

SINUMERIK 810 GA3  
SINUMERIK 820 GA3  
Programmieren PLC

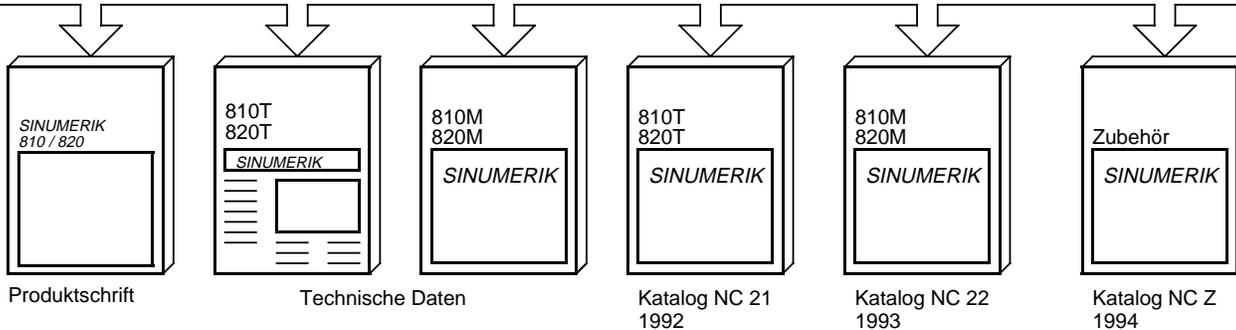
Projektierungsanleitung

Ausgabe 07.94

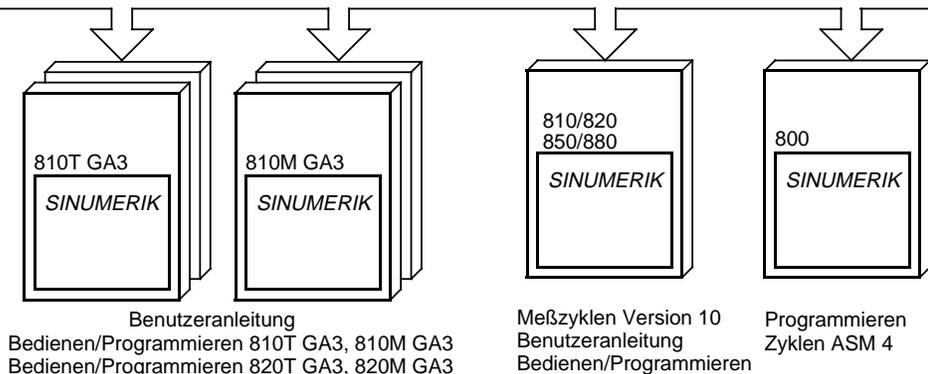
Hersteller-Dokumentation

# SINUMERIK 810/820, Grundausführung 3 (GA3)

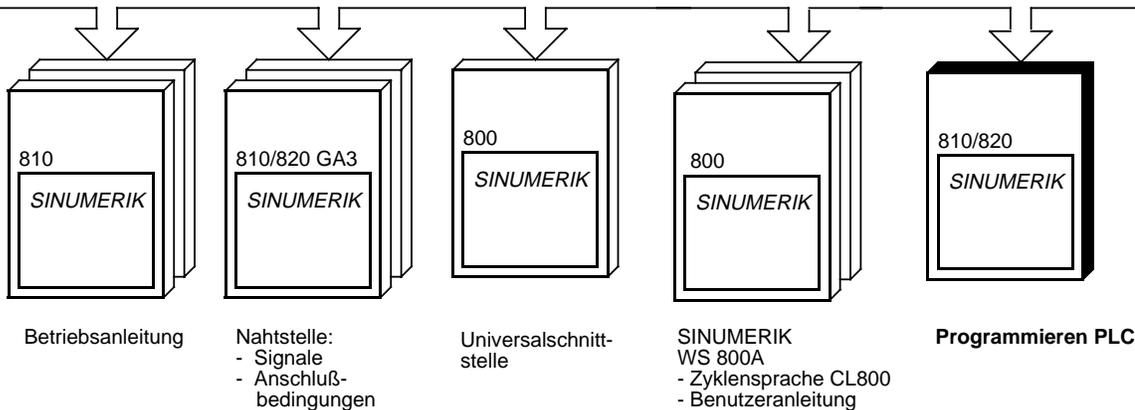
## Allgemeine Dokumentation



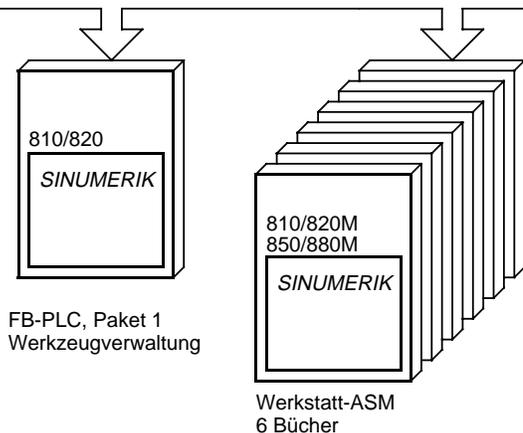
## Anwender-Dokumentation



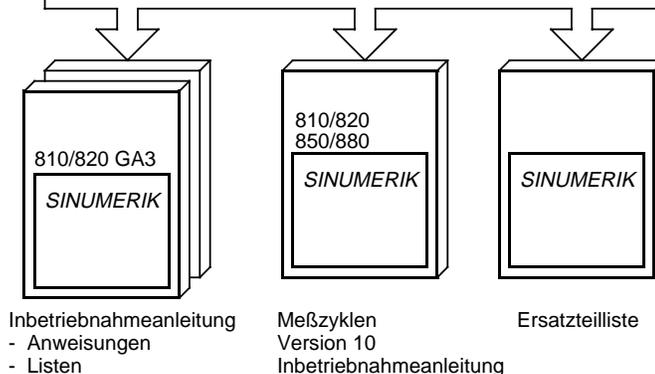
## Hersteller-Dokumentation



## Hersteller-Dokumentation



## Service-Dokumentation



# SINUMERIK 810 GA3 SINUMERIK 820 GA3 Programmieren PLC

**Projektierungsanleitung**

**Hersteller-Dokumentation**

**Gültig für:**

<i>Steuerung</i>		<i>Softwarestand</i>
SINUMERIK 810T/TE	Grundauführung 3	ab 01
SINUMERIK 810M/ME	Grundauführung 3	ab 01
SINUMERIK 820T/TE	Grundauführung 3	ab 01
SINUMERIK 820M/ME	Grundauführung 3	ab 01
SINUMERIK 810 G	Grundauführung 3	ab 01
SINUMERIK 820 G	Grundauführung 3	ab 01
SINUMERIK 810 N	Grundauführung 3	ab 01
SINUMERIK 820 N	Grundauführung 3	ab 01

**Ausgabe 07.94**

## Auflagenschlüssel

Die nachfolgend aufgeführten Ausgaben sind bis zu der vorliegenden Ausgabe erschienen.

In der Spalte "Bemerkung" ist durch Buchstaben gekennzeichnet, welchen Status die bisher erschienenen Ausgaben besitzen.

*Kennzeichnung des Status in Spalte "Bemerkung":*

- A** ... Neue Dokumentation.
- B** ... Unveränderter Nachdruck mit neuer Bestell-Nummer.
- C** ... Überarbeitete Ausgabe mit neuem Ausgabestand.  
Hat sich der auf einer Seite dargestellte technische Sachverhalt gegenüber dem vorherigen Ausgabestand geändert, wird dies durch den veränderten Ausgabestand in der Kopfzeile der jeweiligen Seite angezeigt.

<b>Ausgabe</b>	<b>Bestell-Nr.</b>	<b>Bemerkung</b>
11.90	6ZB5 410-0DW01-0AA0	<b>A</b>
09.91	6ZB5 410-0DW01-0AA1	<b>C</b>
01.93	6ZB5 410-0DW01-0AA2	<b>C</b>
07.94	6ZB5 410-0DW01-0AA3	<b>C</b>

Geprüfte Siemens-Qualität für Software und Training  
nach DIN ISO 9001, Reg. Nr. 2160-01

Es können weitere, in dieser Dokumentation nicht beschriebene Funktionen in der Steuerung lauffähig sein. Es besteht jedoch kein Anspruch auf diese Funktionen bei Neulieferung bzw. im Servicefall.

Die Erstellung dieser Unterlage erfolgte mit dem Siemens-Bürosystem 5800 Office.  
Technische Änderungen vorbehalten.

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

# Vorbemerkung

## Lesehinweise

Die vorliegende Dokumentation wendet sich an den Hersteller von Werkzeugmaschinen mit SINUMERIK 810 Grundausführung 3 (GA3) oder SINUMERIK 820 GA3.

Die Druckschrift beschreibt ausführlich die Programmstruktur und den Befehlsvorrat der PLC (**P**rogrammable **L**ogic **C**ontrol). Ihnen wird erklärt, wie PLC-Grund- und Anwenderprogramme – bestehend aus Datenbausteinen, Funktionsbausteinen und Organisationsbausteinen – aufgebaut werden. An einigen Stellen in der Dokumentation wird unterschieden zwischen COP und ACOP. Folgender Zusammenhang ist gegeben:

COP :	SINUMERIK 810 T/M GA3 SW1
	SINUMERIK 820 T/M GA3 SW1
	SINUMERIK 810 N GA3 SW1
	SINUMERIK 820 N GA3 SW1
ACOP :	Sonstige SINUMERIK 810/820 GA3 Versionen

Die SINUMERIK-Dokumentation ist in 4 Ebenen gegliedert:

- Allgemeine-Dokumentation,
- Anwender-Dokumentation,
- Hersteller-Dokumentation und
- Service-Dokumentation.

Die **Hersteller-Dokumentation** für die Steuerung **SINUMERIK 810/820**, Grundausführung 3 ist in folgende Teile gegliedert:

- Betriebsanleitung,
- Nahtstelle
  - Teil 1: Signale
  - Teil 2: Anschlußbedingungen
- Programmieren PLC
- FB-PLC, Paket 1: Werkzeugverwaltung
- Werkstatt-ASM: Funktionsbausteine
  - Konturberechnung
  - Menübäume
  - Messen mit Bedienung im JOG-Betrieb
  - Technologieberechnung
  - Zyklen Bohren und Fräsen

Darüberhinaus gibt es SINUMERIK-Druckschriften, die für alle SINUMERIK-Steuerungen gelten (z. B. Universalschnittstelle, Meßzyklen, Zyklensprache CL-800).

Nähere Informationen erhalten Sie von Ihrer Siemens Zweigniederlassung.

Erläuterungen	1
Programmbausteine	2
Datenbausteine	3
Funktionsbausteine	4
Organisationsbausteine	5
Residente Funktionsbausteine	6
Übergabeparameter und Betriebssystem-Maschinendaten	7
Fehleranalyse	8
STEP 5-Befehlsvorrat mit Programmierbeispielen	9
Regeln zur Kompatibilität zwischen den Darstellungsarten KOP, FUP und AWL	10

# Inhalt

	Seite
<b>1 Erläuterungen</b>	<b>1-1</b>
1.1 Anwendungsbereich	1-1
1.2 Programmiersprache STEP 5	1-2
1.3 Programmierung	1-3
1.3.1 Programmstruktur	1-3
1.3.2 Programmorganisation	1-4
1.3.3 Programmbearbeitungsvarianten	1-5
1.3.4 Einflußfaktor: Anwenderprogrammlänge	1-6
<b>2 Programmbausteine</b>	<b>2-1</b>
2.1 Programmierung von Programmbausteinen	2-1
2.2 Aufruf von Programmbausteinen	2-2
<b>3 Datenbausteine</b>	<b>3-1</b>
3.1 Programmierung von Datenbausteinen	3-1
3.2 Aufruf von Datenbausteinen	3-2
<b>4 Funktionsbausteine</b>	<b>4-1</b>
4.1 Allgemeines	4-1
4.2 Aufbau von Funktionsbausteinen	4-2
4.2.1 Bausteinkopf	4-2
4.2.2 Bausteinrumpf	4-2
4.3 Aufruf und Parametrierung	4-3
4.3.1 Aufrufanweisung	4-3
4.3.2 Parameterliste	4-3
4.4 Programmierung von Funktionsbausteinen	4-4
4.4.1 Bibliotheksnummer	4-4
4.4.2 Name des Funktionsbausteins	4-4
4.4.3 Formaloperand (Name des Bausteinparameters)	4-5
4.4.4 Art der Bausteinparameter	4-6
<b>5 Organisationsbausteine</b>	<b>5-1</b>
5.1 Allgemeines	5-1
5.2 Zyklische Bearbeitungsebene	5-2
5.2.1 Unterbrechbarkeit	5-2
5.2.2 Reaktionszeit und Zykluszeit	5-3
5.2.3 Programmierung der zyklischen Bearbeitung	5-3
5.2.4 Schnittstelle zwischen Systemprogramm und zyklischer Bearbeitung	5-4
5.2.5 Grobgliederung des Programms	5-5

5.3	Alarmgesteuerte Bearbeitungsebene .....	5-7
5.3.1	Programmierung der Prozessalarm-Bearbeitung .....	5-7
5.3.2	Schnittstelle zwischen Systemprogramm und Prozessalarm-Bearbeitung .....	5-7
5.3.3	Reaktionszeit .....	5-8
<b>6</b>	<b>Residente Funktionsbausteine .....</b>	<b>6-1</b>
6.1	Allgemeines .....	6-1
6.2	Erläuterungen der FB-Bezeichnungen .....	6-2
6.3	Übersicht Funktionsmakros .....	6-3
	FB 11 EINR-DB Einrichten von Datenbausteinen .....	6-4
	FB 40 ... FB 55 siehe Projektierungsanleitung Paket 1: Werkzeugverwaltung (siehe Kapitel 6, Allgemeines)	
	FB 56 VORSCHUB Bestimmung des Vorschubwertes .....	6-6
	FB 57 SCHLEPPA Bestimmung des axialen Schleppabstandes .....	6-8
	FB 58 SPINDIST Bestimmung der Spindeldrehzahl .....	6-10
	FB 60 BLOCK-TR Blocktransfer .....	6-12
	FB 61 NCD-LESE NC-Daten lesen .....	6-14
	FB 62 NCD-SCHR NC-Daten schreiben .....	6-14
	FB 65 M STACK Übergabemerker in Merkerstack .....	6-25
	FB 66 STACK M Merkerstack in Übergabemerkerbereich .....	6-26
	FB 100 ... FB 104 siehe Projektierungsanleitung Grund- und Programmierungsfunktion für WF 725/726 in SINUMERIK 810/820 (siehe Kapitel 6, Allgemeines)	
	FB 190 K-LEITPC Signalaustausch mit Leit-PLC .....	6-27
<b>7</b>	<b>Übergabeparameter und Betriebssystem-Maschinendaten .....</b>	<b>7-1</b>
7.1	Übergabeparameter .....	7-1
7.2	PLC-Maschinendaten SINUMERIK 810 GA3/820 GA3 .....	7-3
7.3	PLC-Maschinendaten-Bits SINUMERIK 810 GA3/820 GA3 .....	7-4
<b>8</b>	<b>Fehleranalyse .....</b>	<b>8-1</b>
8.1	Fehlervarianten .....	8-1
8.2	Unterbrechungsstack .....	8-2
8.3	Fehlerfeincodierung .....	8-5
8.3.1	Anzeige über PG .....	8-5
8.3.2	Anzeige über NC-Bildschirm .....	8-7
8.4	Alarmliste .....	8-8

<b>9</b>	<b>STEP 5-Befehlsvorrat mit Programmierbeispielen</b> .....	9-1
9.1	Speicheraufteilung .....	9-1
9.1.1	Verändern der Segmentweiche .....	9-3
9.1.2	Bausteinlisten .....	9-3
9.2	Allgemeine Hinweise .....	9-4
9.2.1	Zahlendarstellungen .....	9-4
9.2.2	Ergebnisanzeigen der PLC .....	9-6
9.2.3	STEP 5-Befehlsdarstellung .....	9-7
9.3	Grundoperationen .....	9-9
9.3.1	Verknüpfungsoperationen, binär .....	9-9
9.3.2	Speicheroperationen .....	9-12
9.3.3	Lade- und Transferoperationen .....	9-15
9.3.4	Zeit- und Zähloperationen .....	9-17
9.3.5	Vergleichsoperationen .....	9-26
9.3.6	Bausteinaufrufe .....	9-30
9.3.7	Codeoperationen .....	9-32
9.3.8	Arithmetische Operationen .....	9-33
9.3.9	Sonstige Operationen .....	9-34
9.4	Ergänzende Operationen (nur FB) .....	9-34
9.4.1	Verknüpfungsoperationen, binär .....	9-35
9.4.2	Setzoperationen .....	9-35
9.4.3	Zeit- und Zähloperationen .....	9-36
9.4.4	Freigabeoperationen für Zeit- und Zähloperationen .....	9-38
9.4.5	Bitstestoperationen (nur FB) .....	9-39
9.4.6	Lade- und Transferoperationen .....	9-40
9.4.7	Verknüpfungsoperationen, digital .....	9-41
9.4.8	Schiebeoperationen .....	9-41
9.4.9	Umwandlungsoperationen .....	9-42
9.4.10	Dekrementieren/Inkrementieren .....	9-42
9.4.11	Sprungoperationen .....	9-43
9.4.12	Bearbeitungsoperationen .....	9-46
9.4.13	STEP 5-Befehle mit direktem Speicherzugriff .....	9-48
9.4.14	Sonstige Operationen .....	9-49
9.5	STEP 5-Befehlslaufzeiten .....	9-50
<b>10</b>	<b>Regeln zur Kompatibilität zwischen den Darstellungsarten KOP, FUP und AWL</b> .....	10-1
10.1	Allgemeines .....	10-1
10.2	Kompatibilitätsregeln bei graphischer Programmiereneingabe (KOP, FUP) .....	10-2
10.3	Kompatibilitätsregeln bei Programmiereneingabe in Anweisungsliste .....	10-4

# 1 Erläuterungen

## 1.1 Anwendungsbereich

Die PLC ist eine leistungsfähige speicherprogrammierbare Anpaßsteuerung. Sie verfügt über keine eigenständige Hardware und findet als "Software-PLC" ausschließlich in den SINUMERIK Steuerungen 810 GA2, 820 und SINUMERIK 810/820 GA3 Verwendung.

Die Aufgabe der PLC ist die Steuerung von maschinenbezogenen Funktionsabläufen, z. B.

- Steuern von Hilfsachsen,
- Steuern von Werkzeugwechseleinrichtungen und
- Erfassen der Signale der Überwachungseinrichtungen an der Maschine.

### **Besonderheiten:**

Bei der Steuerung SINUMERIK 810/820 GA3 werden alle NC- und PLC-Aufgaben von einem gemeinsamen Prozessor (Intel 80186) bearbeitet. Für die schnelle Ausführung von binären Befehlen<sup>1)</sup> des STEP 5-Anwenderprogramms (z. B. U E 3.0, S A 8.1, usw.) ist ein speziell dafür entwickelter Coprozessor (COP/ACOP) zuständig.

Die Ein-Prozessor-Struktur zwingt zu einer strengen Prioritierung der NC- und PLC-Aufgaben. An vorderster Stelle der Prioritäten stehen NC-Funktionen, wie Lageregelung und Satzaufbereitung, aber auch die PLC-Funktion "Alarmbearbeitung" (OB 2). Die Bearbeitung des zyklischen Programms (OB 1) ist dagegen von untergeordneter Bedeutung.

Damit auch in Extremfällen, wie langes STEP 5-Anwenderprogramm oder häufige Alarmbearbeitung, die wichtigsten Forderungen an die NC (hohe Positioniergenauigkeit, kurze Satzwechselzeiten) ordnungsgemäß erfüllt werden, ist die Prozessorbeanspruchung seitens der PLC auf maximal 20% der Gesamtzeit beschränkt. Darüberhinaus ist es möglich, über Maschinendaten die Art der Abarbeitung des zyklischen Anwenderprogramms zu beeinflussen (siehe Kapitel 1.3.4).

---

1) Der ACOP arbeitet auch die Mehrzahl der Wortbefehle ab.

## 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, abhängig von dem verwendeten Programmiergerät (PG), die drei Darstellungsarten Funktionsplan (FUP), Kontaktplan (KOP) und Anweisungsliste (AWL) (Bild 1.1) möglich, so daß die Programmiermethode dem jeweiligen Anwendungsfall angepaßt werden kann. Der von den Programmiergeräten erzeugte Maschinencode (MC5) ist bei den drei Darstellungsarten identisch. Bei Berücksichtigung bestimmter Programmierregeln (siehe Kapitel 10) kann, je nach verwendetem Programmiergerät, das Anwenderprogramm von einer Darstellungsart in eine andere übersetzt werden.

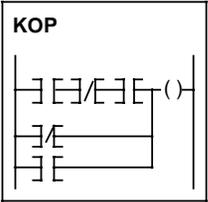
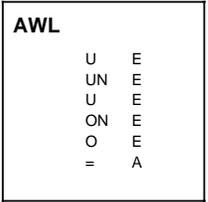
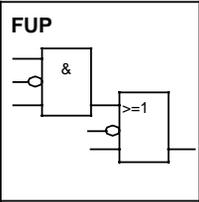
Kontaktplan	Anweisungsliste	Funktionsplan
Programmieren mit grafischen Symbolen wie Stromlaufplan  entspricht DIN 19239 (Entwurf)	Programmieren mit mnemotechnischen Abkürzungen der Funktionsbezeichnungen  entspricht DIN 19239 (Entwurf)	Programmieren mit grafischen Symbolen  entspricht IEC 117-15 DIN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

Bild 1.1 Darstellungsarten der Programmiersprache STEP 5

## 1.3 Programmierung

### 1.3.1 Programmstruktur

Das PLC-Gesamtprogramm besteht aus dem Systemprogramm, dem Grundprogramm und dem Anwenderprogramm. Das Systemprogramm enthält die Gesamtheit aller Anweisungen und Vereinbarungen geräteinterner Betriebsfunktionen. Das Grundprogramm hat eine flexible Grenze zum Systemprogramm, es besteht aus technologiespezifischen Funktionen und Grundfunktionen (z. B. Einrichten von Datenbausteinen, NC-PLC-Nahtstellenversorgung, Signalaus-tausch mit E/A-Modulen). Zusätzlich sind bewährte, in STEP 5 programmierte Funktionsbausteine, zu Funktionsmakros assembliert, in das Grundprogramm mit eingebunden.

Das System- und Grundprogramm ist ein fester Bestandteil der PLC, es wird als EPROM ausgeliefert und darf vom Anwender nicht verändert werden.

Das Anwenderprogramm ist die Gesamtheit aller vom Anwender programmierten Anweisungen und Vereinbarungen bzw. Daten.

Die Anpaßsteuerung PLC zwingt aufgrund ihrer 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 Änderungsmöglichkeiten,
- einfacher Programmtest und
- 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, bevorzugt zur Bearbeitung von Ablaufketten.
- Datenbausteine (DB)  
Sie dienen zum Abspeichern von Daten oder Texten. Dieser Bausteintyp unterscheidet sich in seiner Funktion und seinem Aufbau grundsätzlich von allen anderen Bausteinen.

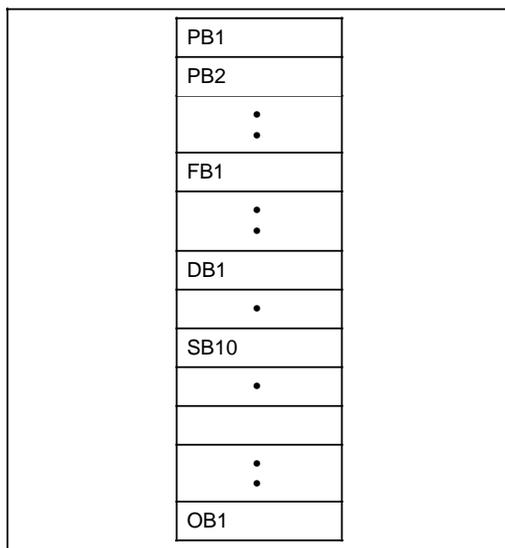
Von jedem Bausteintyp – mit Ausnahme der Organisationsbausteine – können maximal 255 Bausteine programmiert werden. Die Anzahl der Organisationsbausteine beträgt 64 wobei jedoch nur die Organisationsbausteine OB 20, OB 1, OB 2 vom Systemprogramm bedient werden (siehe Kapitel 5).

Alle programmierten Bausteine werden vom Programmiergerät in beliebiger Reihenfolge im Programmspeicher bzw. Datenbausteinspeicher hinterlegt (Bild 1.2).

### 1.3.2 Programmorganisation

Mit der Programmorganisation wird festgelegt, ob und in welcher Reihenfolge die Programm-, Funktions- und Schrittbausteine bearbeitet werden. Dazu werden in Organisationsbausteinen entsprechende Aufrufe (bedingt oder unbedingt) der gewünschten Bausteine programmiert (siehe Kapitel 5.4).

Die Organisationsbausteine werden wie die anderen Bausteine ebenfalls im Anwenderspeicher hinterlegt.



Für die verschiedenen Möglichkeiten der Programmbearbeitung sind unterschiedliche Organisationsbausteine vorgesehen (siehe Abschnitt 5.2).

Von Organisations-, Programm-, Funktions- und Schrittbausteinen können weitere Programm-, Funktions- und Schrittbausteine aufgerufen werden. Organisationsbausteine können vom Anwenderprogramm nicht aufgerufen werden.

Die maximal zulässige Schachtelungstiefe beträgt mit dem Organisationsbaustein 12 Bausteine (Bild 1.3), wobei ein verwendeter Datenbaustein nicht mitgezählt wird.

Bild 1.2 Ablage der Bausteine in beliebiger Reihenfolge in den Programmspeicher

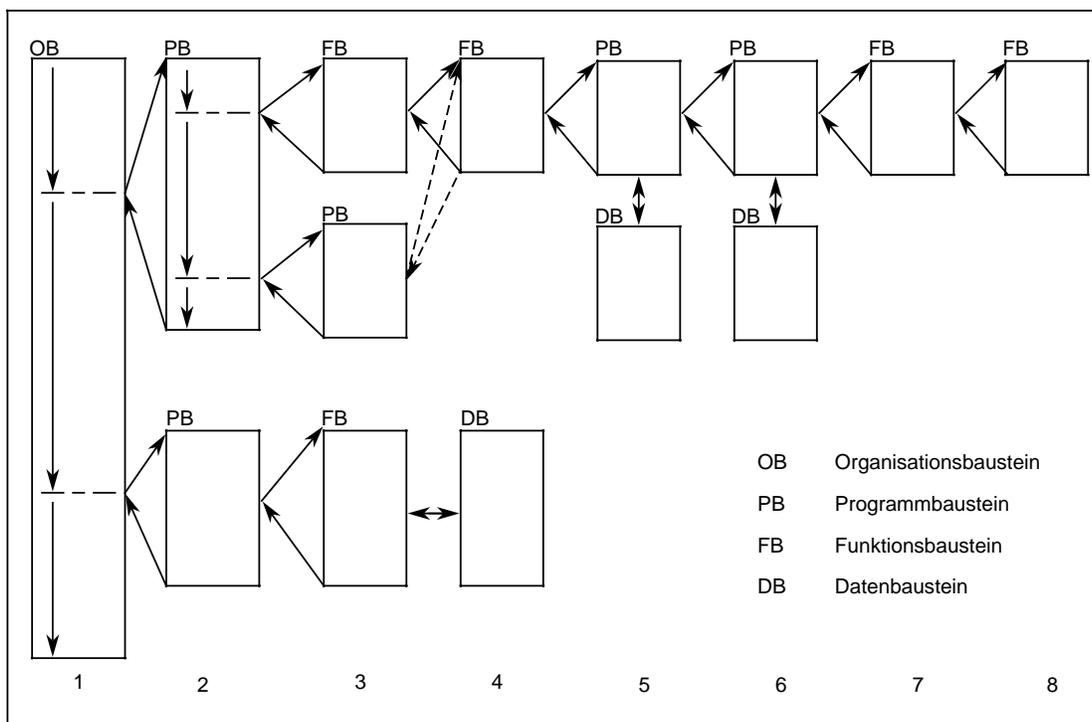


Bild 1.3 Beispiel einer Programmorganisation in der Programmiersprache STEP 5 Schachtelungstiefe=8

### 1.3.3 Programmbearbeitungsvarianten

Das Anwenderprogramm kann auf 3 Arten bearbeitet werden:

- 1) einmalige Programmbearbeitung im Anlaufzweig der Steuerung (nach "Netz ein" bzw. "RESET"). Die Anweisungen müssen innerhalb des OB 20 programmiert werden.
- 2) zyklische Bearbeitung der im OB 1 hinterlegten Anweisungen in einem Zeitraster, das von MD 155 abhängig ist. Bei der Standardeinstellung beträgt das minimale Aufrufraster des OB 1 durch das Systemprogramm 60 ms, ansonsten ergeben sich die nachfolgenden Werte:

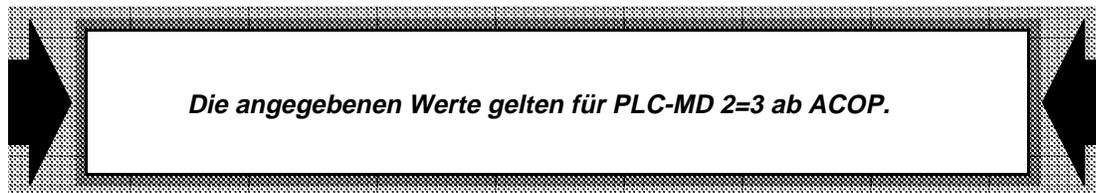
MD 155	Aufrufraster OB 1, wenn PLC-MD 2=1 <sup>1)</sup>	Aufrufraster OB 1, wenn PLC-MD 2=2 <sup>1)</sup>	Aufrufraster OB 1, wenn PLC-MD 2=3 <sup>1)</sup> (Standardeinstellung)
0	16 ms	32 ms	48 ms
1	18 ms	36 ms	54 ms
2	20 ms	40 ms	60 ms
3	22 ms	44 ms	66 ms
4	24 ms	48 ms	72 ms
5	26 ms	52 ms	78 ms

- 3) ereignisgesteuerte, einmalige Programmbearbeitung der im OB 2 stehenden Anweisungen z. B. aufgrund eines Alarmes. Voraussetzung hierfür sind:
  - abgeschlossener Anlauf der Steuerung
  - Freigabe der Alarmbearbeitungsebene über MD (MD 2002, Bit 0 =0)
  - eine Änderung in dem per MD festgelegten Alarmeingangsbyte (PLC-MD 0).

Die zyklische Programmbearbeitung kann am Ende jedes STEP 5-Befehls von der alarmgesteuerten Programmbearbeitung von höherpriorären NC-Funktionen unterbrochen bzw. der Beginn bis zur Beendigung des OB 2 verzögert werden.

1) gilt nur für ACOP

### 1.3.4 Einflußfaktor: Anwenderprogrammlänge



Wie bereits in Kapitel 1.1 erwähnt, sollte die CPU-Belastung durch die STEP 5-Programmbearbeitung einen Wert von 15 % nicht überschreiten. Zusätzlich darf das alarmgesteuerte Anwenderprogramm im OB 2 aufgrund seiner höchsten Prioritätseinstufung eine bestimmte Laufzeit nicht überschreiten, weil ansonsten wichtige NC-Funktionen nur mit einer unzulässigen Verzögerung zum Ablauf kommen würden.

Zu diesem Zweck werden die STEP 5-Programmlaufzeiten in der zyklischen und alarmgesteuerten Bearbeitungsebene hardwaremäßig gemessen. Die jeweils zulässigen Laufzeitwerte kann man über die PLC-MD-Worte 1 und 3 einstellen, wobei bei der Standardeinstellung eine unzulässige CPU-Belastung durch die "integrierte PLC" ausgeschlossen wird. Bei einer Überschreitung dieser Werte wird in einen per MD 2003, Bit 0 bzw. Bit 1 festlegbaren PLC-Betriebszustand verzweigt. Bei der Standardeinstellung ist dies jeweils der Stoppzustand der "integrierten PLC".

Überschreiten die Anwenderprogramme nur geringfügig die zulässigen Laufzeitwerte, so kann durch Änderung dieser Maschinendatenbits die Programmbearbeitung weiterhin aufrecht erhalten werden. Eine Beeinträchtigung der NC-Funktionen kann dann aber die Konsequenz sein.

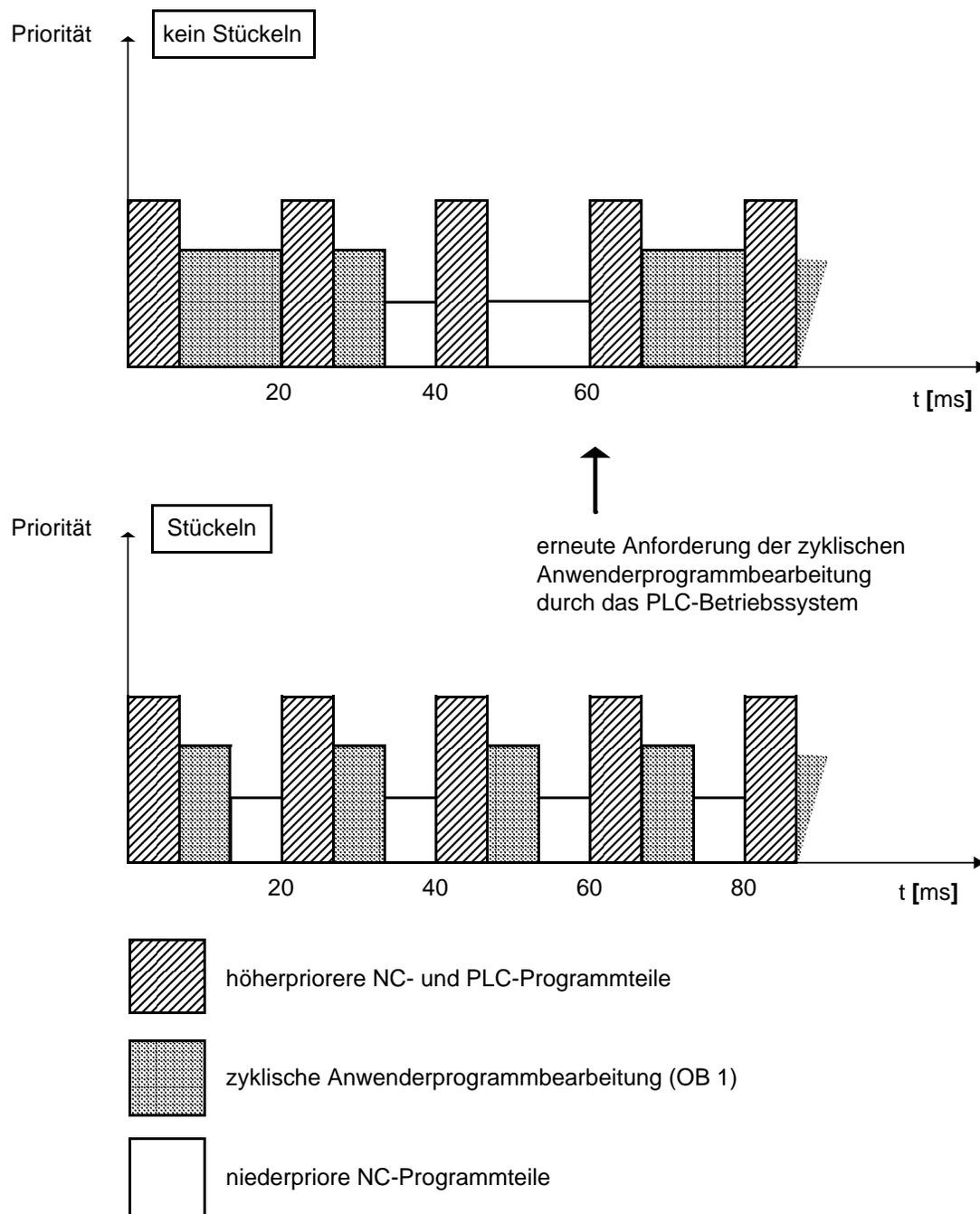
Damit auch lange zyklische Anwenderprogramme (bis max. 8 K-Worte ohne PLC-Speichererweiterung) ablauffähig sind ohne die CPU-Belastung nennenswert zu überschreiten, gibt es zwei Bearbeitungsvarianten:

- Stückeln der zyklischen Bearbeitung durch eigenständiges Unterbrechen
- zyklische Bearbeitung kann nur durch höherprioritäre Programmteile unterbrochen werden

Über PLC-MD 2003, Bit 6 ist die jeweilige Bearbeitungsart einzustellen. Der Prozentsatz von MD 1001 erhält damit folgende Laufzeitbedeutung:

- kein Stückeln (MD 2003, Bit 6=0 Standardeinstellung):  
15 % von dem Aufrufaster der zyklischen Programmbearbeitungsebene ergibt bei der Standardeinstellung (Aufrufaster: alle 60 ms) einen zulässigen Laufzeitwert von ca. 9 ms.
- Stückeln (MD 2003, Bit 6=1):  
15 % von 1/3 des Aufrufasters des zyklischen Bearbeitungsteils ergeben bei der Standardeinstellung (Aufrufaster: alle 60 ms) einen Laufzeitwert von ca. 3 ms. Danach unterbricht sich die zyklische Bearbeitung selbständig, um im nächsten Drittel erneut mit der Programmbearbeitung fortzufahren.

Grafische Übersicht über die beiden Varianten der zyklischen Anwenderprogrammbearbeitung:



Das Aufraster der zyklischen Programmbearbeitung beträgt wie bereits erwähnt bei der Standardeinstellung  $12 \cdot \text{Abtastzeit}$  ( $12 \cdot 5 \text{ ms}$ ).

Durch Vergrößern von PLC-MD 1 (CPU-Belastung in %) kann die zulässige Laufzeit in der zyklischen Bearbeitungsebene für beide Bearbeitungsvarianten vergrößert werden.

Für die zulässige Laufzeit in der alarmgesteuerten Bearbeitungsebene gilt (PLC-MD 3), daß dieser Wert unabhängig von der zyklischen Bearbeitungsvariante in einem Zeitintervall von  $4 \cdot \text{Abtastzeit}$  ( $4 \cdot 5 \text{ ms} = 20 \text{ ms}$ ) einen Laufzeitwert von  $2000 \mu\text{s}$  nicht überschreiten darf.

Diese Laufzeit kann entweder von einer einmaligen OB 2-Bearbeitung beansprucht werden bzw. durch eine im Extremfall 4malige OB 2-Bearbeitung, weil im Zeitraster der Abtastzeit die Abfrage einer Flankenänderung im Alarmeingangsbyte erfolgt. Reicht der Standardlaufzeitwert im OB 2 nicht aus, muß entweder PLC-MD 3 vergrößert werden (maximal auf  $2500 \mu\text{s}$ ) oder durch Veränderung von MD 2003 Bit 1 der standardmäßige Stoppzustand bei einer Laufzeitüberschreitung im OB 2 abgewählt werden. Das abschließende Rechenbeispiel soll bei einem zyklischen Anwenderprogramm von  $25 \text{ ms}$  Laufzeit noch einmal zahlenmäßig die Unterschiede der beiden Bearbeitungsvarianten herausstellen.

#### **Kein Stückeln angewählt:**

Die zyklische Anwenderprogrammbearbeitung kann lediglich von höherprioren NC- und PLC-Programmteilen unterbrochen werden (z. B. Lageregelung, Interpolation, Überwachungsfunktionen, Programmier- und Testfunktionen, alarmgesteuerte Anwenderprogrammbearbeitung (OB 2)). Ansonsten wird sie solange fortgesetzt, bis das zyklische Anwenderprogramm und das Hinterlegen der erzeugten Signale an ihren Bestimmungsorten (NC, Peripherie) abgeschlossen ist. Erst danach können die niederprioren Funktionen zum Ablauf kommen.

