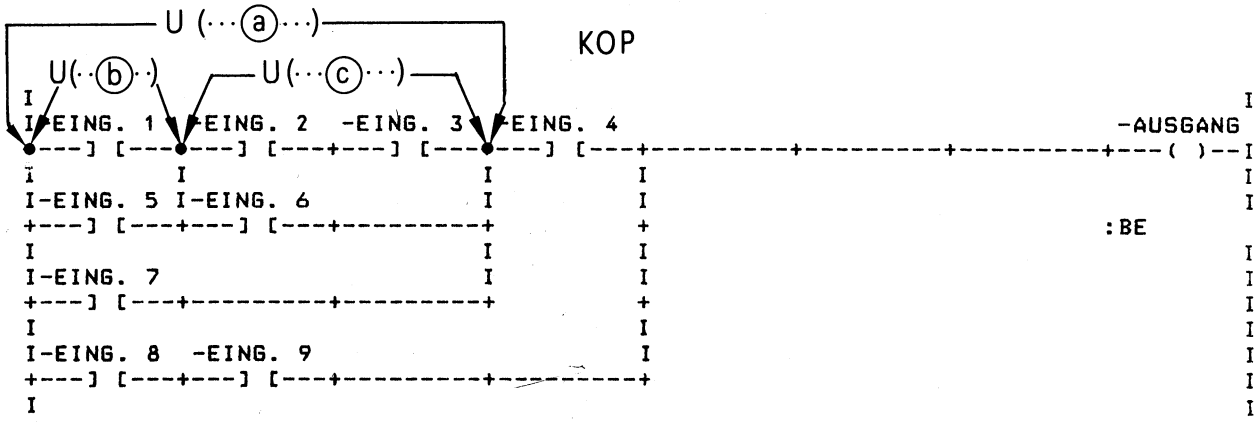
Beispiel 1: KOP/AWL

Fall 1: UND (Kontakt in Reihe)
Fall 2: ODER (nur 1 Kontakt in einem Parallelzweig)
Fall 3: UND-vor-ODER (mehrere Kontakte in einem Parellelzweig)
Fall 4: ODER-vor-UND (Klammerung)

AWL

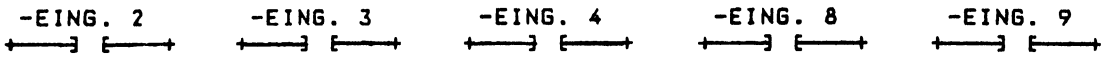
```
      :U(
      :U(
(b)   :O    -EING. 1
      :O    -EING. 5
      :)
(a)   :U(
      :U    -EING. 2
(c)   :U    -EING. 3
      :O    -EING. 6
      :)
      :O    -EING. 7
      :)
      :U    -EING. 4
      :O
      :U    -EING. 8
      :U    -EING. 9
      :=    -AUSGANG
```

KOP

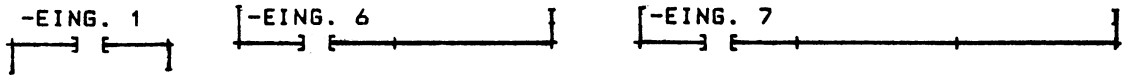


7.3 Kompatibilitätsregeln bei Programmiereingabe in Anweisungsliste

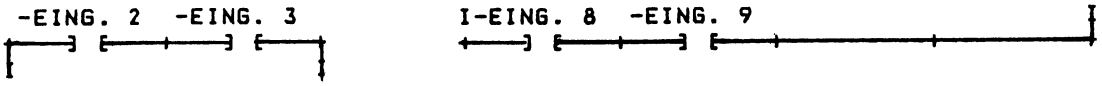
Fall 1:



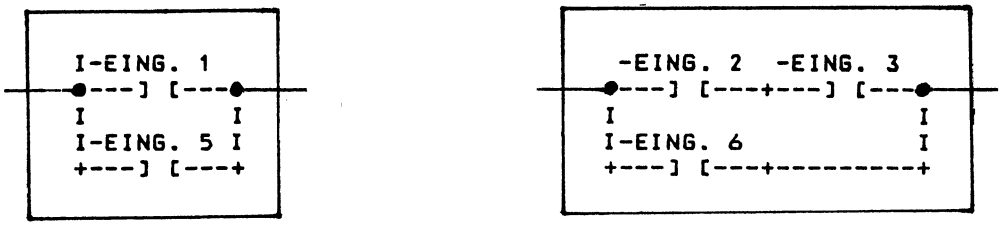
Fall 2:



Fall 3:



Fall 4:



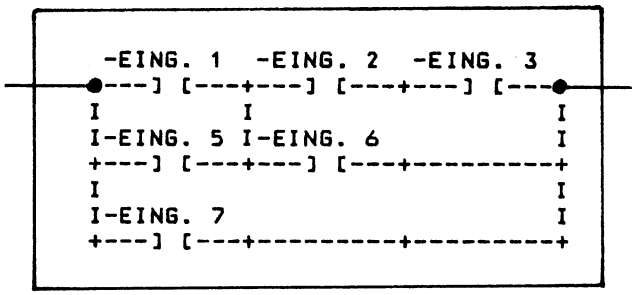


Bild 36
Beispiel 1: KOP/AWL

Jeder unbeschaltete Ein- oder Ausgang muß mit NOP O versorgt werden.

Ausnahme: S, TW bei Zeiten und S und ZW bei Zählern müssen stets gemeinsam beschaltet sein.

Bei der Programmierung in AWL sind die komplexen Glieder in derselben Reihenfolge zu programmieren, wie sie am Bildschirm in graphischer Darstellungsart parametriert werden.

Ausnahme: Zeit- und Zählwert. Der entsprechende Wert muß zuerst im Akkumulator durch einen Ladebefehl hinterlegt werden.

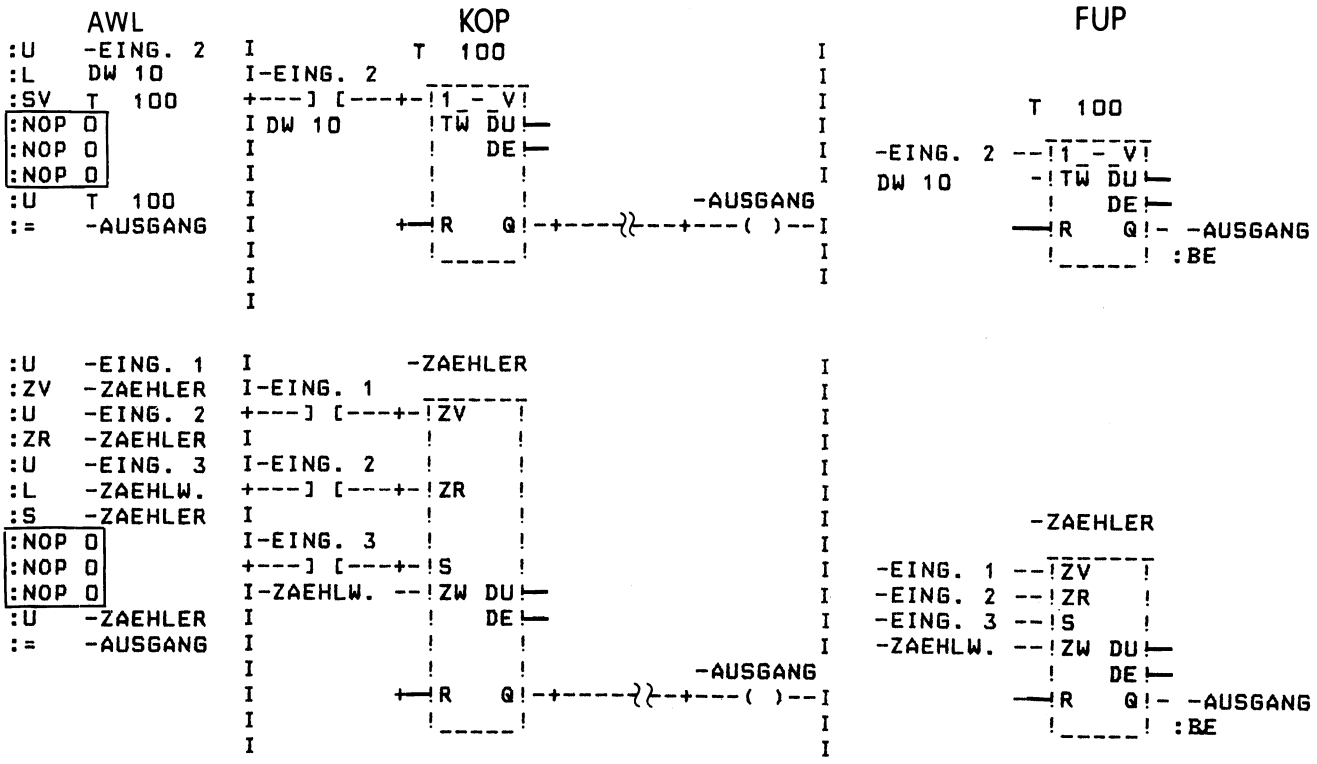


Bild 37
Beispiel zur Versorgung unbeschalteter Ein- und Ausgänge

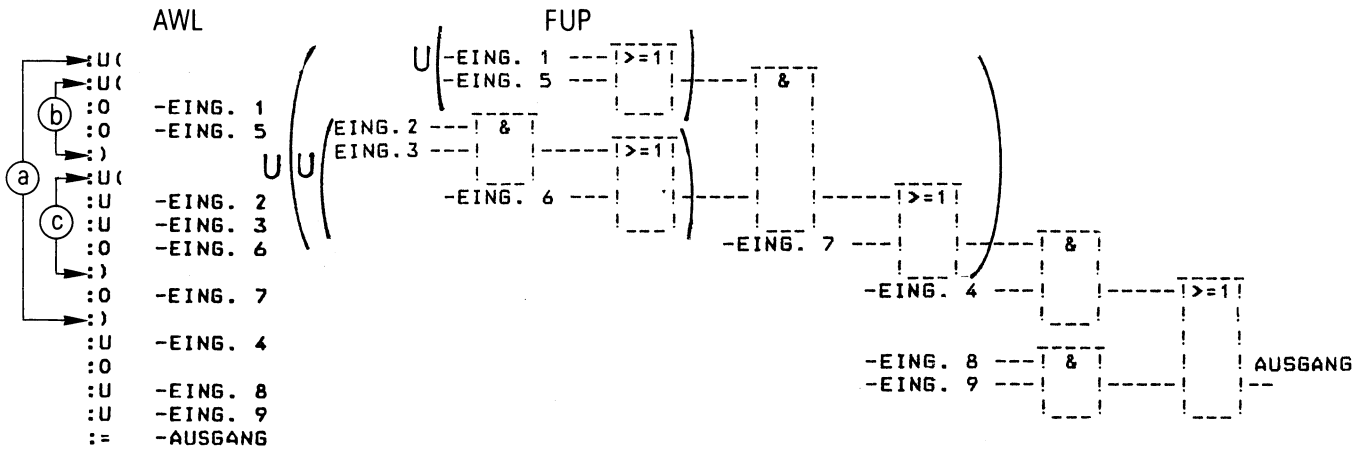
Wichtiger Hinweis: Pro Segment ist nur ein komplexes Funktionsglied zulässig.

Die folgenden Beispiele zeigen die vier vorgeführten Fälle in einer komplexen binären Verknüpfung einmal in den Darstellungsarten KOP und AWL und einmal in FUP und AWL.

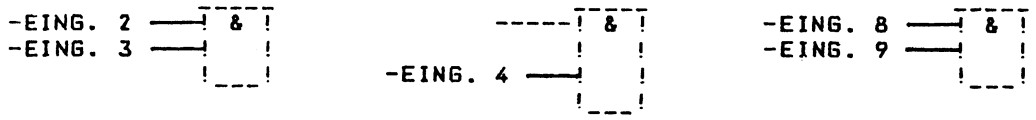
7.3 Kompatibilitätsregeln bei Programmiereingabe in Anweisungsliste

Beispiel 2: FUP/AWL

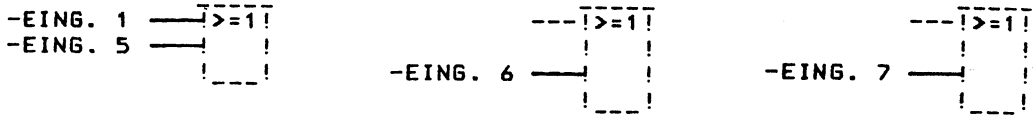
Fall 1: UND (Eingang eines UND-Kastens)
Fall 2: ODER (Eingang eines ODER-Kastens)
Fall 3: UND-vor-ODER (UND-Kasten vor ODER-Kasten)
Fall 4: ODER-vor-UND (ODER-Kasten vor UND-Kasten)



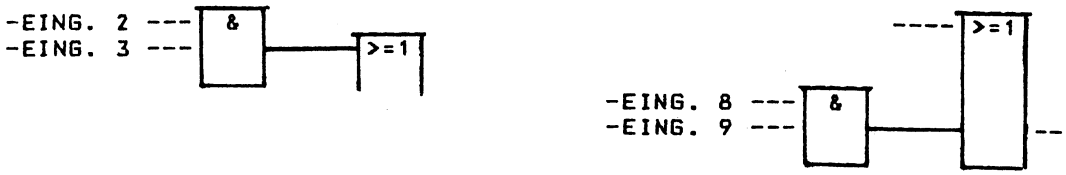
Fall 1:



Fall 2:



Fall 3:



Fall 4:

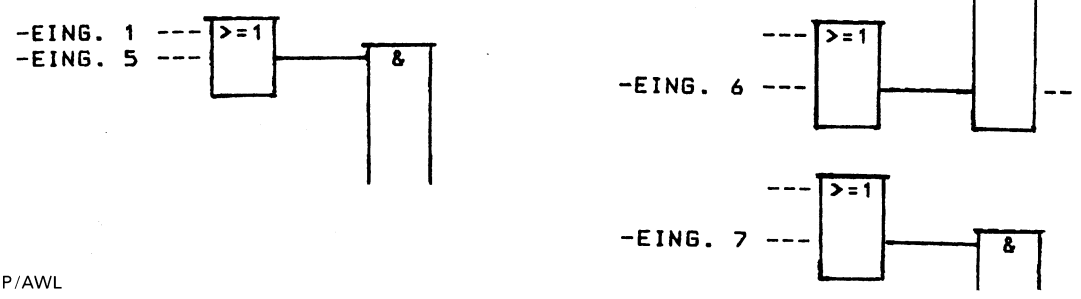


Bild 38
Beispiel 2: FUP/AWL

Regel 5: Konnektoren

Der Klarheit wegen werden die Regeln für Konnektoren getrennt für die Darstellungsarten KOP und FUP aufgelistet. Anschließend folgt ein gemeinsames Beispiel.

a) Konnektoren bei KOP

Ein Konnektor merkt sich das Verknüpfungsergebnis als Zwischenspeicher aus den Operationen, die vor ihm in der eigenen Stromschiene programmiert worden sind. Dabei gelten folgende Regeln:

Bild 39
Der Konnektor in KOP und AWL

- Konnektor in Reihe (in Serie mit anderen Konnektoren): Ein Konnektor wird in diesem Fall wie ein normaler Kontakt behandelt.

- Konnektor in einem Parallelzweig: Innerhalb eines Parallelzweiges wird ein Konnektor wie ein normaler Kontakt behandelt. Zusätzlich muß der gesamte Parallelzweig in eine Klammerung der Type O (. . .) eingeschlossen werden.

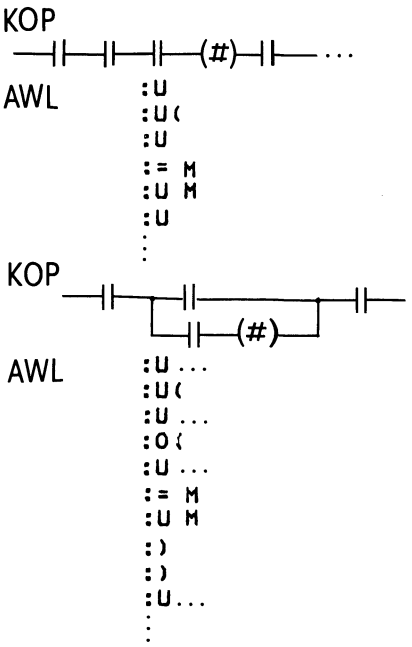


Bild 40
Konnektor-Regler für KOP

- Ein Konnektor darf nie unmittelbar nach der Stromschiene (Konnektor als erster Kontakt) oder direkt nach einer Eröffnung einer Stromschiene (Konnektor als erster Kontakt in einem Parallelzweig) stehen.

b) Konnektoren bei FUP

Ein Konnektor merkt sich das Verknüpfungsergebnis als Zwischenspeicher der gesamten binären Verknüpfung vor diesem Konnektor. Dabei gelten folgende Regeln:

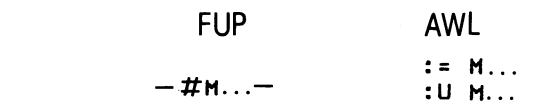


Bild 41
Der Konnektor in FUP und AWL

- Konnektor am ersten Eingang eines UND- bzw. ODER-Kastens: Der Konnektor wird ohne Klammerung abgesetzt.
- Konnektor nicht am ersten Eingang eines ODER-Kastens: Die gesamte binäre Verknüpfung vor dem Eingang wird in eine Klammerung des Typs O (. . .) eingeschlossen.
- Konnektor nicht am ersten Eingang eines UND-Kastens: Die gesamte binäre Verknüpfung vor dem Eingang wird in eine Klammerung des Typs U (. . .) eingeschlossen. (Nur bei FUP erlaubt, bei KOP graphisch nicht darstellbar)!