Da nach 60 ms die zyklische Anwenderprogrammbearbeitung erneut vom Betriebssystem angefordert wird, kann dieses verbleibende Zeitintervall in Abhängigkeit von der Länge des STEP 5-Programms im OB 1 sehr klein werden. Die Konsequenz ist, daß für die niederprioreren Funktionen zu wenig Zeit übrig bleibt. Nach außen hin macht sich dieser Umstand durch eine sehr träge "Bedieneroberfläche" sofort bemerkbar. Der Bildaufbau erfolgt sehr langsam, ebenso die Reaktion der Steuerung aufgrund von Tastenbetätigungen. Im Extremfall kommen die niederprioreren Funktionen fast gar nicht mehr zum Ablauf.

Die CPU-Belastung alleine durch die STEP 5-Programmbearbeitung würde in dem gewählten Beispiel auf einen Wert x von

$$x = \frac{25 \text{ ms}}{60 \text{ ms}} \cdot 100 \% = 41,7\% \text{ ansteigen, vorausgesetzt, die zyklische Bearbeitung}$$

wird innerhalb eines Aufrasterers von 60 ms beendet. Ist dies nicht der Fall, wird im nächsten Zeitintervall die Bearbeitung beendet und – vorausgesetzt es stehen keine höherprioreren Funktionen an – sofort wieder neu gestartet. In dem gewählten Beispiel ist die stets auftretende Grundlast aufgrund des Datentransfers vor und nach dem OB 1-Aufruf von etwa 5 % noch unberücksichtigt geblieben.

Man sieht sehr leicht ein, daß eine CPU-Belastung von fast 47 % für die Gesamtfunktion der Steuerung nicht akzeptabel ist.

#### **Stückeln:**

Bei dieser Bearbeitungsvariante unterbricht sich die zyklische Bearbeitung selbständig. Die CPU-Belastung beträgt bei Standardeinstellung auf ein Aufraster der zyklischen Bearbeitung bezogen:

$$x = \frac{3 \text{ ms} \cdot 3}{60 \text{ ms}} \cdot 100 \% = 15 \%$$

Hinzu käme noch die zyklische Grundlast von etwa 5 %, so daß die gesamte zyklische CPU-Belastung stets 20 % beträgt. Kleiner als 20 % wird der Wert, wenn die zyklische Bearbeitung nicht in einem Aufraster beendet wird. Bei sehr langen zyklischen Anwenderprogrammen kann es zum Ansprechen der Zykluszeitüberwachung (Standardeinstellung: 70 ms) kommen.

Die maximale einstellbare Zykluszeitüberwachung ist 320 ms (PLC MD 5).

Durch Anwählen der Diagnosefunktion (MD 2003 Bit 7=1) kann sich der Anwender einen Überblick über die zu erwartenden Laufzeiten seiner STEP 5-Programme sowie der auftretenden Zykluszeit verschaffen. Er kann an Hand dieser Ergebnisse die Konfiguration seiner MD so einstellen, daß ein Optimum erreicht wird, was seine Anforderungen und den Verwendungszweck der Steuerung betrifft.

### Anstoßmöglichkeiten der Diagnosefunktion

1) einmalige Anzeige über die PG-Funktion "STATUS VARIABLE"

```
STATUS  VARIABLE
DB  1
DW  0    KF = <aktuelle Zykluszeit in ms>
DW  1    KF = <Interpreterlaufzeit im OB 1, OB 2 in µs> 1)
DW  2    KF = <zyklische Interpreterlaufzeit in µs> 1)
DW  3    KF = <alarmgesteuerte Interpreterlaufzeit in µs> 1)

DW  9    KH = 8000
```

**Beachte:**

Das Anstoßbit in DW 9 wird einmalig über das PG gesetzt und vom PLC-Betriebssystem zu Beginn des Zyklus zurückgesetzt, wenn die aktuellen Werte in den DB 1 übertragen werden.

2) periodische Anzeige über die PG-Funktion "STATUS" und ein entsprechendes Anstoßprogramm im OB 1

```
OB1
A  DB1
U  M 0.1
=  D 9.15
} Anstossprogramm

:
:
} eigentliches
Anwenderprogramm

BE
```

**Beachte:**

Auf diese Art lassen sich die Schwankungen der Laufzeiten gut abschätzen. (Zykluszeit sowie Interpreterlaufzeit sind keine konstanten Größen!) Außerdem ist nur so die alarmgesteuerte Interpreterlaufzeit (OB 2) sowie die Anzahl der aufgetretenen Alarmbearbeitungen darstellbar (z. B. dadurch, daß eine Änderung im Alarmeingangsbyte hardwaremäßig simuliert wird).

Überschreitet die Interpreterlaufzeit einen Wert von 32768 µs, so muß am PG die Zahlenformatdarstellung von KF auf KH geändert werden, ansonsten erscheint eine fehlerhafte Anzeige.

1) Werte sind auf die aktuelle Zykluszeit bezogen.

## 2 Programmbausteine

### 2.1 Programmierung von Programmbausteinen

Die folgende Beschreibung gilt für die Programmierung von Organisations-, Programm- und Schrittbausteinen. Diese drei Bausteintypen unterscheiden sich hinsichtlich ihrer Programmierung nicht (Programmierung von Datenbausteinen siehe Kapitel 3, Programmierung von Funktionsbausteinen siehe Kapitel 4). Sie können in den drei Darstellungsarten (AWL, KOP, FUP) der Programmiersprache STEP 5 mit den Grundoperationen programmiert werden.

Die Programmierung eines Programmbausteins (PB) beginnt mit der Angabe einer Programmbaustein-Nummer zwischen 0 und 255 (Beispiel: PB 25). Danach folgt das eigentliche Steuerungsprogramm, das mit der Anweisung "BE" abgeschlossen wird.

Ein S5-Baustein besteht aus zwei Teilen:

- Bausteinkopf
- S5-Befehlsteil (Bausteinrumpf)

Der Bausteinkopf, den das Programmiergerät automatisch zum Programmbaustein generiert, belegt 5 Worte im Programmspeicher.

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

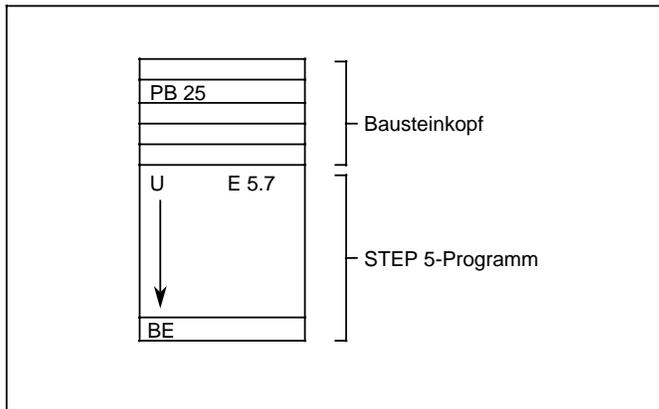


Bild 2.1 Aufbau eines Programmbausteins

## 2.2 Aufruf von Programmbausteinen

Durch Bausteinaufrufe werden die Bausteine zum Bearbeiten freigegeben (Bild 2.2).

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.

### 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.

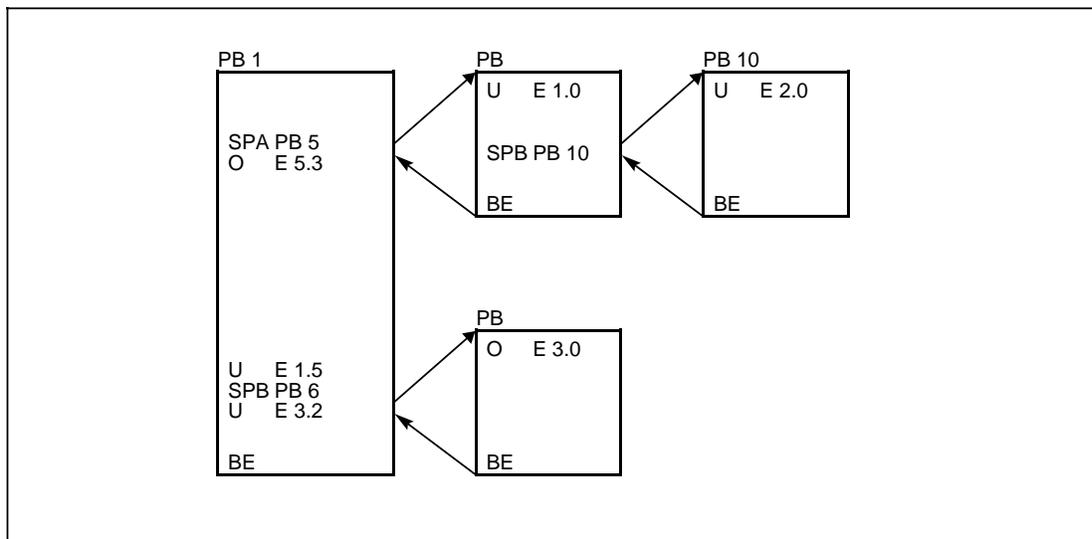


Bild 2.2 Bausteinaufrufe, die die Bearbeitung eines Programmbausteins freigeben

## 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 mit dem PG beginnt mit der Angabe einer Datenbaustein-Nummer zwischen 1 und 255. Jeder Datenbaustein (Beispiel: DB 99) kann aus bis zu 2k Datenworten (16 Bit) bestehen (Bild 3.1). Das Eingeben von Daten muß in aufsteigender Reihenfolge der Datenworte, beginnend mit Datenwort 0 erfolgen.

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.

Die Datenbausteine DB 1 (Diagnose-DB), DB 36 (Zustand der Auftragsabwicklung für NC-Daten lesen bzw. schreiben), DB 37 (Anstoß der beiden V.24-Schnittstellen), DB 39 (Option Software-Nocken), DB 40 (Anzeigefunktionen) und optional DB 71 (ADAR-Interface Kopplung) werden vom Systemprogramm automatisch im Anlauf der Steuerung eingerichtet. Das Programmiergerät (PG) verhindert das Löschen dieser Bausteine (Meldung 70: DB im EPROM).

Über die Programmiergerätefunktion "Steuern Variable" läßt sich der Inhalt aber zu Testzwecken anschauen bzw. verändern (gilt nicht bei vom System eingerichteten Datenbausteinen).

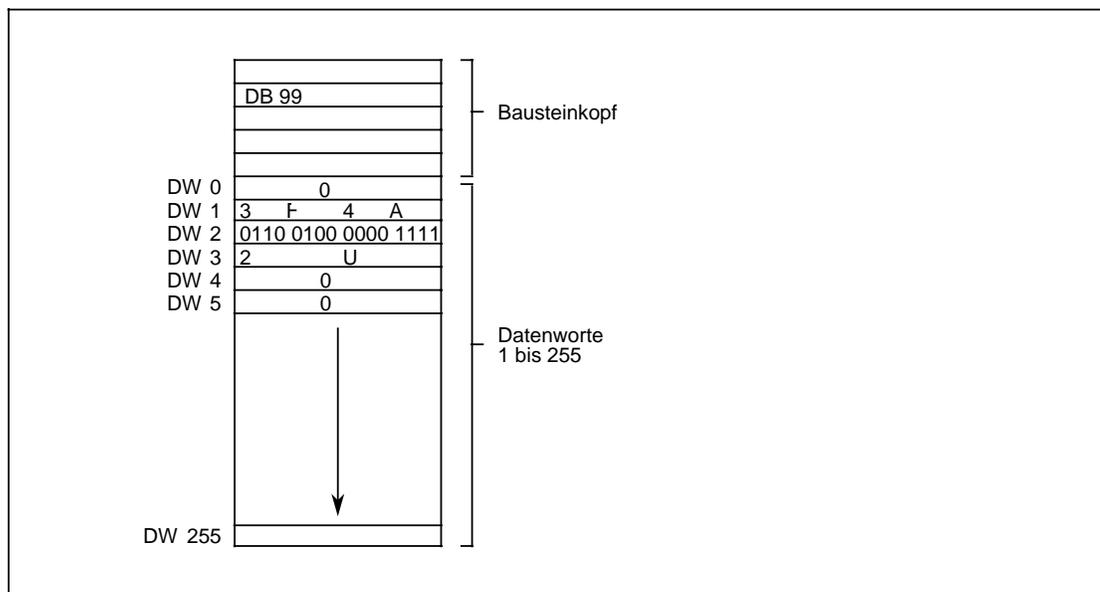


Bild 3.1 Aufbau eines Datenbausteins

### 3.2 Aufruf von Datenbausteinen

Datenbausteine können nur unbedingt aufgerufen werden. Die Anwahl eines Datenbausteins bleibt solange gültig, bis ein neuer Datenbaustein aufgerufen wird.

Anwender-Datenbausteine sollten nicht mit den vom System benötigten Datenbausteinen kollidieren.

Der Aufruf eines Datenbausteins kann innerhalb eines Organisations-, Programm-, Funktions- oder Schrittbausteins programmiert werden. Er erfolgt mit dem Befehl

"A DB xxx" .

#### Beispiel 1:

Es soll der Inhalt des Datenwortes 1 vom Datenbaustein 10 in das Datenwort 1 des Datenbausteins 20 transferiert werden (Bild 3.2).

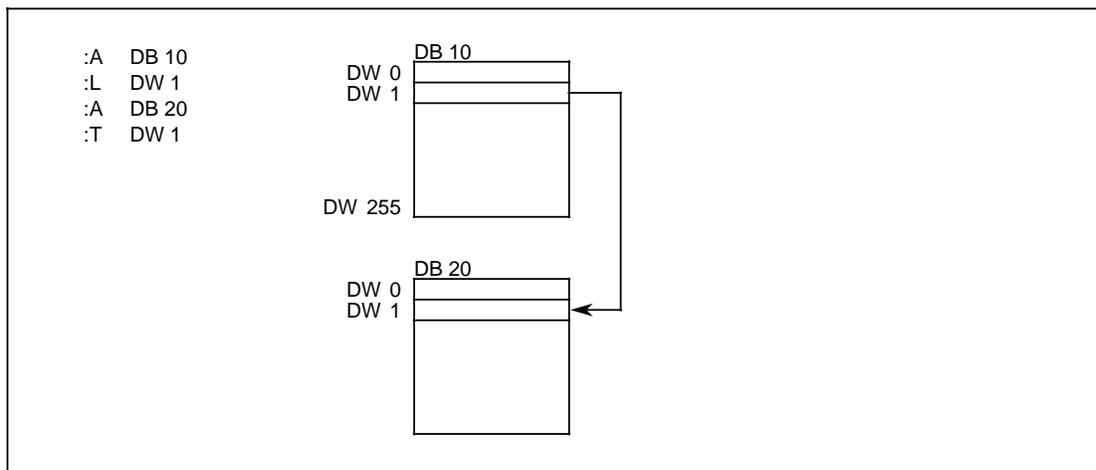


Bild 3.2 Aufruf eines Datenbausteins

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 aufrufenden Programmbaustein gilt wieder der alte Datenbaustein (Bild 3.3).

**Beispiel 2:**

Im Programmbaustein 7 wird der Datenbaustein 10 aufgerufen. 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 Aufschlagen 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.

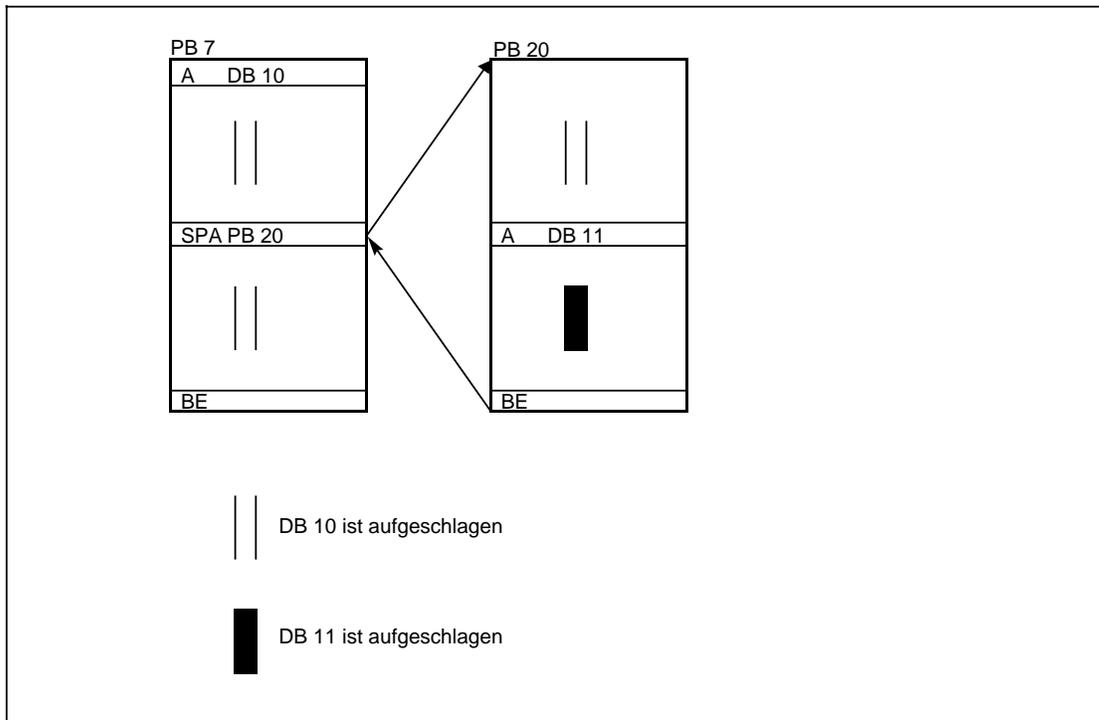


Bild 3.3 Gültigkeitsbereich eines angewählten Datenbausteins

## 4 Funktionsbausteine

### 4.1 Allgemeines

Mit den Funktionsbausteinen werden häufig wiederkehrende oder sehr komplexe Funktionen realisiert.

Funktionsbausteine (FB) sind ebenso Teile des Anwenderprogramms wie z. B. Programm- bausteine. Sie weisen jedoch gegenüber den Organisations-, Programm- und Schrittbausteinen drei wesentliche Unterschiede auf:

- Funktionsbausteine können parametrisiert werden, d. h. es lassen sich die Aktualoperanden, mit denen ein Funktionsbaustein gerufen wird, an die Formalparameter des Funktionsbausteins übergeben.
- Funktionsbausteine können mit einem gegenüber den Organisations-, Programm- und Schrittbausteinen erweiterten Operationsvorrat programmiert werden. Dieser erweiterte Operationsvorrat besteht aus den ergänzenden Operationen der Programmiersprache STEP 5 (siehe auch Kapitel 9.4), die jedoch nur in Funktionsbausteinen programmiert werden können.
- Das Programm eines Funktionsbausteines lässt sich nur als Anweisungsliste erstellen und dokumentieren.

Funktionsbausteine stellen innerhalb des Anwenderprogramms eine komplexe, abgeschlossene Funktion dar. Ein in der Programmiersprache STEP 5 erstellter Funktionsbaustein kann vom Anwender selbst programmiert oder als Softwareprodukt von Siemens bezogen werden. Zusätzlich hat man auch bewährte, technologiespezifische Funktionsbausteine, die in STEP 5 programmiert und erprobt wurden, zu Funktionsmakros assembliert und in das Grundprogramm eingebunden. Sie können vom Anwender wie Funktionsbausteine aufgerufen, aber nicht verändert werden. Man bezeichnet sie auch als "residente Assemblerfunktionsbausteine" (siehe Kapitel 6.1).

## 4.2 Aufbau von Funktionsbausteinen

Ein Funktionsbaustein besteht aus dem Bausteinkopf, der Namen und Parameterdeklaration und dem Bausteinrumpf (Bild 4.1).

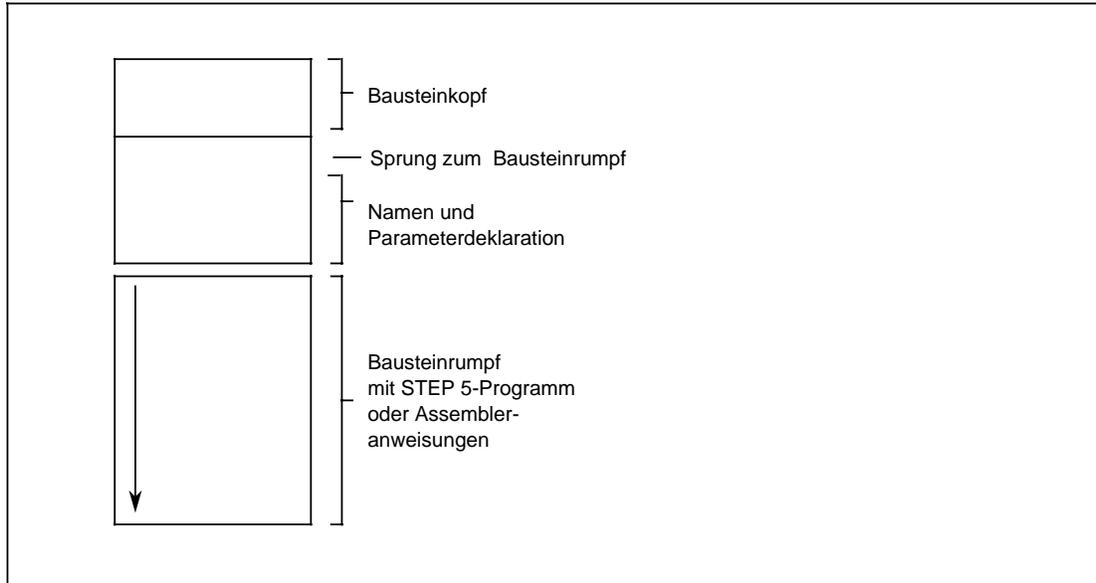


Bild 4.1 Aufbau eines Funktionsbausteins

### 4.2.1 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.

### 4.2.2 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.

Bei dem Aufruf von residenten Assembler-Funktionsbausteinen werden der Bausteinname und die Parameterdeklaration vom Programmiergerät wiedergegeben.

Im Bausteinrumpf bewirkt der STEP 5-Befehl "ASM" (Umschalten auf Assemblercode) als "erste ausführbare Anweisung" des Bausteins, daß der Prozessor die nachfolgenden Assembleranweisungen direkt bearbeitet.

## 4.3 Aufruf und Parametrierung

Funktionsbausteine (FB) stehen nur einmal im Speicher. Sie können von einem übergeordneten Baustein einmal oder mehrfach aufgerufen werden, wobei bei jedem Aufruf andere Parameter verwendet werden können.

Funktionsbausteine werden unter Angabe einer Bausteinnummer programmiert bzw. aufgerufen (FB 0 bis 255).

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

### 4.3.1 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.

### 4.3.2 Parameterliste

Die Parameterliste steht direkt nach der Aufrufanweisung (Bild 4.2). In ihr werden die Eingangs-, Ausgangsvariablen und Daten definiert. 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 Parameter- liste 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 Worte, jeder Parameter ein weiteres Speicherwort.

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 oder im Grundprogramm der Anpaßsteuerung als Funktionsmakro vorhanden sein (Näheres siehe Bedienungsanleitung).



### 4.4.3 Formaloperand (Name des Bausteinparameters)

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

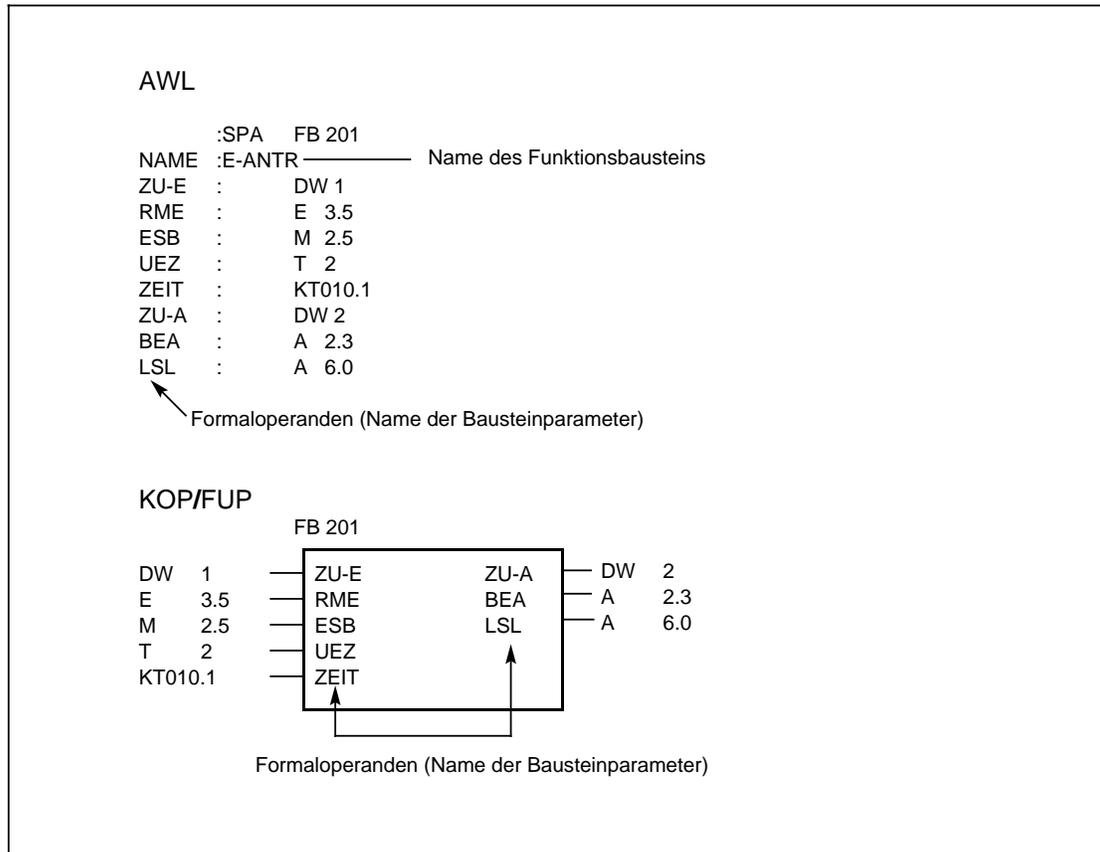


Bild 4.3 Beispiel für einen Funktionsbausteinaufruf



## 5 Organisationsbausteine

### 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, sondern nur programmieren (Bild 5.1).

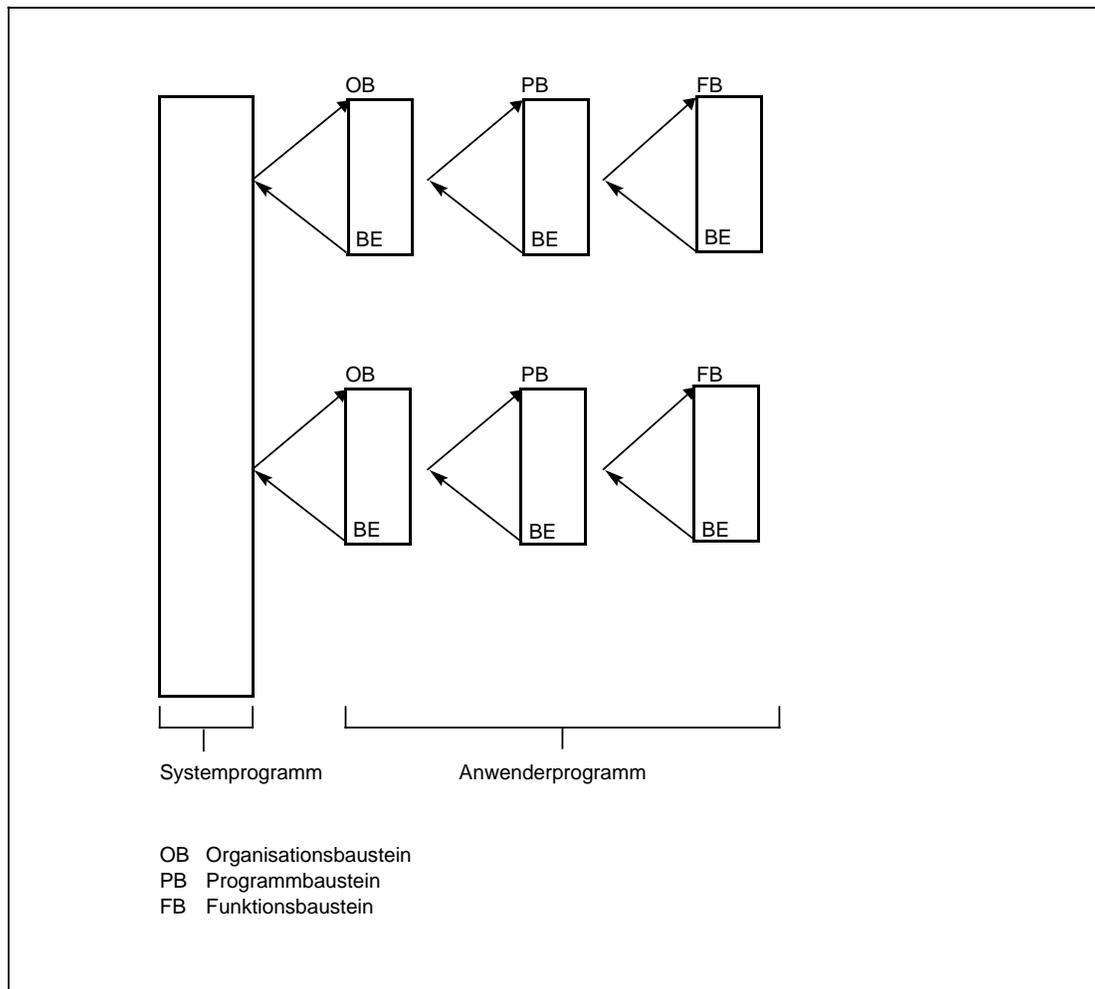


Bild 5.1 Gesamtprogramm einer Anpaßsteuerung

Bei entsprechender Programmierung der Organisationsbausteine können folgende Bearbeitungen durchgeführt werden:

- einmalige Bearbeitung nach Anlauf der Steuerung (OB 20)
- zyklische Bearbeitung (OB 1)  
(siehe "Programmierung der zyklischen Bearbeitung" Seite 5-6)
- alarmgesteuerte Bearbeitung (OB 2)  
(siehe "Programmierung der Prozeßalarm-Bearbeitung" Seite 5-10)

Der Organisationsbaustein OB 2 muß bei Freigabe der Alarmbearbeitung (MD 2002 Bit 0=0) stets vorhanden sein, andernfalls geht die PLC in den Stoppzustand. Die Organisationsbausteine OB 20 und OB 1 können je nach Bedarf in die Steuerung eingegeben werden. Fehlt einer der beiden OBs, so wird dies vom Betriebssystem ignoriert.

Die Organisationsbausteine werden wie Programm- oder Schrittbausteine programmiert, die Programmierung und die Dokumentation ist in allen drei Darstellungsarten (Anweisungsliste AWL, Funktionsplan FUP, Kontaktplan KOP) möglich.

## **5.2 Zyklische Bearbeitungsebene**

### **5.2.1 Unterbrechbarkeit**

Das zyklische Anwenderprogramm kann nach jedem MC5-Befehl unterbrochen werden. Der letzte Befehl wird noch vollständig abgearbeitet, am Ende des Befehls werden anstehende Interrupts (aufgrund eines anstehenden Prozeßalarms bzw. höherpriorer NC-Programmteile) freigegeben.

Bei Unterbrechung aufgrund einer erkannten Flankenänderung im Alarmeingangsbyte werden sämtliche MC5-Register gerettet und nach Beendigung wird der alte Zustand wieder hergestellt. So erhält auch der Datenbaustein wieder seine Gültigkeit, der vor dem Unterbrechungsereignis aufgeschlagen war.

## 5.2.2 Reaktionszeit und Zykluszeit

### a) Reaktionszeit:

Die Reaktionszeit auf ein Prozeßsignal bzw. ein Signal der NC ist folgendermaßen definiert:

Einlesen des aktuellen Signals durch das PLC-Betriebssystem in die Anwenderschnittstelle (E, M-Bereich), Aufruf und Bearbeitung des zyklischen Anwenderprogrammes (OB 1), Transferieren des durch den OB 1 aktualisierten Prozeßausgangsabbildes an Peripherie und NC.

Die Reaktion ist abhängig von:

- 1) Art und Länge des zyklischen Anwenderprogrammes in OB 1
- 2) Bearbeitungsvariante des zyklischen Anwenderprogrammes (MD 2003 Bit 6: Stückeln, kein Stückeln)
- 3) Aufrufraster der zyklischen Anwenderprogrammbearbeitung durch das PLC-Betriebssystem (MD 155: Abtastzeit)
- 4) Laufzeit der höherpriorien NC-Programmteile sowie einer eventuellen Anwenderalarmbearbeitung (OB 2)
- 5) Zeitpunkt: Prozeßabbildaktualisierung durch das PLC-Betriebssystem und tatsächlicher Signalzustandswechsel (Peripherie, interne NC-PLC-Schnittstelle)

### b) Zykluszeit:

Die Zykluszeit ist die Zeitspanne, die zwischen zwei Aktualisierungen des Eingangsprozeßabbildes durch das PLC-Betriebssystem verstreicht. Da die zyklische Anwenderprogrammbearbeitung nicht unmittelbar nach ihrer Beendigung erneut gestartet wird, sondern in einem bestimmten Aufrufraster (abhängig von MD 155) durch das PLC-Betriebssystem aufgerufen wird, kann sie nie kleiner als 60 ms werden (Standardeinstellung: 60 ms).

Die Zykluszeit ist wie die Reaktionszeit von den Punkten 1 ... 4 abhängig. Sie läßt sich über die Diagnosefunktion (MD 2003 Bit 7=1) messen und wird auf die Einhaltung eines Maximalwertes (PLC-MD 5-Standardeinstellung 70 ms) überwacht. Ansonsten verzweigt die PLC in die Stoppschleife.

## 5.2.3 Programmierung der zyklischen Bearbeitung

Die zyklische Bearbeitung ist die "normale" Bearbeitung bei speicherprogrammierbaren Steuerungen (Bild 5.4). 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.

## 5.2.4 Schnittstelle zwischen Systemprogramm und zyklischer Bearbeitung

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 Programmstart des zyklischen Programms.

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 Kapitel 1.3.2).

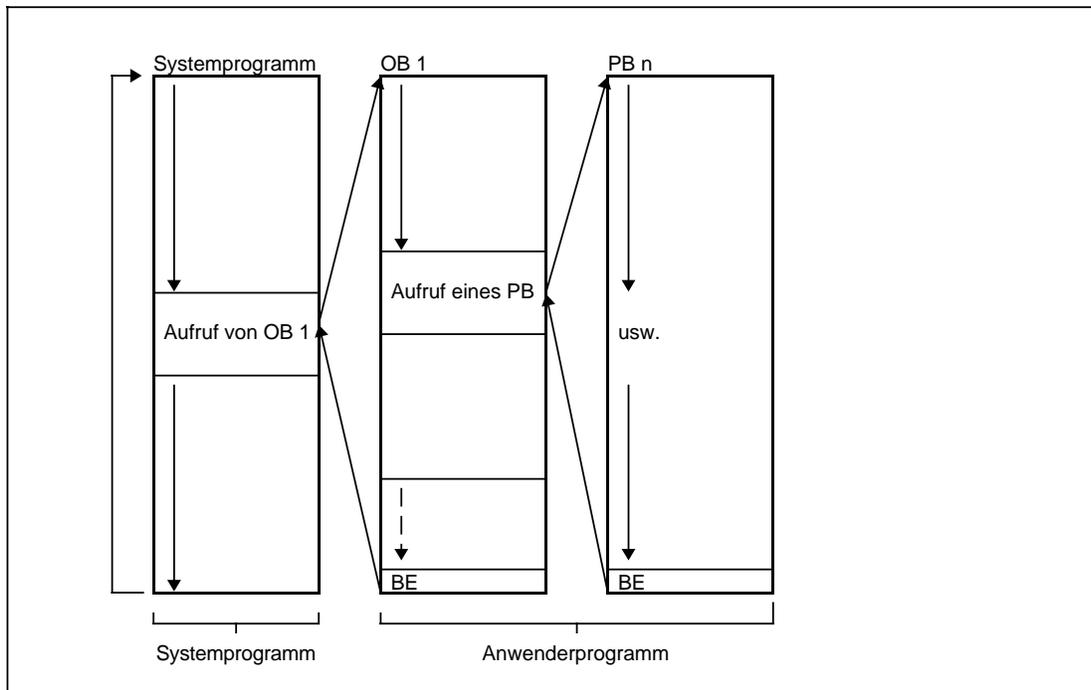


Bild 5.4 Zyklische Programmbearbeitung

Erste Anweisung des STEP 5-Programms.

Erster Aufruf eines Programmbausteins. In diesem Baustein können auch weitere Aufrufe stehen (siehe Kapitel 1.3.2).

Rücksprung vom letzten bearbeitenden Programm- oder Funktionsbaustein.

Der Organisationsbaustein wird mit BE abgeschlossen.

Rücksprung ins Systemprogramm.

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.

## 5.2.5 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 5.5) bzw. programmtechnisch zusammenhängende Anlagenteile hervorheben (Bild 5.6).

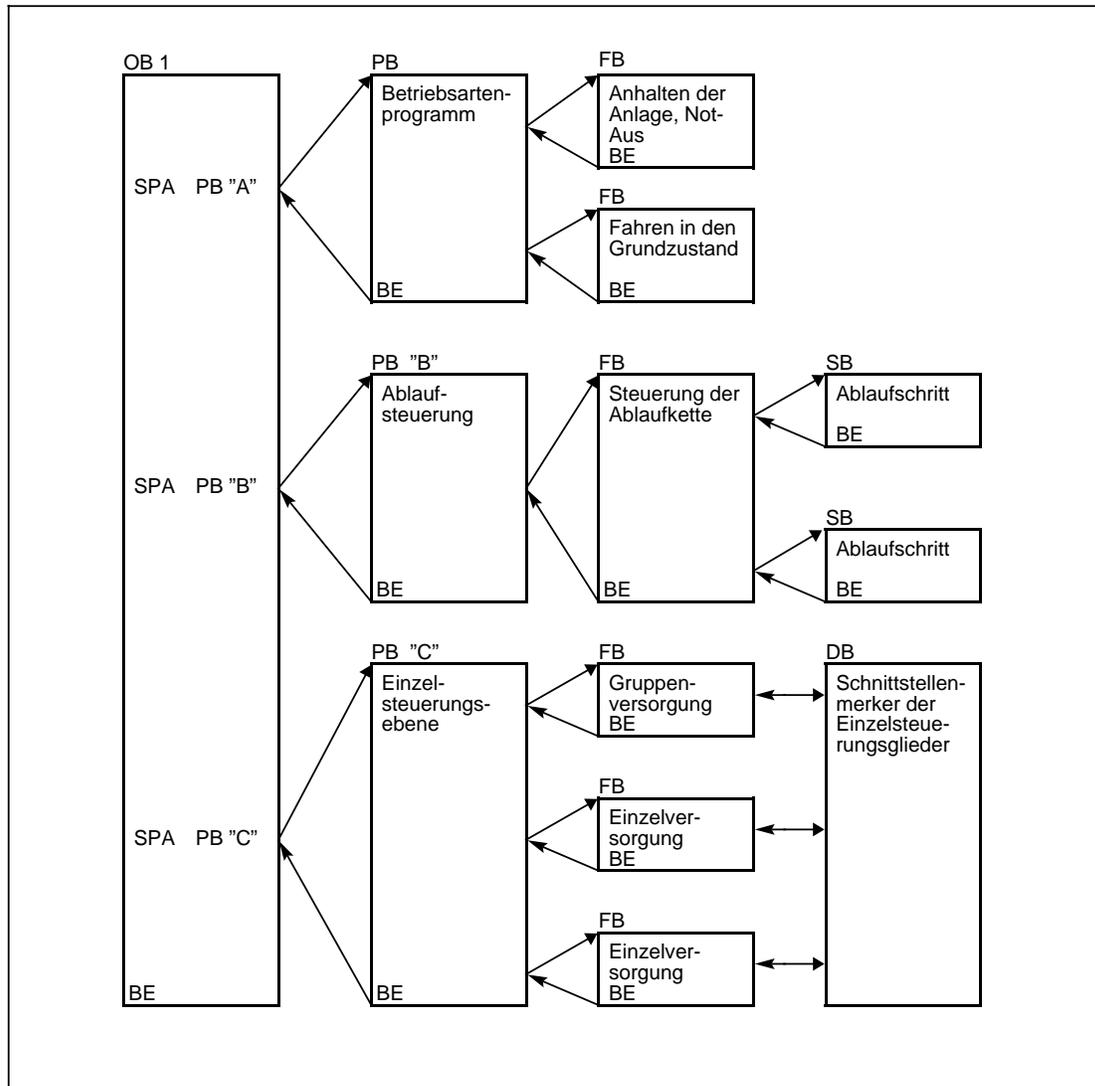


Bild 5.5 Gliederung des Anwenderprogramms nach Programmstruktur

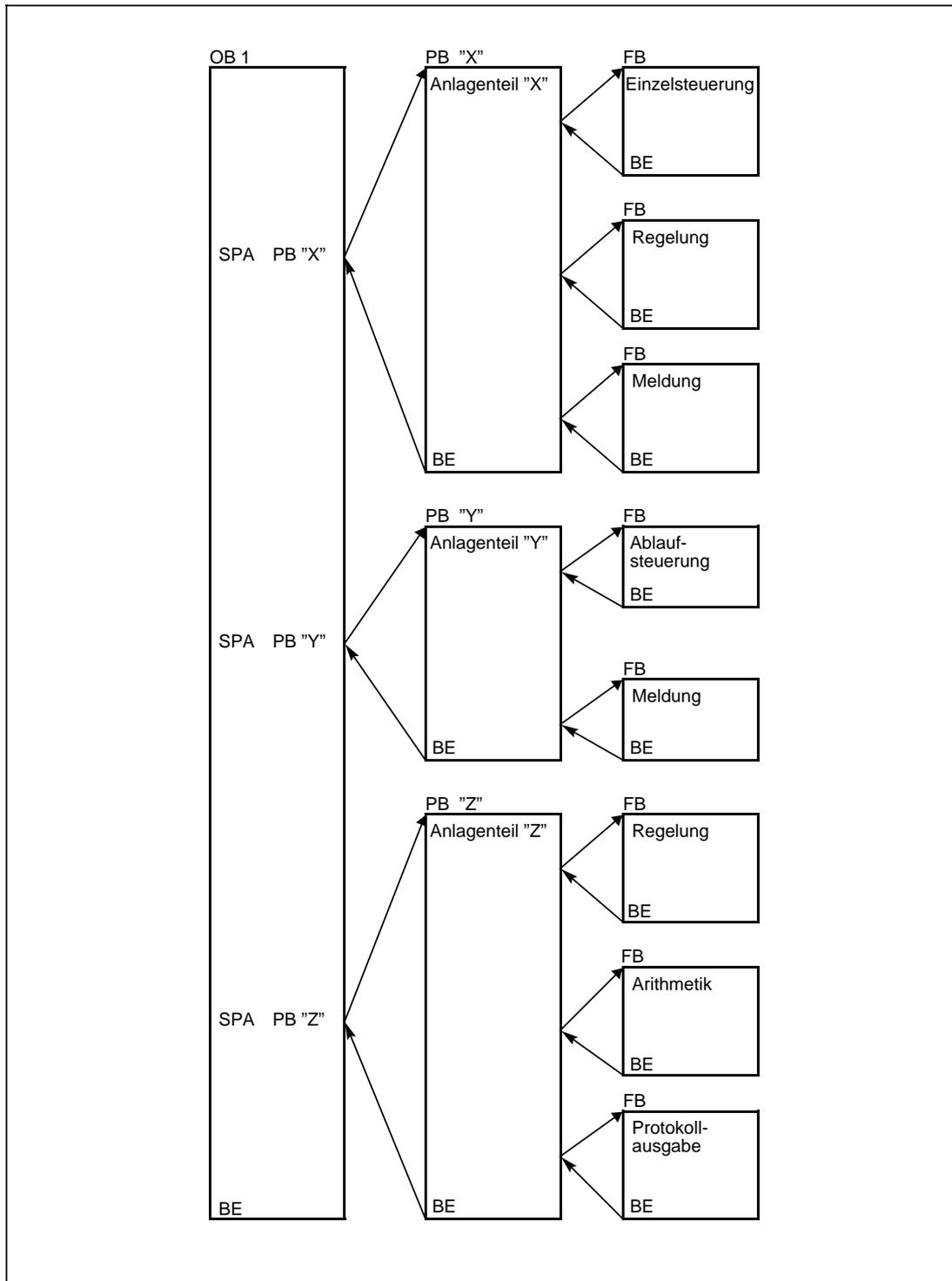


Bild 5.6 Gliederung des Anwenderprogramms nach Anlagenstruktur

## 5.3 Alarmgesteuerte Bearbeitungsebene

### 5.3.1 Programmierung der Prozeßalarm-Bearbeitung

Mit der Anpaßsteuerung PLC kann eine "alarmgesteuerte" Bearbeitung durchgeführt werden.

Bei dieser Betriebsart wird die zyklische Programmbearbeitung unterbrochen und ein spezifisches Programm bearbeitet (OB 2). Nach der Bearbeitung dieses Programms kehrt der Prozessor zur Unterbrechungsstelle zurück und setzt dort die zyklische Bearbeitung fort.

Die Prozeßalarm-Bearbeitung wird aufgrund eines Signalzustandswechsels ausgewählter Eingangsbits (flankengesteuert) angestoßen.

Die Prozeßalarm-Bearbeitung aufgrund eines Signalzustandswechsel ermöglicht dem Anwender die unmittelbare Reaktion auf Prozeßsignale. 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.

Da das Prozeßabbild nach Beendigung des OB 2 vom PLC-Betriebssystem auch nicht an die Peripherie transferiert wird, muß der Anwender über die STEP 5-Befehle T PB bzw. T PW die Prozeßperipherie beeinflussen.

In den Ausgangsbytes AB 90 (Programmbeeinflussung Kanal 1) und AB 99 (Programmbeeinflussung Kanal 2) können der NC über die Alarmbearbeitungsebene schnell Signale übermittelt werden, die von der NC im 20 ms-Raster (abhängig von MD 155) in die interne NC-PLC-Schnittstelle transferiert wird.

#### **Bemerkung:**

Bisher keine Weiterverarbeitung auf der NC-Seite.

### 5.3.2 Schnittstelle zwischen Systemprogramm und Prozeßalarm-Bearbeitung

Der Organisationsbaustein OB 2 bildet die Schnittstelle zwischen Systemprogramm und alarmgesteuerter Bearbeitung.

Der OB 2 wird immer dann aufgerufen, wenn sich der Signalzustand eines Bits im Alarmeingangsbyte ändert.

Dieses Eingangsbyte kann vom Anwender selber ausgewählt und über Maschinendatum vorgegeben werden. Außerdem muß die Alarmbearbeitung über MD 2002 Bit 0=0 freigegeben sein.

Der Signalzustandswechsel eines Bits von "0" nach "1" (positive Flanke) oder von "1" nach "0" (negative Flanke) eines der ausgewählten Bits veranlaßt die Alarmbearbeitung; d. h. das Systemprogramm ruft den OB 2 auf, womit das vom Anwender in diesem Baustein hinterlegte Alarm-Programm abgearbeitet wird.

Das Systemprogramm fragt im 5 ms-Raster (abhängig von MD 155) ab, ob während der Bearbeitung Alarmbytes gesetzt wurden und leitet gegebenenfalls eine Alarmbearbeitung ein.

Die Art (positive oder negative Flanke) des Signalzustandswechsels wird als Übergabeparameter in den Merkerbytes MB 12 und MB 16 festgehalten.

Nach Beendigung des OB 2 werden die Merkerbytes MB 12 und MB 16 vom PLC-Betriebssystem wieder gelöscht.

### 5.3.3 Reaktionszeit

Die Reaktionszeit der Alarmbearbeitungsebene auf eine Flankenänderung im Alarmeingangsbyte ist von folgenden Einflußfaktoren abhängig:

1. Alarmeingangsbyte zentrales bzw. dezentrales Peripheriebyte
2. eingestellte Servoabtastzeit (MD 155)
3. Zeitpunkt der Flankenänderung im Alarmeingangsbyte und der Abfrage einer Änderung durch das PLC-Betriebssystem

Dabei ergeben sich folgende Werte:

zentrales Alarmeingangsbyte:

Die Reaktionszeit beträgt zwischen 2 ms und 7,5 ms.

dezentrales Alarmeingangsbyte:

Die Reaktionszeit beträgt zwischen 3,5 ms und 9 ms.

## 6 Residente Funktionsbausteine

### 6.1 Allgemeines

Die Funktionsmakros sind Funktionsbausteine, die in Assembler-Code geschrieben wurden und im Systemprogramm integriert sind.

Diese Bausteine können vom Anwender wie STEP 5-Bausteine aufgerufen und parametrierbar werden. Sie sind für zeitkritische Funktionen vorgesehen und erfordern gegenüber vergleichbaren STEP 5-Bausteinen eine wesentlich geringere Bearbeitungszeit. Deshalb sollten diese Bausteine vom Anwender bevorzugt verwendet werden.

Bei der Ausgabe dieser Bausteine am Programmiergerät wird **nur** der Bausteinkopf angezeigt.

Bei der Einbindung der Bausteine in das Anwenderprogramm ist folgendes zu beachten:

1. Bei der Erstellung eines Anwenderprogrammes im ON-LINE-Betrieb (Eingabe in AG) können diese Bausteine von anderen Bausteinen aufgerufen werden. Die entsprechende Parameterliste erscheint auf dem PG und wird vom Anwender versorgt.
2. Bei der Programmierung eines Anwenderprogrammes auf Diskette muß vor Aufruf eines Funktionsmakros dieser auch auf der Diskette vorhanden sein. Dazu überträgt man die einzelnen Makros von der PLC auf eine Arbeitsdiskette. Es wird dabei nur der Bausteinkopf mit Parameterliste übertragen.
3. Beim Aufruf dieses FBs im Anwenderprogramm wird dann nur die Parameterliste versorgt.

Laden der PLC-Bausteine:

STEP 5-Funktionsbausteine mit der gleichen FB-Nummer wie ein Funktionsmakro können nicht in die Steuerung übertragen werden (PG-Meldung: "Baustein bereits im EPROM").

Die integrierten Funktionsbausteine für die Werkzeugverwaltung und für WF 725/726 in SINUMERIK 810/820 GA3 sind im Systembereich eingebunden. Die FBs sind keine Funktionsmakros. Die Bausteine können durch Aktivierung der zugehörigen Option verwendet werden. Dabei belegen bei Bearbeitung die zugehörigen variablen Datenbausteine den PLC-Anwenderspeicher. Ausführliche Funktionsbeschreibungen bieten Ihnen die zugehörigen Projektierungsanleitungen.

## 6.2 Erläuterungen der FB-Bezeichnungen

FB-Aufruf	Art des Parameters	Typ des Parameters	zugelassene Aktualoperanden
	E Eingang A Ausgang	BI Operand mit Bitadresse  BY Operand mit Byteadresse  W Operand mit Wortadresse	E n.m Eingang A n.m Ausgang M n.m Merker  EB n Eingangsbyte AB n Ausgangsbyte MB n Merkerbyte DL n Datenbyte links DR n Datenbyte rechts PB n Peripheriebyte  EW n Eingangswort AW n Ausgangswort MW n Merkerwort DW n Datenwort PW n Peripheriewort
	B Befehl	entfällt	DB n Datenbaustein FB n Funktionsbaustein PB n Programmbaustein SB n Schnittbaustein
	T Zeit	entfällt	T n Nr. des Zeitgliedes
	Z Zähler	entfällt	Z n Nr. des Zählers
	D Datum	KM Binärmuster, 16 Stellen KY zwei byteweise Betragszahlen von 0 bis 255 KH Sedezimalzahl (Hexadezimalzahl), max. 4 Stellen KC zwei alphanumerische Zeichen KT Zeitwert (1.1 bis 999.3) KZ Zählwert (0 bis 999) KF Festpunktzahl (-32768 bis +32767)	
	E, BI E, BI -Q E, BI - /		Eingangssignal, statisch wirkend Eingangssignal, das vom FB quittiert wird Eingangssignal, dessen steigende Flanke ausgewertet wird
	§E, ... \$... *... - % 1 - %v1		Keine DW als Parameter zulässig Eingangssignal, das vor Aufruf des FB versorgt werden muß festgelegtes Eingangssignal, das nicht versorgt werden muß
	A, BI Q- A, BI I- A, BI \$... *... % 1 %v1		Ausgangssignal, statisch wirkend Ausgangssignal, das vom Anwender quittiert werden muß Ausgangssignal für einen Zyklus (Impuls) Ausgangssignal auf festgelegtem Merker oder Datenwort, das direkt hinter dem FB ausgewertet werden kann festgelegtes Ausgangssignal, z. B. NC-Signal Fehlernummer AKKU 2 bei Systemstop (STS); AKKU 1 FB-Nummer zusätzl. Angabe der Nr. des Nahst.bytes im HIGH-Byte AKKU 2

### 6.3 Übersicht Funktionsmakros

FB-Nr.	FB-BEZ.	FB-NAME	
11	EINR-DB	Einrichten von Datenbausteinen	
56	VORSCHUB	Bestimmung des Vorschubwertes <sup>1)</sup>	
57	SCHLEPPA	Bestimmung des achsialen Schleppabstandes <sup>1)</sup>	
58	SPINDIST	Bestimmung der Spindeldrehzahl <sup>1)</sup>	
60	BLOCK-TR	Blocktransfer	
61	NCD-LESE	NC-Daten lesen	
62	NCD-SCHR	NC-Daten schreiben	
,			
,			
65	M STACK	Übergabemerker    Merkerstack	
66	STACK M	Merkerstack    Übergabemerker	
,			
,			
80			
81			
82			
83			
84			
85			
86			
87			
88			
,			
,			
,			
190	K-LEITPC	Datenaustausch mit Leit-PLC	

<sup>1)</sup> nur SINUMERIK 810G/820G GA3, ab Softwarestand 1.7

## **FB 11 EINR-DB** **Einrichten von Datenbausteinen**

### **1. Beschreibung**

Mit dem Funktionsbaustein "Einrichten Datenbaustein" können Datenbausteine für variable Daten im RAM-Bereich der PLC eingerichtet werden.

Der Funktionsbaustein EINR-DB generiert z. B. im Neustartzweig (OB 20) den vorgegebenen Datenbaustein im RAM-Speicher, sofern dieser noch nicht in der Adreßliste eingetragen und eine sinnvolle Datenwort-Nr. (DWNr 255) vorgegeben ist.

Die Übergabemerker (MW 246, MW 248) werden ausgewertet, wenn Parameteranteil AN=0 ist.

Die PLC verzweigt in die Stoppschleife, wenn einer der folgenden Parametrierfehler vorliegt:

- DW-Nr. >255 bzw.<0
- Datenbausteinnummer >255
- Datenbausteinnummer =0
- Nicht mehr ausreichender RAM-Bereich im AG
- Datenbaustein bereits mit anderer Länge im AG vorhanden

Eine detaillierte Fehlerkennung wird im AKKU 2 hinterlegt (AKKU 1: Bausteinnummer). Der Inhalt des eingerichteten Datenbausteines wird mit Nullen vorbesetzt.

### **Anmerkung:**

Die Datenbausteine für das Grundprogramm (Nahtstellendatenbausteine) werden im Neustart automatisch eingerichtet.

### **2. Ergänzende Angaben**

Bib.-Nr.:

Zu ladende FBs: keine

Zu ladende DBs: keine

Art des FB-Aufrufs: absolut oder bedingt (SPA FB 11 oder SPB FB 11)

Einzugebende DBs: keine

Fehlermeldungen: %2 DB-Nr. ist größer als 255

%3 angegeb. DWNr kleiner 0

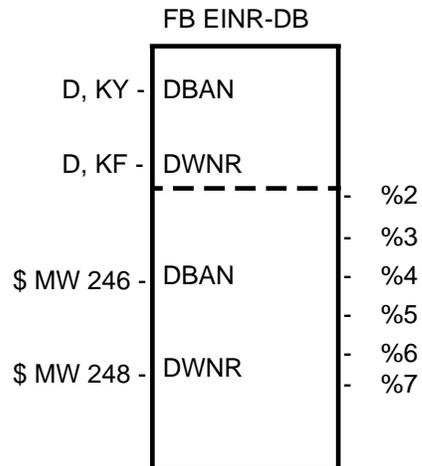
%4 Länge des einzurichtenden DBs ist ungleich der Länge des bereits im AG vorhandenen DBs

%5 Speicherplatz im AG nicht mehr ausreichend

%6 angegebene DWNr größer als 255

%7 DB 0 nicht einrichtbar

### 3. Bausteinaufruf



### 4. Signalbeschreibung

**DBAN** Nummer und Anzahl der Datenbausteine, die eingerichtet werden sollen.

**Sonderfall:**

Ist der Parameteranteil AN mit 0 vorgegeben, so kann der Funktionsbaustein über das MW 246, 248 parametrieren werden.

High-Byte: DB-Nr.

Low-Byte: Anzahl der Datenbausteine (mindestens 1)

**DWNR** Nr. des letzten DW

### 5. Beispiel



## FB 56 VORSCHUB Bestimmung des Vorschubwertes

(nur SINUMERIK 810G/820G GA3, ab Softwarestand 1.7)

### 1. Beschreibung

Mit dem FB 56 kann der Wert des Vorschubs gelesen werden. Dem Baustein wird über MB 224 der Vorschubtyp (G94/G95/G98) und die Vorschubart (Bahnvorschub oder achsialer Vorschub) usw. mitgeteilt.

Der Wert des Vorschubs steht dann als BCD-Wert im MW 225 und MW 227.

### 2. Ergänzende Angaben

Bib.-Nr.:

Zu ladende FBs: keine

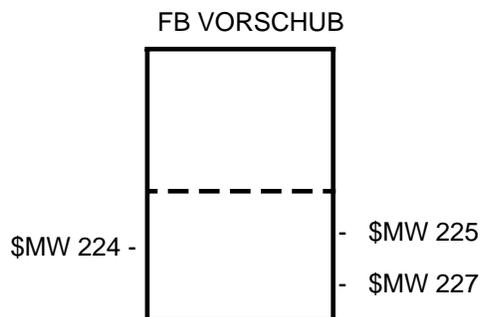
Zu ladende DBs: keine

Art des FB-Aufrufs: absolut oder bedingt (SPA FB 56 oder SPB FB 56)

Einzugebende DBs: keine

Fehlermeldungen: siehe Signalbeschreibung (AKKU1 und AKKU2)

### 3. Bausteinaufruf



#### 4. Signalbeschreibung

Aufbau des MB 224:

MB 224							
Bit:7	6	5	4	3	2	1	0

0	0	0	0					Bahnvorschub
0	0	0	1					achsialer Vorschub für die 1. Achse
0	0	1	0					achsialer Vorschub für die 2. Achse
0	0	1	1					achsialer Vorschub für die 3. Achse
0	1	0	0					achsialer Vorschub für die 4. Achse
0	1	0	1					achsialer Vorschub für die 5. Achse
0	1	1	X					Alarm 6114 "Funktionsmakro", AKKU1=56, AKKU2=2
1	X	X	X					Alarm 6114 "Funktionsmakro", AKKU1=56, AKKU2=2
				0	0			G94
				0	1			G95
				1	0			G98
				X	1			Alarm 6114 "Funktionsmakro", AKKU1=56, AKKU2=1
				0	1			Vorschub (Bahn- oder achsialer Vorschub) vom 1. Kanal
				1	0			Vorschub (Bahn- oder achsialer Vorschub) vom 2. Kanal
				1	1			Alarm 6114 "Funktionsmakro", AKKU1=56, AKKU2=1
				0	0			Bei einem achsialen Vorschub wird der Kanal automatisch bestimmt, bei Bahnvorschub wird der Alarm 6114 (AKKU1=56, AKKU2=1) generiert.

#### Hinweise zu Bit 0 und Bit 1

Zur Bestimmung des Bahnvorschubs muß dem FB 56 der Kanal mitgeteilt werden. Beim achsialen Vorschub kann der FB 56 den Kanal in dem die Achse programmiert wurde automatisch bestimmen.

Aufbau des MW 225 und MW 227:

	DL		DR		
MW 225	6	7	5	E	
MW 227	4	2	0	1	E=Dezimalpunkt (variabel)

Beispiel: Anzeige des Vorschubwertes von 1425.67

#### 5. Beispiel

```

:
: L KH 0010      Lesen des achsialen Vorschubwertes G94 der 1. Achse
: T MB 224
: SPA FB 56
Name : VORSCHUB
: L MW 225      Ergebnis in
: T MW 175      Merkerwörtern
: L MW 227      retten
: T MW 177      (zur weiteren Verarbeitung)

```

## FB 57 SCHLEPPA

### Bestimmung des achsialen Schleppabstandes

(nur SINUMERIK 810G/820G GA3, ab Softwarstand 1.7)

#### 1. Beschreibung

Der achsiale Schleppabstand kann mit dem FB 57 gelesen werden. Dem Baustein muß über MB 224 die Achsnummer (1 bis 5) mitgeteilt werden. Der Wert des achsialen Schleppabstandes (in Lageregelfeinheiten) wird für die entsprechende Achse im MW 225 und MW 227 ausgegeben.

#### 2. Ergänzende Angaben

Bib.-Nr.:

Zu ladende FBs: keine

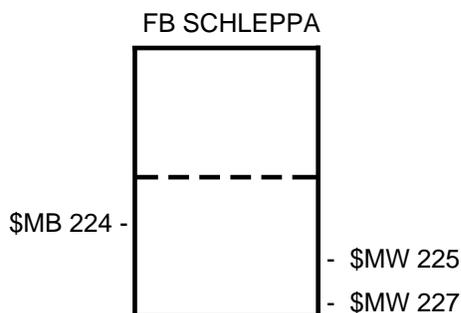
Zu ladende DBs: keine

Art des FB-Aufrufs: absolut oder bedingt (SPA FB 57 oder SPB FB 57)

Einzugebende DBs: keine

Fehlermeldungen: siehe Signalbeschreibung (AKKU1 und AKKU2)

#### 3. Bausteinaufruf



#### 4. Signalbeschreibung

Aufbau des MB 224:

MB 224							
Bit:7	6	5	4	3	2	1	0

0	0	0	0	0	0	0	0	Alarm 6114 "Funktionsmakro", AKKU1=57, AKKU2=1
0	0	0	0	0	0	0	1	achsialer Schleppabstand für die 1. Achse
0	0	0	0	0	0	1	0	achsialer Schleppabstand für die 2. Achse
0	0	0	0	0	0	1	1	achsialer Schleppabstand für die 3. Achse
0	0	0	0	0	1	0	0	achsialer Schleppabstand für die 4. Achse
0	0	0	0	0	1	0	1	achsialer Schleppabstand für die 5. Achse
0	0	0	0	0	1	1	1	Alarm 6114 "Funktionsmakro", AKKU1=57, AKKU2=1

Aufbau des MW 225 und MW 227:

	DL	DR
MW 225	15	0
MW 227	31	16

## 5. Beispiel

```

:
: L KH 0001      Lesen des Schleppabstandes der 1. Achse
: T MB 224
: SPA FB 57
Name : SCHLEPPA
: L MW 225      Ergebnis in
: T MW 180      Merkerwörtern
: L MW 227      retten
: T MW 182      (zur weiteren Verarbeitung)
:

```

## FB 58 SPINDIST

### Bestimmung der Spindeldrehzahl

(nur SINUMERIK 810G/820G GA3, ab Softwarestand 1.7)

#### 1. Beschreibung

Mit dem FB 58 kann die Spindeldrehzahl gelesen werden. Dem Baustein muß über MB 224 die Spindelnummer (1 oder 2) mitgeteilt werden. Die Spindeldrehzahl der entsprechenden Spindel wird im MW 225 und MW 227 ausgegeben.

#### 2. Ergänzende Angaben

Bib.-Nr.:

Zu ladende FBs: keine

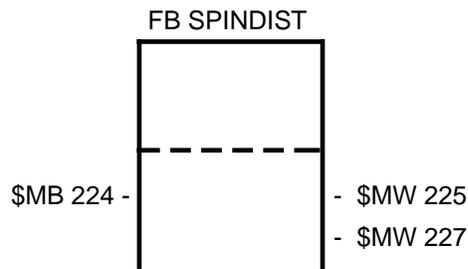
Zu ladende DBs: keine

Art des FB-Aufrufs: absolut oder bedingt (SPA FB 58 oder SPB FB 58)

Einzugebende DBs: keine

Fehlermeldungen: siehe Signalbeschreibung (AKKU1 und AKKU2)

#### 3. Bausteinaufruf



#### 4. Signalbeschreibung

Aufbau des MB 224:

MB 224							
Bit:7	6	5	4	3	2	1	0

0	0	0	0	0	0	0	0	Alarm 6114 "Funktionsmakro", AKKU1=58, AKKU2=1
0	0	0	0	0	0	0	1	Spindeldrehzahl der Spindel 1
0	0	0	0	0	0	1	0	Spindeldrehzahl der Spindel 2
0	0	0	0	0	0	1	1	Alarm 6114 "Funktionsmakro", AKKU1=58, AKKU2=1

Aufbau des MW 225 und MW 227:

	DL	DR
MW 225	15	0
MW 227	31	16

**5. Beispiel**

```
:  
: L KH 0001      Lesen der Spindeldrehzahl von Spindel 1  
: T MB 224  
: SPA FB 58  
Name : SPINDIST  
: L MW 225      Ergebnis in  
: T MW 190      Merkerwörtern  
: L MW 227      retten  
: T MW 192      (zur weiteren Verarbeitung)  
:
```

## FB 60 BLOCK-TR Blocktransfer

### 1. Beschreibung

Der Funktionsbaustein BLOCK-TR kopiert aus einem Quelldatenbaustein im RAM oder EPROM die vorgegebene Anzahl der Datenworte in einen im RAM liegenden Zieldatenbaustein.

Der Beginn des zu kopierenden Blockes im Quell-DB und das Anfangsdatenwort im Ziel-DB sind frei wählbar.

Abgeprüft wird ob:

- der Quell-DB in der benötigten Länge im AG liegt,
- der Ziel-DB in der benötigten Länge im AG liegt,
- der Ziel-DB im RAM liegt und
- die Anzahl der zu übertragenden Datenworte >0 und <129 ist.

Bei Fehler verzweigt die PLC in die Stoppschleife. Eine detaillierte Fehlerkennung wird im AKKU 2 hinterlegt (AKKU 1>Bausteinnummer). Wird dem Parameter "DB QZ" eine Null zugewiesen, werden die Merkerwörter 250 ... 254 abgefragt (Parametrierung: 0,1 bzw. 1,0 bzw. 0,0).

### 2. Ergänzende Angaben

Bib.-Nr.:

Zu ladende FBs: keine

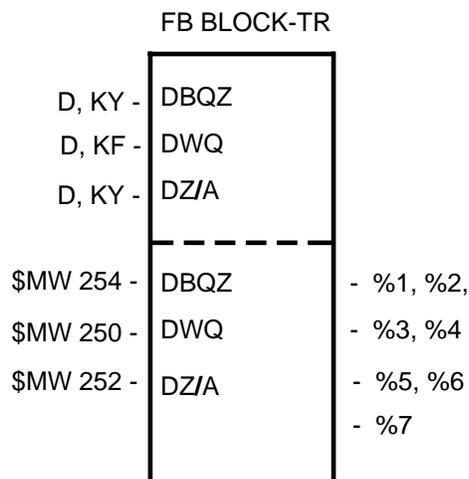
Zu ladende DBs: keine

Art des FB-Aufrufs: absolut oder bedingt (SPA FB 60 oder SPB FB 60)

Einzugebende DBs: keine

Fehlermeldungen: %1 Anzahl der zu übertragenden DW ist größer 128  
 %2 Anzahl der zu übertragenden DW=0  
 %3 Quell-DB fehlt  
 %4 Ziel-DB fehlt  
 %5 Ziel-DB ist zu kurz  
 %6 Ziel-DB im EPROM  
 %7 Quell-DB ist zu kurz

### 3. Bausteinaufruf



#### 4. Signalbeschreibung

**DBQZ** Nummer der Quell- und Ziel-DBs

High-Byte: Quell-DB

Low-Byte : Ziel-DB

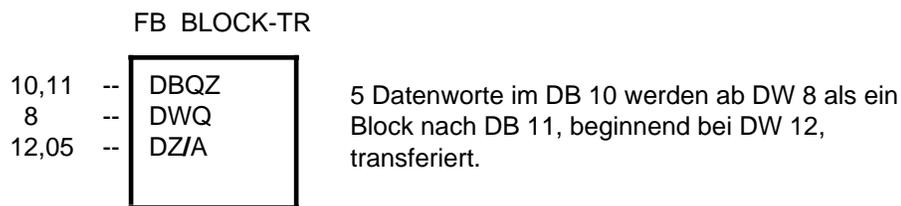
**DWQ** Anfang des Datenblockes im DBQ, ab dem kopiert werden soll.

**DZ/A** Anfang des Datenblockes im DBZ in den kopiert werden soll und Anzahl der zu kopierenden Datenwörter.

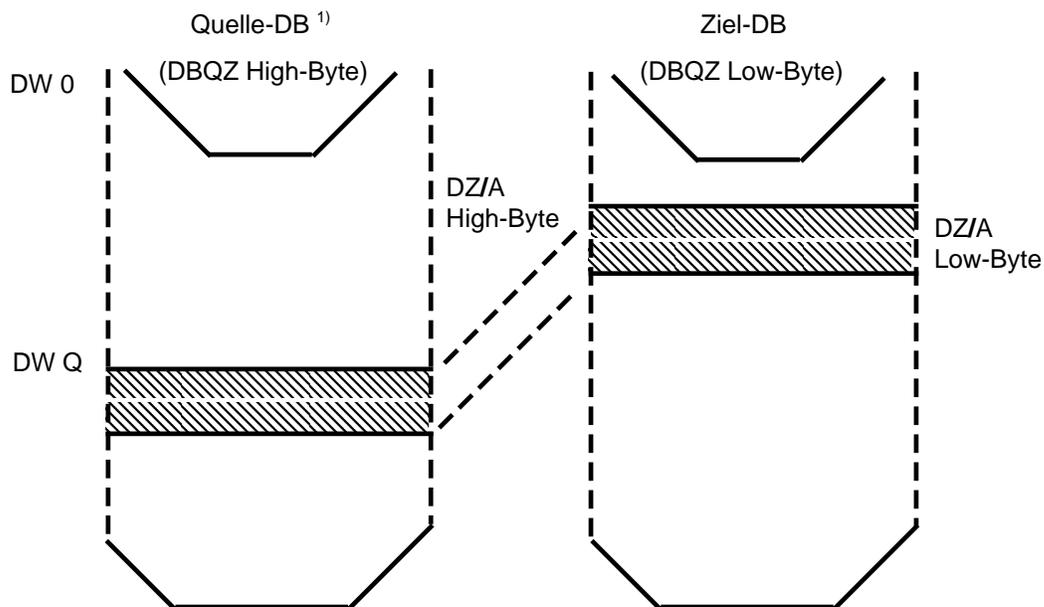
High-Byte: Anfang des Datenblocks

Low-Byte : Anzahl der zu kopierenden Datenworte

#### 5. Beispiel



#### 6. Übersichtsbild Datentransfer



1) Der Datenbaustein "Quelle" kann im RAM oder EPROM liegen.

## **FB 61 NCD-LESE NC-Daten lesen**

## **FB 62 NCD-SCHR NC-Daten schreiben**

### **1. Beschreibung**

Mit den Funktionsbausteinen FB 61 und FB 62 können NC-Daten von der PLC gelesen bzw. geschrieben werden. Voraussetzung dafür ist, daß der auftragsgesteuerte Datenaustausch NC PLC per Optionsbit freigegeben ist (ansonsten: Fehler 8).

Die Datenquelle in der NC bzw. PLC und das Datenziel in der PLC bzw. NC müssen den Funktionsbausteinen über Parameter mitgeteilt werden. Den Funktionsbausteinen muß über den Parameter NSBY auf der Nahtstelle im DB 36 ein Byte zugeordnet werden, dem der aktuelle Bearbeitungsstand der Datenübergabe entnommen werden kann. Die Funktion der Datenübergabe NC/PLC ist detailliert im Kapitel 7 beschrieben.

Ist der Datentyp der NC unbekannt, dann wechselt die PLC in den Stop-Zustand und dem Anwender wird eine PLC-Fehlermeldung angezeigt (Nr. 6114) . Werden beim Schreiben in die NC falsche Werte (z.B. zu groß) vorgegeben, so wird dem Anwender der NC-Alarm 3016 gemeldet.

### **2. Ergänzende Angaben**

Bib-Nr. FB 61:

FB 62:

Zu ladende FBs: keine

Zu ladende DBs: keine

Art des FB-Aufrufs: absolut oder bedingt

Einzugebende DBs: keine

Fehlermeldungen: AKKU 1 (FB-Nr.)=61 oder 62

AKKU 2 high-Byte=Nr. Nahtstellenbyte, d. h. die zum Fehler führende Auftrags-Nr. (außer bei Fehler 8).

AKKU 2 low-Byte=detaillierte Fehlernummer:

0: Anz. >1 unzulässig

1: Nahtstellenbyte unzulässig

2: Adressiertes Datenwort fehlt/DB fehlt oder DB-Nr. bzw. MW-Nr. unzulässig

3: Datentyp unzulässig

4: \*ANZ 0 bzw. >80 (bei NC-Alarm >127)

5: Lesen unzulässig/Schreiben unzulässig

6: Zahlenformat unzulässig

7: Wert 3 bei ZOA bzw. ZOFA ungleich 0 bzw. 1

8: NC-Daten lesen/schreiben per Option nicht freigegeben

---

\* s. Sonderfälle b. Parameter ANZ

### 3. Bausteinaufruf

#### FB 61: NCD-LESE

E, BI	LESE	
E, BY	NSBY	
D, KF	ANZ	
D, KC	DTY1	
D, KC	DTY2	
D, KC	DTY3	
D, KF	WER1	
D, KF	WER2	
D, KF	WER3	
D, KC	ZFPN	
D, KY	ZIEL	
-----		
\$MB 242	NSBY	% 0
\$MB 243	ANZ	% 1
\$MW 244	WER1	% 2
\$MW 248	WER2	% 3
\$MW 250	WER3	% 4
\$MW 252	ZFPN	% 5
\$MW 254	ZIEL	% 6
		% 7

#### FB 62: NCD-SCHR

E, BI	SCHR	
E, BY	NSBY	
D, KF	ANZ	
D, KC	DTY1	
D, KC	DTY2	
D, KC	DTY3	
D, KF	WER1	
D, KF	WER2	
D, KF	WER3	
D, KC	ZFPN	
D, KY	QUEL	
-----		
\$MB 242	NSBY	% 0
\$MB 243	ANZ	% 1
\$MW 244	WER1	% 2
\$MW 248	WER2	% 3
\$MW 250	WER3	% 4
\$MW 252	ZFPN	% 5
\$MW 254	QUEL	% 6
		% 7
		% 8

### 4. Signalbeschreibung

**LESE** Kommando für Datenübertragung

**SCHR** Mit dem Signal LESE bzw. SCHR=1 werden die Parameter in das FIFO (Auftragspuffer) übernommen.

Das Signal "LESE" bzw. "SCHR" muß bei absolutem Bausteinaufruf am Ende der Übertragung vom Anwender zurückgesetzt werden.

Bei bedingtem Aufruf ist es mit "1" (M 0.1) zu beschalten (siehe auch Impulsdiagramme). Eine Aktivierung der Datenübergabe ist nur möglich, wenn das Signal DATENÜBERGABE BELEGT des parametrisierten Nahtstellenbytes (NSBY) 0-Signal führt.

**NSBY** Dem Funktionsbaustein muß auf der Nahtstelle (DB 36) ein Byte zugeordnet werden, dem der aktuelle Bearbeitungszustand der Datenübergabe entnommen werden kann.

**Sonderfall:** Variable Versorgung siehe nachfolgende Übersicht.

**Hinweis:**

Wird als Nahtstellenbyte das Byte DL 32 (=Nummer 65) angegeben, wird der damit deklarierte Auftrag "alarmgesteuert" bearbeitet, d. h. der Auftrag wird zwischen den Aufträgen eingeschoben, die bereits im Auftragspuffer stehen. Bei Auftragsnummer 65 mit ANZ>1 wird der Alarmauftrag **vollständig** bearbeitet.

**ANZ** Anzahl der zu übertragenden Daten. Wird mehr als ein Wort übertragen (ANZ >1) werden Quell- und Zieladresse inkrementiert. Wird dabei z. B. ein nicht mehr vorhandenes Datenwort adressiert, geht die PLC mit %2 auf STOP.

Dabei ist zu beachten, daß abhängig vom Parameter ZFPN pro übertragenem Wert eine unterschiedliche Anzahl von Datenworten benötigt wird.

**Sonderfälle:**

Bei ANZ=0 und ANZ=128 werden die Parameter NSBY, ANZ, WER1 ... WER3, ZFPN und ZIEL/QUEL über Merkerworte versorgt. Der Unterschied zwischen ANZ=0 und ANZ=128 kann der Übersicht variable Versorgung entnommen werden.

**DTY1** 2 ... 6 ASCII-Zeichen; Mnemonik entspr. CL 800.

**DTY3** Die Werte sind ab DTY 1 einzugeben.

**Achtung:**

Fehlende Zeichen sind mit einem "Blank" aufzufüllen (siehe Tabelle Abschnitt 7).

**ZFPN** Zahlenformat PLC/NC  
Übersicht Zahlenformate NC/PLC Abschnitt 6.

**ZIEL** Datenziel in der PLC (FB 61)

**QUEL** Datenquelle in der PLC (FB 62)

- High-Byte: DB-Nummer (2 ... 255), Ausnahme: 0=Merker
- Low-Byte: – Wort-Nummer (bei Datenbausteinen)
- Byte-Nummer (bei Merkern), (zulässig 136 ... 255)

**Achtung:**

1. Es ist die Anzahl der benötigten Bytes entsprechend dem Parameter ZFPN zu beachten.
2. Soll z. B. ein 16 Bit-Wort an die NC übergeben werden, so muß bei den Zahlenformaten FO-FF der zu übertragende Wert bei der Datenübergabe aus Datenworten in das parametrisierte **Wort k+1** geschrieben werden.

Das parametrisierte Wort k ist mit 0 zu laden (siehe Abschnitt 6).

**5. Übersicht variable Versorgung**

Bei der variablen Versorgung müssen die Werte im selben Format in die Merkerworte geschrieben werden, wie es für die direkte Versorgung am FB erforderlich ist.