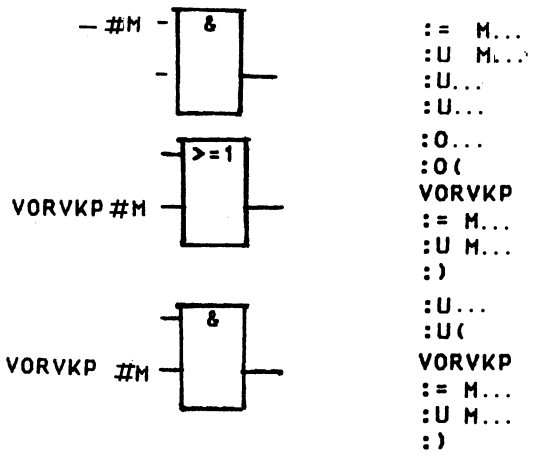
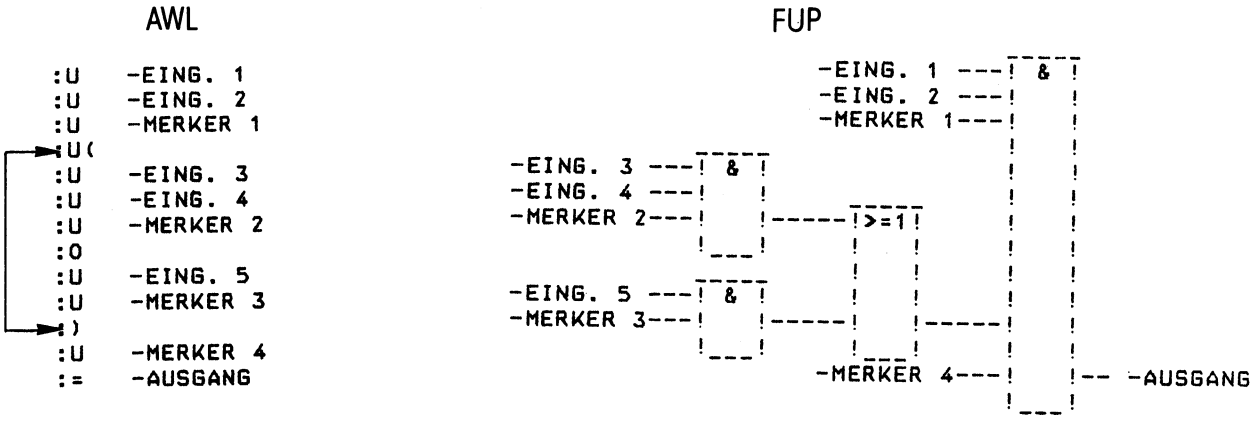


Bild 42
Konnektor-Regeln für FUP

Beispiele zu den Konnektoren
Es werden zwei Beispiele gegeben, eins ohne und eins mit Konnektoren.



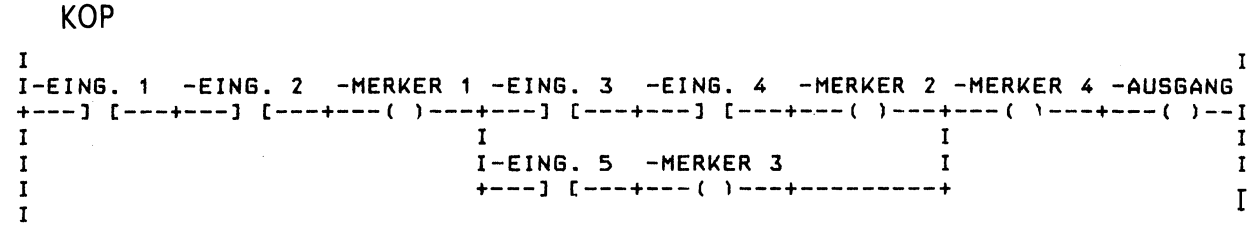


Bild 43
Beispiel ohne Konnektoren

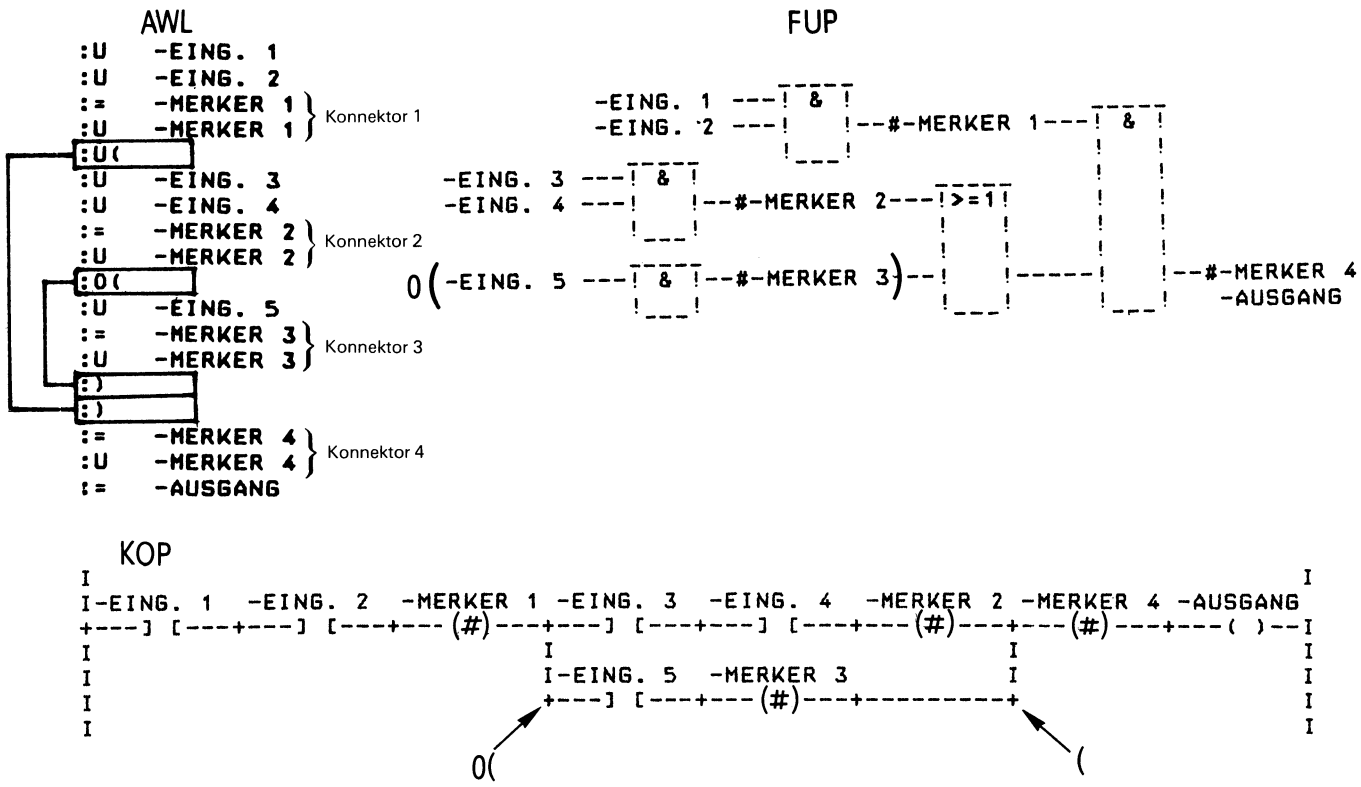


Bild 44
Beispiel mit Konnektoren

Konnektor 1: Verknüpfungsergebnis von U EING 1
U EING 2

Konnektor 2: Verknüpfungsergebnis von U EING 3
U EING 4

Konnektor 3: Verknüpfungsergebnis von U EING 5

Konnektor 4: Verknüpfung von der gesamten binären VKPF

8 Hinweise für die Abschätzung des erforderlichen Speicherplatzes

Das Automatisierungsgerät S5-150 ermöglicht einen Ausbau des Anwenderspeichers bis zu maximal 24 K Wörter. Unabhängig vom gesteckten Speicher sind davon für das Gerät intern 1,1 K Wörter zum Aufbau der Adreßlisten erforderlich, so daß der Anwender maximal 22,9 K Wörter zum Programmieren von Programm-, Funktions-, Schritt- und Datenbausteinen verwenden kann.

Der für ein Programm erforderliche Speicherplatz kann wie folgt grob abgeschätzt werden:

Anweisungen in Programm- und Schrittbausteinen:

Ⓐ $\Sigma(\text{PB-AW} + \text{SB-AW}) \approx 8 \times \Sigma(\text{E} + \text{A}) + 12$
(Σ Antriebe + Σ Ablaufketten + Σ Regelkreise)

Anweisungen in Funktionsbausteinen:

Ⓑ $\Sigma\text{FB-AW} \approx$ (Anzahl der FB) x 150

Datenwörter:

Ⓒ $\Sigma\text{DW} \approx 2 \times \Sigma$ Antriebe + Σ Schritte (Ablaufkette) + 10 x Σ Regelkreise + 10 x Σ Meldungen + 256 (für Protokollierung)

Anweisungen in Organisationsbausteinen:

Ⓓ $\Sigma\text{OB-AW}: \frac{8 \times \Sigma(\text{E} + \text{A})}{150}$

Erforderlicher Speicherplatz $\approx$ A + B + C + D

Grundfunktionen

Operation		Parameter	Bearbeitungszeit in µs
Verknüpfungsoperationen, binär Seite 65 und 66			
U	E	0,0 bis 127,7	2,1
U	A	0,0 bis 127,7	2,1
U	M	0,0 bis 255,7	2,1
U	D[1)]	0,0 bis 255,15	8,9
U	T	0 bis 255	2,1
U	Z	0 bis 255	2,1
UN	E	0,0 bis 127,7	2,1
UN	A	0,0 bis 127,7	2,1
UN	M	0,0 bis 255,7	2,1
UN	D[1)]	0,0 bis 255,15	8,9
UN	T	0 bis 255	2,1
UN	Z	0 bis 255	2,1
O	E	0,0 bis 127,7	2,1
O	A	0,0 bis 127,7	2,1
O	M	0,0 bis 255,7	2,1
O	D[1)]	0,0 bis 255,15	8,9
O	T	0 bis 255	2,1
O	Z	0 bis 255	2,1
ON	E	0,0 bis 127,7	2,1
ON	A	0,0 bis 127,7	2,1
ON	M	0,0 bis 255,7	2,1
ON	D[1)]	0,0 bis 255,15	8,9
ON	T	0 bis 255	2,1
ON	Z	0 bis 255	2,1
)			1,2
U(			1,5
O(			1,5
O			1,2
Vergleichsfunktionen Seite 73			
! =F			1,5
>< F			1,5
> F			1,5
>= F			1,5
< F			1,5
<= F			1,5
! =D			1,8
>< D			1,8
> D			1,8
>= D			1,8
< D			1,8
<= D			1,8
! =G			4,6
>< G			4,6
> G			4,6
>= G			4,6
< G			4,6
<= G			4,6

Operation		Parameter	Bearbeitungszeit in µs
Speicheroperationen Seite 67			
S	E	0,0 bis 127,7	3,0
S	A	0,0 bis 127,7	3,0
S	M	0,0 bis 255,7	3,0
S	D[1)]	0,0 bis 255,15	8,9
R	E	0,0 bis 127,7	2,1
R	A	0,0 bis 127,7	2,1
R	M	0,0 bis 255,7	2,1
R	D[1)]	0,0 bis 255,15	8,9
=	E	0,0 bis 127,7	2,5
=	A	0,0 bis 127,7	2,5
=	M	0,0 bis 255,7	2,5
=	D[1)]	0,0 bis 255,15	8,9

[1)] Wort 1: B2 + Bitadresse; B3 + relative Adresse

Ladefunktionen Seite 67 bis 71			
L	EB	0 bis 127	2,1
L	EW	0 bis 126	3,3
L	ED	0 bis 124	5,1
L	AB	0 bis 127	2,1
L	AW	0 bis 126	3,3
L	AD	0 bis 124	5,1
L	MB	0 bis 255	2,1
L	MW	0 bis 254	3,3
L	MD	0 bis 252	5,1
L	DL	0 bis 255	3,0
L	DR	0 bis 255	3,0
L	DW	0 bis 255	3,0
L	DD	0 bis 254	3,9
L	T	0 bis 255	2,1
L	Z	0 bis 255	2,1
L	PB	0 bis 127 128 bis 255	2,1
L	PW	0 bis 126 128 bis 254	3,3
L	QB	0 bis 255	2,1
L	QW	0 bis 254	3,3
LC	T	0 bis 255	12,5
LC	Z	0 bis 255	12,5
L	KB	0 bis 255	1,8
L	KC	2 alphanum. Zeichen	2,1
L	KM	Bitmuster (16 Bit)	2,1
L	KH	0 bis FFFF	2,1
L	KF	0 bis (2^{16}-1)	2,1
L	KY	0 bis 255, jedes Byte	2,1
L	KT	0,0 bis 999,3	2,1
L	KZ	0 bis 999	2,1
L	KG	$+0{,}1469368\text{-}10^{-38}$ bis $+0{,}1701412\text{-}10^{39}$)	3,0

Operation		Parameter	Bearbeitungszeit in µs
Zeit- und Zähloperationen Seite 68			
SI	T	0 bis 255	6,5
SV	T	0 bis 255	6,5
SE	T	0 bis 255	6,5
SS	T	0 bis 255	6,5
SA	T	0 bis 255	6,5
R	T	0 bis 255	3,3
S	Z	0 bis 255	5,1
R	Z	0 bis 255	3,3
ZV	Z	0 bis 255	3,8
ZR	Z	0 bis 255	3,9
Transferfunktionen Seite 72			
T	EB	0 bis 127	2,4
T	EW	0 bis 126	3,3
T	ED	0 bis 124	5,7
T	AB	0 bis 127	2,4
T	AW	0 bis 126	3,3
T	AD	0 bis 124	5,7
T	MB	0 bis 255	2,4
T	MW	0 bis 254	3,3
T	MD	0 bis 252	5,7
T	DR	0 bis 255	4,2
T	DL	0 bis 255	4,2
T	DW	0 bis 255	2,7
T	DD	0 bis 254	4,2
T	PB	0 bis 127 128 bis 255	3,3 2,4
T	PW	0 bis 126 128 bis 254	6,3 4,2
T	QB	0 bis 255	2,4
T	QW	0 bis 254	3,3
Bausteinaufrufe Seite 75			
SPA	PB	0 bis 255	7,0
SPA	FB	0 bis 255	7,0
SPA	SB	0 bis 255	7,0
SPB	PB	0 bis 255	7,0
SPB	FB	0 bis 255	7,0
SPB	SB	0 bis 255	7,0
A	DB	0 bis 255	4,5
BE			7,6
BEB			7,6
BEA			7,6

Gesamtübersicht über STEP-5-Befehle

Grundfunktionen (Fortsetzung)

Operation	Parameter	Bearbeitungszeit in µs
Arithmetische Operationen Seite 74		
+ F		1,9
– F		1,9
* F		8,0
/ F		10,0
+ G		14,0
– G		14,0
* G		13,0
/ G		16,0
Sonstige Funktionen Seite 76		
NOP 0		1,2
NOP 1		1,2
STP		3,9
BLD		–
Verknüpfungsfunktionen, wortweise Seite 76		
UW		1,2
OW		1,2
XOW		1,2
Zeit- und Zählfunktionen Seite 77 und 78		
F T	0 bis 255	4,0
F Z	0 bis 255	4,0
F =	Formaloperand	4,6
SI =	Formaloperand	4,6
SE =	Formaloperand	4,6
SVZ =	Formaloperand	4,6
SSV =	Formaloperand	4,6
SAR =	Formaloperand	4,6
RD =	Formaloperand	4,9
Lade- und Transferfunktionen Seite 82		
L =	Formaloperand	4,7
LC =	Formaloperand	4,6
LW =	Formaloperand	3,0
LD =	Formaloperand	4,2
T =	Formaloperand	4,7
L BS	0 bis 255	2,1
L BT	0 bis 255	2,1
L BA	0 bis 255	2,1
L BB	0 bis 255	2,1
T BA	0 bis 255	4,7
T BB	0 bis 255	2,4

Ergänzende Funktionen

Operation	Parameter	Bearbeitungszeit in µs
Verknüpfungsfunktionen, binär Seite 76		
U =	Formaloperand	6,3
UN =	Formaloperand	6,3
O =	Formaloperand	6,3
ON =	Formaloperand	6,3
Bittestfunktionen Seite 78 und 79		
P E	0,0 bis 127,7	6,1
P A	0,0 bis 127,7	6,1
P M	0,0 bis 255,7	6,1
P TB	0,0 bis 255,15	6,1
P ZW	0,0 bis 255,15	6,1
P D	0,0 bis 255,15	6,4
P BS	0,0 bis 255,15	6,1
P BT	0,0 bis 255,15	6,1
P BA	0,0 bis 255,15	6,1
P BB	0,0 bis 255,15	6,1
PN E	0,0 bis 127,7	6,1
PN A	0,0 bis 127,7	6,1
PN M	0,0 bis 255,7	6,1
PN T	0,0 bis 255,15	6,1
PN Z	0,0 bis 255,15	6,1
PN D	0,0 bis 255,15	6,4
PN BS	0,0 bis 255,15	6,1
PN BT	0,0 bis 255,15	6,1
PN BA	0,0 bis 255,15	6,1
PN BB	0,0 bis 255,15	6,1
Umwandlungsfunktionen Seite 83		
KEW		1,2
KZW		1,2
KZD		2,0
DEF		7,5
DUF		12,0
DED		24,0
DUD		35,7
FDG		12,5
GFD		10,0
Schiebefunktionen Seite 84		
SLW	0 bis 15	10,8
SRW	0 bis 15	10,8
SLD	0 bis 32	18,8
SVD	0 bis 32	18,8
RLD	0 bis 32	26,2
RRD	0 bis 32	26,2
SVW	0 bis 15	7,7

Operation	Parameter	Bearbeitungszeit in µs
Setzfunktionen Seite 80 und 81		
SU E	0,0 bis 127,7	6,1
SU A	0,0 bis 127,7	6,1
SU M	0,0 bis 255,7	6,1
SU T	0,0 bis 255,15	6,1
SU Z	0,0 bis 255,15	6,1
SU D	0,0 bis 255,15	6,4
SU BA	0,0 bis 255,15	6,1
SU BB	0,0 bis 255,15	6,1
RU E	0,0 bis 127,7	6,1
RU A	0,0 bis 127,7	6,1
RU M	0,0 bis 255,7	6,1
RU T	0,0 bis 255,15	6,1
RU Z	0,0 bis 255,15	6,1
RU D	0,0 bis 255,15	6,4
RU BA	0,0 bis 255,15	6,1
RU BB	0,0 bis 255,15	6,1
S =	Formaloperand	6,3
RB =	Formaloperand	6,3
= =	Formaloperand	6,3
Sprungfunktionen Seite 85		
SPA =	Symboladresse	2,6
SPB =	Symboladresse	2,6
SPZ =	Symboladresse	3,1
SPN =	Symoladresse	3,1
SPP =	Symboladresse	3,1
SPM =	Symboladresse	3,1
SPO =	Symboladresse	3,1
SPS[1] =	Symboladresse	3,5