**Beispiel:**

Soll der Parameter ZFPN mit F1 parametrisiert werden, so ist zu programmieren:

```
.  
. .  
. .  
L KC F1  
T MW 252  
. .  
. .  
. .
```

**Ausnahme:**

Der Parameter NSBY kann auf zwei Arten versorgt werden.

- Bei ANZ=0 ist die Nummer des Nahtstellenbytes im **MB 242** einzutragen (siehe Druckschrift Nahtstellenbeschreibung, Teil 1, Übersicht, Kapitel 1.5).

Nr. Nahtstellenbyte		Byte-Nummer
1	≙^	DLO
2	≙^	DRO
.		.
.		.
.		.
64	≙^	DR 31
65 (alarmgesteuert)	≙^	DL 32

**Beispiel:**

Soll als NSBY das DL 15 verwendet werden, muß programmiert werden:

:

L KB 31  
T MB 242

- Bei ANZ=128 muß der Operationscode mit Parameter von L DL xx bzw. L DR xy im **MW 241** eingetragen werden. Diese Art der Parametrierung eignet sich besonders, wenn z. B. der FB 61 als Unterprogramm eines Funktionsbausteines aufgerufen wird; wenn am aufrufenden FB der Parameter NSBY im Datenformat "E/BY" deklariert wurde.

Das PG setzt bei der Parametrierung des aufrufenden FBs in der Parameterliste schon den richtigen Code ab, so daß dieser mit der Befehlsfolge

:

LW =ABCD ist ein Parameter (Typ: E/BY) des übergeordneten FBs.  
T MW 241

:

an den FB 61/FB 62 übergeben werden kann.

Beispiel für Operationscode:

Nr. Nahtstellenbyte	Byte-Nr.	Operationscode (Hexa)
.		
.		
.		
31	DL 15	220F
32	DR 15	2A0F

**Beispiel:**

```

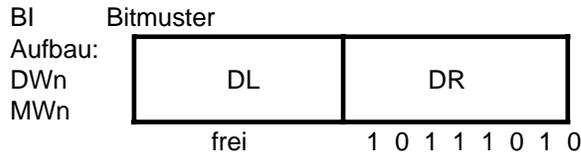
FB 140          SPRM-A          BIB=14050

NETZWERK 1
NAME           : STAZ
BEZ           : DBZW  E/A/D/B/T/Z: B
BEZ           : BEAR  E/A/D/B/T/Z: E          BI/BY/W/D: BI
BEZ           : NSBY  E/A/D/B/T/Z: E          BI/BY/W/D: BY
BEZ           : P-NR  E/A/D/B/T/Z: D          KM/KH/KY/KC/KT/KZ/KG/: KF
:
: LW          =DBZW !          PARAMETERVERSORGUNG
: T          MB 254 ZIEL, DB-Nr. FB 61 KOMMANDO TOS
:                               (WZ-Korrektur lesen)
: L          KB 40 !
: T          MB 255 ! ZIEL, DW-Nr.
: L          KC F5 !
: T          MW 252 ! ZFPN
: A          DB 36 !          MW 190 soll ein freies Anwender-Merker-
: L          =NSBY !          wort darstellen.
: T          MB 190 !          Der Lese-Befehl für den nachfolgenden
: U          = BEAR !          FB 61 wird gebildet aus:
: UN         M 190.3 !          Leseanforderung=BEAR
: =          M 191.0          Datenschnittstelle nicht belegt (M 190.3)
: RB         = BEAR !

: LW         =NSBY !          Der Operationscode des Parameters
:                               "NSBY" vom übergeordneten FB 140
:                               wird im MW 241 abgelegt.
: T          MW 241 ! NSBY
: L          KB 1 !
: T          MB 243 ! ANZ
: LW         =P-Nr. !
: T          MW250 ! WER3
: L          DR 54 !
: T          MW 244 ! WER1
: L          DR 51 !
: T          MW 248 ! WER2
: SPA        FB 61
NAME          : NCD  LESE
LESE         :      M 191.0
NSBY         :      MB 242          (jede andere Adresse zulässig)
ANZ          :      KF+128
DTY1         :      KCTO
DTY2         :      KCS
DTY3         :      KC
WER1         :      KF+0
WER2         :      KF+0
WER3         :      KF+0
ZFPN        :      KC 00
ZIEL         :      KY 0,0
:
    
```

## 6. Übersicht Zahlenformate NC/PLC

Mit dem Parameter "Zahlenformat" wird das Format der Zahlen bei der Datenübergabe PLC gewandelt. Die für jeden Datentyp zulässigen Zahlenformate können der Tabelle für die Datentypen entnommen werden. Prinzipiell sind folgende Zahlenformate in der PLC möglich.



### Beispiel:

R100 = 10111010

Sind einzelne Ziffern im R-Parameter #1 bzw. 0 wird der Fehler "Zahlenformat" ausgegeben.

BO BCD-Zahl mit Vorzeichen und Komma

:

B9

BG

### Aufbau:

	DL		DR		
DWn/ MWn	frei		VZ: 1	8	VZ=0 pos.
DWn+1/ MWn+2	7	6	5	4	
DWn+2/ MWn+4	E	3	2	1	

### Beispiel:

R100=- 87654.321

Parameter ZFPN=B4

### Hinweis:

Der erste Buchstabe des Parameters (hier B) gibt an, daß es sich bei der Datenquelle bzw. dem Datenziel in der PLC um BCD-Werte handelt. Der zweite Wert (hier 4) sagt aus, daß die BCD-Zahl in der PLC mit 3 Stellen abgelegt wird (FB 61).

Beim Transfer PLC NC (FB 62) wird damit festgelegt, wo – unabhängig vom Dezimalpunkt in der PLC – der Dezimalpunkt in der NC liegt.

Steht in der PLC in Datenwörtern der Wert 1234.56 wird beim PLC-Transfer R100 ZFPN mit B2 parametrier, wird in R100 die Zahl 1234.5 eingetragen.

Eine Übersicht über die möglichen Parameter gibt die folgende Tabelle:

ZFPN	Bedeutung
B0/F0	Wert ohne Dezimalp. (z. B. 1234)
B1/F1	Wert mit Dezimalp. (z. B. 1234.)
B2/F2	1 Stelle hinter dem Dezimalpunkt
B3/F3	2 Stellen hinter dem Dezimalpunkt
B4/F4	3 Stellen hinter dem Dezimalpunkt
B5/F5	4 Stellen hinter dem Dezimalpunkt
B6/F6	5 Stellen hinter dem Dezimalpunkt
B7/F7	6 Stellen hinter dem Dezimalpunkt
B8/F8	7 Stellen hinter dem Dezimalpunkt
B9/F9	8 Stellen hinter dem Dezimalpunkt
FA	Linearpositionswert *
FB	Rundpositionswert *
FC	Vorschubwert linear *
FD	Vorschubwert rund *
FE	Vorschubwert Umdrehung *
FF	Drehzahlwert *
BG	Wert wie abgelegt
BI	Bitmuster

F0 Festpunktzahl mit 32 Bit  
 :  
 FF

**Aufbau:**

	DL	DR
DWn/ MWn	31	16
DWn+1/ MWn+2	15	0

**Beispiel:**

R100=1234.567 führt bei ZFPN=F4 zum PLC-Datum 1234567; d. h. im Gegensatz zu BCD wird der Dezimalpunkt ausgeblendet (FB 61).

Beim Datentransfer PLC NC (FB 62) wird angegeben, wo der Dezimalpunkt in der NC abgelegt wird.

PLC-Datum=121457 ZFPN=F2 führt zum NC-Datum 12145.7.

\* Bei diesen Zahlenformaten wird abhängig vom Eingabesystem (über Maschinendatum eingestellt) nur **die Position des** Dezimalpunktes innerhalb des dimensionslosen R-Parameter-Wertes bestimmt. Es findet **keine** Abprüfung auf Obergrenzen statt.

## 7. Tabelle für Datenübergabe NC/COM PLC-Datenwörter/Merker

Funktionsbeschreibung	Datentyp (DTY1-DTY3)	Grenzwert	Wert (WER1 - WER3)	Zahlen- format (ZFPN) FB 61 *1	Zahlen- format (ZFPN) FB 62 *1	Maxi. wert Para- meter <u>ANZ</u> *2	
<b>Maschinendaten</b>							
Maschinendaten-NC	MDN <Adresse>	0...4999	1	B0, F0 (*3)	B0, F0 (*3)	80	
Maschinendaten-NC-Bytes	MDNBY <Adresse>	5000...9999	1	BI	BI	80	
<b>Settingdaten</b>							
Settingdaten-NC	SEN <Adresse>	0...4999	1	B0, F0 (*3)	B0, F0 (*3)	80	
Settingdaten-NC-Bytes	SENBY <Adresse>	5000...9999	1	BI	BI	30	
<b>Werkzeug-Korrekturen</b>							
Werkzeug-Korrektur	T0S <TO-Bereich>	0	1	B0-B9, F0-F9, BG, BI	B0-B9, F0-F9, BG, BI		
	<D-Nr.>	1 - 99	2				
	<P-Nr.>	0 - 9 (15 bei G)	3			10 T/M 16 G	
Werkzeug-Korrektur add.	T0A				B0-B9, F0-F9		
	<D-Nr.,>	1 - 99	1		BG, BI		
	<P-Nr.>	0 - 9 (15 bei G)	2			1	
<b>Nullpunktverschiebungen</b>							
Einstellbare Nullpunkt- verschiebung (G54 - G57) grob/fein	Z0A <Gruppe> <Achs-Nr.> <g/f>	1 - 4 1 - 5 0/1	1 2 3	B0-B9, F0-F9	B0-B9, F0-F9 BG	1	
	Programmierbare Nullpunkt- verschiebung (G58, G59)	Z0PR <Gruppe> <Achs-Nr.>	1 - 2 1 - 5	1 2	B0-B9, F0-F9	B0-B9, F0-F9 BG	1
		Einstellbare Nullpunkt- verschiebung additiv nur schreiben grob/fein	Z0FA <Gruppe> <Achs-Nr.>  <g/f>	1 - 4 1 - 5  0/1	1 2 3		B0-B9, F0-F9 BG
Externe Nullpunktverschb. vom PLC	Z0E <Achs-Nr.>	1 - 5	1	B0-B9, F0-F9	B0-B9, F0-F9 BG	1	
PRESET-Verschiebung	Z0PS <Achs-Nr.>	1 - 5	1	B0-B9, BG, F0-F9	B0-B9, BG, F4	1	
Summen-Verschiebung	Z0S <Achs-Nr.>	1 - 5	1	B0-B9, BG,F0		1	

\*1 Ist ein Zahlenformat nicht angegeben, ist vom entsprechenden FB kein Datentransfer möglich.

\*2 Hat ein Datentyp mehr als einen Parameter, steht die Anzahl in der Zeile, in der der Parameter WERT steht, der inkrementiert wird.

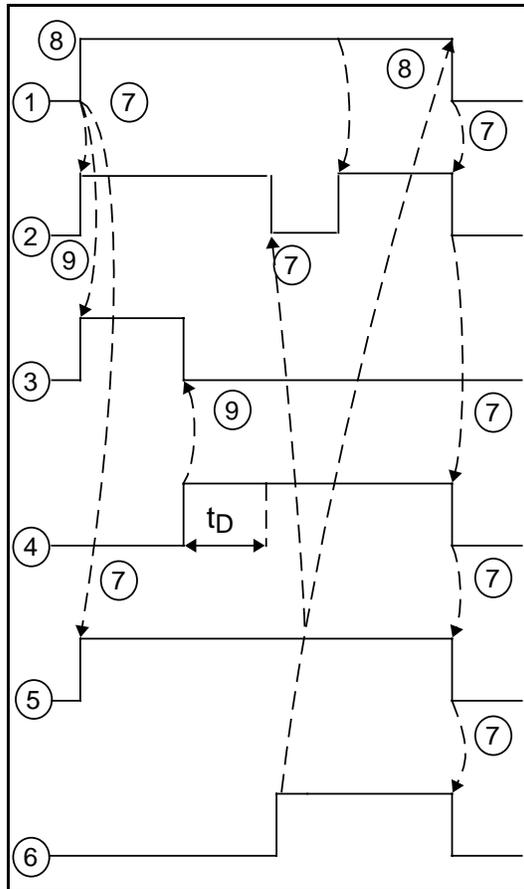
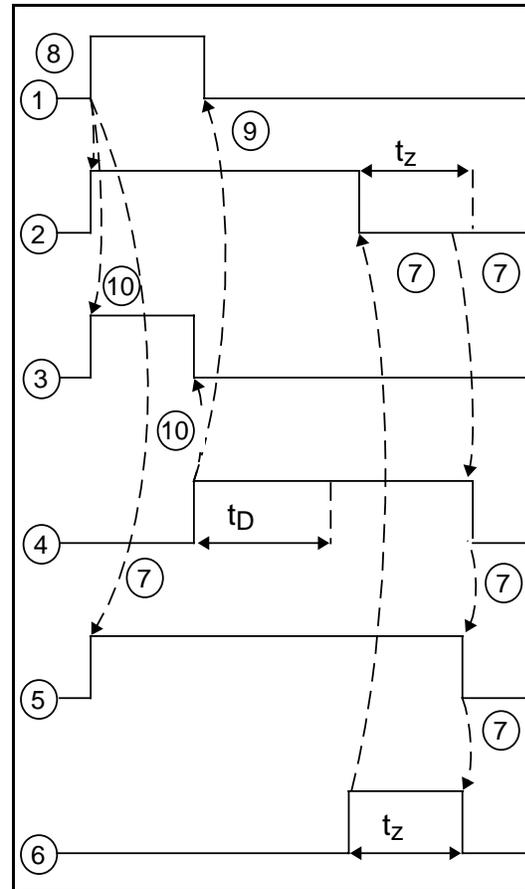
Eine Anzahl >1 ist nur unter folgenden Randbedingungen möglich:

- Datenblock in der NC **geschlossen** vorhanden
- PLC-Quell- bzw. Zieladresse in ausreichender Länge vorhanden:
  - bei B0-B9/BG drei Worte pro Wert
  - bei F0-FF zwei Worte pro Wert
  - bei BI ein Wort pro Wert

\*3 Die Ein-/Ausgabe erfolgt entsprechend dem Format des Maschinen-/Settingdatums.

Funktionsbeschreibung	Datentyp (DTY1-DTY3)	Grenzwert	Wert (WER1 - WER3)	Zahlen- format (ZFPN) FB 61 *1	Zahlen- format (ZFPN) FB 62 *1	Maxi. wert Para- meter <u>ANZ*2</u>
<b>Istwerte</b>						
Achs-Position-Ist, werkstückbezogen	ACPW<Achs-Nr.>	1 - 5	1	B0-B9/BG, F0		1
Achs-Position-Ist, maschinenbezogen	ACPM<Achs-Nr.>	1 - 5	1	B0-B9/BG, F0		1
<b>Externe Sollwerte</b>						
Externer Bahnvorschub	EXBF <Kan.-Nr.> <lin/Umdr>	1 - 2 0/1	1 2	B4, F4	B4, F4	1
<b>Programmdaten</b>						
Programmzeiger für akt. Satz	PP <Kan.-Nr.> <Ebene>	1 - 2 0 - 3	1 2	B0, F0		1 *4
<b>Programmanwahl</b>						
Anwahl eines NC-Programmes	INITMP <Kan.-Nr.>	1 - 2	1		B0, F0	1
Anwahl eines NC-Unterprogrammes	INITSP <Kan.-Nr.>	1 - 2	1		B0, F0	1
<b>R-Parameter</b>						
R-Parameter NC-Kanal	RPNC <Kan.-Nr.> <Paramet.>	1 - 2 0 - 499	1 2	B0-B9, F0-FF BG, BI	B0-B9, F0-FF BG, BI	80
R-Parameter zentral	RPNC <Kan.-Nr.> <Paramet.>	0 900 - 999	1 2	B0-B9, F0-FF BG, BI	B0-B9, F0-FF BG, BI	80
<b>NC-Alarme</b>						
NC-Alarme*5	NCAL			B0,F0		127

- \*1 Ist ein Zahlenformat nicht angegeben, ist vom entsprechenden FB kein Datentransfer möglich.
- \*2 Hat ein Datentyp mehr als einen Parameter WERT, steht die Anzahl in der Zeile, in der der Parameter Wert steht, der inkrementiert wird.  
 Eine Anzahl >1 ist nur unter folgenden Randbedingungen möglich:
- Datenblock in der NC **geschlossen** vorhanden
  - PLC-Quell- bzw. Zieladresse in ausreichender Länge vorhanden:
    - bei B0-B9/BG drei DW pro Wert
    - bei F0-FF zwei DW pro Wert
    - bei BI ein DW pro Wert
- \*3 Der Wert wird so übergeben, wie er intern aufgrund der NC-Maschinendaten für die Eingabefinheit festgelegt ist.
- \*4 Beim Auslesen des Programmzeigers werden bei ANZ=1 drei Daten geliefert:
- bei Ebene 0: Programmtyp (0=kein Programm angewählt, 1=Hauptprogramm, 2=Unterprogramm), Programmnummer, Satznummer
  - ab Ebene 1: Unterprogrammnummer, Durchlaufzahl, Satznummer
- \*5 In Vorbereitung

**8a Impuldiagramme der Nahtstellensignale****8b Bedingter Bausteinaufruf**

- 1: LESEN/SCHREIBEN
- 2: DATENÜBERGABE ANGEFORDERT
- 3: FIFO VOLL
- 4: DATENÜBERGABE LÄUFT
- 5: DATENÜBERGABE BELEGT
- 6: DATENÜBERGABE BEENDET+  
evtl. FEHLER
- 7: Signalwechsel durch FB
- 8: Signalwechsel durch Anwender
- 9: Signalwechsel durch FB;  
entfällt, wenn FIFO noch nicht voll ist
- t<sub>D</sub>: Belegung interne Schnittstelle durch  
Datenübergabe

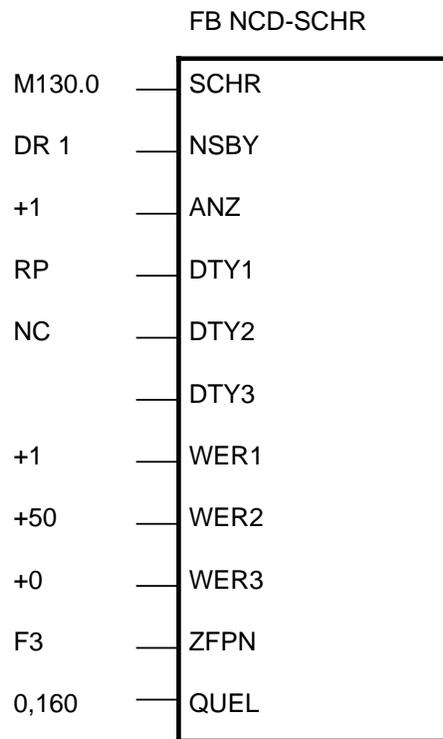
- 1: LESEN/SCHREIBEN
- 2: DATENÜBERGABE ANGEFORDERT
- 3: FIFO VOLL
- 4: DATENÜBERGABE LÄUFT
- 5: DATENÜBERGABE BELEGT
- 6: DATENÜBERGABE BEENDET+  
evtl. FEHLER
- 7: Signalwechsel durch FB
- 8: Signalwechsel durch Anwender
- 9: Anwender ruft Baustein nicht mehr  
auf
- 10: Signalwechsel durch FB;  
entfällt, wenn FIFO noch nicht voll ist
- t<sub>z</sub>: PLC-Zykluszeit
- t<sub>D</sub>: Belegung interne Schnittstelle durch  
Datenübergabe

### 9. Parametrierbeispiel für FB 62:

Übertragung des R-Parameters 50 an Kanal 1 aus MW 160, MW 162.

Zahlenformat PLC: Festpunkt

Zahlenformat NC: 2 Stellen nach dem Dezimalpunkt



## FB 65 M STACK

### Übergabemerker in Merkerstack

#### 1. Beschreibung

Mit Hilfe des Funktionsbausteines FB 65 kann der Merkerbereich MB 224 ... MB 255 bei Bedarf in den Merkerstack gerettet werden. So können Zwischenergebnisse und Übergabemerker der Funktionsbausteine beispielsweise bei einem verschachtelten Aufruf (ein FB ruft seinerseits einen anderen FB, der mit dem gleichen Merkerbereich arbeitet) vor Überschreiben geschützt werden. Der FB 65 kann nur in Verbindung mit dem FB 66 aufgerufen werden.

#### 2. Ergänzende Angaben

Bib.-Nr.:

Zu ladende FBs: keine

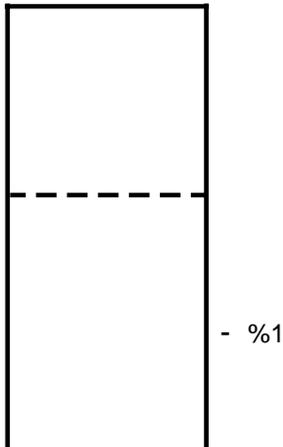
Art des FB-Aufrufs: absolut, bedingt

Einzugebende DBs: keine

Fehlermeldungen: %1 Stackpointerüberlauf bei Merkereintrag

#### 3. Bausteinaufruf

FB M STACK



## FB 66 STACK M

### Merkerstack in Übergabemerkerbereich

#### 1. Beschreibung

Der Funktionsbaustein FB 66 schreibt den Merkerbereich MB 224 ... MB 255, den der FB 65 in den Merkerstack transferiert hat, wieder an seinen Ursprungsort zurück. Damit ist gewährleistet, daß der augenblicklich ablaufende Funktionsbaustein mit seinen richtigen Vergangenheitswerten versorgt wird.

Der FB 66 kann nur in Verbindung mit dem FB 65 aufgerufen werden.

#### 2. Ergänzende Angaben

Bib.-Nr.:

Zu ladende FBs: keine

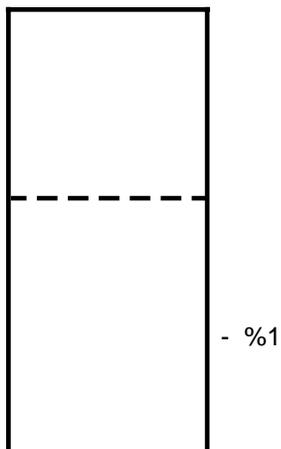
Art des FB-Aufrufs: absolut, bedingt

Einzugebende DBs: keine

Fehlermeldungen: %1 Stackpointerunterschreitung bei Merkeraustrag

#### 3. Bausteinaufruf

FB STACK M



## FB 190 K-LEITPC Signalaustausch mit Leit-PLC

### 1. Beschreibung

Der Funktionsbaustein FB 190 kopiert aus einem parametrierbaren Datenbaustein eine ebenfalls parametrierbare Anzahl von Worten in einen Koppelbaustein. Auf diesen Koppelbereich kann eine Leit-PLC zugreifen.

Die Signalfußrichtung ist über den Parameter RI/A wählbar (in den Koppel- bzw. aus dem Koppelbaustein in den Datenbereich der 810 GA3/820 GA3).

Die Fehlermeldung wird in ein frei wählbares Ausgangsbyte geschrieben.

DB-Koppelbereich

0	belegt
1	frei
.	
-	
-	
.	
126	belegt
127	

Die Nahtstelle für den Anwender ist sowohl auf der SINUMERIK- als auch auf SIMATIC-Seite ein **frei wählbarer Datenbaustein**.

### 2. Ergänzende Angaben

Bib.-Nr:

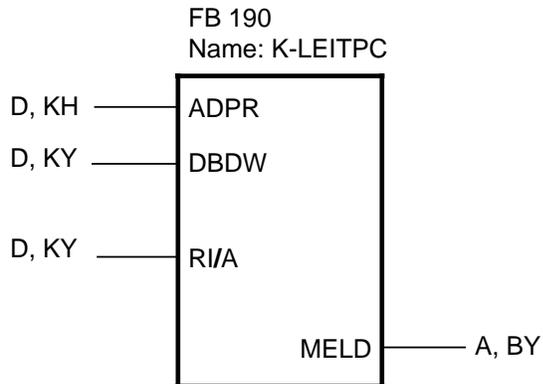
zu ladende FBs: keine

zu ladende DBs: keine

Art des FB-Aufrufs: absolut oder bedingt (SPA FB 190, SPB FB 190)

Einzugebende DBs: keine

### 3. Bausteinaufruf



### 4. Signalbeschreibung

**ADPR** Anfangsadresse im DUAL-PORT-RAM; wird nicht ausgewertet (nur aus Kompatibilitätsgründen zum Funktionsbaustein FB 190 für die SIMATIC-Seite)

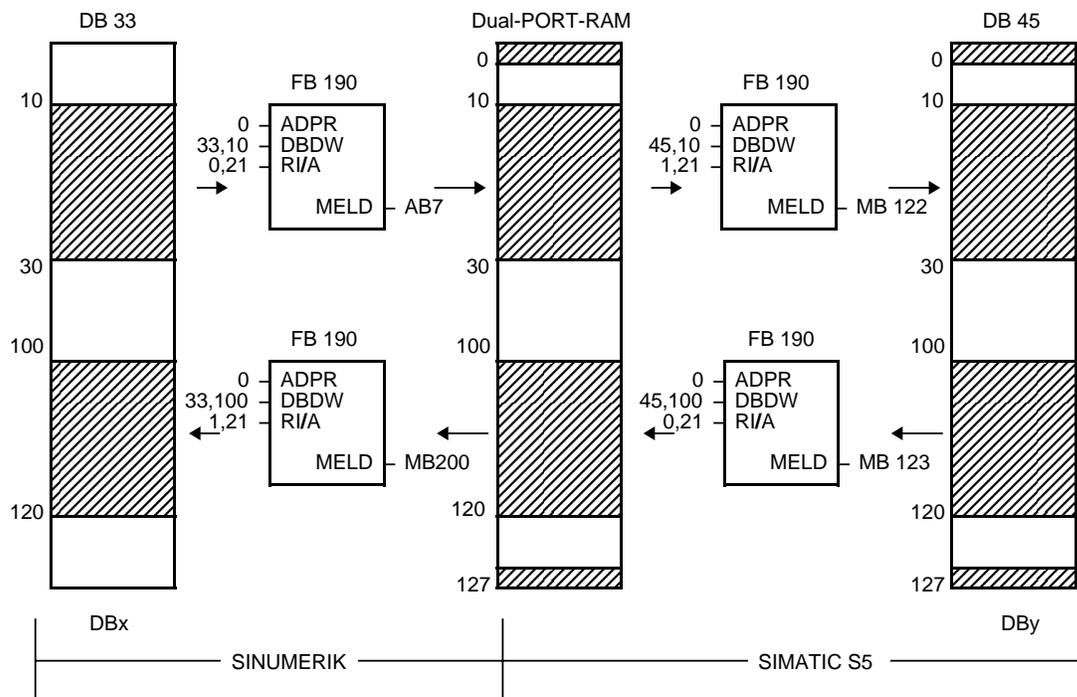
**DBDW** High-Byte: Datenbausteinnummer  
Low-Byte: Nr. des Anfangswortes und gleichzeitig Anfangsadresse im Koppelbereich

**R/A** High-Byte: Richtungssinn  
0 Transfer in Richtung Koppelbereich  
1 Transfer in Richtung Datenbaustein  
Low-Byte: Anzahl der zu übertragenden Worte (<127)

**MELD** Fehlermeldungen:

- 1 Schnittstellen Störung
- 2 DB nicht vorhanden
- 3 DB zu klein
- 4 DB im EPROM
- 5 Datenwort nicht zulässig (0 DW 127)
- 6 falsche Richtung
- 255 kein Fehler

## 5. Beispiel



Wie im Bild zu sehen ist, überträgt je ein Standardfunktionsbaustein FB 190 den mit I gekennzeichneten Datenblock vom DB 33 in der SINUMERIK zum DB 45 in der SIMATIC und den mit II gekennzeichneten Block in umgekehrter Richtung.

Die gewünschte Blocklänge wird durch den Parameter R/I/A festgelegt.

Die Lage des Datenblocks im Koppelbereich wird durch die Anfangswörter des Datenbausteins bestimmt. **Nicht** verwendet werden darf DW 0 und DW 127. Dazwischen kann der Datenblock frei wählbar positioniert werden.

Die Versorgung der Datenbausteine obliegt dem Anwender. Er kann per Programm mit Lade- und Transferbefehlen den Datenblock I des DB 33 beschreiben und dadurch festlegen, welche Funktionen und Meldungen zur SIMATIC S5 übertragen und dort verarbeitet werden sollen. Rückmeldungen von der SIMATIC S5 erfolgen über den Datenblock II.

Grundsätzlich soll die interne PLC der SINUMERIK nicht durch die angekoppelte SIMATIC S5 ersetzt werden. Die SIMATIC S5 hat bei Transferstraßen die Funktion einer Leitsteuerung. Es sollen z. B. nach dem Einschalten der Anlage alle SINUMERIK-Steuerungen vom zentralen Bedienpult auf Betriebsart "Referenzpunktfahren" geschaltet und danach das Automatik-Programm der SINUMERIK-Steuerungen vorgewählt und gestartet werden können.

Das Bild zeigt den Datenfluß zwischen dem Funktionsmakro auf SINUMERIK-Seite und dem Funktionsbaustein auf SIMATIC-Seite über den Koppelbereich.

### Wichtig:

Die Anzahl der zu übertragenden Datenwörter, sowohl bei dem Funktionsmakro FB 190 auf SINUMERIK-Seite als auch bei dem Funktionsbaustein FB 190 auf der SIMATIC-Seite, müssen bei den beiden miteinander kommunizierenden Funktionsbausteinen FB 190 gleich sein.

Die Nummer des ersten Datenwortes auf der SINUMERIK-Seite muß identisch mit der Nummer des ersten Datenwortes auf der SIMATIC-Seite sein.

# 7 Übergabeparameter und Betriebssystem-Maschinendaten

## 7.1 Übergabeparameter

In den Merkerbytes MB 0 bis MB 24 werden während des Anlaufs und der Programm-bearbeitung einzelne Bits gesetzt.

M	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bemerkung
0	1 Hz <sup>1)</sup>						Eins	Null	Grundsignal
1	OB-Nr. der aktuellen Bearbeitungsebene								
2						OB 2	OB 1		Grundstellung
3						OB 2	OB 1	OB 20	Neustart
4									
5									
6						OB 2			Bearbeitungszeitverzug
7									Bereitschaft
8								Störungssammelbit	
9									
10									
11									
12	Prozeßalarmbyte								Negative Flanke
13									
14									
15									
16	Prozeßalarmbyte								Positive Flanke
17									Blockweises Nachladen
18									
19	----- Anzahl der Sätze (Hex) des 1. Programmes -----								
20									
21	----- Anzahl der Sätze (Hex) des 2. Programmes -----								
22									
23	Ringpuffer voll	Ringpuffer leer						V.24 läuft	Überwachung
24	Meßtaster betätigt 2	Meßtaster betätigt 1			NC-BB2		Batt.	NC-Alarm	

Tabelle 7.1

1) gilt für IPO-Takt = 20 ms, sonst 50 x IPO-Takt

## **Blinktakt**

Das Systemprogramm stellt ein Blinksignal der Frequenz 0,5 Hz (bei IPO-Takt = 20 ms) zur Verfügung. Das Puls-Pausen-Verhältnis beträgt 1:1.

## **Grundstellung**

Die Grundstellungssignale werden für den ersten vollständigen Durchlauf einer Bearbeitungsebene nach erfolgtem Neustart gesetzt; sie werden wieder gelöscht, wenn die Bearbeitung des Anwenderprogramms durch den jeweiligen OB ein erstes Mal vollständig ausgeführt wurde.

## **Neustart**

Die Neustartsignale M3.1 bis M3.7 zeigen einen vorangegangenen Neustart während des Grundstellungsdurchlaufs an, d. h. sie sind solange gesetzt, bis die jeweilige Bearbeitungsebene zum ersten Mal nach Neustart vollständig ausgeführt wurde.

Das Neustartsignal M3.0 des OB 20 ist gleich Null, wenn der OB 20 im Wiederanlaufzweig aufgerufen wird. Wird der Wiederanlauf des OB 20 durch einen Netzausfall unterbrochen, wird dieser Organisationsbaustein nochmals mit Neustart von Anfang an durchlaufen, wobei M3.0 = 1 ist.

## **Bearbeitungszeitverzug**

Wird die alarmgesteuerte Bearbeitung vor ihrer Beendigung neu angefordert, so wird das entsprechende Bit im MB 6 gesetzt. In diesem Fall geht die Steuerung in den STOP-Zustand, im MB 6 erfolgt kein Eintrag. Bei Neustart und Wiederanlauf werden alle Bits wieder gelöscht.

## **Negative bzw. positive Flanke**

Tritt an einem Bit der zur Prozeßalarm-Bearbeitung ausgewählten Eingangsbytes ein Signalzustandswechsel auf, so wird bei einer negativen Flanke (Wechsel von Zustand 1 auf Null) das entsprechende Bit im Merkerbyte MB 12 gesetzt, bei einer positiven Flanke (Wechsel von Zustand Null auf 1) im MB 16.

## **Überwachung**

Die entsprechenden Bits werden gesetzt bei

- Meßtaster 2 betätigen (M24.7)
- Meßtaster 1 betätigen (M24.6)
- NC-Alarm mit Bearbeitungsstillstand (M24.3)
- Statisch auf "1" (M24.2)
- Batterieausfall (M24.1)
- NC-Alarm (M24.0)

Von der Funktion "Blockweises Nachladen" werden die MB 19 bis MB 23 belegt. Die Bedeutung der Signale ist in der Nahtstellenbeschreibung Teil 1 erläutert.

**7.2 PLC-Maschinendaten SINUMERIK 810 GA3/820 GA3**

## PLC-WERTE

MD-Nr.	Bezeichnung	Standardwert	Max. Wert	Eingabefinheit
0	Nr. für alarmverarbeitete Eingabebyte	7	31	---
1	Prozentuale PLC-Prozessorbelastung (OB 1+OB 2)	15 %	20 %	1 %
2 <sup>1)</sup>	PLC-Aufruf OB 1	0	3	IPO-Takt
3	Max. Interpreterlaufzeit (PLC-alarmgest.)	2000	2500	µs
4	Reserviert WS 510 P			
5	Zykluszeitüberwachung	70	320	1 ms
6	Nr. der letzten aktiven MC 5-Zeit	15	31	---
7				
8	Schnittstelle für DB 37	1	2	---
9 <sup>2)</sup>	Erweiterung DB-Speicher	8	32	1 kbyte
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				

1) gilt nur für ACOP

2) gilt nur für T/M ab SW2.3 und N ab SW2 nund setzt die Option PLC-Speichererweiterung voraus.

### 7.3 PLC-Maschinendaten-Bits SINUMERIK 810 GA3/820 GA3

#### PLC-BITS

MD-NR.	7	6	5	4	3	2	1	0
2000	PLC-MD 1007 BCD-codiert	PLC-MD 1006 BCD-codiert	PLC-MD 1005 BCD-codiert	PLC-MD 1004 BCD-codiert	PLC-MD 1003 BCD-codiert	PLC-MD 1002 BCD-codiert	PLC-MD 1001 BCD-codiert	PLC-MD 1000 BCD-codiert
2001	H-Nr. Kanal 1/2 BCD-codiert	T-Nr. Kanal 1/2 BCD-codiert	S-Nr. Kanal 1/2 BCD-codiert	M-Nr. Kanal 1/2 BCD-codiert	Reserviert S 810 GA 1	Reserviert S 810 GA 1	Reserviert S 810 GA 1 1)	Reserviert S 810 GA 1 1)
2002	M-Dekod. mit erweiterter Adr.		Vorsch./Eilig Korr. wirksam Achse 3-n <sup>2</sup> )	Anschluß dezentraler Peripherie	MSTT von E- nach A-Abbild übertragen	Reserviert S 810 GA 1	2. Achswahl- schalter vorhanden	keine Alarm- verarbeitung (OB 2)
2003	Freigabe Diagnose DB 1	Stückeln (Step 5- Bearbeitung)	DEMO- Betr.	Freigabe Step 5- System- Befehle	Wiederanlauf bei Spannungs- wiederkehr	Stop dezentrale Peripherie gestört	Stop bei Laufzeitüber- schr. OB 1	Stop bei Lauf- zeitüberschr. OB2 bzw. Be- arbeitungs- verzug OB2
2004								
2005								
2006								
2007								
2008								
2009								
2010								

1) nur 810G/820G:  
 2) n=5 bei SW 1  
 n=7 bei SW 2

Spindel-Kor- rekturschal- ter f. S2 vorhanden	1. Spindel- Korr.schalter f. S1 und S2 wirksam
--	---

## 8 Fehleranalyse

### 8.1 Fehlervarianten

Das PLC-Systemprogramm kann interne Fehler, eine fehlerhafte Programmierung durch den Anwender bzw. eine nicht zulässige Vorbesetzung von Maschinendaten erkennen.

Die Fehler können im Anlauf oder während des zyklischen Betriebes auftreten. Als Resultat kann entweder der PLC-Stopzustand eintreten oder der Anwender bekommt das Störereignis gemeldet und zwar auf zweierlei Art:

1. visuell auf dem NC-Bildschirm als PLC-Alarm
2. durch Setzen eines Sammelbits im Grundsignalmerkerbereich (M8.0) sowie eines Fehlerfeincodierungsbits im DB 1. Der Anwender kann dann programmtechnisch auf diesen Störfall reagieren. Die Bits werden vom PLC-Betriebssystem nicht nach einem Zyklus automatisch gelöscht,

#### DB 1:

	Bit	.7	.6	.5	.4	.3	.2	.1	.0
DL 8						Ausfall der Leit-PLC		Laufzeit OB 2 über- schritten	Laufzeit OB 1 über- schritten
DR 8						Verände- rung dez. Peripherie	Über- temperatur im EG	dez. Über- tragung gestört	EG reagiert nicht

sondern müssen vom Anwender bei Bedarf zurückgesetzt werden.

Die PLC-Störmeldung wird, wenn kein höherpriorer Alarm ansteht (NC, PLC) automatisch in der obersten Bildschirmzeile durch Alarmnummer und entsprechenden Text angezeigt. Der Alarm läßt sich bei Bedarf quittieren und verschwindet dann. Bei anstehenden höherprioreren Alarmen kann nur die Betätigung der Menütaste "PLC-Alarme" Auskunft über das Störereignis geben.

Führt das Störereignis zum PLC-Stopzustand, so erscheint stets in der obersten Bildschirmzeile der Alarm 3 "PLC-Stop".

Eine detaillierte Fehlerauswertung ist im PLC-Stopzustand auf 3 Arten möglich, auf die in den Kapiteln 8.2 und 8.3 näher eingegangen wird.

## 8.2 Unterbrechungsstack

Eine Analyse des aufgetretenen Fehlers ist mit Hilfe der Programmiergeräte-Auskunftsfunktion "Ausgabe USTACK" möglich. Es erfolgt die Anzeige der Steuerbits auf dem Bildschirm.

Abhängig vom Stand der verwendeten PG-Software kann an den nicht benutzten Stellen eine Bitkennung angezeigt werden, die für die PLC 150U aber irrelevant ist. Andererseits können die mit \* versehenen Kennungen als NB (nicht benutzt) angezeigt werden, bekommen für die PLC 150U aber eine bestimmte Fehlerbedeutung.