[1] Wort 1: Sprungweite (2 Byte)

Sonstige Funktionen

Seite 86

BAF		1,5
BAS		1,5
AF		4,5
AS		4,5
AAF		4,2
AAS		4,2
AFF		3,9
AFS		3,9
ENT		1,5
D	0 bis 255	2,4
I	0 bis 255	2,4
B =	Formaloperand	5,9
B DW	0 bis 255	4,8
B MW	0 bis 25	4,4

Verknüpfungsoperationen, binär

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
														UND-Verknüpfung mit
U	E	0,0 bis 127,7	C0	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Einganges auf Signalzustand ,,1''
U	A	0,0 bis 127,7	C0	80	—	—	N	N	—	—	—	—	2,1	Abfrage eines Ausganges auf Signalzustand ,,1''
U	M	0,0 bis 255,7	80	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Merkers auf Signalzustand ,,1''
U	D[1]	0,0 bis 255,15	78	3F	00	00	N	N	—	—	—	—	8,9	Abfrage eines Datums auf Signalzustand ,,1''
U	T	0 bis 255	F8	00	—	—	N	N	—	—	—	—	2,1	Abfrage einer Zeit (Time) auf Signalzustand ,,1''
U	Z	0 bis 255	B8	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Zählers auf Inhalt ,,> 0''
UN	E	0,0 bis 127,7	E0	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Einganges auf Signalzustand ,,0''
UN	A	0,0 bis 127,7	E0	80	—	—	N	N	—	—	—	—	2,1	Abfrage eines Ausganges auf Signalzustand ,,0''
UN	M	0,0 bis 255,7	A0	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Merkers auf Signalzustand ,,0''
UN	D[1]	0,0 bis 255,15	78	3F	20	00	N	N	—	—	—	—	8,9	Abfrage eines Datums auf Signalzustand ,,0''
UN	T	0 bis 255	FC	00	—	—	N	N	—	—	—	—	2,1	Abfrage einer Zeit (Time) auf Signalzustand ,,0''
UN	Z	0 bis 255	BC	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Zählers auf Inhalt ,,0''
				+ relative Adresse										
			+ Bitadresse											

[1] Wort 1: B2 + Bitadresse; B3 + relative Adresse

Verknüpfungsoperationen, binär (Fortsetzung)

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
														ODER-Verknüpfung mit
O	E	0,0 bis 127,7	C8	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Einganges auf Signalzustand ,,1''
O	A	0,0 bis 127,7	C8	80	—	—	N	N	—	—	—	—	2,1	Abfrage eines Ausganges auf Signalzustand ,,1''
O	M	0,0 bis 255,7	88	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Merkers auf Signalzustand ,,1''
O	D[1)]	0,0 bis 255,15	78	3F	10	00	N	N	—	—	—	—	8,9	Abfrage eines Datums auf Signalzustand ,,1''
O	T	0 bis 255	F9	00	—	—	N	N	—	—	—	—	2,1	Abfrage einer Zeit (Time) auf Signalzustand ,,1''
O	Z	0 bis 255	B9	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Zählers auf Inhalt ,,> 0''
ON	E	0,0 bis 127,7	E8	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Einganges auf Signalzustand ,,0''
ON	A	0,0 bis 127,7	E8	80	—	—	N	N	—	—	—	—	2,1	Abfrage eines Ausganges auf Signalzustand ,,0''
ON	M	0,0 bis 255,7	A8	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Merkers auf Signalzustand ,,0''
ON	D[1)]	0,0 bis 255,15	78	3F	30	00	N	N	—	—	—	—	8,9	Abfrage eines Datums auf Signalzustand ,,0''
ON	T	0 bis 255	FD	00	—	—	N	N	—	—	—	—	2,1	Abfrage einer Zeit (Time) auf Signalzustand ,,0''
ON	Z	0 bis 255	BD	00	—	—	N	N	—	—	—	—	2,1	Abfrage eines Zählers auf Inhalt ,,0''
				+ relative Adresse										
			+ Bitadresse											
)			BF	00	—	—	N	J	—	—	—	—	1,2	Klammer zu
U(			BA	00	—	—	N	J	—	—	—	—	1,5	UND-Verknüpfung von Klammerausdrücken
O(			BB	00	—	—	N	J	—	—	—	—	1,5	ODER-Verknüpfung von Klammerausdrücken
O			FB	00	—	—	N	J	—	—	—	—	1,2	ODER-Verknüpfung von UND-Funktionen

[1)] Wort 1: B2 + Bitadresse; B3 + relative Adresse

Speicheroperationen

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
														Setzen
S	E	0,0 bis 127,7	D0	00	—	—	J	J	—	—	—	—	3,0	eines Einganges (im PAE)
S	A	0,0 bis 127,7	D0	80	—	—	J	J	—	—	—	—	3,0	eines Ausganges (im PAA)
S	M	0,0 bis 255,7	90	00	—	—	J	J	—	—	—	—	3,0	eines Merkers
S	D[1)]	0,0 bis 255,15	78	3F	40	00	J	J	—	—	—	—	8,9	eines Datums
														Rücksetzen
R	E	0,0 bis 127,7	F0	00	—	—	J	J	—	—	—	-	2,1	eines Einganges (im PAE)
R	A	0,0 bis 127,7	F0	80	—	—	J	J	—	—	—	—	2,1	eines Ausganges (im PAA)
R	M	0,0 bis 255,7	B0	00	—	—	J	J	—	—	—	—	2,1	eines Merkers
R	D[1)]	0,0 bis 255,15	78	3F	50	00	J	J	—	—	—	—	8,9	eines Datums
														Zuweisen
=	E	0,0 bis 127,7	D8	00	—	—	N	J	—	—	—	—	2,5	eines Einganges (im PAE)
=	A	0,0 bis 127,7	D8	80	—	—	N	J	—	—	—	—	2,5	eines Ausganges (im PAA)
=	M	0,0 bis 255,7	98	00	—	—	N	J	—	—	—	—	2,5	eines Merkers
=	D[1)]	0,0 bis 255,15	78	3F	60	00	N	J	—	—	—	—	8,9	eines Datums

+ relative Adresse

+ Bitadresse

[1)] Wort 1: B2 + Bitadresse; B3 + relative Adresse

Zeit- und Zähloperationen

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
SI	T	0 bis 255	34	00	—	—	J	J	—	—	—	—	6,5	Starten einer Zeit als Impuls
SV	T	0 bis 255	1C	00	—	—	J	J	—	—	—	—	6,5	Starten einer Zeit als verlängerter Impuls
SE	T	0 bis 255	24	00	—	—	J	J	—	—	—	—	6,5	Starten einer Zeit als Einschaltverzögerung
SS	T	0 bis 255	2C	00	—	—	J	J	—	—	—	—	6,5	Starten einer Zeit als speichernde Einschaltverzögerung
SA	T	0 bis 255	14	00	—	—	J	J	—	—	—	—	6,5	Starten einer Zeit als Ausschaltverzögerung
R	T	0 bis 255	3C	00	—	—	J	J	—	—	—	—	3,3	Rücksetzen einer Zeit
S	Z	0 bis 255	5C	00	—	—	J	J	—	—	—	—	5,1	Setzen eines Zählers
R	Z	0 bis 255	7C	00	—	—	J	J	—	—	—	—	3,3	Rücksetzen eines Zählers
ZV	Z	0 bis 255	6C	00	—	—	J	J	—	—	—	—	3,8	Vorwärtszählen eines Zählers
ZR	Z	0 bis 255	54	00	—	—	J	J	—	—	—	—	3,9	Rückwärtszählen eines Zählers
				+ relative Adresse										

Lade- und Transferfunktionen

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
													Laden
L EB	0 bis 127	4A	00	—	—	N	N	—	—	—	—	2,1	eines Eingabebytes (vom PAE)
L EW	0 bis 126	52	00	—	—	N	N	—	—	—	—	3,3	eines Eingabewortes (vom PAE)
L ED	0 bis 124	5A	00	—	—	N	N	—	—	—	—	5,1	eines Eingabe-Doppelwortes (vom PAE)
L AB	0 bis 127	4A	80	—	—	N	N	—	—	—	—	2,1	eines Ausgabebytes (vom PAA)
L AW	0 bis 126	52	80	—	—	N	N	—	—	—	—	3,3	eines Ausgabewortes (vom PAA)
L AD	0 bis 124	5A	80	—	—	N	N	—	—	—	—	5,1	eines Ausgabe-Doppelwortes (vom PAA)
L MB	0 bis 255	0A	00	—	—	N	N	—	—	—	—	2,1	eines Merkerbytes
L MW	0 bis 254	12	00	—	—	N	N	—	—	—	—	3,3	eines Merkerwortes
L MD	0 bis 252	1A	00	—	—	N	N	—	—	—	—	5,1	eines Merker-Doppelwortes
L DL	0 bis 255	22	00	—	—	N	N	—	—	—	—	3,0	eines Datums (linkes Byte)
L DR	0 bis 255	2A	00	—	—	N	N	—	—	—	—	3,0	eines Datums (rechtes Byte)
L DW	0 bis 255	32	00	—	—	N	N	—	—	—	—	3,0	eines Datums (Wort)
L DD	0 bis 254	3A	00	—	—	N	N	—	—	—	—	3,9	eines Datums (Doppelwort)
			+ relative Adresse										

Lade- und Transferfunktionen (Fortsetzung)

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
														Laden
L	T	0 bis 255	02	00	—	—	N	N	—	—	—	—	2,1	eines Zeitwertes
L	Z	0 bis 255	42	00	—	—	N	N	—	—	—	—	2,1	eines Zählwertes
L	PB	0 bis 127 128 bis 255	72	00	—	—	N	N	—	—	—	—	2,1	eines Peripheriebytes der Digital-Eingaben eines Peripheriebytes der Analog-Eingaben
L	PW	0 bis 126 128 bis 254	74	00	—	—	N	N	—	—	—	—	3,3	eines Peripheriewortes der Digital-Eingaben eines Peripheriewortes der Analog-Eingaben
L	QB	0 bis 255	5F	00	—	—	N	N	—	—	—	—	2,1	eines Bytes der erweiterten Peripherie
L	QW	0 bis 254	57	00	—	—	N	N	—	—	—	—	3,3	eines Wortes der erweiterten Peripherie
LC	T	0 bis 255	0C	00	—	—	N	N	—	—	—	—	12,5	eines Zeitwertes (BCD-codiert)
LC	Z	0 bis 255	4C	00	—	—	N	N	—	—	—	—	12,5	eines Zählwertes (BCD-codiert)
				+ relative Adresse										

Lade- und Transferfunktionen (Fortsetzung)

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
													Laden einer Konstanten
L KB	0 bis 255	28	00	—	—	N	N	—	—	—	—	1,8	1 Byte
			+ Konstante (1 Byte)										
L KC	2 alphanum. Zeichen	30	10	00	00	N	N	—	—	—	—	2,1	2 ASCII-Zeichen
L KM	Bitmuster (16 Bit)	30	80	00	00	N	N	—	—	—	—	2,1	als Bitmuster
L KH	0 bis FFFF	30	40	00	00	N	N	—	—	—	—	2,1	im Hexa-Code
L KF	0 bis (2^{16}-1)	30	04	00	00	N	N	—	—	—	—	2,1	als Festpunktzahl
L KY	0 bis 255, jedes Byte	30	20	00	00	N	N	—	—	—	—	2,1	2 Byte
L KT	0,0 bis 999,3	30	02	00	00	N	N	—	—	—	—	2,1	als Zeitwert
L KZ	0 bis 999	30	01	00	00	N	N	—	—	—	—	2,1	als Zählwert
				+ Konstante (1 Wort)									
L KG	$\pm 0{,}1469368 \cdot 10^{-38}$ bis $\pm 0{,}1701412 \cdot 10^{39}$)	38	00	00	00	N	N	—	—	—	—	3,0	als Gleitpunktzahl
				00	00								
				+ Konstante (Doppelwort: Wort 1, Wort 2)									

Lade- und Transferfunktionen (Fortsetzung)

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
														Transferieren
T	EB	0 bis 127	4B	00	—	—	N	N	—	—	—	J	2,4	zu einem Eingabebyte (im PAE)
T	EW	0 bis 126	53	00	—	—	N	N	—	—	—	J	3,3	zu einem Eingabewort (im PAE)
T	ED	0 bis 124	5B	00	—	—	N	N	—	—	—	J	5,7	zu einem Eingabedoppelwort (im PAE)
T	AB	0 bis 127	4B	80	—	—	N	N	—	—	—	J	2,4	zu einem Ausgangsbyte (im PAA)
T	AW	0 bis 126	53	80	—	—	N	N	—	—	—	J	3,3	zu einem Ausgabewort (im PAA)
T	AD	0 bis 124	5B	80	—	—	N	N	—	—	—	J	5,7	zu einem Ausgabe-Doppelwort (im PAA)
T	MB	0 bis 255	0B	00	—	—	N	N	—	—	—	J	2,4	zu einem Merkerbyte
T	MW	0 bis 254	13	00	—	—	N	N	—	—	—	J	3,3	zu einem Merkerwort
T	MD	0 bis 252	1B	00	—	—	N	N	—	—	—	J	5,7	zu einem Merker-Doppelwort
T	DR	0 bis 255	2B	00	—	—	N	N	—	—	—	J	4,2	zu einem Datum (rechtes Byte)
T	DL	0 bis 255	23	00	—	—	N	N	—	—	—	J	4,2	zu einem Datum (linkes Byte)
T	DW	0 bis 255	33	00	—	—	N	N	—	—	—	J	2,7	zu einem Datum (Wort)
T	DD	0 bis 254	3B	00	—	—	N	N	—	—	—	J	4,2	zu einem Datum (Doppelwort)
T	PB	0 bis 127 128 bis 255	73	00	—	—	N	N	—	—	—	J	3,3 2,4	zu einem Peripheriebyte der Digital-Ausgaben mit Nachführen des PAA zu einem Peripheriebyte der Analog-Ausgaben ohne Nachführen des PAA
T	PW	0 bis 126 128 bis 254	7B	00	—	—	N	N	—	—	—	J	6,3 4,2	zu einem Peripheriewort der Digital-Ausgaben mit Nachführen des PAA zu einem Peripheriewort der Analog-Ausgaben ohne Nachführen des PAA
T	QB	0 bis 255	7F	00	—	—	N	N	—	—	—	J	2,4	zu einem Peripheriebyte der erweiterten Peripherie
T	QW	0 bis 254	77	00	—	—	N	N	—	—	—	J	3,3	zu einem Peripheriewort der erweiterten Peripherie
				+ relative Adresse										

Vergleichsfunktionen

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
													Vergleich-Festpunktzahlen
! =F		21	80	—	—	N	N	J	J	—	—	1,5	auf gleich
>< F		21	60	—	—	N	N	J	J	—	—	1,5	auf ungleich
>F		21	20	—	—	N	N	J	J	—	—	1,5	auf größer
>=F		21	A0	—	—	N	N	J	J	—	—	1,5	auf größer-gleich
<F		21	40	—	—	N	N	J	J	—	—	1,5	auf kleiner
<=F		21	C0	—	—	N	N	J	J	—	—	1,5	auf kleiner-gleich
													Vergleich-Festpunkt-Doppelwort
! =D		39	80	—	—	N	N	J	J	—	—	1,8	auf gleich
>< D		39	60	—	—	N	N	J	J	—	—	1,8	auf ungleich
>D		39	20	—	—	N	N	J	J	—	—	1,8	auf größer
>=D		39	A0	—	—	N	N	J	J	—	—	1,8	auf größer-gleich
<D		39	40	—	—	N	N	J	J	—	—	1,8	auf kleiner
<=D		39	C0	—	—	N	N	J	J	—	—	1,8	auf kleiner-gleich
													Vergleich-Gleitkommazahlen
! =G		31	80	—	—	N	N	J	J	—	—	4,6	auf gleich
>< G		31	60	—	—	N	N	J	J	—	—	4,6	auf ungleich
>G		31	20	—	—	N	N	J	J	—	—	4,6	auf größer
>=G		31	A0	—	—	N	N	J	J	—	—	4,6	auf größer-gleich
<G		31	40	—	—	N	N	J	J	—	—	4,6	auf kleiner
<=G		31	C0	—	—	N	N	J	J	—	—	4,6	auf kleiner-gleich

Arithmetische Operationen

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
+F		79	00	—	—	N	N	J	J	J	J	1,9	Festpunkt-Addition
−F		59	00	—	—	N	N	J	J	J	J	1,9	Festpunkt-Subtraktion
* F		60	04	—	—	N	N	J	J	J	J	8,0	Festpunkt-Multiplikation
/ F		60	00	—	—	N	N	J	J	J	J	10,0	Festpunkt-Division
+G		60	0F	—	—	N	N	J	J	J	J	14,0	Gleitpunkt-Addition
−G		60	0B	—	—	N	N	J	J	J	J	14,0	Gleitpunkt-Subtraktion
* G		60	07	—	—	N	N	J	J	J	J	13,0	Gleitpunkt-Multiplikation
/ G		60	03	—	—	N	N	J	J	J	J	16,0	Gleitpunkt-Division