### STEUERBITS

NB	NB	BSTSCH	SCHTAE	ADRBAU	SPABBR	NAUAS	NB
NB	NB	NNN*	PERUNKL*	NB	NB	NB	NB
STOZUS	STOANZ	NEUSTA	WIEDAN	BATPUF	NB	NB	NB
KEINPS*	UAFEHL	MAFEHL	EOVH	NB	NB	OBWIED	NB
NB	NB	KOPFNI	NB	WECKFE	PADRFE	ASPLUE	RAMADFE
EAADFE*	SYNFEH	NINEU	NIEWIED	RUFBST	QVZNIN	SUMF	URLAD
NB	NB	STS*	STP*	TBWFEH	LIR/TIR	NB	NB
NB	LPTP*	NB	NB	NB	NB	NB	NB

Im einzelnen haben die Steuerbits folgende Bedeutung:

<b>BSTSCH</b>	Koordinierungsbit Baustein schieben
<b>SCHTAE</b>	Speicherinhalt zusammenschieben tätig
<b>ADRBAU</b>	Erfolgreicher Adresslistenaufbau nach Neustart/Wiederanlauf
<b>SPABBR</b>	Speicherinhalt zusammenschieben abgebrochen
<b>NAUAS</b>	Dezentraler Netzspannungsausfall
<b>NNN</b>	Nicht interpretierbarer Operationscode/unzulässiger Befehlsparameter
<b>PERUNKL</b>	Peripherie unklar (im Erweiterungsgerät)
<b>STOZUS</b>	AG im Stopzustand (darf nur vom System gesetzt bzw. gelöscht werden)
<b>STOANZ</b>	Anzeige des Stopzustandes
<b>NEUSTA</b>	Neustartkennung
<b>WIEDAN</b>	Wiederanlaufkennung
<b>BATPUF</b>	Netzgerät ist batteriegepuffert
<b>KEINPS</b>	Anwenderprogrammspeicher oder Anwenderdatenspeicher nicht bestückt
<b>UAFEHL</b>	STOP-Zustand wurde durch ein Unterbrechungsereignis erreicht
<b>MAFEHL</b>	Anlauffehleranzeige (Steuerung konnte nicht in Normalbetrieb übergehen)
<b>EOVH</b>	Eingangsbytes für Prozeßalarmanschluß (vier aufeinanderfolgende Eingänge) vorhanden
<b>OBWIED</b>	Organisationsbaustein für Wiederanlauf (OB 20) aktiv
<b>KOPFNI</b>	Bausteinkopf im Anwenderspeicher nicht interpretierbar

<b>WECKFE</b>	Sammelanzeige für Laufzeitfehler eines Anwender-Programms (OB 1, OB 2)
<b>PADFRE</b>	Fehler im EPROM-Speicher
<b>ASPLUE</b>	Anwenderspeicher lückenhaft adressiert
<b>RAMADFE</b>	Fehler im RAM-Speicher
<b>EAADFE</b>	Fehler im Peripheriebereich
<b>SYNFEH</b>	Unzulässige Bausteinlänge im Anwenderspeicher oder Synchronisationswort eines Bausteins zerstört
<b>NINEU</b>	Neustart nicht möglich (Urladen notwendig)
<b>NIWIED</b>	Wiederanlauf nicht möglich (Neustart notwendig)
<b>RUFBST</b>	Aufruf eines nicht vorhandenen Bausteines
<b>QVZNIN</b>	Quittungsverzug
<b>SUMF</b>	Summenfehler
<b>URLAD</b>	Urlöschen mit anschließendem Urladen des Anwenderspeichers nötig
<b>STS</b>	Direkter Stop durch Befehl STS
<b>STP</b>	Unterbrechungsanzeige nach Befehl STP
<b>TBWFEH</b>	Quittungsverzug beim Blocktransferbefehl
<b>LIRTIR</b>	Quittungsverzug beim Befehl LIR oder TIR
<b>LPTP</b>	Quittungsverzug beim Direktzugriff auf die Peripherie durch Anwenderbefehle

Als nächste Anzeige läßt sich folgendes Bild ausgeben:

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

```

TIEFE:      01
BEF-REG:  0000   SAZ:    0000   DB-ADR: 0000   BA-ADR: 000
BST-STP:  0000   XX-NR.:      DB-NR.:      YY-Nr.:
                REL-SAZ:      DBL-REG:
VEK-ADR:  0000   UAMK:  0000   UALW:  0000
AKKU1:    0000 0000 AKKU2: 0000 0000 AKKU3: 0000 0000 AKKU4: 0000 0000
ERGEBNISANZEIGE:  ANZ1  ANZO  OVFL  OVFLS  ODER  STATUS VKE ERAB
STÖRUNGSURSACHE: STOPS STUEB NAU QVZ ZYK BAU SUF STUEU ADF PARI TRAF

```

Die Anzeigen AKKU 3 und AKKU 4 haben bei der PLC keine Bedeutung, da die Akkus hier nicht vorhanden sind. Für die mit "-" gekennzeichneten Bits kann, wie bei den Steuerbits, eine irrelevante Bitkennung angezeigt werden. Zur Bedeutung der Ergebnisanzzeigebits siehe Kapitel 9.2.2.

<b>TIEFE</b>	Bei der PLC wird immer das jüngste Unterbrechungsereignis angezeigt (TIEFE=01)
<b>BEF-REG</b>	Befehlsdarstellung (hexadezimal)

<b>SAZ</b>	Stepadresszähler Die Befehlsdarstellung wird durch einen Zugriff in den Anwenderprogrammspeicher über den SAZ ermittelt. Da jedoch der SAZ stets auf den nächsten auszuführenden Befehl zeigt, wird der Befehl interpretiert, der an der um zwei Bytes niedrigeren Adresse steht. Daher liefert die Anzeige BEF-REG keine richtige Information bei Doppelwortbefehlen sowie unmittelbar nach Sprungbefehlen.
<b>DB-ADR</b>	Bausteinanfangsadresse
<b>BST-STP</b>	Bausteinstackpointer
<b>XX-NR.</b>	Aktueller Baustein, dessen Bearbeitung die Unterbrechung verursacht
<b>DB-NR.</b>	Aktueller Datenbaustein
<b>YY-NR.</b>	Baustein, von dem aus der aktuelle Baustein (XX) aufgerufen wird
<b>REL-SAZ</b>	Relativer Stepadresszähler im aktuellen Baustein
<b>DBL-REG</b>	Register, in dem die Datenbausteinlänge geführt wird
<b>VEK-ADR</b>	Vektoradresse für Externspeicher
<b>UAMK</b>	Steuerregister 1 (hexadezimal)
<b>UALW</b>	Steuerregister 2 (existiert nicht)
<b>STOPS</b>	ohne Bedeutung
<b>STUEB</b>	Bausteinstack-Überlauf (maximal 12 Einträge möglich)
<b>NAU</b>	Netzspannungsausfall (Steuerung bleibt im STOP-Zustand, bis dezentraler Ausfall beseitigt)
<b>QVZ</b>	Quittungsverzug
<b>ZYK</b>	Zykluszeitüberschreitung
<b>BAU</b>	Batterieausfall
<b>SUF</b>	Substitutionsfehler
<b>STUEU</b>	Unterbrechungsstack-Überlauf (maximal 2 Ebenen möglich)
<b>ADF</b>	Adressierfehler
<b>PARI</b>	Parity Fehler
<b>TRAF</b>	Transferfehler

## 8.3 Fehlerfeincodierung

### 8.3.1 Anzeige über PG

Über die Auskunftsfunktionen des Programmiergerätes lassen sich durch Eingabe der Pseudo-  
adresse F000<sub>hex</sub> weitere Informationen zur Unterbrechungsanalyse ausgeben.

Auf dem PG-Bildschirm erscheint das folgende Bild:

ADR	WERT	ADR	WERT	ADR	WERT	ADR	WERT
F000	00xx	F001	61yy	F002	xxx1	F003	xxx2
F004	xxx3	F005	zob2	F006	0000	F007	0000
F008	0000	F009	0000	F00A	0000	F00B	0000
F00C	0000	F00D	0000	F00E	0000	F00F	0000
F010	0000	F011	0000	F012	0000	F013	0000
F014	0000	F015	0000	F016	0000	...	

Die Bedeutung der Adressen F00C bis F011 wird in Abschnitt 9.1.2 erläutert, die nachfolgenden Anzeigen haben keine Bedeutung mehr.

Für die Adressen F000 bis F005 gilt:

- xx** = interne Fehlerfeincodierung (FEHLCODE bei PLC-Stoppzustand)
- 61yy** = NC-Alarmnummer 6100 ... 6163
- xxx1** = Ergänzende Fehlerangabe, Wort 1
- xxx2** = Ergänzende Fehlerangabe, Wort 2
- xxx3** = Ergänzende Fehlerangabe, Wort 3
- zob2** = Ereigniszähler Bearbeitungszeitverzug im OB 2

Die ergänzenden Fehlerangaben dienen der genaueren Analyse eines Quittungsverzugsfehlers bzw. eines Parametrierfehlers. Bei einem Neustart werden alle ergänzenden Fehlerangaben gelöscht.

#### 1. PLC-QVZ:

Im Wort 1 (xxx1) wird der Opcode des Befehls, der den Quittungsverzug erzeugt hat, hinterlegt. xxx2 bzw. xxx3 nehmen den folgenden Inhalt an:

- Beim Quittungsverzug durch die Befehle LIR, TIR, TNB oder TNW:
  - xxx2** Offsetadresse
  - xxx3** Segmentnummer  
des nicht adressierten Speichers
- Bei Quittungsverzug durch Substitutionsbefehle:
  - xxx2** der substituierte Befehl
- Bei Quittungsverzug durch die Befehle LPB, LPW, TPB, TPW:
  - xxx2** = 000E (Quittungsverzug beim Laden von den Eingangssignalformen)
  - = 000A (Quittungsverzug beim Transportieren zu den Ausgangssignalformen)
  - xxx3** = Byteadresse (BCD-codiert) des Befehlsparameters

#### 2. Parametrierfehler

- xxx3** = Nummer (BCD-codiert) des unzulässigen Befehlsparameters  
(Peripheriebyte-Nr., Segment-Nr., Baustein-Nr., Zeit-/Zähler-Nr.)

Darstellung des Steuerregisters UAMK (binär):

Datenbit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Funktion
Bit 0 und 1 X X X X X X X X X X X X X X 0 0 X X X X X X X X X X X X X X 1 1	Hardwarevariante
Bit 2 X X X X X X X X X X X X X 0 X X X X X X X X X X X X X X X 1 X X	Batteriealarm (1=ok.)
Bit 3 X X X X X X X X X X X X 0 X X X X X X X X X X X X X X X 1 X X X	NMI vom Wartungsfeld
Bit 4 X X X X X X X X X X 0 X X X X X X X X X X X X X X 1 X X X X	NMI vom Netzteil
Bit 5 X X X X X X X X X 0 X X X X X X X X X X X X X X 1 X X X X X	Freigabe System NMI (AUX 0) NMI-WF wirkt nur auf eigener CPU NMI-WF wird anderen CPU's als System NMI (AUX 0) signalisiert
Bit 6 X X X X X X X X 0 X X X X X X X X X X X X X X 1 X X X X X X	Parity-Überwachung
Bit 7 X X X X X X X 0 X X X X X X X X X X X X X X 1 X X X X X X X	Wartungsfeld (1=steckt)
Bit 8 X X X X X X 1 X X X X X X X	MPC EOB Indikator (EOB = 1)
Bit 9 X X X X X 0 X X X X X X X X X X X X X 1 X X X X X X X X	MPC Fehlerindikator (Fatal Error = 1)
Bit 10 X X X X X 0 X X X X X X X X X X X X X X 1 X X X X X X X X X	MPC Flag 0 Flag = 0 Befehlsausgabesperre im EG Flag wird von *OUTDS gesteuert
Bit 11 X X X X 0 X X X X X X X X X X X X X X 1 X X X X X X X X X X	MPC Flag 1 FLI = 0 FLI = 1 MPC - RESET im EG
Bit 12 X X X X 0 X X X X X X X X X X X X X X 1 X X X X X X X X X X	Taktfrequenz (0 = 8 Mhz, 1 = 16 Mhz)
Bit 14 X 0 X X X X X X X X X X X X X X 1 X X X X X X X X X X X X X	Parity-Bit setzen high Byte
Bit 15 0 X X X X X X X X X X X X X X 1 X X X X X X X X X X X X X X	Parity-Bit setzen low Byte

Der "Ereigniszähler Bearbeitungszeitverzug" erfasst, wie oft eine begonnene Bearbeitung des alarmgesteuerten Programms (Aufruf durch OB 2) vor Beendigung der ursprünglichen Alarmbearbeitung erneut angefordert wurden.

In der ergänzenden Fehlerangabe, Wort 1 (xxx1) wird die Nummer des Organisationsbausteins hinterlegt, durch den der Bearbeitungszeitverzug ausgelöst wurde.

Zwischen der internen Fehlerfeincodierung xx und der NC-Alarmnummer 61yy besteht folgender Unterschied:

- xx – Anzeige hexadezimal zwischen 100 und 163 (64h - 13h)
- Wert 00 hat stets einen PLC-Stopzustand zur Folge
- 61yy – Anzeige BCD-codiert zwischen 6100 und 6163 (auch auf dem NC-Bildschirm)
- aus einem Wert 0000 resultiert nicht unbedingt ein PLC-Stopzustand

Die nachfolgende Tabelle gibt eine Übersicht über die einem Fehlereignis zugeordnete Fehlernummer und zwar für die beiden Fehlerkennungen xx und 61yy:

Fehlerkennung Fehlerereignis	XX	61yy
Fehler während der STEP 5-Programmbe- arbeitung	100-115 (64h-73h)	6100-6115
Fehler im Anlauf	116-143 (74h-8Fh)	6116 -6143
Fehler während dem zyklischen Betrieb	144-149 (90h-95h)	6144-6149
interrumpgesteuerte Fehlermeldungen	150-163 (96h-A3h)	6150-6163

### 8.3.2 Anzeige über NC-Bildschirm

Bei einem Stopzustand der PLC erscheint stets der Alarm 3 "PLC-Stop" als höchstpriorer PLC-Alarm auf dem NC-Bildschirm. Die Anzeige erfolgt in der Alarmzeile, vorausgesetzt es steht kein höherpriorerer Alarm an.

Über die Menütasten "Diagnose" und "PLC-Alarm" kann man außer der Sammelstörmeldung auch eine detailliertere Information über das Störereignis erhalten. Mit einer Alarmnummer zwischen 6100 und 6163 und einem entsprechenden Text wird dem Anwender die Ursache des PLC-Stopzustandes mitgeteilt. In der Anzeige können zusätzlich noch aufgetretene Anwenderalarme erscheinen, wobei die Reihenfolge vom zeitlichen Auftreten der jeweiligen Alarme abhängt. So erscheint der Betriebssystemalarm in der Regel als letzter Anzeigewert. Sowohl der Alarm 3 als auch der differenziertere Fehleralarm kann nur durch POWER-ON-RESET oder durch das Kommando – AG-Start – über ein Programmiergerät gelöscht werden – vorausgesetzt die Fehlerursache ist behoben. Auf den folgenden Seiten ist eine Zusammenstellung der möglichen PLC-Betriebssystemalarme aufgelistet.

## 8.4 Alarmliste

**ALARMLISTE SYSTEM 810 GA3/820 GA3  
PLC-BETRIEBSSYSTEM- UND ANWENDERALARME**

Alarm-Nr.	Bezeichnung	Satz-/Reih. bez.	Verriegelung von:					aufgerufen von	
						Be- arb	NC- Start	NC- BB2	PLC
6000	Anwenderalarme								Meldung
	”								Meldung
6063	Anwenderalarme								Meldung
6100	Fehlender Signalumformer								STOP
6101	Unzulässiger MC5-Code								STOP
6102	Unzul. MC5-Parameter								STOP
6103	Transfer in fehlenden DB								STOP
6104	Substitutionsfehler								STOP
6105	Fehlender MC-5-Baustein								STOP
6106	Fehlender DB								STOP
6107	Unzul. Segment LIR/TIR								STOP
6108	Unzul. Segment Block-transfer								STOP
6109	Überlauf B-Stack								STOP
6110	Überlauf U-Stack								STOP
6111	MC5-Befehl STS								STOP
6112	MC5-Befehl STP								STOP
6113	Unzul. MC5-Zeit/Zähler								STOP
6114	Funktionsmakro								STOP
6115	Systembefehle gesperrt								STOP
6116	MD 0000: Alarmbyte-Nr.								STOP
6117	MD 0001: CPU-Belastung								STOP
6118	MD 0003: Alarmlaufzeit								STOP
6119	MD 0005: Zykluszeit								STOP
6120									
6121	MD 0006: Letzte MC5-Zeit								STOP

<b>ALARMLISTE SYSTEM 810 GA3/820 GA3</b> <b>PLC-BETRIEBSSYSTEM- UND ANWENDERALARME</b>
---

Alarm-Nr.	Bezeichnung	Satz-/Reih. bez.	Verriegelung von:					aufgerufen von	
						Be- arb	NC- Start	NC- BB2	PLC
6122	Unzulässige Brückenrangierung								Meldung
6123	Unzulässige Servoabstastzeit								STOP
6124	Lücke im MC5-Speicher								STOP
6125	Doppelbelegte Eingänge								STOP
6126	Doppelbelegte Ausgänge								STOP
6127	Alarmbyte fehlt								STOP
6128	Falsche Peripherierangierung								STOP
6129									
6130	Synch.-Fehler Grundprogr.								STOP
6131	Synch.-Fehler MC5-Progr.								STOP
6132	Synch.-Fehler MC5-Daten								STOP
6133	Unzul. Baust.Grundprogr.								STOP
6134	Unzul.Baustein MC5-Progr.								STOP
6135	Unzul.Baustein MC5-Daten								STOP
6136	Summenfehler MC5-Baust.								STOP
6137	Summenfehler Grundprogr.								STOP
6138	EG reagiert nicht								*
6139	EG-Übertragungsfehler								*
6140									
6141									
6142									
6143	Dekodier-DB nicht vorhand.								STOP
6144	Lange Dekodier-DB nicht Modulo 6								STOP
6145	Falsche Anzahl Dekodier-Einheiten								STOP
6146	Dekodier-DB zu kurz								STOP
6147	Veränderg. i. d. dezentralen Peripherie								*

**ALARMLISTE SYSTEM 810 GA3/820 GA3**  
**PLC-BETRIEBSSYSTEM- UND ANWENDERALARME**

Alarm-Nr.	Bezeichnung	Satz-/Reih. bez.	Verriegelung von:					aufgerufen von	
						Be arb	NCSt Start	NC BB2	PLC
6148	Übertemperatur im EG								Meldung
6149	Stop über Softkey PG								STOP
6150	QVZ: Anwenderspeicher								STOP
6151	QVZ: Koppelspeicher								STOP
6152	QVZ: LIR / TIR								STOP
6153	QVZ: TNB / TNW								STOP
6154	QVZ: LPB / LPW / TPB / TPW								STOP
6155	QVZ: Substitutionsbefehl								STOP
6156	QVZ nicht interpretierbar								STOP
6157	QVZ SPA FB / SPB FB								STOP
6158	QVZ bei Ein-/Aus-Transfer								STOP
6159	Laufzeit-Überschr. Step5-Progr.								*
6160	Laufzeit-Überschreitg. OB2								*
6161	Zykluszeit-Überschreitung								STOP
6162	Bearbeitungsverzug OB2								*
6163	Ausfall der Leit-PLC								Meldung
6164									
7000	PLC-Anwenderbetriebs-meldungen								Meldung
	”								Meldung
7063	”								Meldung

\* PLC-Verhalten ist von der MD-Voreinstellung abhängig (Stopp bzw. lediglich Meldung)

## 9 STEP 5-Befehlsvorrat mit Programmierbeispielen

### 9.1 Speicheraufteilung

Um mit einem Programmiergerät (z. B. PG 675, PG 685) über die Funktion „Auskunft Adresse“ auch die Bereiche erreichen zu können, die außerhalb des Adressbereiches liegen, welcher über die „normalen“ STEP 5-Befehle adressierbar ist (z. B. Prozeßabbild), benötigt man die sogenannte „Segmentweiche“. Sie ist eine Hilfsgröße, mit der bestimmte Bereiche – die Segmente – innerhalb des 1 Megabyte großen Adressbereiches des Prozessors angewählt werden können.

Jede Adresse setzt sich aus der Offsetadresse (Stellung im Segment) und der Segmentadresse (abhängig von der Stellung der Segmentweiche) zusammen.

Die folgenden Segmente können vom Anwender über eine Segmentnummer angewählt werden – und zwar einmal über die Diagnosefunktion „Auskunft Adresse“ eines Programmiergerätes bzw. über spezielle STEP 5-Systembefehle (siehe Kapitel 9.4.13):

Segmentnummer	Bezeichnung	Segmentadresse	Offsetadresse (wortorientiert)	Zugriffsmöglichkeit
1	PLC-Systemsegment	D00 <sub>hex</sub> (bei COP) C00 <sub>hex</sub> (bei ACOP)	0000 <sub>hex</sub> ...2C00 <sub>hex</sub>	lesen und schreiben
5	PLC-Anwenderprogrammsegment	1000 <sub>hex</sub> 1400 <sub>hex</sub> + var. (bei ACOP und SW2.3 bei T/M bzw. SW2 bei N)	0000 <sub>hex</sub> ...4000 <sub>hex</sub> 0000 <sub>hex</sub> ... var. (bei ACOP und SW2.3 bei T/M bzw. SW2 bei N)	lesen und schreiben
6	PLC-Anwenderdaten-Segment	1400 <sub>hex</sub> 1280 <sub>hex</sub> (bei ACOP und SW2.3 bei T/M bzw. SW2 bei N)	0000 <sub>hex</sub> ...2E20 <sub>hex</sub> 0000 <sub>hex</sub> ...var. (bei SW2.3 bei T/M bzw. SW2 bei N)	lesen und schreiben

Wird für die Segmentweiche bzw. Segmentnummer ein anderer Wert vorgegeben, so wird dieser entweder ignoriert (bei Auskunftfunktion eines Programmiergerätes) bzw. führt zu einer Fehlermeldung (bei STEP 5-Systembefehlen LIR, TIR, TNB, TNW).

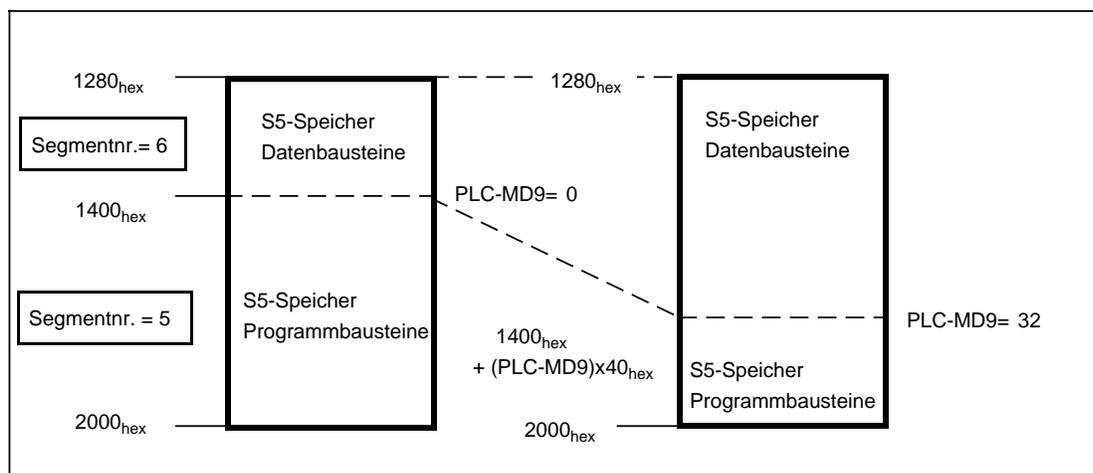
Bei einem Neustart wird automatisch das Anwenderprogrammsegment mit der Stellung 5 der Segmentweiche durch das PLC-Betriebssystem angewählt.

### Speicheraufteilung

Segment-  
nummer

1	1)	PLC-Systemdaten:            Prozeßabbilder, Bausteinlisten, Speicherverwaltung, BSTACK, USTACK u.s.w. (            )
5	2)	Anwenderprogramm-            max. 16 kB <sup>3)</sup> für STEP 5-Anwender- speicher:                            programm (OB, PB, SB, FB) Max. 48 kB bei PLC-Speichererweiterung <sup>4)</sup>
6	2)	Anwenderdaten-                max. 6 kB für STEP 5-Anwenderdaten- speicher:                            bausteine und 0,3 kB für Datenbausteine, die vom PLC-Betriebssystem im Anlauf eingerichtet werden. Max. 38 kB bei PLC-Speichererweiterung <sup>4)</sup>

Bei ACOP ab SW2.3 bei SINUMERIK 810 T/M GA3, 820 T/M GA3 bzw. SW2 bei SINUMERIK 810 N GA3, 820 N GA3 mit Option PLC-Speichererweiterung gilt folgende Speicheraufteilung :



- 1) interner RAM-Bereich auf der CPU-Baugruppe
- 2) gepufferter RAM-Bereich auf Maschinendatenkärtchen
- 3) max. 12 kB bei COP, max. 16 kB ab ACOP ohne PLC-Speichererweiterung
- 4) bei PLC-Speichererweiterung (erst ab SW2.3 bei T/M bzw. SW2 bei N) entfällt die Option C49 (Programmspeichererweiterung auf 128 kByte)

## 9.1.1 Verändern der Segmentweiche

Der folgende Ablauf gilt für das Programmiergerät PG 675:

- Aufruf der Auskunftsfunktionen mit der Taste F7
- Aufruf von beliebigen Speicherbereichen mit der Taste F8
- Eingabe der Pseudoadresse  $E000_{hex}$
- Betätigen der Übernahmetaste, dann
- sofortiges Betätigen der Abbruchtaste

Es wird als erstes Wort die momentane Stellung der Segmentweiche angezeigt.

- Betätigen der Korrekturtaste
- Eingabe der neuen Segmentadresse

## 9.1.2 Bausteinlisten

Durch Eingabe der Pseudoadresse  $F000_{hex}$  lassen sich die Anfangsadressen der Bausteinlisten ausgeben (zur Bedeutung der Adressen  $F000$  bis  $F009$  siehe Kapitel 8.2):

**F00C** Anfangsadresse der OB-Bausteinliste  
**F00D** Anfangsadresse der FB-Bausteinliste  
**F00E** Anfangsadresse der DB-Bausteinliste  
**F010** Anfangsadresse der SB-Bausteinliste  
**F011** Anfangsadresse der PB-Bausteinliste

Bei der Ausgabe der Bausteinlisten über die Adreßliste ist zu beachten:

- Die erste Adresse des Adreßlisteneintrags ist die Offsetadresse, die zweite ist die Segmentadresse.
- In den angegebenen Adressen sind High- und Low-Byte vertauscht.
- Die Offsetadressen sind wortorientiert; Ausnahme: die Einträge der DB-Liste sind byteorientiert. Sie sind nach der High-Low-Vertauschung durch zwei zu dividieren<sup>3)</sup>.

Werden die Bausteine über die Buchhalterfunktion des Programmiergerätes ausgegeben, so erscheinen Datenbausteine mit einer um  $2000_{hex}$ <sup>1)</sup> erhöhten Offset-Adresse. Residente Funktionsmakros erscheinen mit einer um  $6000_{hex}$  erhöhten Offset-Adresse<sup>2)</sup>. Ab SW2.3 bei SINUMERIK T/M gilt für Programm-,Organisations-,Schritt- und Funktionsbausteine die Offset-Adresse  $2000_{hex}$ <sup>4)</sup>.

1) bei ACOP  $2000_{hex}$   
 bei COP  $8000_{hex}$   
 ab SW2.3 bei T/M bzw. SW2 bei N  $1400_{hex}$

2) bei ACOP  $6000_{hex}$   
 bei COP  $8000_{hex}$   
 ab SW2.3 bei T/M bzw. SW2 bei N  $800_{hex}$

3) bei ACOP sind alle Offsetadressen byteorientiert

4) bei aktiver PLC-Speichererweiterung gilt als Offset-Adresse  $2000_{hex} + (MD9) \times 20_{hex}$

## 9.2 Allgemeine Hinweise

Der STEP 5-Befehlsvorrat teilt sich in Grundoperationen und ergänzende Operationen auf.

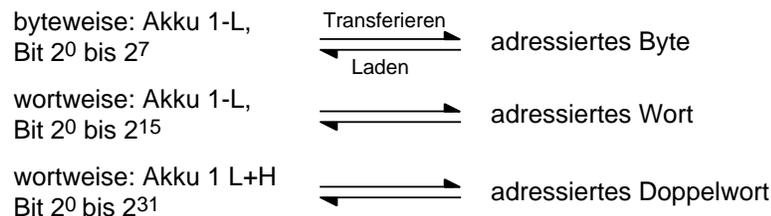
Die Grundoperationen sind für die Ausführungen von einfachen binären Funktionen vorgesehen. Sie können in der Regel in den drei Darstellungsarten (KOP, FUP und AWL) der STEP 5-Sprache am Programmiergerät ein- bzw. ausgegeben werden.

Die ergänzenden Operationen sind für komplexe Funktionen (z. B. Regeln, Melden, Protokollieren) vorgesehen, wobei sie nicht graphisch darstellbar sind und nur in Anweisungsliste (AWL) am Programmiergerät ein- bzw. ausgegeben werden können.

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<sup>31</sup> ... 2<sup>16</sup>    2<sup>15</sup> ... 2<sup>0</sup>  
                          Akku-H      Akku-L

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



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

### 9.2.1 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:    +0 ... 32 767  
                          - 1 ... - 32 768  
 Steht in Akku-L und wird als eine 16-Bit-Dualzahl in 2er-Komplement-Darstellung aufgefaßt (höchstwertiges Bit=Vorzeichenbit).
- II) Festpunkt-Doppelwort:    +0 ... 2 147 483 647  
                                      - 1 ... 2 147 483 648  
 Steht in Akku und wird als eine 32-Bit-Dualzahl in 2er-Komplement-Darstellung aufgefaßt (höchstwertiges Bit=Vorzeichenbit).



## 9.2.2 Ergebnisanzeigen der PLC

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 anzeigensetzende Befehle und anzeigenauswertende Befehle.

Entsprechend den beiden Befehlsgruppen gibt es "Bit-Anzeigen" und "Wort-Anzeigen". Das Anzeigenbyte bei der PLC sieht folgendermaßen aus

2 <sup>7</sup> Wort-Anz.				Bit-Anz. 2 <sup>0</sup>				
Anz. 1	Anz. 0	OV	OS	OR	—	VKE	ER	—

Zu den Bit-Anzeigen:

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

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

**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.

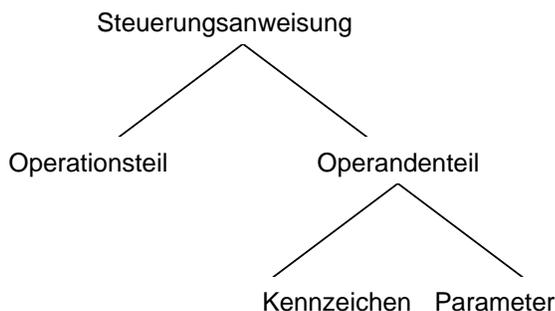
Anz 1, Anz 0 sind codierte-Ergebnis-Anzeigen, deren Interpretation aus der folgenden Tabelle ersichtlich wird.

Wort-Ergebnis-Anzeigen			Festpunkt-rechnung-Ergebnis	Bool Ergebnis	Vergleich Inhalte von Akku 1+Akku 2	Schieben geschobenes Bit	Gleitpunkt-rechnung-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").

### 9.2.3 STEP 5-Befehlsdarstellung

Eine Anweisung ist demnach folgendermaßen aufgebaut:



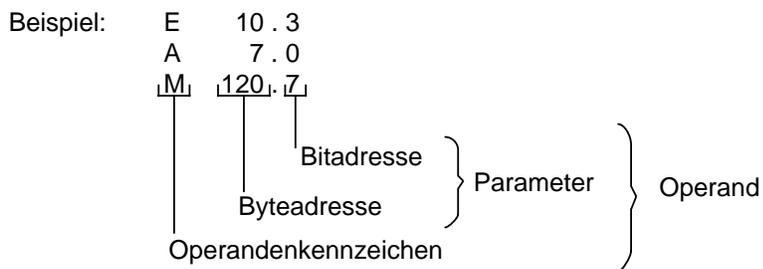
Der **Operationsteil** beschreibt die auszuführende Funktion. Er sagt aus, *was* der Prozessor tun soll.

Der Operandenteil enthält die für die Ausführung der Operation notwendigen Angaben. Er sagt aus, **womit** der Prozessor etwas tun soll. Die Programmiersprache STEP 5 kennt folgende Operandenbereiche:

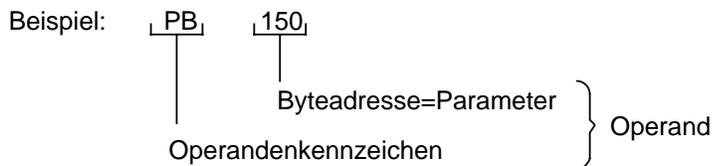
Eingänge E	sie stellen die Schnittstelle vom Prozeß zum Automatisierungsgerät dar (Prozeßabbild),
Ausgänge A	sie stellen die Schnittstelle vom Automatisierungsgerät zum Prozeß dar (Prozeßabbild),
Merker M	sie sind zur Speicherung binärer Zwischenergebnisse vorgesehen,
Daten D	sie sind zur Speicherung digitaler Zwischenergebnisse vorgesehen,
Zeiten T	mit ihnen werden Zeitfunktionen realisiert,
Zähler Z	mit ihnen werden Zählerfunktionen realisiert,
Peripherie P, Q	es wird die Prozeßperipherie (Ein-/Ausgabebaugruppen) direkt angesprochen,
Konstanten K	sie stellen eine fest vorgegebene Zahl dar,
Bausteine OB, PB, FB, DB	sie dienen zum Strukturieren des Programms.

Die Bezeichnung der Operandenbereiche ist das **(Operanden-)Kennzeichen**. Um einen bestimmten Operanden in einem Operandenbereich anzusprechen ist die Angabe eines Parameters notwendig.

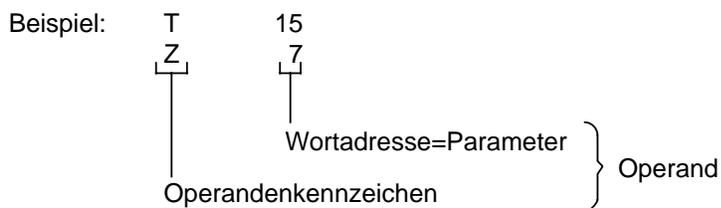
Der **Parameter** gibt die Adresse eines Operanden an. Die Operandenbereiche Eingänge E, Ausgänge A und Merker M werden byteweise adressiert, d. h. die Zahlenangabe bezieht sich auf ein bestimmtes Byte dieser Operandenbereiche (Byteadresse). Diese Operandenbereiche können auch bitweise angesprochen werden. Die Bitadresse ist von der Byteadresse durch einen Punkt getrennt.



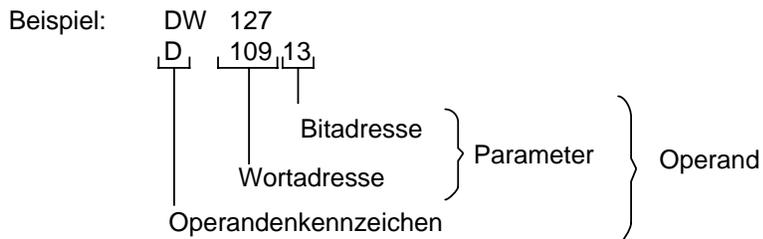
Der Operandenbereich Peripherie P wird ebenfalls byteweise adressiert, hat jedoch keine Bitadresse.



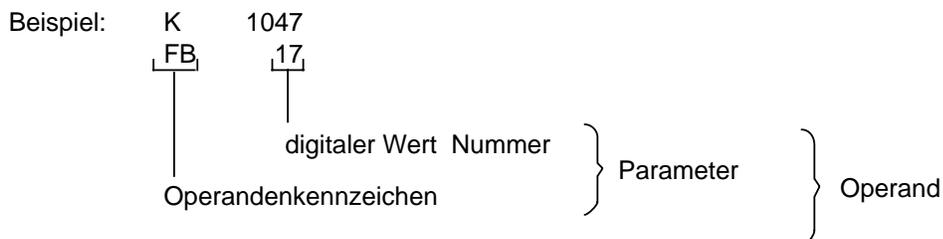
Die Operandenbereiche Zeiten T und Zähler Z werden wortweise adressiert (Wortadresse). Diese Operandenbereiche haben keine Bitadresse.



Der Operandenbereich Daten D kann wortweise und bitweise adressiert werden.



Der Parameter des Operandenbereichs Konstante K stellt die Zahl dar, die als digitaler Wert weiterverarbeitet werden soll. Die Parameter der Operandenbereiche OB, PB, FB und DB geben die Nummer der Operanden in diesem Bereich an.



## 9.3 Grundoperationen

Die Grundoperationen sind in Programm-, Schritt-, Organisations-, und Funktionsbausteinen programmierbar. Sie können in Programm-, Schritt- und Organisationsbausteinen in den drei Darstellungsarten (KOP, FUP und AWL) ein- und ausgegeben werden.

Ausnahmen:

1. Lade-, Transfer- und Codeoperationen. Sie können nur indirekt und begrenzt im Zusammenhang mit Zeit- und Zähloperationen graphisch programmiert werden.
2. Rechenoperationen und der Stop-Befehl (STP) können nur in Anweisungsliste programmiert werden.

### 9.3.1 Verknüpfungsoperationen, binär

Operation	Parameter	Funktion
) U( O( O		Klammer zu UND-Verknüpfung von Klammersausdrücken ODER-Verknüpfung von Klammersausdrücken ODER-Verknüpfung von UND-Funktionen
U    □ □ O    □ □ ↑ ↑ E A M D N E N A N M N D T N T Z N Z	0.0 bis 127.7 0.0 bis 127.7 0.0 bis 255.7 0.0 bis 255.15 0.0 bis 127.7 0.0 bis 127.7 0.0 bis 255.7 0.0 bis 255.15 1 bis 31 1 bis 31 1 bis 31 1 bis 31	UND-Verknüpfung mit ODER-Verknüpfung mit Abfrage eines Eingangs auf Signalzustand "1" Abfrage eines Ausgangs auf Signalzustand "1" Abfrage eines Merkers auf Signalzustand "1" Abfrage eines Datums auf Signalzustand "1" Abfrage eines Eingangs auf Signalzustand "0" Abfrage eines Ausgangs auf Signalzustand "0" Abfrage eines Merkers auf Signalzustand "0" Abfrage eines Datums auf Signalzustand "0" Abfrage einer Zeit (Time) auf Signalzustand "1" Abfrage einer Zeit (Time) auf Signalzustand "0" Abfrage eines Zählers auf Inhalt>0 Abfrage eines Zählers auf Inhalt=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, O=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.

### UND-Verknüpfung

Vorlage	STEP 5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>                 U E 1.1                 U E 1.3                 U E 1.7                 = A 3.5             </pre>		

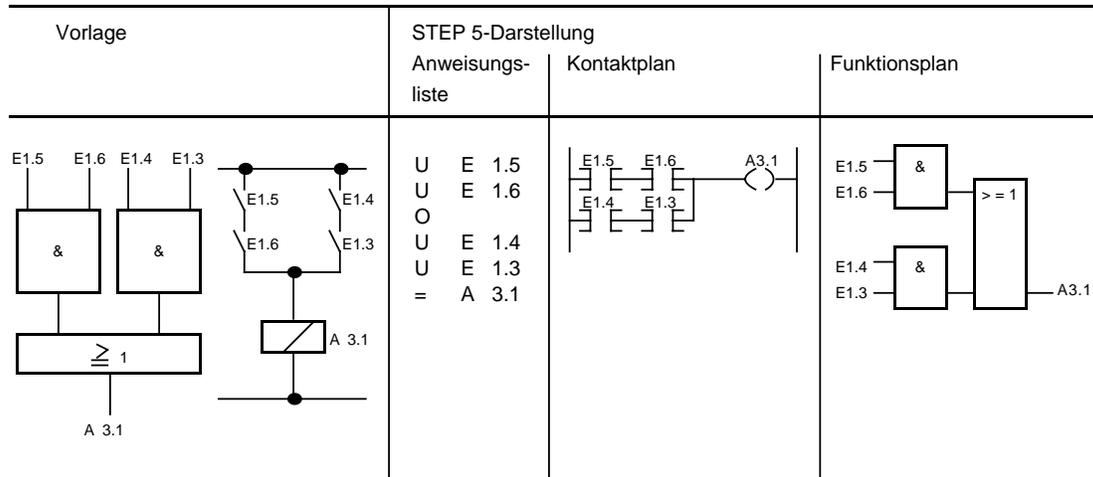
Am Ausgang A 3.5 erscheint Signalzustand "1", wenn alle Eingänge gleichzeitig den Signalzustand "1" aufweisen. Am Ausgang A 3.5 erscheint Signalzustand "0", wenn mindestens einer der Eingänge den Signalzustand "0" aufweist. Die Anzahl der Abfragen und die Reihenfolge der Programmierung ist beliebig.

### ODER-Verknüpfung

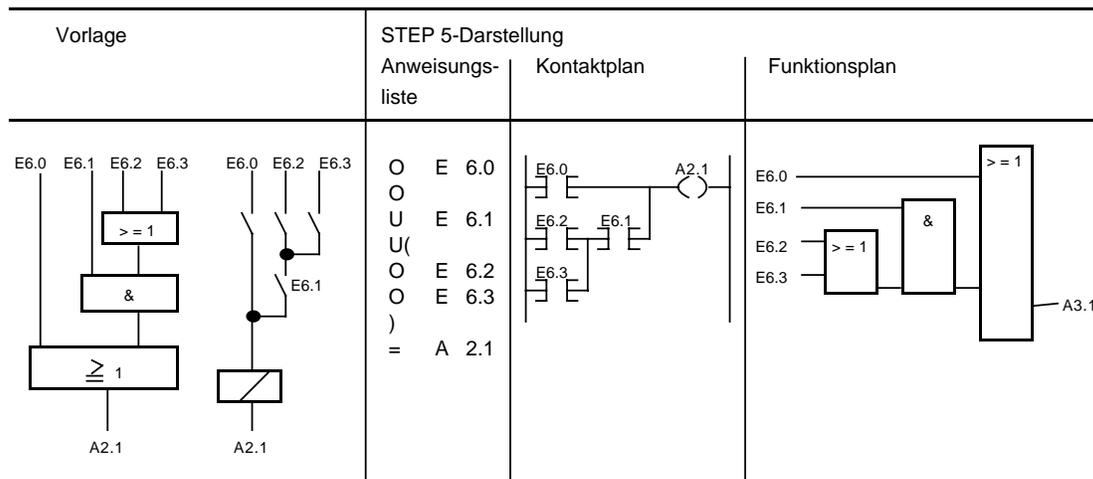
Vorlage	STEP 5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>                 O E 1.2                 O E 1.7                 O E 1.5                 = A 3.2             </pre>		

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**

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**

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.

### ODER-vor-UND-Verknüpfung

Vorlage	STEP 5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U( O  E 1.4 O  E 1.5 ) U( O  E 2.0 O  E 2.1 ) =  A 3.0                     </pre>		

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		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U  E 1.5 UN E 1.6 =  A 3.0                     </pre>		

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.

### 9.3.2 Speicheroperationen

Operation	Parameter	Funktion
S <input type="checkbox"/>		Setzen
R <input type="checkbox"/>		Rücksetzen
= <input type="checkbox"/>		Zuweisen
<input type="checkbox"/>		
<input type="checkbox"/>	0.0 bis 127.7	eines Einganges
<input type="checkbox"/>	0.0 bis 127.7	eines Ausganges
<input type="checkbox"/>	0.0 bis 255.7	eines Merkers
<input type="checkbox"/>	0.0 bis 255.15	eines Datenwortes

**RS-Speicherglied für speichernde Signalausgabe**

Vorlage	STEP 5-Darstellung															
	Anweisungs- liste	Kontaktplan	Funktionsplan													
		<table border="1"> <tr><td>U</td><td>E</td><td>2.7</td></tr> <tr><td>S</td><td>A</td><td>3.5</td></tr> <tr><td>U</td><td>E</td><td>1.4</td></tr> <tr><td>R</td><td>A</td><td>3.5</td></tr> </table>	U	E	2.7	S	A	3.5	U	E	1.4	R	A	3.5		
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 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															
	Anweisungs- liste	Kontaktplan	Funktionsplan													
		<table border="1"> <tr><td>U</td><td>E</td><td>2.6</td></tr> <tr><td>S</td><td>M</td><td>1.7</td></tr> <tr><td>U</td><td>E</td><td>1.3</td></tr> <tr><td>R</td><td>M</td><td>1.7</td></tr> </table>	U	E	2.6	S	M	1.7	U	E	1.3	R	M	1.7		
U	E	2.6														
S	M	1.7														
U	E	1.3														
R	M	1.7														

Signalzustand "1" am Eingang E 2.6 bewirkt das Setzen des Speicherglieds. 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 Speicherglieds. 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.

### Nachbildung eines Wischrelais

Vorlage	STEP 5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>                     U E 1.7                     UN M 4.0                     = M 2.0                     U M 2.0                     S M 4.0                     UN E 1.7                     R M 4.0                 </pre>		

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		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>                     U E 1.0                     UN M 1.0                     = M 1.1                     U M 1.1                     S M 1.0                     UN E 1.0                     R M 1.0                     U M 1.1                     U A 3.0                     = M 2.0                     U M 1.1                     UN A 3.0                     UN M 2.0                     S A 3.0                     U M 2.0                     R A 3.0                 </pre>		

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.

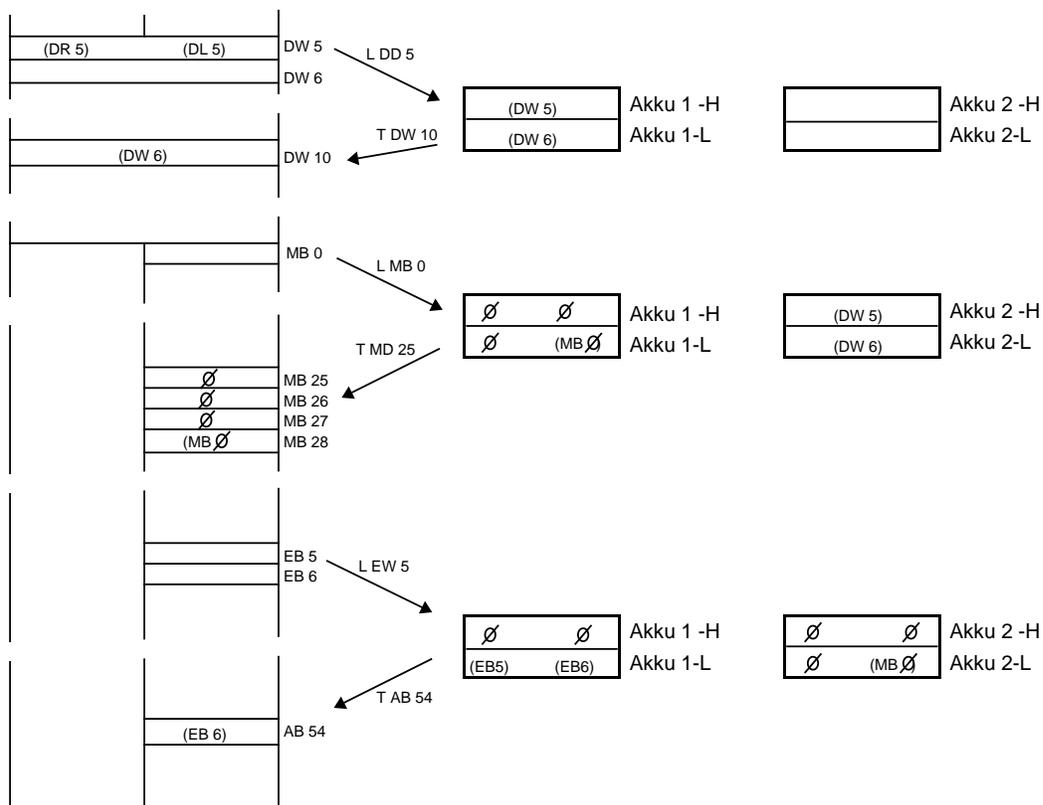
### 9.3.3 Lade- und Transferoperationen

Operation	Parameter	Funktionen
<b>L</b> <input type="checkbox"/> <input type="checkbox"/> <b>T</b> <input type="checkbox"/> <input type="checkbox"/> 		Laden Transferieren
<b>E B</b>	0 bis 127	eines Eingabebytes vom PAE <sup>2)</sup>
<b>E W</b>	0 bis 126	eines Eingabewortes vom PAE
<b>E D</b>	0 bis 124	eines Eingabedoppelwortes vom PAE
<b>A B</b>	0 bis 127	eines Ausgabebytes vom PAA <sup>3)</sup>
<b>A W</b>	0 bis 126	eines Ausgangswortes vom PAA
<b>A D</b>	0 bis 124	eines Ausgangs-Doppelwortes vom PAA
<b>M B</b>	0 bis 225	eines Merkerbytes
<b>M W</b>	0 bis 254	eines Merkerwortes
<b>M D</b>	0 bis 252	eines Merkerdoppelwortes
<b>D R</b>	0 bis 255	eines Datums (rechtes Byte)
<b>D L</b>	0 bis 255	eines Datums (linkes Byte)
<b>D W</b>	0 bis 255	eines Datenwortes
<b>D D</b>	0 bis 254	eines Datum-Doppelwortes
<b>T</b> <sup>2)</sup>	0 bis 31	eines Zeitwertes (dual)
<b>Z</b> <sup>2)</sup>	0 bis 31	eines Zählwertes (dual)
<b>P B</b>	0 bis 127	eines Peripheriebytes der Digital-Eingaben bzw. -Ausgaben
<b>P W</b>	0 bis 126	eines Peripheriewortes der Digital-Eingaben bzw. -Ausgaben
<b>K M</b> <sup>1)</sup>	16-Bit-Muster	einer Konstanten als Bitmuster
<b>K H</b> <sup>1)</sup>	0 bis FFFFH	einer Konstanten im Hexa-Code
<b>K F</b>	0 bis $+(2^{16}-1)$	einer Konstanten als Festpunktzahl
<b>K Y</b> <sup>1)</sup>	0 bis 255 für jedes Byte	einer Konstanten, 2 Byte
<b>K B</b> <sup>1)</sup>	0 bis 255	einer Konstanten, 1 Byte
<b>K C</b> <sup>1)</sup>	2 alphanumerische Zeichen	einer Konstanten, 2 ASCII-Zeichen
<b>K G</b> <sup>1)</sup>	$\pm 0,1469368 \cdot 10^{-38}$ bis $\pm 0,1701412 \cdot 10^{+39}$	einer Konstanten als Gleitpunktzahl
<b>K T</b>	0.1 bis 999.3	eines Zeitwertes (Konstante)
<b>K Z</b> <sup>1)</sup>	0 bis 999	eines Zählwertes (Konstante)
1) nicht bei Transferieren 2) PAE Prozeßabbild für Eingänge 3) PAA Prozeßabbild für Ausgänge		

Die Lade- und Transferoperationen sind absolute Befehle, das heißt, sie werden unabhängig vom Verknüpfungsergebnis ausgeführt. Ladeoperationen von Daten aus einem DB überprüfen nicht, ob die Länge des DB ausreicht.

Die Lade- und Transferoperationen können nur indirekt im Zusammenhang mit Zeit- oder Zähloperationen graphisch programmiert werden, sonst nur in Anweisungsliste.

**Beispiel: Lade- und Transferfunktion**



**Laden und Transferieren eines Zeitwertes (auch Zeit- und Zähloperation Seite ...).**

Vorlage	STEP 5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre>                     U   E 5.0                     L   EW 22                     SI  T 10                     T   AW 64                 </pre>		

Der Ausgang DU vom Zeitglied wurde, bei der graphischen Eingabe, mit AW 64 belegt. Das Programmiergerät hinterlegt daraufhin selbsttätig den entsprechenden Lade- und Transferbefehl im Anwenderprogramm. So wird der Inhalt der mit T10 adressierten Speicherzelle in den Akkumulator (AKKU 1) geladen.

Anschließend wird der Akkumulatorinhalt (AKKU 1) in das mit AW 64 adressierte Prozessbild transferiert.

### 9.3.4 Zeit- und Zähloperationen

Standardmäßig sind die Zeiten T0 ... T15 und die Zähler Z0 ... Z31 freigegeben. Bei der Programmierung eines größeren Parameters wird vom Interpreter der Fehler "Nicht zulässige MC5-Zeit/Zähler" gemeldet.

Über PLC-MD 6 kann die Zahl der Zeiten bis maximal auf einen Wert von 31 vergrößert werden. Es muß zu diesem Zweck über das erwähnte MD die Nummer der letzten, "aktiven", daß heißt vom Betriebssystem berücksichtigten MC5-Zeit, angegeben werden. Die Zahl ist nach Neustart wirksam.

Die Art der Zeitzellenaktualisierung durch das Betriebssystem erlaubt keine sinnvolle Programmierung des 0.01 s-Zeitrasters (z. B. L KT 10.0).

Da die Zeitzellenaktualisierung außerdem von der "Servoabtastrate", daß heißt von MD 155 abhängig ist, ergeben sich Ungenauigkeiten in den Zeiten, die in der nachfolgenden Tabelle dargestellt sind.

Programmiertes Zeitraster				
MD 155	100 ms		1 s	10 s
	SW1	SW2		
0	-20%	0%	-5%	0%
1	-10%	-1%	-1%	<1%
2	0%	0%	0%	0%
3	+10%	-2%	-1%	< 1 %

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

Sinnvoll sind folgende Ladeoperationen: <sup>1)</sup>

für Zeiten: L KT, L EW, L AW, L MW, L DW

für Zähler: L KZ, L EW, L AW, L MW, L DW

1) Zeit- oder Zähloperationen verändern den Inhalt von Akku 1 nicht.

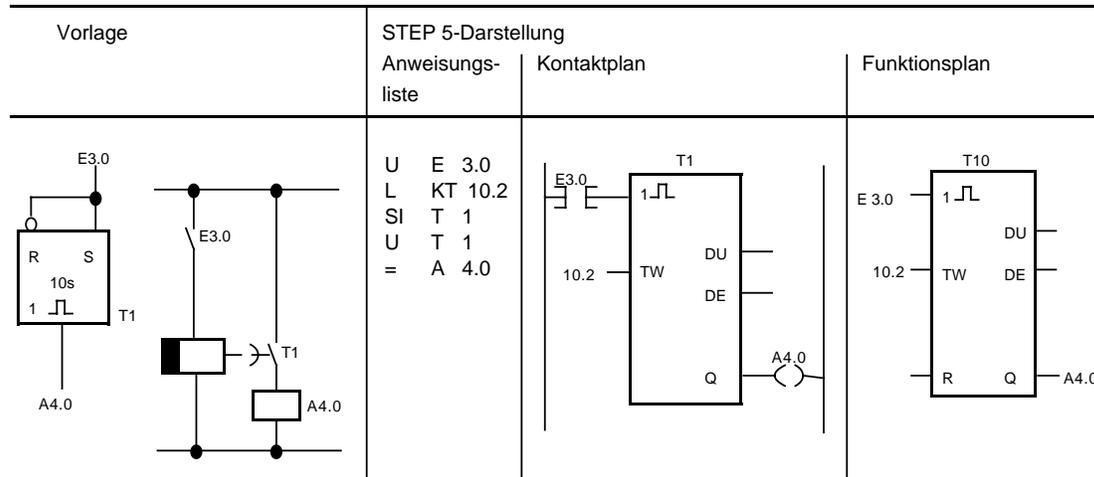
Operation	Parameter	Funktion
SI T	0 bis 31	Starten einer Zeit als Impuls
SV T	0 bis 31	Starten einer Zeit als verlängerter Impuls
SE T	0 bis 31	Starten einer Zeit als Einschaltverzögerung
SA T	0 bis 31	Starten einer Zeit als speichernde
SA T	0 bis 31	Einschaltverzögerung
R T	0 bis 31	Starten einer Zeit als Ausschaltverzögerung
S Z	0 bis 31	Rücksetzen einer Zeit
R Z	0 bis 31	Setzen eines Zählers
ZV Z	0 bis 31	Rücksetzen eines Zählers
ZR Z	0 bis 31	Vorwärtszählen eines Zählers Rückwärtszählen eines Zählers

**Wichtiger Hinweis:**

Da die Zeit- und Zählerbefehle vom COP/ACOP unterstützt werden, und dieser keine Parameterprüfung kennt, können u. U. die Steuerbits von nicht in diesem Befehl programmierten Zeit- bzw. Zählerzellen beeinflusst werden.

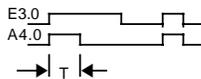
**Beispiel:**

Durch ein B MW-Befehl ist der Befehl SE T 0 (substituiert: SE T 33) abzuarbeiten.  
 Bei VKE=0: die Bits ZWG, ZKS, FMS bei T1 werden gelöscht.

**Impuls**

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.

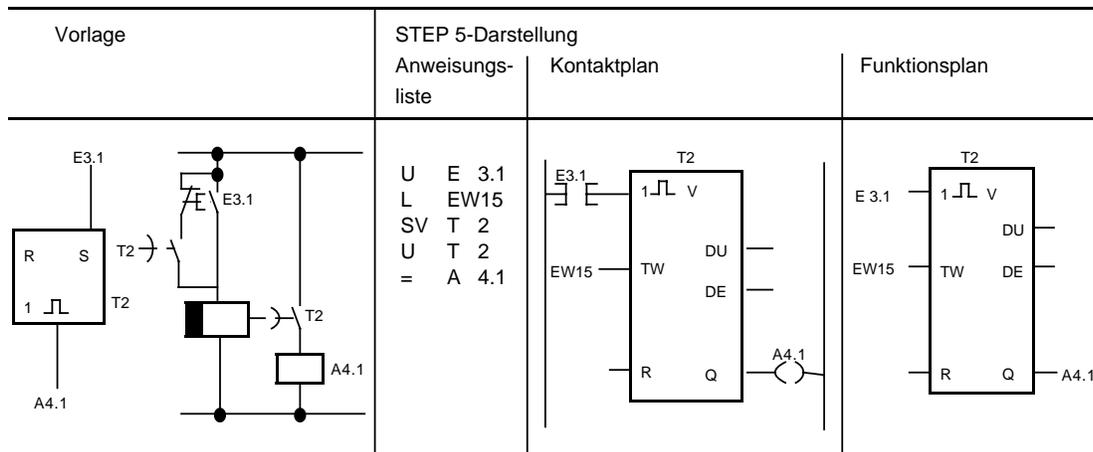
DU und DE sind digitale Ausgänge der Zeitzelle. Am Ausgang DU steht der Zeitwert dual-codiert, am Ausgang DE BCD-codiert, mit Zeitraster an.

**KT10.2:**

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

- 1= 0,1 s
- 2= 1 s
- 3= 10 s

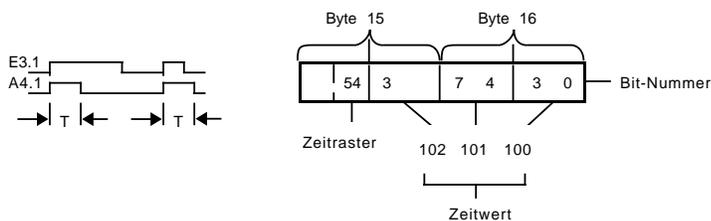
### Verlängerter Impuls



Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied gestartet.

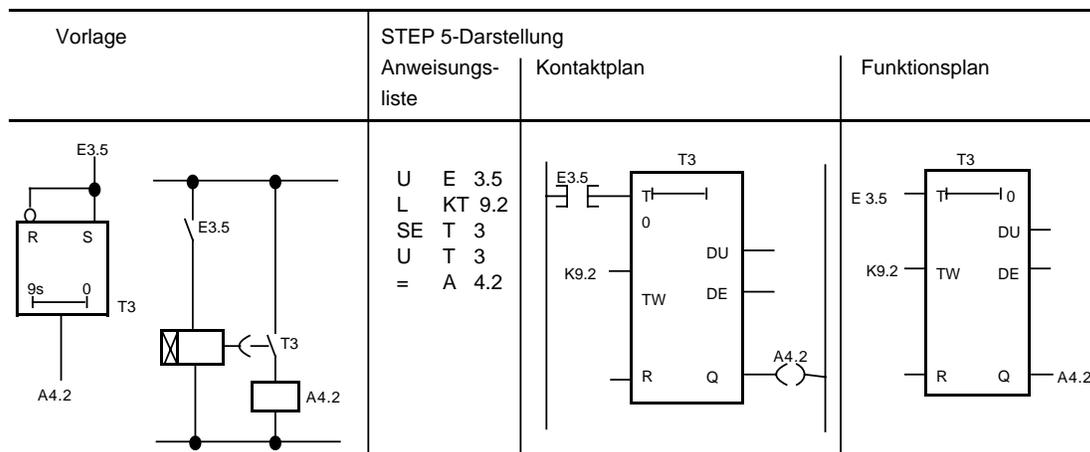
Bei Verknüpfungsergebnis "0" bleibt das Zeitglied unbeeinflusst.

Bei Abfragen U T oder O T liefern Signalzustand "1", solange die Zeit läuft.



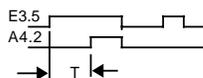
EW 15:

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

**Einschaltverzögerung**

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:

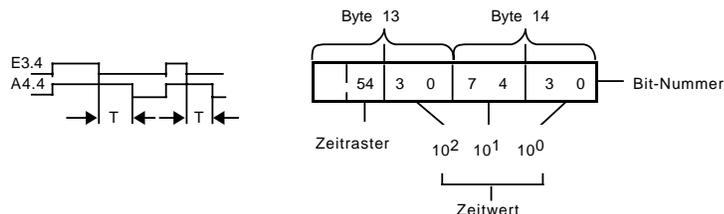
- 1= 0,1s
- 2= 1 s
- 3= 10 s

### Ausschaltverzögerung

Vorlage	STEP 5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	<pre>                 UN E 3.4                 L  MW13                 SA T 5                 U  T 5                 =  A 4.4             </pre>		

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).

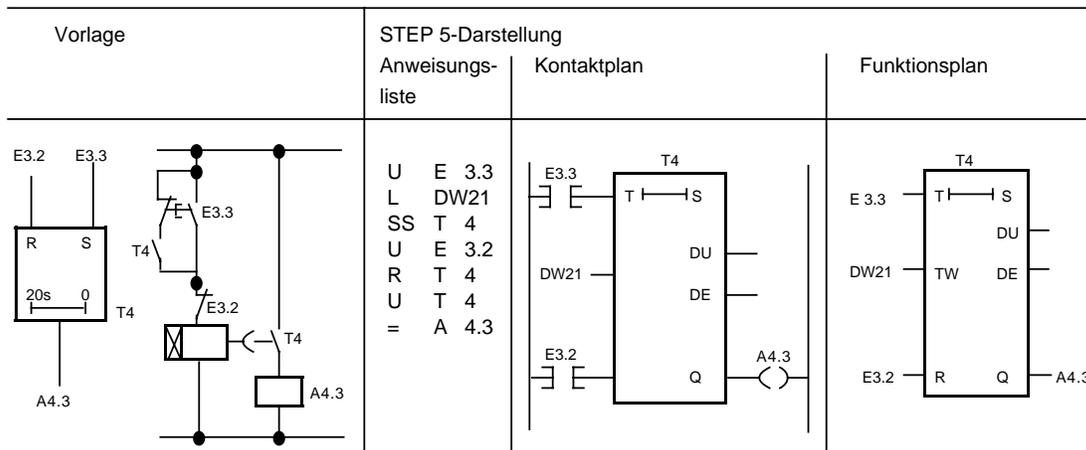
Die Abfragen U T bzw. O T liefern den Signalzustand "1", wenn die Zeit läuft oder das Verknüpfungsergebnis am Eingang noch ansteht.



MW 13:

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

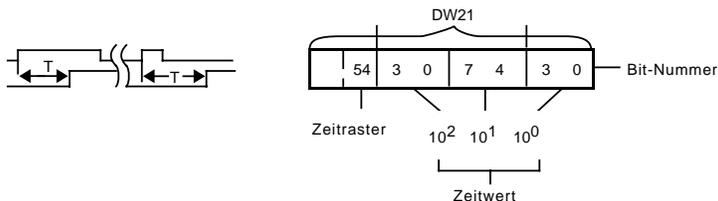
### Speichernde Einschaltverzögerung



Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied gestartet. Bei Verknüpfungsergebnis "0" wird das Zeitglied unbeeinflusst.

Die Abfragen U T bzw. O T liefern Signalzustand "1", wenn die Zeit abgelaufen ist.

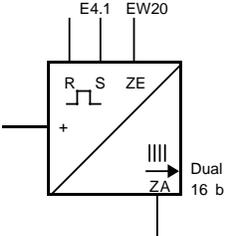
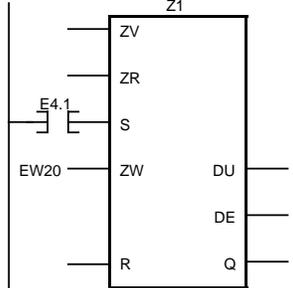
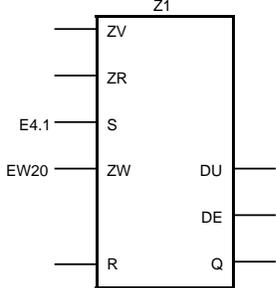
Der Signalzustand wird erst dann "0", wenn das Zeitglied mit der Funktion R T zurückgesetzt wurde.



DW 21:

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

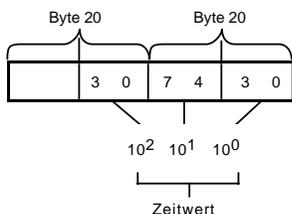
**Zähler setzen**

Vorlage	STEP 5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>                     U  E 4.1                     L  EW20                     S  Z 1                 </pre>		

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).

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

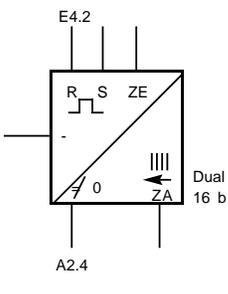
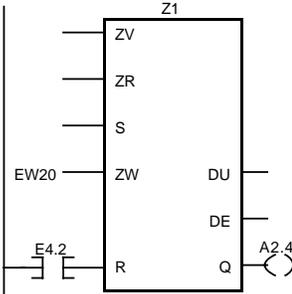
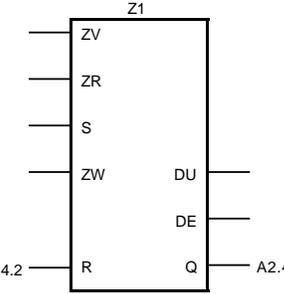
Der für die Flankenauswertung des Setzeingangs erforderliche Merker ist im Zählwort mitgeführt.



EW 20:

Zähler setzen mit dem in BCD-Code vorliegenden Wert der Operanden E, A, M oder D (im Beispiel ist das Eingangswort 20).

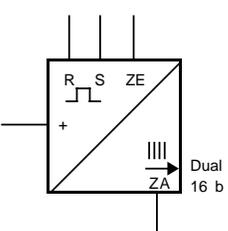
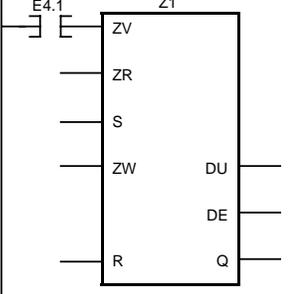
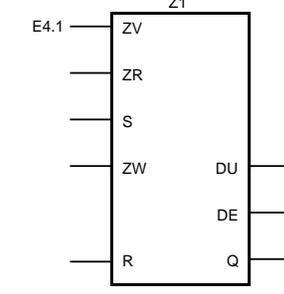
**Zähler zurücksetzen**

Vorlage	STEP 5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 4.2 R Z 1 U Z 1 = A 2.4 </pre>		

Bei Verknüpfungsergebnis "1" wird der Zähler auf Null gesetzt (rückgesetzt).

Bei Verknüpfungsergebnis "0" bleibt der Zähler unbeeinflusst.

**Vorwärts zählen**

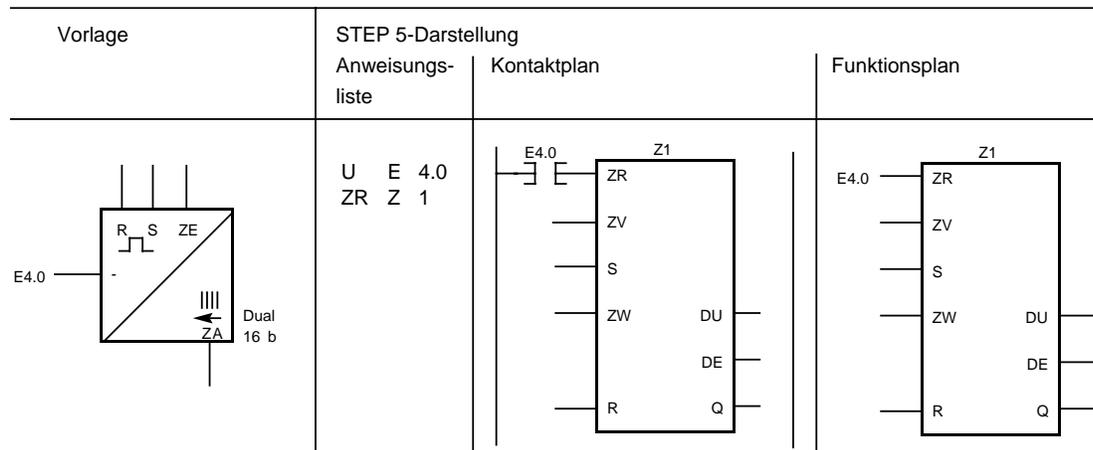
Vorlage	STEP 5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 4.1 ZV Z 1 </pre>		

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 Flankenbewertung der Zählwege 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



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 Zählgänge 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.

### 9.3.5 Vergleichsoperationen

Die Vergleichsoperationen vergleichen den Inhalt des Akkumulators 1 mit dem Inhalt des Akkumulators 2.

Deshalb müssen zuerst die zu vergleichenden Werte in den Akkumulatoren, z. B. mit Ladeoperationen, hinterlegt werden.

Während des Vergleichs bleiben die Akkumulatorinhalte unverändert.

Operation	Parameter	Funktion
! = <input type="checkbox"/>		Vergleich auf gleich
>< <input type="checkbox"/>		Vergleich auf ungleich
> <input type="checkbox"/>		Vergleich auf größer
> = <input type="checkbox"/>		Vergleich auf größergleich
< <input type="checkbox"/>		Vergleich auf kleiner
< = <input type="checkbox"/>		Vergleich auf kleingleich
<input type="checkbox"/>	F	zweier Festpunkt-Zahlen
<input type="checkbox"/>	G	zweier Gleitkomma-Zahlen
<input type="checkbox"/>	D	zweier Festpunkt-Doppelwort-Zahlen

### Vergleich auf gleich

Vorlage	STEP 5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> L EB19 L EB20 ! = F = A 3.0                     </pre>		

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

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

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

Akku 2-H und Akku 1-H bleiben beim Festpunktvergleich an der Operation unbeteiligt. Beim Vergleich wird die Zahlendarstellung der Operanden (hier Festpunktrechnung) berücksichtigt.

0	EB19	Akku 2-L
0	EB19	Akku 1-L

### Vergleich auf ungleich

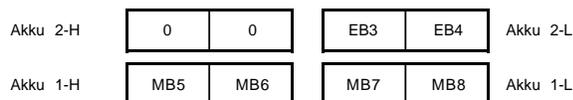
Vorlage	STEP 5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan

### Vergleich auf größer

Vorlage	STEP 5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>L EW3 L MD5 &gt; D = A 3.2</pre>		

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

VKE="1": Vergleich ist erfüllt, Akku 2>Akku 1  
 VKE="0": Vergleich ist nicht erfüllt, Akku 2≠Akku 1



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		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>L DD2 L EB7 &lt; D = A 3.4</pre>		

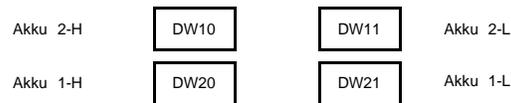
### Vergleich auf größer-gleich

Vorlage	STEP 5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre>L DD10 L DD20 =&gt;G =</pre>		

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

VKE="1": Vergleich ist erfüllt, Akku 2  $\hat{=}$  Akku 1

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



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

### Vergleich auf kleiner-gleichliste

Vorlage	STEP 5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre>L DD2 L EB7 &lt; G = A 3.4</pre>		

### 9.3.6 Bausteinaufrufe

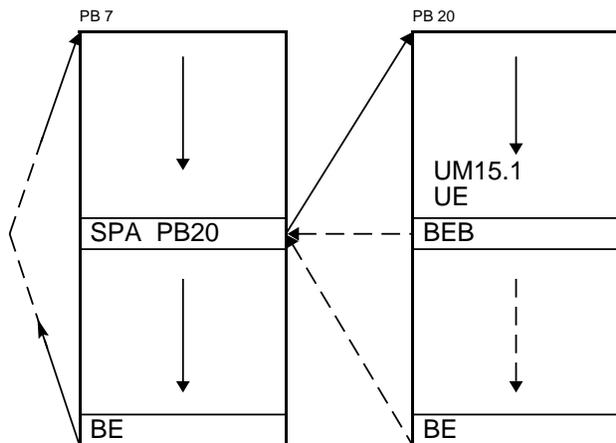
Operation	Parameter	Funktionen
<b>SPA</b> □□ <b>SPB</b> □□ ↑↑ <b>P B</b> <b>F B</b> <b>S B</b>	1 bis 255	Sprung unbedingt Sprung bedingt (abhängig vom VKE)  zu einem Programmbaustein zu einem Funktionsbaustein (Typ FB) zu einem Schrittbaustein
<b>A D B</b>	1 bis 255	Aufruf eines Datenbausteins
<b>BE</b> <b>BEB</b> <b>BEA</b>		Bausteinende Bausteinende bedingt (abhängig vom VKE) Bausteinende absolut

Der Befehl A DB (Aufruf eines Datenbausteins) wird unter Punkt "Aufruf von Datenbausteinen" (Seite 3–1) erläutert.

Die Befehle BE (Bausteinende) und BEB (Bausteinende bedingt) bewirken den Rücksprung zum vorverschachtelten Baustein.

Der Befehl BE muß obligatorisch am Ende eines jeden Bausteins (außer DB) programmiert werden.

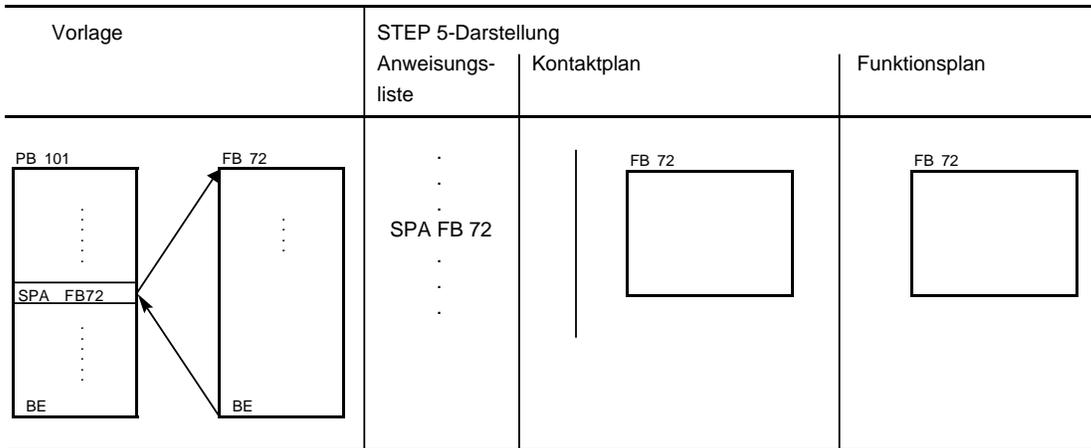
**Beispiel:**



Bei Verknüpfungsergebnis=1 erfolgt der Rücksprung zum PB 7 bereits durch die Bearbeitung des BEB Befehls.

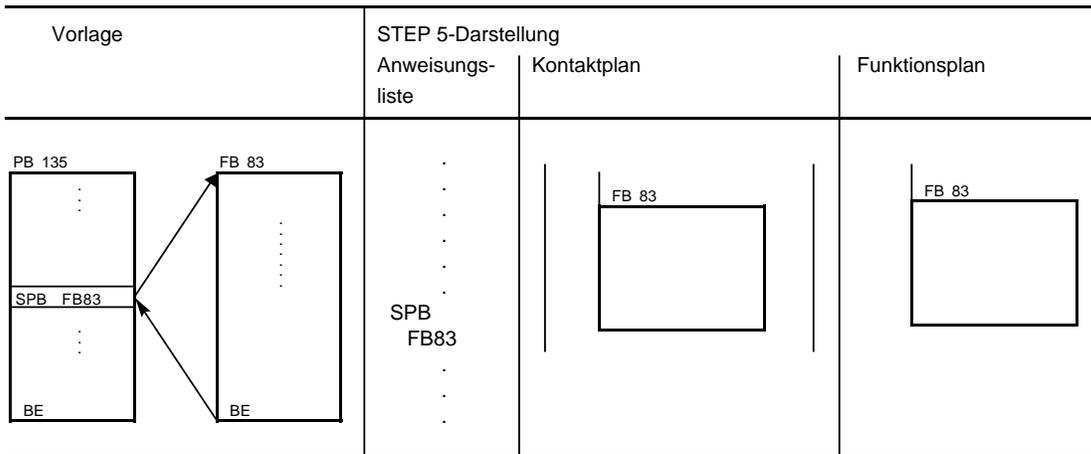
Bei Verknüpfungsergebnis=0 wird der PB 20 bis zum Befehl BE weiter bearbeitet, der dann den Rücksprung zum PB 7 nach vollständiger Bearbeitung des PB 20 bewirkt.

**Absoluter Aufruf eines Funktionsbausteins**



Im gewünschten Programmpunkt wird der absolute Funktionsbausteinaufruf eingegeben. Bei den graphischen Darstellungsarten KOP und FUP wird der aufgerufene Baustein (FB) als ein Kasten dargestellt.