Bausteinaufrufe

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
														Sprung bedingt
SPA	PB	0 bis 255	75	00			N	J	—	—	—	—	7,0	zu einem Programmbaustein
SPA	FB	0 bis 255	3D	00			N	J	—	—	—	—	7,0	zu einem Funktionsbaustein
SPA	SB	0 bis 255	7D	00			N	J	—	—	—	—	7,0	zu einem Schrittbaustein
														Sprung bedingt
SPB	PB	0 bis 255	55	00			J	J	—	—	—	—	7,0	zu einem Programmbaustein
SPB	FB	0 bis 255	1D	00			J	J	—	—	—	—	7,0	zu einem Funktionsbaustein
SPB	SB	0 bis 255	5D	00			J	J	—	—	—	—	7,0	zu einem Schrittbaustein
A	DB	0 bis 255	20	00			N	N	—	—	—	—	4,5	Aufruf eines Datenbausteins
				+ Bausteinnummer (hexadezimal)										
BE			65	00			N	J	—	—	—	—	7,6	Bausteinende
BEB			05	00			J	J	—	—	—	—	7,6	Bausteinende bedingt
BEA			65	01			N	J					7,6	Bausteinende absolut

Sonstige Funktionen

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
NOP 0		00	00			N	N	—	—	—	—	1,2	Nulloperation (alle Bits gelöscht)
NOP 1		FF	FF			N	N	—	—	—	—	1,2	Nulloperation (alle Bits gesetzt)
STP		70	03			N	N	—	—	—	—	3,9	Stop
BLD		10	00			N	N	—	—	—	—	—	Segmentende für Programmierung in Anweisungsliste. Dient zum Bildaufbau im Kontaktplan für das Programmiergerät

Verknüpfungsfunktionen, wortweise (ergänzende Funktionen)

UW		41	00			N	N	X	X	—	—	1,2	UND-Verknüpfung
OW		49	00			N	N	X	X	—	—	1,2	ODER-Verknüpfung
XOW		51	00					X	X	—	—	1,2	Exklusiv-ODER-Verknüpfung von AKKU 1

Verknüpfungsfunktionen, binär (ergänzende Funktionen)

U	=	Formaloperand	07	00			N	N	—	—	—	—	6,3	UND-Funktion, Abfrage eines Formaloperanden auf Signalzustand ,,1''
UN	=	Formaloperand	27	00			N	N	—	—	—	—	6,3	UND-Funktion, Abfrage eines Formaloperanden aus Signalzustand ,,0''
O	=	Formaloperand	0F	00			N	N	—	—	—	—	6,3	ODER-Funktion, Abfrage eines Formaloperanden auf Signalzustand ,,1''
ON	=	Formaloperand	2F	00			N	N	—	—	—	—	6,3	ODER-Funktion, Abfrage eines Formaloperanden auf Signalzustand ,,0''
				+ Parameteradresse									

Zeit- und Zählfunktionen (ergänzende Funktionen)

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
F T	0 bis 255	04	00			J	J	—	—	—	—	4,0	Freigabe einer Zeit für einen Neustart Die Operation wird nur bei steigender Flanke des Verknüpfungsergebnisses ausgeführt. Sie bewirkt einen Neustart der Zeit, wenn bei der Startoperation Verknüpfungsergebnis ,,1'' anliegt.
F Z	0 bis 255	44	00			J	J	—	—	—	—	4,0	Freigabe eines Zählers für einen Neustart Die Operation wird nur bei steigender Flanke des Verknüpfungsergebnisses ausgeführt. Sie bewirkt ein Setzen, Vorwärts- oder Rückwärtszählen des Zählers, wenn in der entsprechenden Operation Verknüpfungsergebnis ,,1'' anliegt.
F =	Formaloperand	06	00			N	N	—	—	—	—	4,6	Freigabe eines Formaloperanden für Neustart (Beschreibung siehe FT bzw. FZ, je nach Formaloperand; Parameterart: T, Z).
SI =	Formaloperand	36	00			N	N	—	—	—	—	4,6	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als Impuls (Parameterart: T).
SE =	Formaloperand	26	00			N	N	—	—	—	—	4,6	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als Einschaltverzögerung (Parameterart: T).
SVZ =	Formaloperand	1E	00			N	N	—	—	—	—	4,6	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als verlängerter Impuls bzw. Setzen eines als Formaloperand vorgegebenen Zählers mit dem nachfolgend angegebenen Zählwert (Parameterart: T, Z).
SSV =	Formaloperand	2E	00			N	N	—	—	—	—	4,6	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als speichernde Einschaltverzögerung bzw. Vorwärtszählen eines als Formaloperand vorgegebenen Zählers (Parameterart: T, Z).
			+ Parameteradresse										

Zeit- und Zählfunktionen (ergänzende Funktionen) (Fortsetzung)

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
SAR =	Formaloperand	16	00			N	N	—	—	—	—	4,6	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als Ausschaltverzögerung bzw. Rückwärtszählen eines als Formaloperand vorgegebenen Zählers (Parameterart: T, Z).
RD =	Formaloperand	3E	00			N	N	—	—	—	—	4,9	Rücksetzen eines Formaloperanden für Zeiten und Zähler (Parameterart: T, Z).
			+ Parameteradresse										

Bittestfunktionen (ergänzende Funktionen)

													Prüfe Bit auf Signalzustand „1"
P	E	0,0 bis 127,7	70	38	C0	00	N	N	—	—	—	—	6,1	eines Einganges (im PAE)
P	A	0,0 bis 127,7	70	38	C0	80	N	N	—	—	—	—	6,1	eines Ausganges (im PAA)
P	M	0,0 bis 255,7	70	49	C0	00	N	N	—	—	—	—	6,1	eines Merkers
P	T	0,0 bis 255,15	70	25	C0	00	N	N	—	—	—	—	6,1	eines Zeitwortes
P	Z	0,0 bis 255,15	70	15	C0	00	N	N	—	—	—	—	6,1	eines Zählwortes
P	D	0,0 bis 255,15	70	46	C0	00	N	N	—	—	—	—	6,4	eines Datenwortes
P	BS	0,0 bis 255,15	70	57	C0	00	N	N	—	—	—	—	6,1	Bereich Systemdaten
P	BT	0,0 bis 255,15	70	QE	C0	00	N	N	—	—	—	—	6,1	Bereich Systemdatenerweiterung
P	BA	0,0 bis 255,15	70	47	C0	00	N	N	—	—	—	—	6,1	Bereich Anschaltungsdaten
P	BB	0,0 bis 255,15	70	1E	C0	00	N	N	—	—	—	—	6,1	Bereich Anschaltungsdatenerweiterung
					+ Bitadresse	+ relative Adresse							

Bittestfunktionen (ergänzende Funktionen) (Fortsetzung)

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
														Prüfe Bit auf Signalzustand ,,0''
PN	E	0,0 bis 127,7	70	38	80	00	N	N	–	–	–	–	6,1	eines Einganges (im PAE)
PN	A	0,0 bis 127,7	70	38	80	80	N	N	–	–	–	–	6,1	eines Ausganges (im PAA)
PN	M	0,0 bis 255,7	70	49	80	00	N	N	–	–	–	–	6,1	eines Merkers
PN	T	0,0 bis 255,15	70	25	80	00	N	N	–	–	–	–	6,1	eines Zeitwortes
PN	Z	0,0 bis 255,15	70	15	80	00	N	N	–	–	–	–	6,1	eines Zählwortes
PN	D	0,0 bis 255,15	70	46	80	00	N	N	–	–	–	–	6,4	eines Datenwortes
PN	BS	0,0 bis 255,15	70	57	80	00	N	N	–	–	–	–	6,1	Bereich Systemdaten
PN	BT	0,0 bis 255,15	70	0E	80	00	N	N	–	–	–	–	6,1	Bereich Systemdatenerweiterung
PN	BA	0,0 bis 255,15	70	47	80	00	N	N	–	–	–	–	6,1	Bereich Anschaltungsdaten
PN	BB	0,0 bis 255,15	70	1E	80	00	N	N	–	–	–	–	6,1	Bereich Anschaltungserweiterung
						+ relative Adresse								
					+ Bitadresse									

Setzfunktionen (ergänzende Funktionen)

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
														Setze Bit unbedingt
SU	E	0,0 bis 127,7	70	38	40	00	N	J	–	–	–	–	6,1	eines Einganges (im PAE)
SU	A	0,0 bis 127,7	70	38	40	80	N	J	–	–	–	–	6,1	eines Ausganges (im PAA)
SU	M	0,0 bis 255,7	70	49	40	00	N	J	–	–	–	–	6,1	eines Merkers
SU	T	0,0 bis 255,15	70	25	40	00	N	J	–	–	–	–	6,1	eines Zeitwortes
SU	Z	0,0 bis 255,15	70	15	40	00	N	J	–	–	–	–	6,1	eines Zählwortes
SU	D	0,0 bis 255,15	70	46	40	00	N	J	–	–	–	–	6,4	eines Datenwortes
SU	BA	0,0 bis 255,15	70	47	40	00	N	J	–	–	–	–	6,1	Bereich Anschaltungsdaten
SU	BB	0,0 bis 255,15	70	1E	40	00	N	J	–	–	–	–	6,1	Bereich Anschaltungsdatenerweiterung
														Rücksetze Bit unbedingt
RU	E	0,0 bis 127,7	70	38	00	00	N	J	–	–	–	–	6,1	eines Einganges (im PAE)
RU	A	0,0 bis 127,7	70	38	00	80	N	J	–	–	–	–	6,1	eines Ausganges (im PAA)
RU	M	0,0 bis 255,7	70	49	00	00	N	J	–	–	–	–	6,1	eines Merkers
RU	T	0,0 bis 255,15	70	25	00	00	N	J	–	–	–	–	6,1	eines Zeitwortes
RU	Z	0,0 bis 255,15	70	15	00	00	N	J	–	–	–	–	6,1	eines Zählwortes
RU	D	0,0 bis 255,15	70	46	00	00	N	J	–	–	–	–	6,4	eines Datenwortes
RU	BA	0,0 bis 255,15	70	47	00	00	N	J	–	–	–	–	6,1	Bereich Anschaltungsdaten
RU	BB	0,0 bis 255,15	70	1E	00	00	N	J	–	–	–	–	6,1	Bereich Anschaltungsdatenerweiterung