**Bedingter Aufruf eines Programmbausteins**

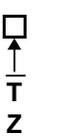


(Vorverknüpfung nur als Beispiel)

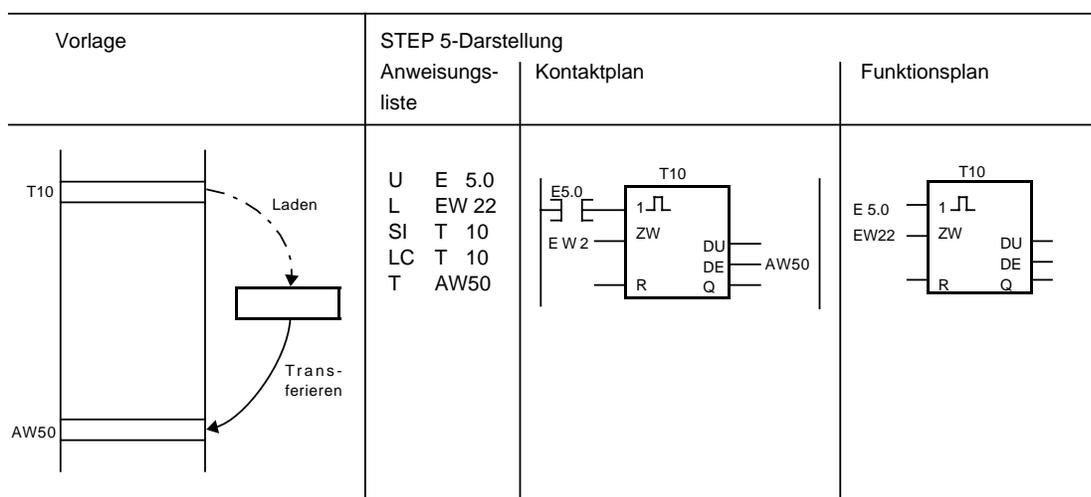
Im gewünschten Programmpunkt wird der bedingte Bausteinaufruf nach der entsprechenden Verknüpfung eingegeben. Ist das Verknüpfungsergebnis=1, so erfolgt der Sprung in den angegebenen Baustein. Bei nicht erfüllter Bedingung wird der Sprung nicht durchgeführt. Bei den graphischen Darstellungsarten KOP und FUP wird der aufgerufene Baustein als ein Kasten dargestellt.

### 9.3.7 Codeoperationen

Die Codeoperationen ermöglichen das Laden eines in dualer Form vorliegenden Zeit- oder Zählwertes "codiert" in den Akkumulator, wobei der entsprechende Wert in BCD-codierter Form zur weiteren Verarbeitung zur Verfügung steht.

Operation	Parameter	Funktion
<b>LC</b> 	0 bis 31 0 bis 31	Lade codiert von Zeitwerten von Zählwerten

#### Laden eines Zeitwertes (codiert)



Der Inhalt der mit T10 adressierten Speicherzelle wird in den Akkumulator codiert geladen. Die anschließende Transferoperation transferiert den Inhalt vom Akkumulator in die durch AW 50 adressierte Speicherzelle der Prozeßabbilder. Bei den graphischen Darstellungsarten KOP und FUP kann eine Codieroperation nur indirekt durch die Belegung des Ausgangs DE einer Zeit- bzw. Zählerzelle erfolgen. Bei der Darstellungsart AWL kann dieser Befehl jedoch isoliert werden.

### 9.3.8 Arithmetische Operationen

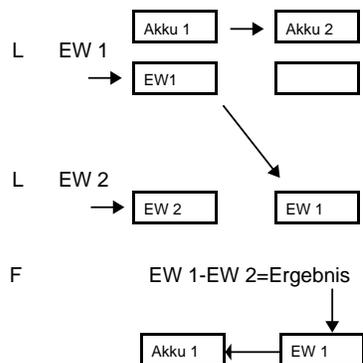
Die arithmetischen Operationen können nur in der Anweisungsliste dargestellt werden.

Sie verarbeiten die Inhalte von Akkumulator 1 und 2, wobei z. B. entsprechende Ladeoperationen erforderlich sind.

Operation	Parameter	Funktion
+ <input type="checkbox"/>		Addition
- <input type="checkbox"/>		Subtraktion
x <input type="checkbox"/>		Multiplikation
: <input type="checkbox"/>		Division
↑ F		zweier Festpunktzahlen
G		zweier Gleitpunktzahlen

Durch zwei Ladeoperationen können die beiden Akkus 1 und 2 entsprechend den Operanden der Ladeoperationen geladen werden. Anschließend lassen sich mit den Inhalten beider Akkus arithmetische Operationen ausführen.

#### Beispiel:



Die anschließende Transferoperation transportiert das Ergebnis, das im Akku 1 abgelegt ist, zu dem bei der Transferoperation abgegebenen Operanden.

Erfolgt beim Rechnen mit Festpunktzahlen ein Überlauf (0 V=1), wird der Akku 1-H gelöscht.

Bei Multiplikation und Division von Gleitpunktzahlen wird nur mit 16 Bit Mantisse gerechnet. Daraus ergibt sich eine kleinere Genauigkeit:

Multiplikation: mindestens 12 Bit der Mantisse sind exakt

Division: mindestens 11 Bit der Mantisse sind exakt.

Bei der Subtraktion von Gleitpunktzahlen kann das Bit 2-24 falsch sein, wenn die Differenz der beiden Exponenten größer als 24 ist.

### 9.3.9 Sonstige Operationen

Die folgenden Operationen sind nur in der Anweisungsliste darstellbar.

Operation	Parameter	Funktion
<b>STP</b>		Stop
<b>NOP 0</b>		Nulloperation (alle Bits gelöscht)
<b>NOP 1</b>		Nulloperation (alle Bits gesetzt)
<b>BLD</b>	0 bis 255	Bildschirmbefehl

Der STOP-Befehl wird dann eingesetzt, wenn z. B. bei bestimmten kritischen Zuständen der Anlage oder bei Auftreten eines Gerätefehlers der Übergang des Automatisierungsgerätes in den Stopzustand gewünscht wird.

Die Nulloperationen dienen z. B. zur Freihaltung oder Überschreibung von Speicherplätzen

Der Bildschirmbefehl bestimmt die Einteilung von Programmteilen in Segmente innerhalb eines Bausteins. Er wird selbstständig vom Programmiergerät in das Programm hinterlegt und von der Anpaßsteuerung wie eine Nulloperation behandelt.

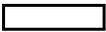
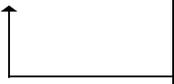
### 9.4 Ergänzende Operationen (nur FB)

Funktionsbausteine können gegenüber den Programmbausteinen 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. Zusätzlich sind die Kombinationsmöglichkeiten der Substitutionsbefehle mit den Aktualoperanden angegeben.

### 9.4.1 Verknüpfungsoperationen, binär

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

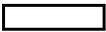
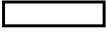
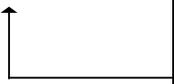
#### Beispiel:

```

: U    =EIN
: UN   =STOP
: UN   =END
: O    =BETR
:=     =LAUF

```

### 9.4.2 Setzoperationen

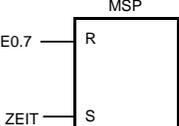
Operation	Beschreibung
<b>S =</b> 	Setzen (binär) eines Formaloperanden.
<b>RB=</b> 	Rücksetzen (binär) eines Formaloperanden.
<b>==</b>  	Zuweisen des Verknüpfungsergebnisses an einen Formaloperanden.  <b>Formaloperand einsetzen</b> Als Aktualoperand sind binär adressierte Eingänge, Ausgänge und Merker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

#### Beispiel:

```

U E 0.7
RB =MSP
U   =ZEIT
S   =MSP

```



### 9.4.3 Zeit- und Zähloperationen

Operation	Beschreibung
<b>RD=</b> <input type="text"/>	Rücksetzen (digital) eines Formaloperanden (Parameterart: T, Z).
<b>SI =</b> <input type="text"/>	Starten einer als Formaloperand vorgegebenen Zeit mit dem Akku hinterlegten Wert als Impuls (Parameterart: T).
<b>SE=</b> <input type="text"/>	Starten einer als Formaloperand vorgegebenen Zeit mit dem Akku hinterlegten Wert als Einschaltverzögerung (Parameterart: T).
<b>SVZ=</b> <input type="text"/>	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).
<b>SSV=</b> <input type="text"/>	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).
<b>SAR=</b> <input type="text"/>	Starten einer als Formaloperand vorgegebenen Zeit mit dem im Akku hinterlegten Wert als Ausschaltverzögerung bzw. Rückwärtszählen eines als Formaloperand vorgegebenen Zählers (Parameterart: T, Z).  <b>Formaloperand einsetzen</b> (s. S. 8). Als Aktualoperand sind Zeiten und Zähler zugelassen; Ausnahme: Bei SI und SE nur Zeiten. Der Zeit- bzw. Zählwert kann wie bei den Grundoperationen oder als Formaloperand wie folgt vorgeben 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 Datum (Parameterart: D; Parametertyp: KT, KZ).

**Beispiele:**

Funktionsbausteinaufruf	Programm im Funktionsbaustein	ausgeführtes Programm
: SPA FB203 NAME : BEISPIEL ANNA : E 10.3 BERT : T 17 ANS : 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 KZ100 : 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 := = MAUS	: U E 10.4 : L EW20 : SV T 18 := M 100.7

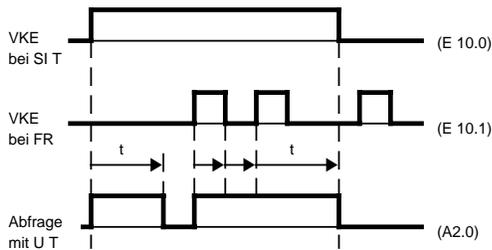
### 9.4.4 Freigabeoperationen für Zeit- und Zähloperationen

Operation	Beschreibung
<b>FR T 0 bis 31</b>	<b>Freigabe einer Zeit für Neustart</b> Die Operation wird nur bei steigender Flanke des Verknüpfungsergebnisses ausgeführt. Sie bewirkt einen Neustart der Zeit, wenn bei der Startoperation Verknüpfungsergebnis "1" anliegt.
<b>FR Z0 bis 31</b>	<b>Freigabe eines Zählers</b> 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.
<b>FR=</b> <input type="checkbox"/>	<b>Freigabe eines Formaloperanden für Neustart,</b> Parameterart (T, Z).

**Beispiel:**

```

: U      E 10.0
: L      KT 500.1
: SI     T 10
: U      E 10.1
: FR     T 10
: U      T 10
:=      A 2.0
    
```



## 9.4.5 Bittestoperationen (nur FB)

Operation	Parameter	Funktion
<b>P</b> <input type="checkbox"/>		Prüfe Bit auf Signalzustand "1"
<b>PN</b> <input type="checkbox"/>		Prüfe Bit auf Signalzustand "0"
<b>SU</b> <input type="checkbox"/>		Setze Bit unbedingt
<b>RU</b> <input type="checkbox"/>		Rücksetze Bit unbedingt
<b>E</b>	0.0 bis 127.7	eines Eingangs
<b>A</b>	0.0 bis 127.7	eines Ausgangs
<b>M</b>	0.0 bis 255.7	eines Merkers
<b>Z</b>	0.0 bis 31.15	eines Zählwortes
<b>T</b>	0.0 bis 31.15	eines Zeitwortes
<b>D</b>	0.0 bis 255.15	eines Datenwortes

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
<b>P</b>	0	0
	1	1
<b>PN</b>	0	1
	1	0

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

### 1. Beispiel

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

```
: A    DB 200
: P    D 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.

### 2. Beispiel

Das Bit 3 von DW 55 soll gesetzt und das Bit 9 von DW 103 soll rückgesetzt werden.

```
: SU   D 55.3
: RU   D 103.9
```

### 9.4.6 Lade- und Transferoperationen

Operation	Beschreibung
<b>L =</b> <input type="text"/>	<b>Laden eines Formaloperanden</b> Der Wert des als Formaloperanden vorgegebenen Operanden wird in den Akku geladen (Parameterart: E, A; Parametertyp: BY, W, D).
<b>LC=</b> <input type="text"/>	<b>Laden codiert eines Formaloperanden</b> Der Wert der als Formaloperand vorgegebenen Zeit- oder Zählzelle wird BCD-codiert in den Akku geladen (Parameterart: T, Z).
<b>LW=</b> <input type="text"/>	<b>Laden des Bitmusters eines Formaloperanden</b> Das Bitmuster des Formaloperanden wird in den Akku geladen (Parameterart: D; Parametertyp: KF, KH, KM, KY, KC, KT, KZ).
<b>LD=</b> <input type="text"/>	<b>Laden des Bitmusters eines Formaloperanden</b> Das Bitmuster des Formaloperanden wird in den Akku geladen (Parameterart: D; Parametertyp: KG).
<b>T =</b> <input type="text"/> 	<b>Transferieren zu einem Formaloperanden</b> Der Akkumulatorinhalt wird zu dem als Formaloperand vorgegebenen Operanden transferiert (Parameterart: E, A; Parametertyp: BY, W, D).  <b>Formaloperanden einsetzen.</b> Als Aktualoperand sind die den Grundoperationen entsprechenden Operanden zugelassen. Bei LW ist ein Datum in Form eines Binärmusters, Hexamusters, 2 byteweise Betragzahlen, Zeichen, Festpunktzahl, Zeitwerte und Zählwerte zugelassen. Bei LD ist eine Gleitkommazahl als Datum zugelassen.

**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

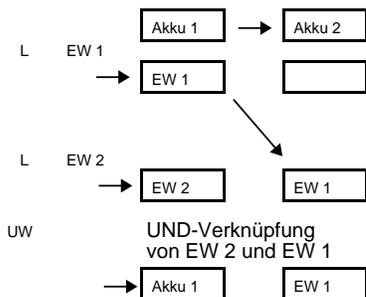
### 9.4.7 Verknüpfungsoperationen, digital

Operation	Beschreibung
<b>UW</b> <b>OW</b> <b>XOW</b>	UND-Verknüpfung digital von Akku 1 und Akku 2 ODER-Verknüpfung digital von Akku 1 und Akku 2 Exklusiv-ODER-Verknüpfung digital von Akku 1 und Akku 2

Durch zwei Ladeoperationen können der Akku 1 und Akku 2 entsprechend den Operanden der Ladeoperation geladen werden.

Anschließend lassen sich die Inhalte beider Akkus digital verknüpfen.

#### Beispiel:



Die anschließende Transferoperation transferiert das Ergebnis, das im Akku 1 abgelegt ist, zu dem bei der Transferoperation angegebenen Operanden.

### 9.4.8 Schiebeoperationen

Operation	Beschreibung
<b>SLW 0 bis 15</b> <b>SRW 0 bis 15</b> <b>SVW 0 bis 15</b>	Schieben links (von rechts werden Nullen nachgezogen) Schieben rechts (von links werden Nullen nachgezogen) Schieben rechts mit Vorzeichen (von links wird das Vorzeichen nachgezogen)
<b>SLD 0 bis 32</b> <b>SVD 0 bis 32</b>	Schieben links Doppelwort (von rechts werden Nullen nachgezogen) Schieben rechts mit Vorzeichen Doppelwort (von links wird das Vorzeichen nachgezogen)

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
:SLW 4
:T DW53 H=4AF0
```

### 9.4.9 Umwandlungsoperationen

Operation	Beschreibung
<b>KEW</b>	1er Komplement von Akku 1-L
<b>KZW</b>	2er Komplement von Akku 1-L
<b>KZD</b>	2er Komplement von Akku 1
<b>DEF</b>	Dezimal-Festpunkt Wandlung
<b>DUF</b>	Festpunkt-Dezimal Wandlung
<b>DED</b>	Dezimal-Festpunkt Doppelwort Wandlung
<b>DUD</b>	Festpunkt Doppelwort-Dezimal Wandlung
<b>FDG</b>	Festpunkt Doppelwort-Gleitpunkt Wandlung
<b>GFD</b>	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 KM=0011111001011011
: KEW
: T DW78 KM=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 KF:+51
: KZW
: T DW51 KF: - 51
```

### 9.4.10 Dekrementieren/Inkrementieren

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

**Beispiel:**

STEP 5-Programm: Belegung der Datenwörter:

```
: L DW7 KH=1010 Die Operationen I16 und D33 beeinflussen nur das
: I 16 rechte Byte. Deshalb bleibt der Wert im linken Byte
: T DW8 KH=1020 (KH=10) konstant.
: D 33
: T DW9 KH=10FF
```

### 9.4.11 Sprungoperationen

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  $\pm 127$  Wörter umfaßt und eine STEP 5-Anweisung aus mehr als einem Wort bestehen kann. Sprünge dürfen nur innerhalb eines Bausteins durchgeführt werden; Sprünge über Netzwerke hinweg sind nicht zulässig.

Operation	Beschreibung
<b>SPA</b> = <input type="text"/>	<b>Sprung unbedingt</b> Der unbedingte Sprung wird unabhängig von Bedingungen ausgeführt.
<b>SPB</b> = <input type="text"/>	<b>Sprung bedingt</b> Der bedingte Sprung wird ausgeführt, wenn das Verknüpfungsergebnis "1" ist. Bei Verknüpfungsergebnis "0" wird die Anweisung nicht ausgeführt und das Verknüpfungsergebnis auf "1" gesetzt.
<b>SPZ</b> = <input type="text"/>	<b>Sprung, wenn Akkumulatorinhalt Null (Zero) ist</b> Der Sprung wird ausgeführt, wenn der Akkumulatorinhalt Null ist. Ist der Akkumulatorinhalt nicht Null, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.
<b>SPN</b> = <input type="text"/>	<b>Sprung, wenn Akkumulatorinhalt nicht Null ist</b> Der Sprung wird ausgeführt, wenn der Akkumulatorinhalt nicht Null ist. Ist der Akkumulatorinhalt Null, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.
<b>SPP</b> = <input type="text"/>	<b>Sprung, wenn Akkumulatorinhalt positiv ist</b> Der Sprung wird ausgeführt, wenn der Akkumulatorinhalt größer als Null ist. Ist der Akkumulatorinhalt Null oder kleiner Null, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.
<b>SPM</b> = <input type="text"/>	<b>Sprung, wenn Akkumulatorinhalt negativ ist</b> Der Sprung wird ausgeführt, wenn der Akkumulatorinhalt kleiner Null ist. Ist der Akkumulatorinhalt Null oder größer Null, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.
<b>SPO</b> = <input type="text"/>	<b>Sprung bei Überlauf (Overflow)</b> Der Sprung wird ausgeführt, wenn ein Überlauf vorliegt. Liegt kein Überlauf vor, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.

Operation	Beschreibung
<p><b>SPS =</b> <input type="text"/></p> 	<p><b>Der Sprung wird ausgeführt, wenn "Überlauf gespeichert" gesetzt ist (OS=1).</b>                  Sonst (OS=0) wird der Sprung nicht ausgeführt.                  "Überlauf gespeichert" wird bei arithmetischen Operationen im Falle eines Überlaufs gesetzt und bleibt solange gespeichert bis die arithmetische Operation unterbrochen wird. Ein Überlauf entsteht, wenn bei gegebener Zahlendarstellung der zulässige Bereich durch eine arithmetische Operation überschritten wird.</p> <p><b>Symboladresse einsetzen</b> (maximal 4 Zeichen).</p>

Die bedingten Sprungoperationen (alle außer SPA) werden in Abhängigkeit vom VKE und von den Anzeigen im Steuerwerk des Automatisierungsgerätes ausgeführt.

**Hinweis:**

Sprunganweisung und Sprungziel müssen in einem Netzwerk liegen. Pro Netzwerk 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
    
```

**Beispiel (Vergleichsoperationen):**

```

: L DW 67
: L DW 107
: !=F
: SPB = GLCH (Sprung, wenn gleich, es kann auch SPZ programmiert werden)
: SPM = KLNR (Sprung, wenn kleiner)
: SPP = GRSS (Sprung, wenn größer)
:
KLNR:
:
GLCH:
:
GRSS:
    
```

**Beispiel (Rechenoperationen):**

```

: L      WERT
: L      -MESS
:+F
: SPZ   = ZERO      (Sprung, wenn Null)
: SPP   = POSI      (Sprung, wenn positiv)
:
POSI :
:
ZERO :

```

**Beispiel (Digitalverknüpfung):**

```

: L      MW25
: L      EW10
:XOW
: SPZ   = ZERO      (Sprung, wenn Akkumulatorinhalt=KH 0000)
: SPN   = NONZ      (Sprung, wenn Akkumulatorinhalt=KH 0000)
:
ZERO :
:
NONZ :

```

**Beispiel (Schiebeoperationen):**

```

: L      AW101
: SLW    10
: SPZ   = NULL      (Sprung, wenn zuletzt hinausgeschobenes Bit=0)
: SPN   = EINS      (Sprung, wenn zuletzt hinausgeschobenes Bit=1)
:
NULL :
:
EINS :
:

```

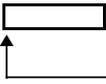
**Beispiel (Umwandlungsoperationen):**

```

: L      PW169
: KZW
: SPO   = OVFL      (Sprung, wenn Überlauf)
: SPN   = NONZ      (Sprung, wenn Akkumulatorinhalt=KH 0000)
:
NONZ :
:
OVFL :
:

```

### 9.4.12 Bearbeitungsoperationen

Operation	Beschreibung																								
<b>B=</b> 	<p><b>Bearbeite Formaloperand</b> (Parameterart: B)  <b>Formaloperand einsetzen</b> (s. S. 8)</p> <p>Nur die Operationen                  A DB                  SPA PB                  SPA SB                  SPA FB                  können substituiert werden (siehe "Typ des Bausteinparameters und zugelassener Aktualoperand Seite 4-5).</p> <p><b>B DW0 bis 254 Bearbeite Datenwort*</b>                  (Operation) Die nachfolgend angegebene Operation wird mit dem im Datenwort angegebenen Parameter kombiniert und ausgeführt.</p> <p><b>B MW0 bis 254 Bearbeite Merkerwort*</b>                  Die nachfolgend angegebene Operation wird mit dem im Merkerwort angegebenen Parameter kombiniert und ausgeführt.</p>																								
<p>* Es lassen sich auch Zweiwortbefehle substituiieren; damit sind z. B. die folgenden Befehlsfolgen möglich:</p> <table style="margin-left: 40px;"> <tr> <td>A</td><td>DB20</td> <td>A</td><td>DB100</td> </tr> <tr> <td>.</td><td>.</td> <td>.</td><td>.</td> </tr> <tr> <td colspan="4" style="text-align: center;">oder</td> </tr> <tr> <td>.</td><td>.</td> <td>.</td><td>.</td> </tr> <tr> <td>B</td><td>MW240</td> <td>B</td><td>DW21</td> </tr> <tr> <td>P</td><td>D 0.0</td> <td>S</td><td>Z 0.0</td> </tr> </table>		A	DB20	A	DB100	.	.	.	.	oder				.	.	.	.	B	MW240	B	DW21	P	D 0.0	S	Z 0.0
A	DB20	A	DB100																						
.	.	.	.																						
oder																									
.	.	.	.																						
B	MW240	B	DW21																						
P	D 0.0	S	Z 0.0																						

Die Adresse des tatsächlich angesprochenen Bits muß dann – wie sonst auch – im entsprechenden Zeigerwort hinterlegt sein (im Beispiel MW 240 bzw. DW 21). Dabei muß die Byte- bzw. Wortadresse rechts, die Bitadresse links abgelegt sein. Etwaige im Highbyte des Zeigers ausserhalb der Bitadresse vorhandene Bits werden ausgeblendet.

Bei der Substituierung von Zweiwortbefehlen im Prozeßabbild ist folgendes zu beachten:

- Die Unterscheidung zwischen Ein- und Ausgängen wird nicht im Opcode, sondern im Adreßteil des Befehls getroffen, d. h. die Vorgabe muß im Zeigerwort getroffen werden.

Zeigerwort



Bit 7 gibt Auskunft über Art der Peripherie:  
 0=Eingang; 1=Ausgang

**Beispiel:**

```
L KH 0104
T MW 250
B MW 250
SU E 0.0 (oder SU A 0.0)
E 4.1 wird gesetzt
```

- Soll der A 4.1 gesetzt werden, so muß im Merkerwort der Wert KH 0184 abgelegt werden.

- Bei der Angabe des Befehls F sind die beiden Ausdrücke SU E 0.0 oder SU A 0.0 vollkommen gleichwertig!
- Eine im Befehl selbst programmierte Bitadresse (z. B. SU E 4.7) wird nicht berücksichtigt.

#### Beispiel: **Bearbeite Datenwort**

Es sollen die Inhalte der Datenwörter DW 20 bis DW 100 gelöscht 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

```

Folgende Operationen können mit B DW bzw. B MW kombiniert werden:

U-, UN-, O-, ON-	Binäre Verknüpfungen
S-, R-, = .	Speicherfunktionen
FRT, R T, SA T, SE T, SI T, SS T, SV T	Zeitfunktionen
FRZ, R Z, S Z, ZR Z, ZV Z	Zählfunktionen
L-, LC-, T-	Lade- und Transferfunktionen
SPA, SPB, SPZ, SPN, SPP, SPM, SPO	Sprungfunktionen
SLW, SRW	Schiebefunktionen
D,	Dekrementieren, Inkrementieren
A DB, SPA-, SPB-	Bausteinaufrufe

Das Programmiergerät 670 prüft die Zulässigkeit der Kombination nicht ab. Es dürfen keine Zwei- bzw. Dreiwort-Befehle und keine Operationen mit Formaloperanden in Funktionsbausteinen kombiniert werden.

#### 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 1.