+ relative Adresse

+ Bitadresse

Setzfunktionen (ergänzende Funktionen) (Fortsetzung)

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
S =	Formaloperand	17	00	—	—	N	J	—	—	—	—	6,3	Setzen (binär) eines Formaloperanden
RB =	Formaloperand	37	00	—	—	N	J	—	—	—	—	6,3	Rücksetzen (binär) eines Formaloperanden
= =	Formaloperand	1F	00	—	—	N	J	—	—	—	—	6,3	Zuweisen des Verknüpfungsergebnisses zu einem Formaloperanden
			+ Parameteradresse										

Lade- und Transferfunktionen (ergänzende Funktionen)

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
L =	Formaloperand	46	00	—	—	N	N	—	—	—	—	4,7	Laden eines Formaloperanden (Parameterart: E, A; Parametertyp: BY, W, D)
LC =	Formaloperand	0E	00	—	—	N	N	—	—	—	—	4,6	Laden codiert eines Formaloperanden (Parameter: T, Z)
LW =	Formaloperand	3F	00	—	—	N	J	—	—	—	—	3,0	Laden des Bitmusters eines Formaloperanden (Parameterart: D; Parametertyp: KF, KH, KM, KY, KC, KT, KZ)
LD =	Formaloperand	56	00	—	—	N	N	—	—	—	—	4,2	Laden des Bitmusters eines Formaloperanden (Parameterart: D; Parametertyp: KG)
T =	Formaloperand	66	00	—	—	N	N	—	—	—	—	4,7	Transferieren zu einem Formaloperanden (Parameterart: E, A; Parametertyp: BY, W, D)
			+ Parameteradresse										
L BS	0 bis 255	62	00	—	—	N	N	—	—	—	—	2,1	Laden eines Wortes aus dem Bereich Systemdaten
L BT	0 bis 255	4F	00	—	—	N	N	—	—	—	—	2,1	Laden eines Wortes aus dem Bereich Systemdatenerweiterung
L BA	0 bis 255	6A	00	—	—	N	N	—	—	—	—	2,1	Laden eines Wortes aus dem Bereich Anschaltungsdaten
L BB	0 bis 255	47	00	—	—	N	N	—	—	—	—	2,1	Laden eines Wortes aus dem Bereich Anschaltungsdatenerweiterung
T BA	0 bis 255	6B	00	—	—	N	N	—	—	—	—	4,7	Transferieren zu einem Wort aus dem Bereich Anschaltungsdaten
T BB	0 bis 255	67	00	—	—	N	N	—	—	—	—	2,4	Transferieren zu einem Wort aus dem Bereich Anschaltungsdatenerweiterung
			+ relative Adresse										

Umwandlungsfunktionen (ergänzende Funktionen)

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
KEW		01	00	—	—	N	N	—	—	—	—	1,2	1-er-Komplementbildung (Festpunkt)
KZW		09	00	—	—	N	N	J	J	J	J	1,2	2-er-Komplementbildung (Festpunkt)
KZD		68	07	—	—	N	N	J	J	J	J	2,0	2-er-Komplementbildung (Festpunkt-Doppelwort)
DEF		68	0C	—	—	N	N	—	—	—	—	7,5	Dezimal (BCD)→Festpunkt Wandlung
DUF		68	08	—	—	N	N	—	—	—	J	12,0	Festpunkt →Dezimal (BCD) Wandlung
DED		68	0E	—	—	N	N	—	—	—	—	24,0	Dezimal (BCD)→Festpunkt-Doppelwort Wandlung
DUD		68	0A	—	—	N	N	—	—	—	J	35,7	Festpunkt-Doppelwort→Dezimal (BCD) Wandlung
FDG		68	06	—	—	N	N	—	—	—	—	12,5	Festpunkt-Doppelwort→Gleitpunkt Wandlung
GFD		68	02	—	—	N	N	—	—	—	J	10,0	Gleitpunkt→Festpunkt-Doppelwort Wandlung

Schiebefunktionen (ergänzende Funktionen)

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
SLW	0 bis 15	61	00	—	—	N	N	J	J	—	—	10,8	Schieben links (16 bit)
SRW	0 bis 15	69	00	—	—	N	N	J	J	—	—	10,8	Schieben rechts (16 bit)
			+ Schiebezahl										
SLD	0 bis 32	29	00	—	—	N	N	J	J	—	—	18,8	Schieben links (32 bit)
SVD	0 bis 32	71	00	—	—	N	N	J	J	—	—	18,8	Schieben rechts (32 bit) mit Vorzeichen
RLD	0 bis 32	64	00	—	—	N	N	J	J	—	—	26,2	Rotieren links (32 bit)
RRD	0 bis 32	74	00	—	—	N	N	J	J	—	—	26,2	Rotieren rechts (32 bit)
			+ Schiebezahl										
SVW	0 bis 15	68	01	—	—	N	N	J	J	—	—	7,7	Schieben rechts (32 bit) mit Vorzeichen
			+ Schiebezahl										

Sprungfunktionen (ergänzende Funktionen)

Operation		Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt				Bearbeitungszeit in µs	Funktion
			Wort 0		Wort 1									
			B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
SPA	=	Symboladresse	2D	00	—	—	N	N	—	—	—	—	2,6	Sprung unbedingt
SPB	=	Symboladresse	FA	00	—	—	J	J	—	—	—	—	2,6	Sprung bedingt (Sprungbedingung: VKE)
SPZ	=	Symboladresse	45	00	—	—	N	N	—	—	—	—	3,1	Sprung bedingt (Sprungbedingung: ANZ 1, ANZ 0)
SPN	=	Symboladresse	35	00	—	—	N	N	—	—	—	—	3,1	Sprung bedingt (Sprungbedingung: ANZ 1, ANZ 0)
SPP	=	Symboladresse	15	00	—	—	N	N	—	—	—	—	3,1	Sprung bedingt (Sprungbedingung: ANZ 1, ANZ 0)
SPM	=	Symboladresse	25	00	—	—	N	N	—	—	—	—	3,1	Sprung bedingt (Sprungbedingung: ANZ 1, ANZ 0)
SPO	=	Symboladresse	0D	00	—	—	N	N	—	—	—	—	3,1	Sprung bedingt (Sprungbedingung: OV)
SPS[1]	=	Symboladresse	60	0C	0	0	N	N	—	—	—	—	3,5	Sprung bedingt (Sprungbedingung: OS)
				+ Sprungweite										

[1] Wort 1: Sprungweite (2 Byte)

Sonstige Funktionen (ergänzende Funktionen)

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort-Anzeigen werden beeinflußt					Funktion
		Wort 0		Wort 1									
		B0	B1	B2	B3			ANZ1	ANZ0	OV	OS		
BAF		FE	00	—	—	J	J	—	—	—	—	1,5	Befehlsausgabe freigeben
BAS		BE	00	—	—	J	J	—	—	—	—	1,5	Befehlsausgabe sperren
AF		08	80	—	—	N	N	—	—	—	—	4,5	Prozeßalarmbearbeitung freigeben
AS		08	00	—	—	N	N	—	—	—	—	4,5	Prozeßalarmbearbeitung sperren
AAF		5E	40	—	—	N	N	—	—	—	—	4,2	Anforderungsalarmbearbeitung freigeben
AAS		5E	00	—	—	N	N	—	—	—	—	4,2	Anforderungsalarmbearbeitung sperren
AFF		78	10	—	—	N	N	—	—	—	—	3,9	Adressierfehlerbearbeitung freigeben
AFS		78	00	—	—	N	N	—	—	—	—	3,9	Adressierfehlerbearbeitung sperren
ENT		60	08	—	—	N	N	—	—	—	—	1,5	Eintrag von Daten in Arithmetikspeicher
D	0 bis 255	19	00	—	—	N	N	—	—	—	—	2,4	Dekrementieren
I	0 bis 255	11	00	—	—	N	N	—	—	—	—	2,4	Inkremetieren
			+ Wert										
B =	Formaloperand	76	00	—	—	N	N	—	—	—	—	5,9	Bearbeite Formaloperand (Parameterart: B)
			+ Parameteradresse										
B DW	0 bis 255	6E	00	—	—	N	N	—	—	—	—	4,8	Bearbeite Datenwort
B MW	0 bis 25	4E	00	—	—	N	N	—	—	—	—	4,4	Bearbeite Merkerwort
			+ relative Adresse										

SIEMENS AKTIENGESELLSCHAFT

Bestell-Nr.: C79000-B8500-C247-1
Printed in the Federal Republic of Germany
AG 882 1.0 HL 88 De