```

      : 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

```

### 9.4.13 STEP 5-Befehle mit direktem Speicherzugriff

Operation	Beschreibung
<b>LIR 0</b>	Lade Akku 1: mit dem Inhalt des durch Akku 1 adressierten Speicherwortes
<b>TIR 2</b>	Transferiere Akku 2: in das durch den Inhalt von Akku 1 adressierte Speicherwort
<b>TNB 0 bis 255</b>	Blocktransfer byteweise, Quelladresse in Akku 2, Zieladresse in Akku 1
<b>TNW 0 bis 25</b>	Blocktransfer blockweise, Quelladresse in Akku 2, Zieladresse in Akku

Die Befehle LIR, TIR, TNB, TNW müssen, wenn sie im Anwenderprogramm verwendet werden, erst per Maschinendatum (MD 2003 Bit 5=1) freigegeben sein. Ansonsten erfolgt eine Fehlermeldung und die PLC verzweigt in die Stoppschleife.

Es werden nur die Befehle LIR0 und TIR2 durchgeführt. Die Parameter werden jedoch nicht überprüft, d. h. es kann ein beliebiger Wert programmiert werden. Eine Fehlermeldung erfolgt nicht.

Für die Programmierung der Befehle mit direktem Speicherzugriff (LIR, TIR, TNB, TNW) gilt:

Es wird für Quelle bzw. Ziel eine Offset- und eine Segmentadresse benötigt. Die Vorgabe des Segmentes geschieht durch Eingabe einer Segmentnummer im Highwort des AKKUs.

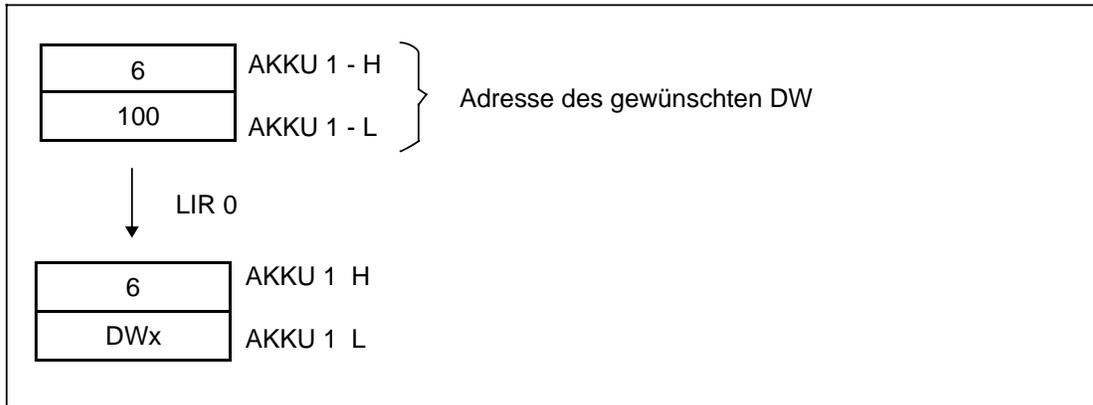
Dabei gilt die folgende Zuordnung:

Segment-Nr	Segment
<b>1</b>	PLC-Systemdaten (z. B. Prozeßabbilder, Bausteinadresslisten)
<b>2</b>	Hardware (z. B. EIA-Signalumformer, MPC-Koppelspeicher)
<b>3</b>	Interne NC-PLC-Schnittstelle
<b>4</b>	Spindelschnittstelle
<b>5</b>	Anwender-Programmspeicher
<b>6</b>	Anwender-Datenspeicher
<b>7</b>	PLC-Systemprogramm
<b>8</b>	residente Funktionsbausteine (z. B. FB 11)

Die Segmentnummern 1 ... 10 sind bei Lesezugriffen (Quelle) zulässig. Als Ziel eines Schreibvorganges bei den Befehlen TIR, TNB, TNW sind nur die Segmentnummern 1 ... 6 erlaubt. Bei Verstoß oder Vergabe einer anderen Nummer geht die PLC mit einer entsprechenden Fehlermeldung in den Stoppzustand.

## 1. Beispiel: Lesezugriff auf bestimmte Adresse

L KB 6 Segmentadresse für DWx AKKU 1\_H  
 T MW 250  
 L KH 0100 Offsetadresse für DWx AKKU 1\_L  
 T MW 252  
 L MD 250  
 LIR 0 Laden von DWx nach AKKU 1\_L



## 2. Beispiel: Schreibvorgang auf bestimmte Adresse

L KB 6 Segmentadresse für Wert x  
 T MW 250  
 L KH 0100 Offsetadresse für Wert x  
 T MW 252  
 L KH FFFF Laden von Wert x  
 L MD 250  
 TIR 2 Transferieren von Wert x Anwenderdatensegment

## 9.4.14 Sonstige Operationen

Operation	Beschreibung
<b>ADD BF-127 bis +127</b>	Addiere Byte-Konstante (Festpunkt) zum Akku 1
<b>ADD KF-32768 bis+32767</b>	Addiere Festpunktkonstante (Wort) zum Akku 1
<b>STS</b> <b>TAK</b>	sofortiger Stop-Befehl Tausche den Inhalt von Akku 1 und Akku 2

## 9.5 STEP 5-Befehlslaufzeiten

BEFEHL	OPCODE	LAUFZEIT [Microsekunden]	
		bei COP	bei ACOP
U UN O ON		0,5 - 0,75	0,5 - 0,75
S R =		0,8	0,75
O		0,3	0,25
NOP 0	0000	3	0,25
NOP 1	FFFF	3	0,25
BLD 255	10FF	3	0,25
FRT	0400	0,75 - 0,8	0,75
FRZ	4400	0,75 - 0,8	0,75
U (	BA00	0,3	0,25
O)	BB00	0,3	0,25
)	BF00	0,3	0,25
L MB	0A00	5,8	0,5
L EB / AB	4A00	7,3	0,5
L MW	1200	5,8	0,75
L EW / AW	5200	5,8	0,75
L MD	1A00	7,6	1,0
L ED / AD	5A00	7,6	1,0
T MB	0B00	6,4	0,5
T EB / AB	4B00	6,4	0,5
T MW	1300	7,0	0,75
T EW / AW	5300	7,0	0,75
T MD	1B00	9,5	1
T ED / AD	5B00	9,5	1

BEFEHL	OPCODE	LAUFZEIT [Microsekunden]	
		bei COP	bei ACOP
L PB	7200	ca. 12	1,0
L PW	7A00	ca. 13	0,75 - 1,0
T PB	7300	ca. 14	1,25
T PW	7B00	ca. 15	2,25
L KB	2800	4,4	0,25
L KH	3000 0000	8,8	0,5
L KG	3800 0000 0000	10,5	0,75
L Z	4200	7,3	0,5
L T	0200	7,3	0,5
LC Z	4C00	14,0 - 38,3	0,5
LC T	0C00	15,2 - 39,5	0,5
L DR	2A00	8,6	0,5
L DL	2200	8,3	0,5
L DW	3200	8,3	0,5
L DD	3A00	10,0	0,75
T DR	2B00	9,6	0,5
T DL	2300	8,6	0,5
T DW	3300	11,3	0,5
T DD	3B00	13,4	0,75
LIR	4000	8,5	8,5
TIR	4800	9,0	9,0
TNB	0300	19,0 - 161,8	19,0 - 161,8
TNW	4300	20,3 - 163,2	20,3 - 163,2
SE T	2400	0,7 - 18,3	0,7 - 18,3
SS T	2C00	0,7 - 18,3	0,7 - 18,3
SI T	3400	0,7 - 18,2	0,7 - 18,3
SV T	1C00	0,7 - 18,2	0,7 - 18,3
SA T	1400	0,7 - 18,2	0,7 - 18,3
R T	3C00	0,7 - 6,3	0,25 - 6,8

BEFEHL	OPCODE	LAUFZEIT [Microsekunden]	
		bei COP	bei ACOP
R Z	7C00	0,7 - 6,3	0,25 - 6,3
S Z	5C00	0,7 - 18,1	0,75 - 18,1
ZV Z	6C00	0,7 - 11,5	0,75 - 11,5
ZR Z	5400	0,7 - 10,7	0,75 - 10,7
F T	0400	0,7	0,75
F Z	4400	0,7	0,75
SPA FB	3D00	35,8	2
SPB FB	1D00	0,3 - 35,8	2
SPA PB	7500	30,0	2
SPB PB	5500	0,3 - 30,0	2
SPA SB	7D00	30,0	2
SPB SB	5D00	0,3 - 30,0	2
BE	6500	8 - 22,8	2,3 - 22,8
BEA	6501	8 - 22,8	2,3 - 22,8
BEB	0500	0,3 - 22,8	2,3 - 22,8
A DB	2000	15,4	1
SPA =	2D00	6,1	0,32
SPB =	FA00	0,3 - 6,1	0,32
SPM =	2500	4,8 - 7,8	0,32
SPN =	3500	4,8 - 7,8	0,32
SPP =	1500	4,8 - 7,8	0,32
SPZ =	4500	4,8 - 7,8	0,32
SPO =	0D00	4,8 - 7,8	0,32
SRW	6900	7,1	1,5
SLW	6100	7,1	1,5
SVD	7100	7,6 - 111,6	2,5
SLD	2900	7,6 - 111,6	2,5

BEFEHL	OPCODE	LAUFZEIT [Microsekunden]	
		bei COP	bei ACOP
>F (GRF)	2120	12,8 - 13,7	0,25
<F (KLF)	2140	12,8 - 14,0	0,25
><F (UGLF)	2160	12,8 - 14,0	0,25
!=F (GLF)	2180	13,1 - 14,4	0,25
>=F (GGF)	21A0	13,1 - 14,4	0,25
<=F (KGF)	21C0	13,6 - 14,0	0,25
>D (GRD)	3920	13,0 - 14,0	0,25
<D (KLD)	3940	13,0 - 14,0	0,25
><D (UGLD)	3960	13,0 - 14,0	0,25
!=D (GLD)	3980	13,0 - 14,0	0,25
>=D (GGD)	39A0	13,0 - 14,0	0,25
<=D (KGD)	39C0	13,0 - 14,0	0,25
>G (GRG)	3120	14,5 - 20,5	14,5 - 20,5
<G (KLG)	3140	14,5 - 20,5	14,5 - 20,5
><G (UGLG)	3160	14,5 - 20,5	14,5 - 20,5
!=G (GLG)	3180	14,5 - 20,5	14,5 - 20,5
<=G (GGG)	31A0	14,5 - 20,5	14,5 - 20,5
>=G (KGG)	31C0	14,5 - 20,5	14,5 - 20,5
+F (ADDF)	7900	7,2 - 9,7	0,25
- F (SUBF)	5900	7,8 - 10,3	0,25
D	1900	3,4	0,25
I	1100	3,4	0,25
ADD BF	5000	3,7	0,25
ADD KF	58 00 0000	8,3	0,25

BEFEHL	OPCODE	LAUFZEIT [Microsekunden]	
		bei COP	bei ACOP
*F (NULP)	6004	13,7 - 16,8	13,7 - 16,8
:F (DIVF)	6000	11,2 - 16,6	11,2 - 16,6
*G (MULG)	6007	17,5 - 27,5	17,5 - 27,5
:G (DIVG)	6003	9,4 - 27,5	9,4 - 27,5
+G (ADDG)	600F	20,0 - 193,0	20,0 - 193,0
- G (SUBG)	600B	20,0 - 200,0	20,0 - 200,0
SPS	600C 0000	13,8 - 16,8	0,56
KEW	0100	2,8	0,25
KZW	0900	6,8 - 7,5	0,25
UW	4100	7,1	0,25
OW	4900	7,1	0,25
XOW	5100	7,1	0,25
SVW	6801	12,2	1,5
KZD	6807	11,3 - 13,5	11,3 - 13,5
DUF	6808	18,0 - 56,2	18 - 56,2
DUD	680A	14,8 - 165,1	14,8 - 165,1
DEF	680C	20,0	20
DED	680E	11,1 - 45,3	11,1 - 45,3
GFD	6002	9,7 - 122,5	9,7 - 122,5
FDG	6806	150,0 - 190,0	150,0 - 190,0
TAK	7002	6,9	0,25
STP	7003	7,9	7,9
STS	7000	----	----
RU Z	7015 0000	23,0	1
SU Z	7015 4000	22,7	1
PN Z	7015 8000	23,8	0,75
P Z	7015 C000	23,6	0,75
RU T	7025 0000	22,6	1

BEFEHL	OPCODE	LAUFZEIT [Microsekunden]	
		bei COP	bei ACOP
SU T	7025 4000	22,3	1
PN T	7025 8000	23,3	0,75
P T	7025 C000	23,3	0,75
RU D	7046 0000	25,8	1
SU D	7046 4000	25,3	1
PN D	7046 8000	24,3	0,75
P D	7046 C000	24,3	0,75
RU E / A	7038 0000	23,6	1
SU E / A	7038 4000	21,5	1
PN E / A	7038 8000	22,1	0,75
P E / A	7038 C000	21,9	0,75
RU M	7049 0000	23,5	1
SU M	7049 4000	21,5	1
PN M	7049 8000	22,2	0,75
P M	7049 C000	21,8	0,75
U D	783F 0000	25,7	0,75
O D	783F 1000	25,7	0,75
UN D	783F 2000	25,7	0,75
ON D	783F 3000	25,7	0,75
S D	783F 4000	26,7	1
R D	783F 5000	27,0	1
= D	783F 6000	30,2	1
B MW 1)	4E00	14,5 - 25,0	0,8
B DW 1)	6E00	16,0 - 26,0	0,6

**Hinweis:**

1) Bearbeitungszeit des substituierten Befehls muß zur angegebenen Zeit addiert werden.

BEFEHL	OPCODE	LAUFZEIT [Microsekunden]	
		bei COP	bei ACOP
S =	1700	11,6 <sup>1)</sup>	0,32 <sup>1)</sup>
RB =	3700	11,6 <sup>1)</sup>	0,32 <sup>1)</sup>
= =	1F00	11,6 <sup>1)</sup>	0,32 <sup>1)</sup>
U =	0700	13,2 <sup>1)</sup>	0,32 <sup>1)</sup>
O =	0F00	13,6 <sup>1)</sup>	0,32 <sup>1)</sup>
UN =	2700	13,6 <sup>1)</sup>	0,32 <sup>1)</sup>
ON =	2F00	13,6 <sup>1)</sup>	0,32 <sup>1)</sup>
L =	4600	9,7 <sup>1)</sup>	0,32 <sup>1)</sup>
T =	6600	10,1 <sup>1)</sup>	0,31 <sup>1)</sup>
LC =	0E00	11,8 <sup>1)</sup>	0,31 <sup>1)</sup>
LW =	3F00	10,1 <sup>1)</sup>	0,5 <sup>1)</sup>
LD =	5600	12,5 <sup>1)</sup>	0,75 <sup>1)</sup>
SI =	3600	11,2 <sup>1)</sup>	0,32 <sup>1)</sup>
SE =	2600	11,2 <sup>1)</sup>	0,32 <sup>1)</sup>
FR =	0600	12,1 <sup>1)</sup>	0,32 <sup>1)</sup>
RD =	3E00	11,9 <sup>1)</sup>	0,32 <sup>1)</sup>
SVZ =	1E00	12,2 <sup>1)</sup>	0,32 <sup>1)</sup>
SSV =	2E00	12,2 <sup>1)</sup>	0,32 <sup>1)</sup>
SAR =	1600	12,2 <sup>1)</sup>	0,32 <sup>1)</sup>
B =	7600	13,5 <sup>1)</sup>	0,32 <sup>1)</sup>

**Hinweis:**

1) Bearbeitungszeit des substituierten Befehls muß zur angegebenen Zeit addiert werden.

# 10 Regeln zur Kompatibilität zwischen den Darstellungsarten KOP, FUP und AWL

## 10.1 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 dieses Abschnittes ist die Aufstellung einiger Regeln, deren Einhaltung eine vollständige Kompatibilität der drei Darstellungsarten gewährleistet.

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.

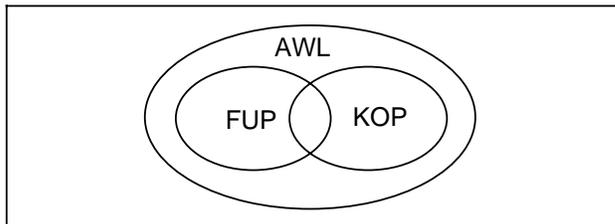


Bild 10.1 Umfang bzw. Begrenzungen der Darstellungsarten der Programmiersprache STEP 5.

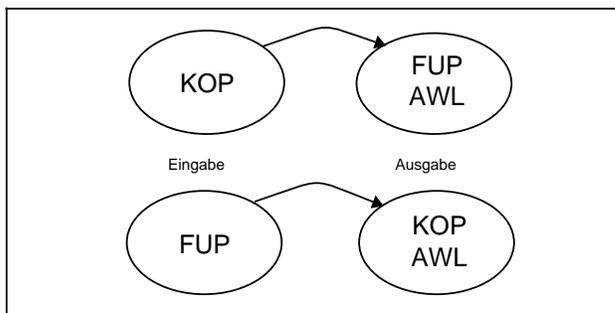


Bild 10.2 Graphische Eingabe

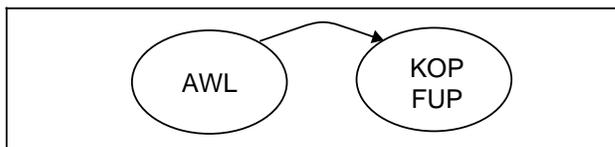


Bild 10.3 Eingabe in Anweisungsliste

## 10.2 Kompatibilitätsregeln bei graphischer Programmeingabe (KOP, FUP)

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

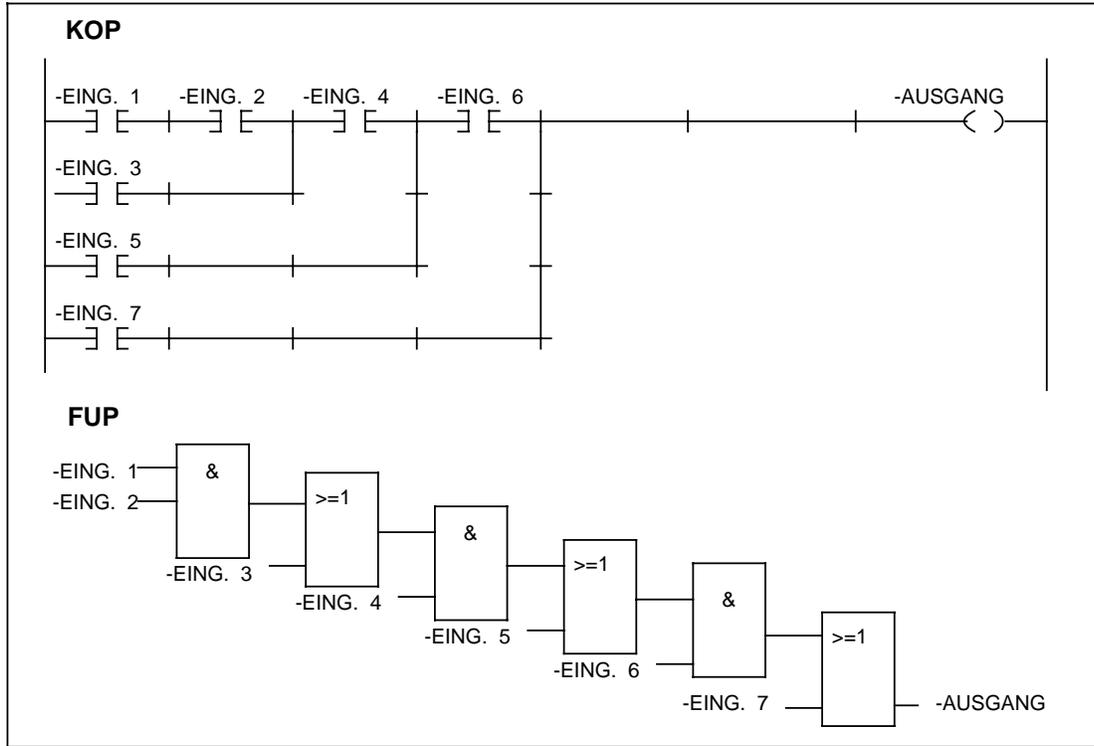


Bild 10.4 Beispiel einer maximalen KOP-Schachtelung zur Ausgabe in FUP

### 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.

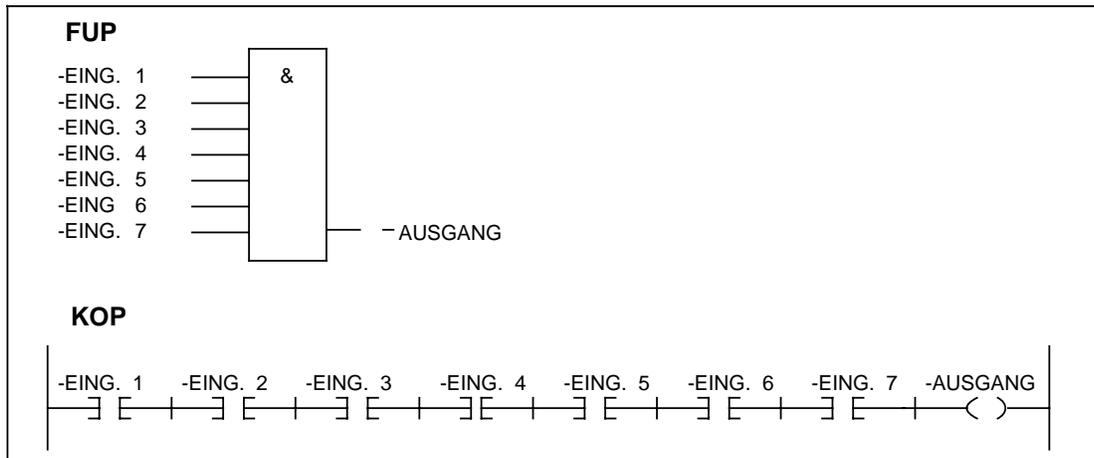


Bild 10.5 Beispiel eines maximalen UND-Kastenausbaus zur Ausgabe in KOP

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

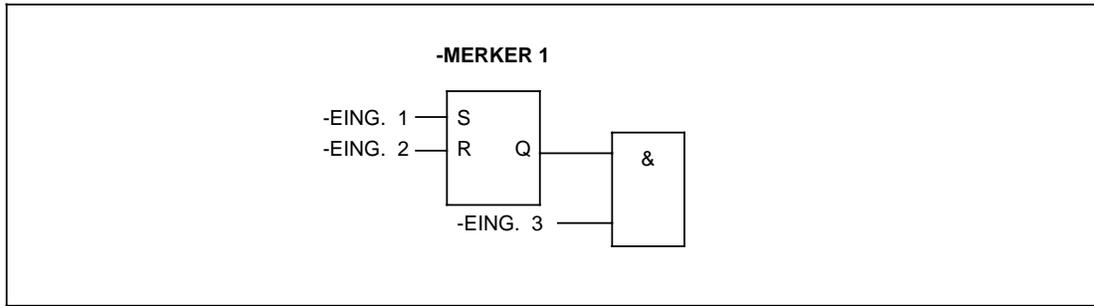


Bild 10.6 Nur UND-Kästen sind nach einem komplexen Glied zulässig.

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

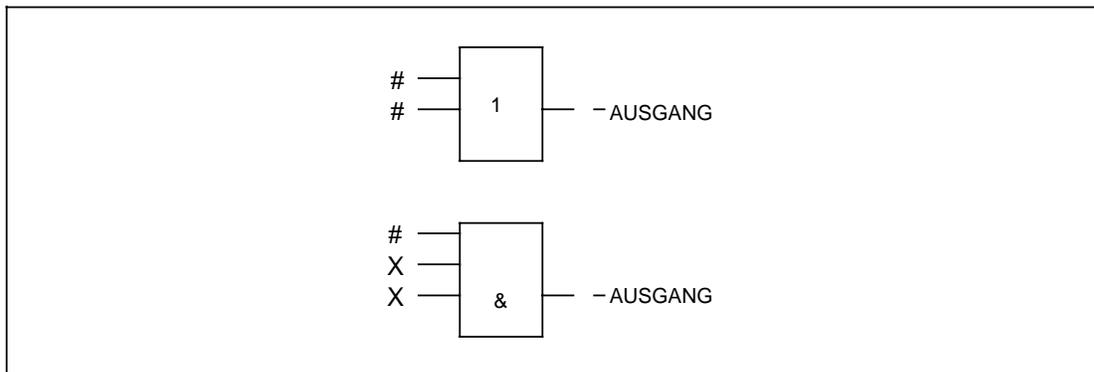


Bild 10.7 Beispiel für die Zulässigkeit der Konnektoren bei ODER- und UND-Kästen

### 10.3 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 . . .

AWL: U . . . .

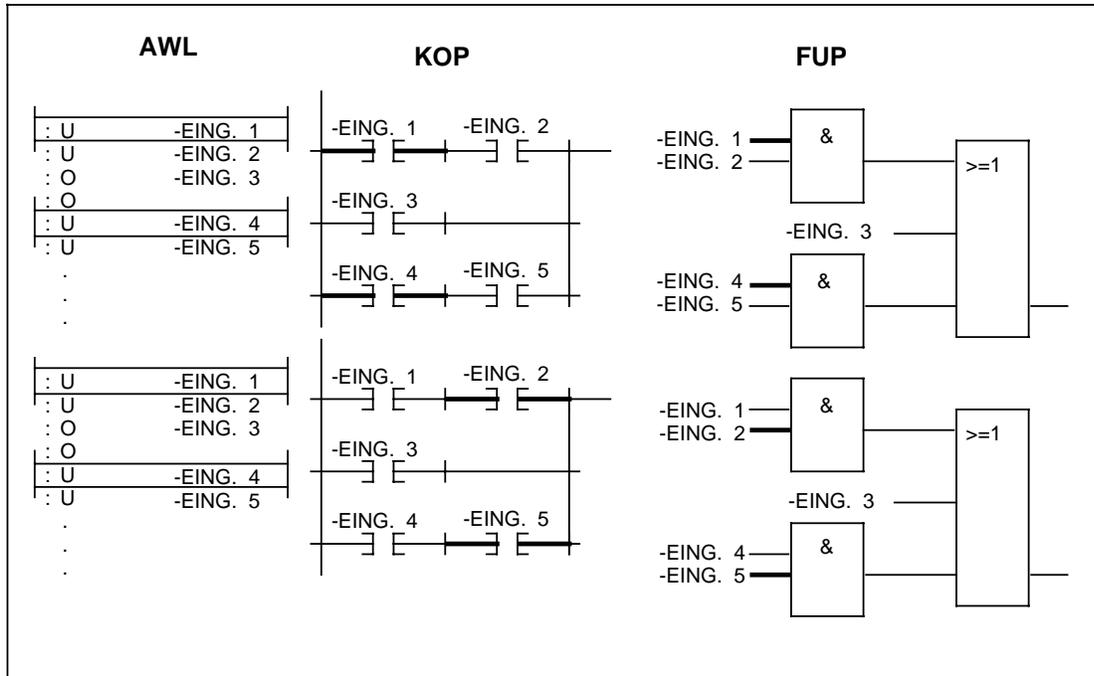


Bild 10.8 Erläuterungen zur Regel UND-Verknüpfung

**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 . . .

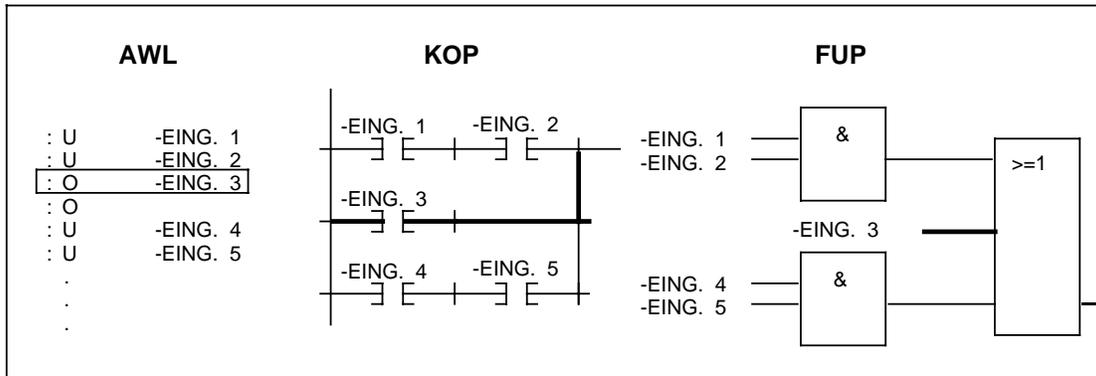
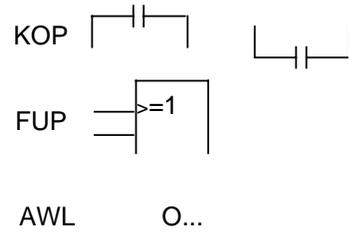


Bild 10.9 Erläuterungen zur Regel ODER-Verknüpfung

**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

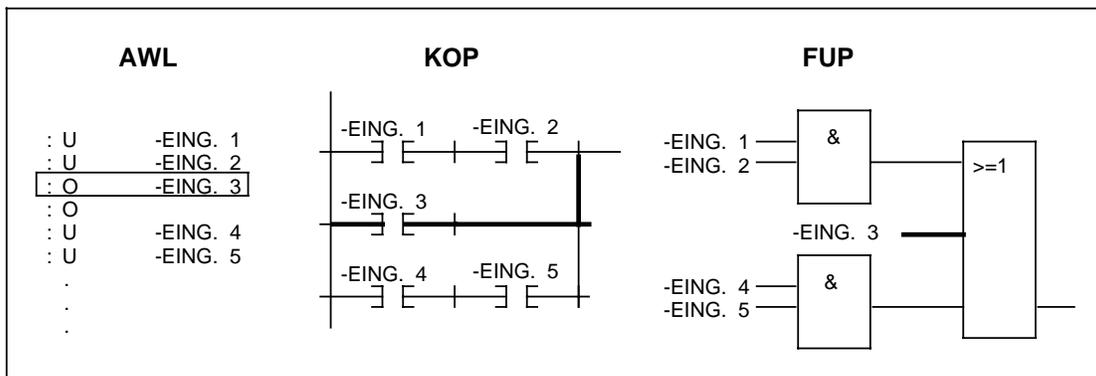
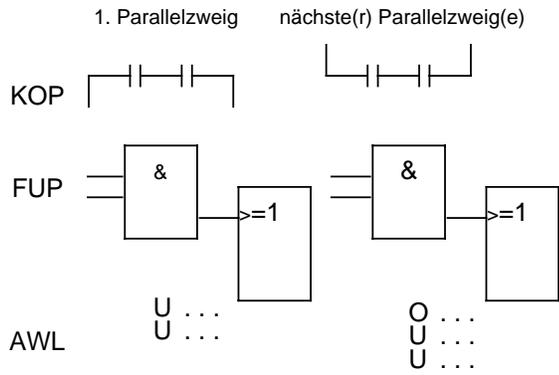
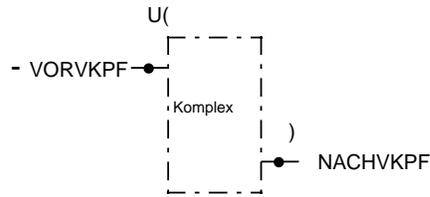


Bild 10.10 Erläuterungen zur Regel UND-vor-ODER-Verknüpfung

**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.

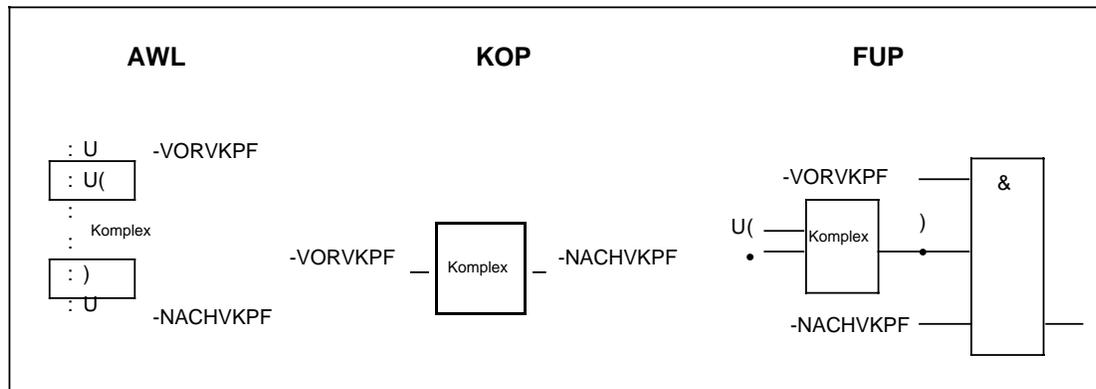


Bild 10.11 Erläuterungen zur Klammerung von komplexen binären Funktionen

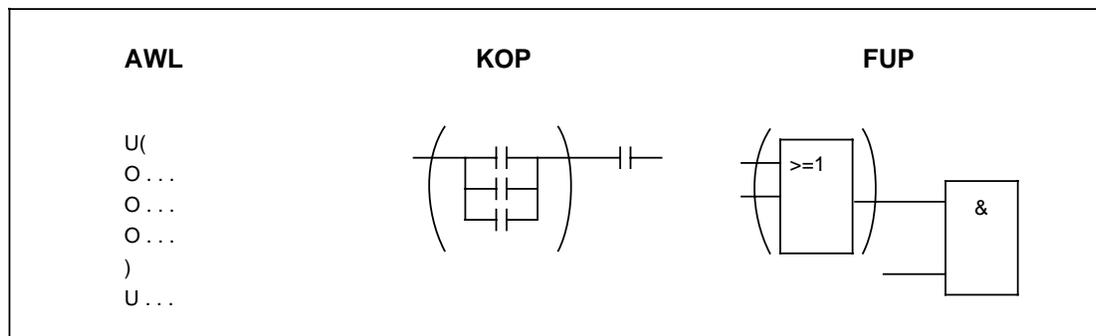


Bild 10.12 Erläuterungen zur Regel ODER-vor-UND-Verknüpfung

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.

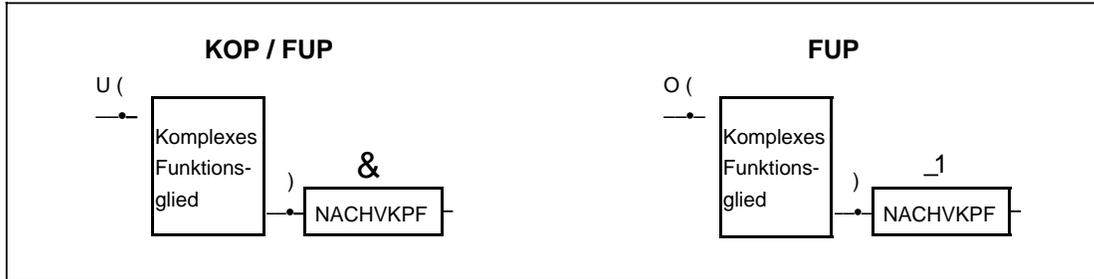


Bild 10.13 Erläuterungen zur Klammerung von komplexen Gliedern

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 Parallelzweig)
- Fall 4: ODER-vor-UND (Klammerung)

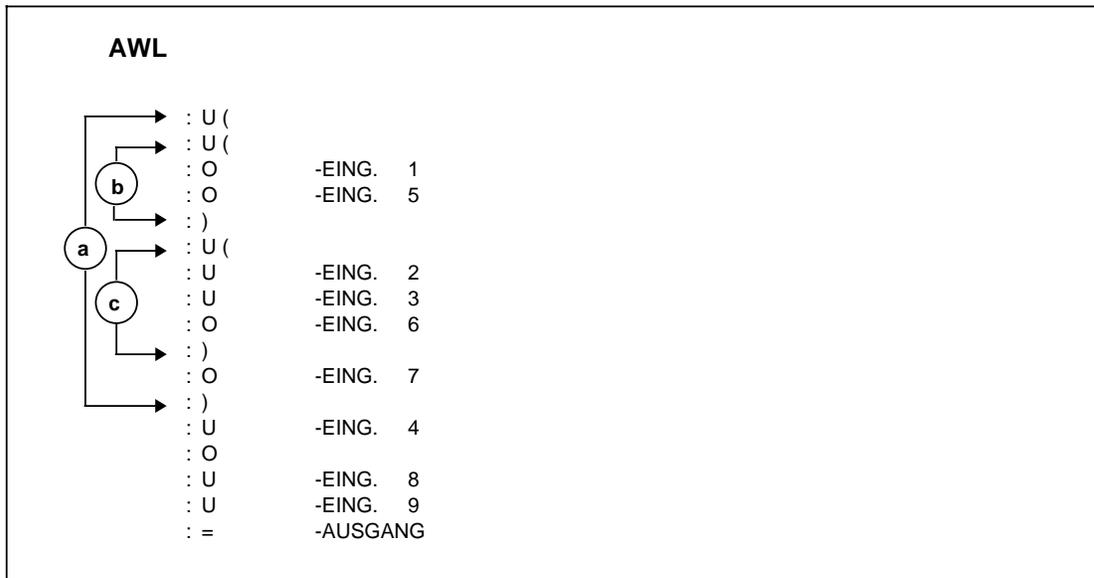


Bild 10.14 Beispiel 1: KOP/AWL (Fortsetzung nächste Seite)

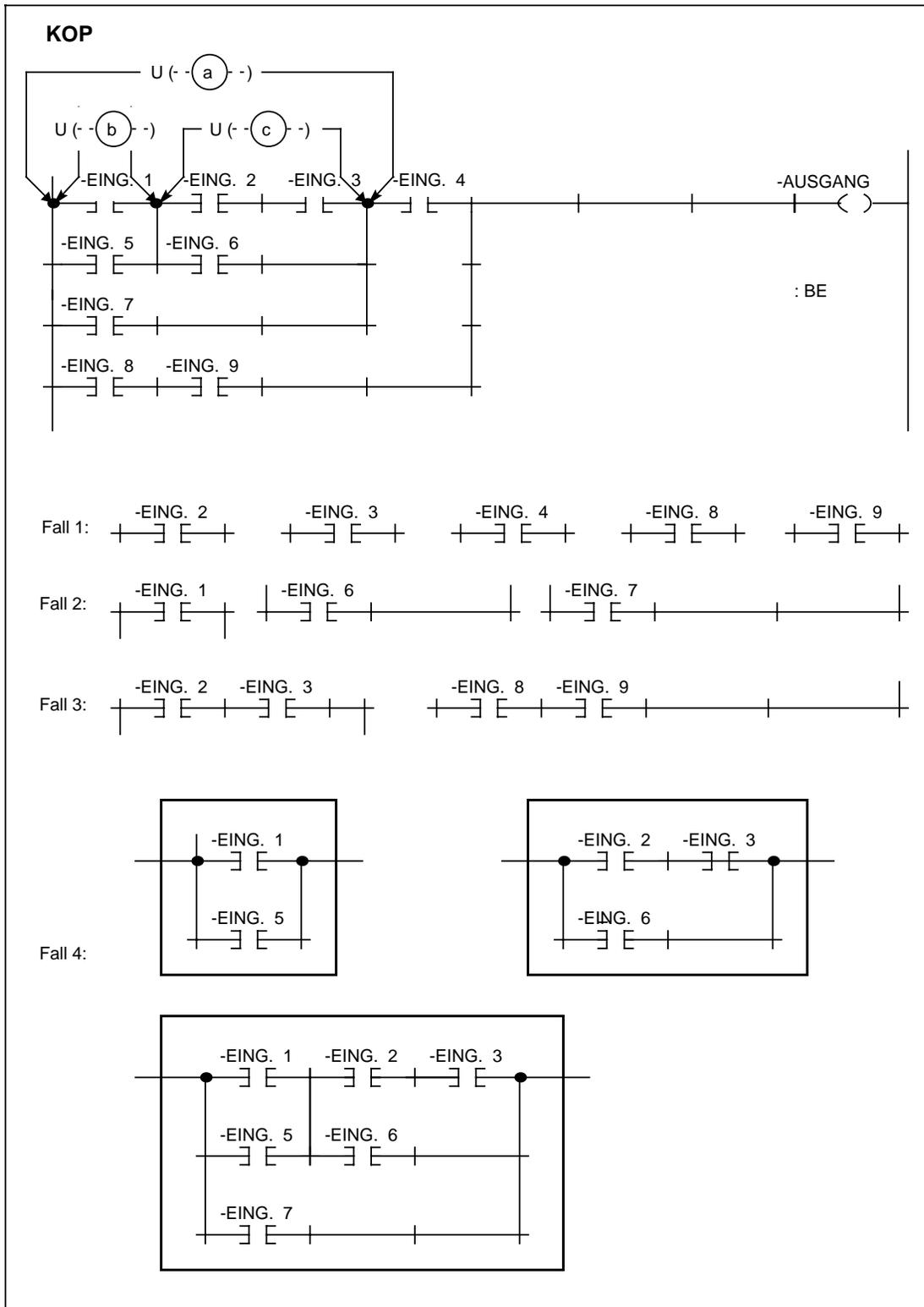


Bild 10.14 Beispiel 1: KOP/AWL (Fortsetzung)

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 parametrisiert werden.

**Ausnahme:**

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

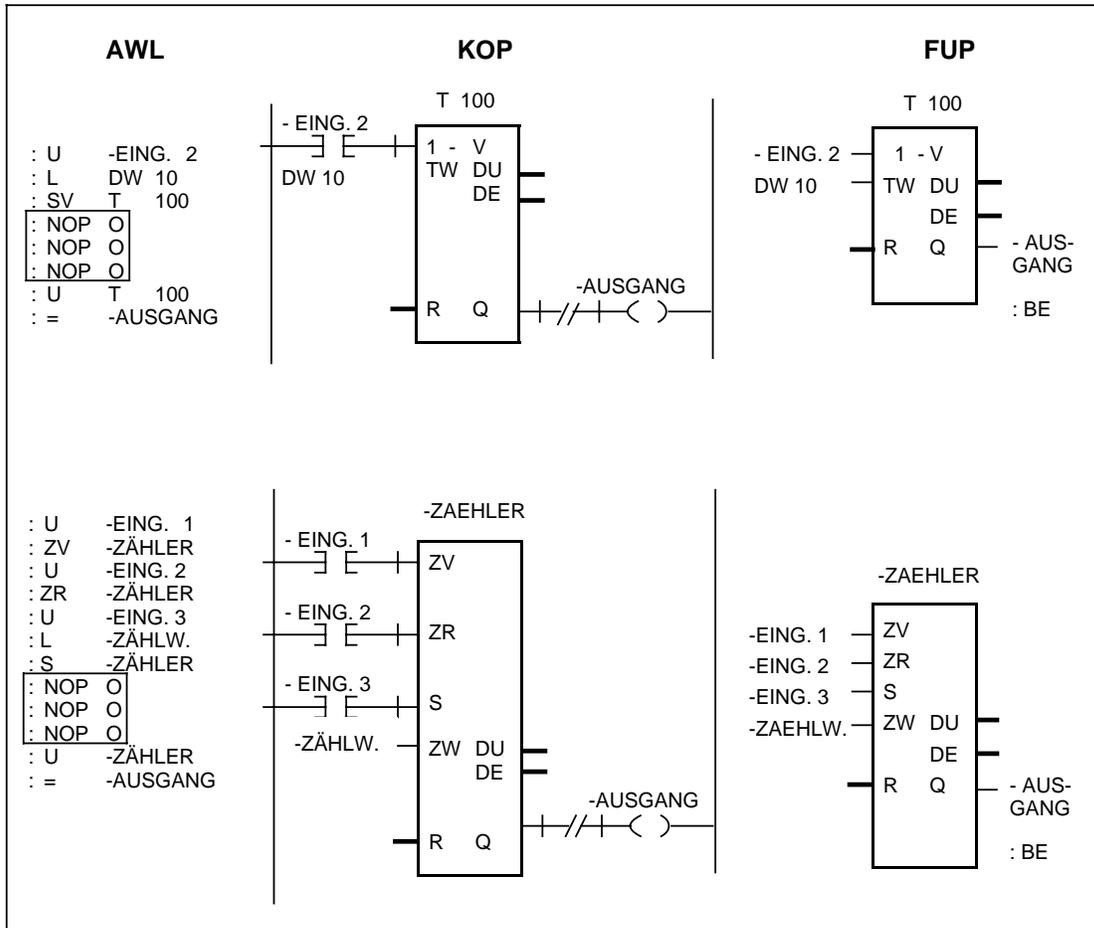


Bild 10.15 Beispiel zur Versorgung unbeschalteter Ein- und Ausgänge

**Wichtiger Hinweis:**

Pro Netzwerk 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.

**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)

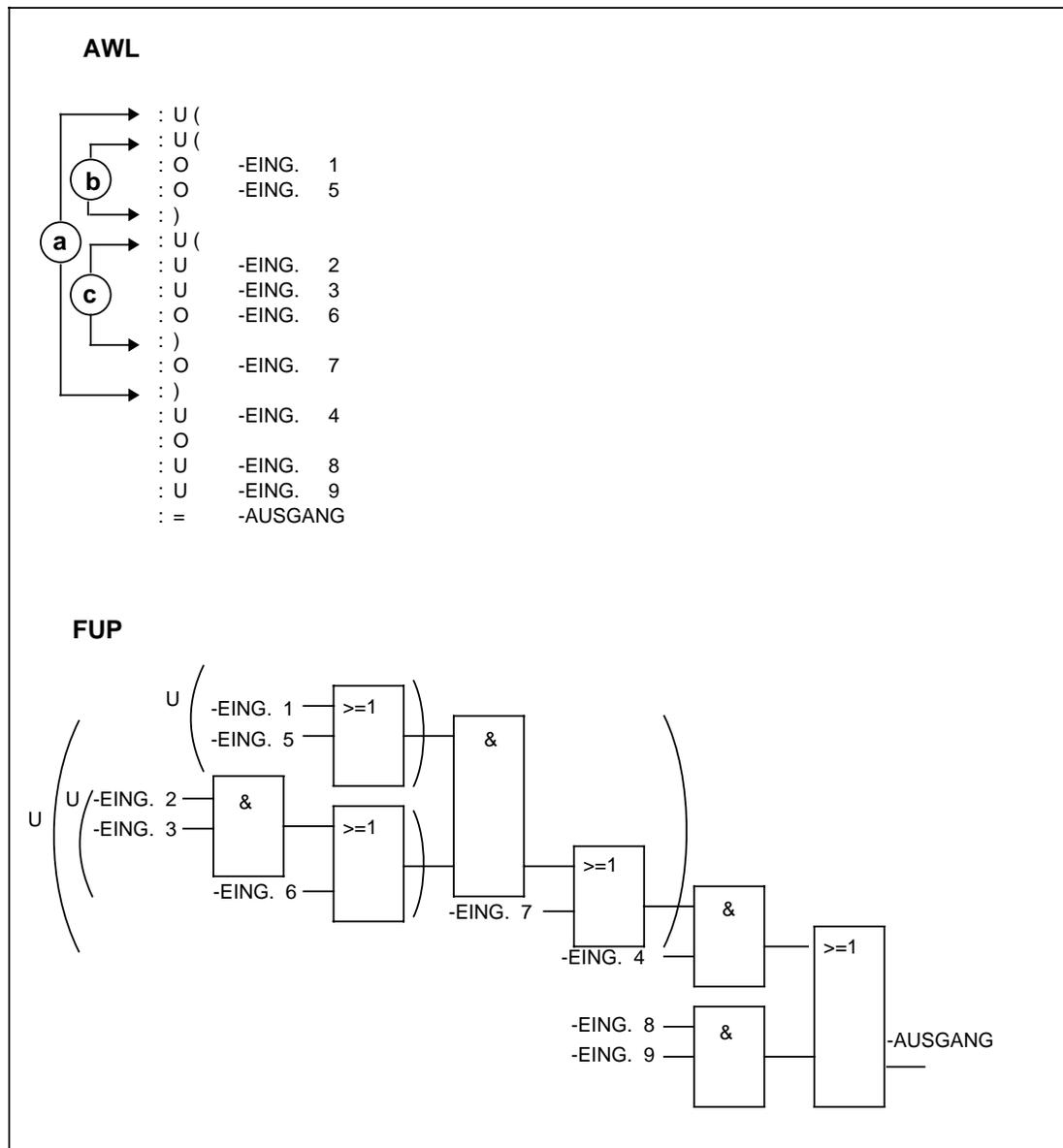


Bild 10.16 Beispiel 2: FUP AWL (Fortsetzung nächste Seite)

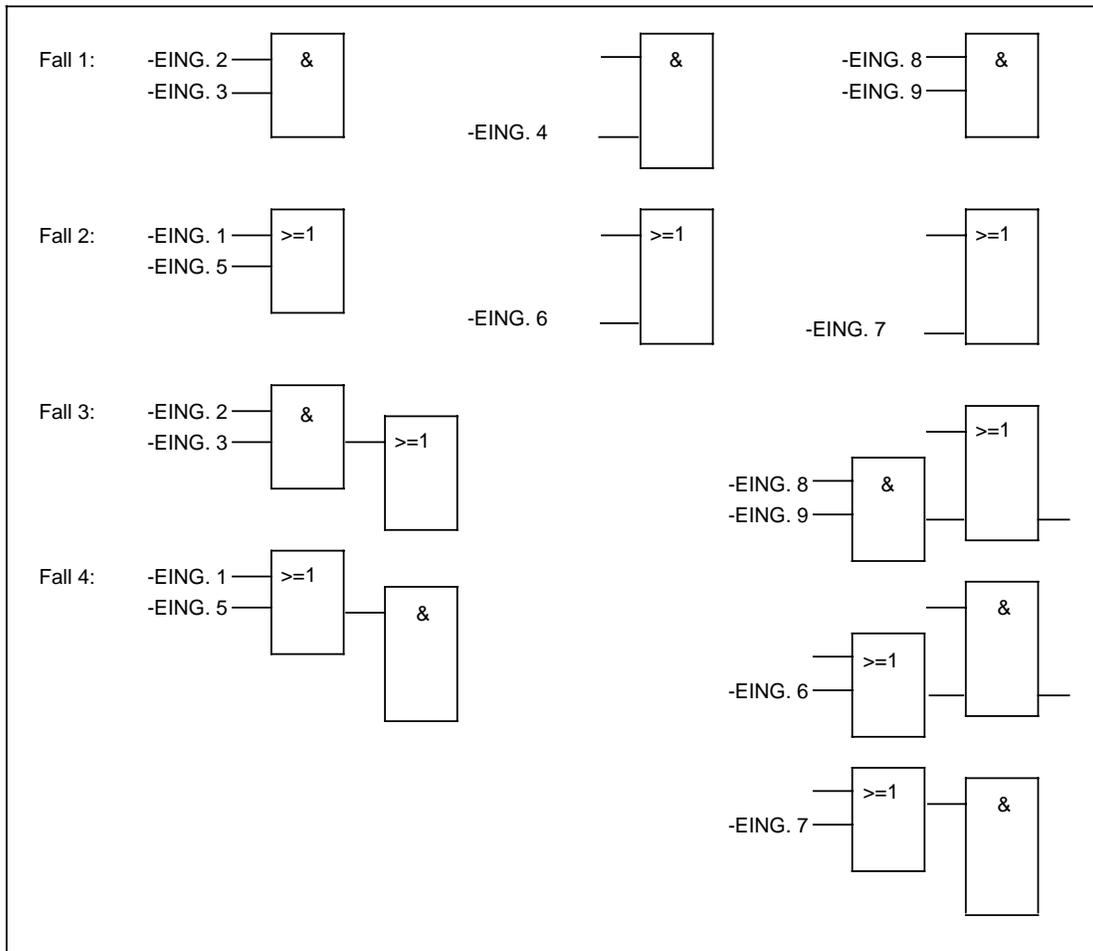


Bild 10.16 Beispiel 2: FUP AWL (Fortsetzung)

Regel 5: Konnektoren

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

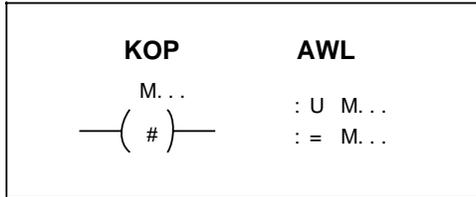


Bild 10.17 Der Konnektor in KOP und AWL

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:

- 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.
- 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.

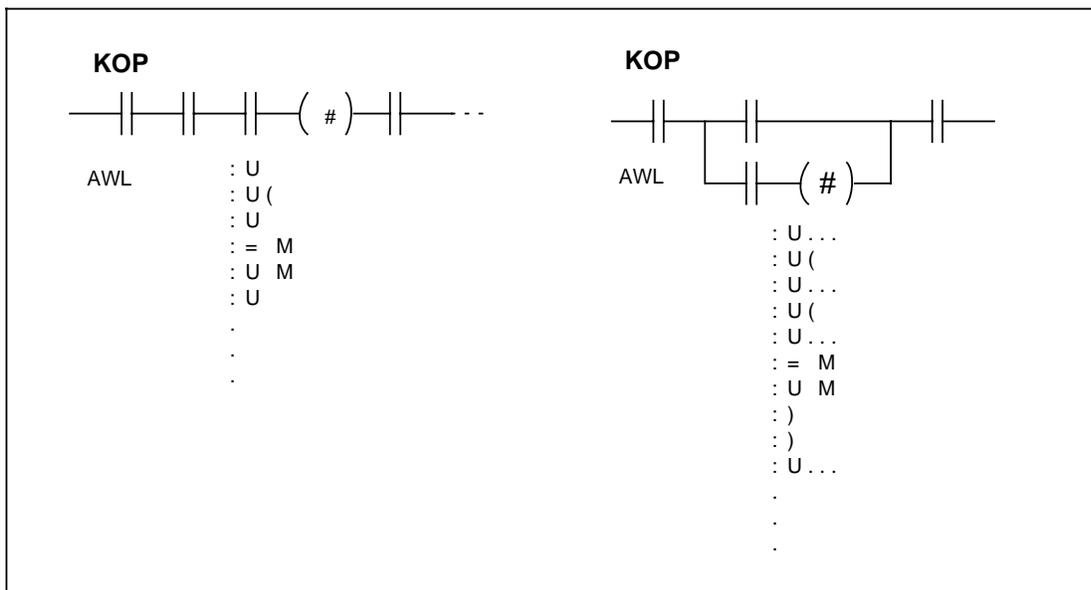


Bild 10.18 Konnektor-Regler für KOP

FUP	AWL
— # M... —	:=M... :U M...

Bild 10.19 Der Konnektor in FUP und AWL

## b) Konnektoren bei FUP

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

- 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 Types 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!)

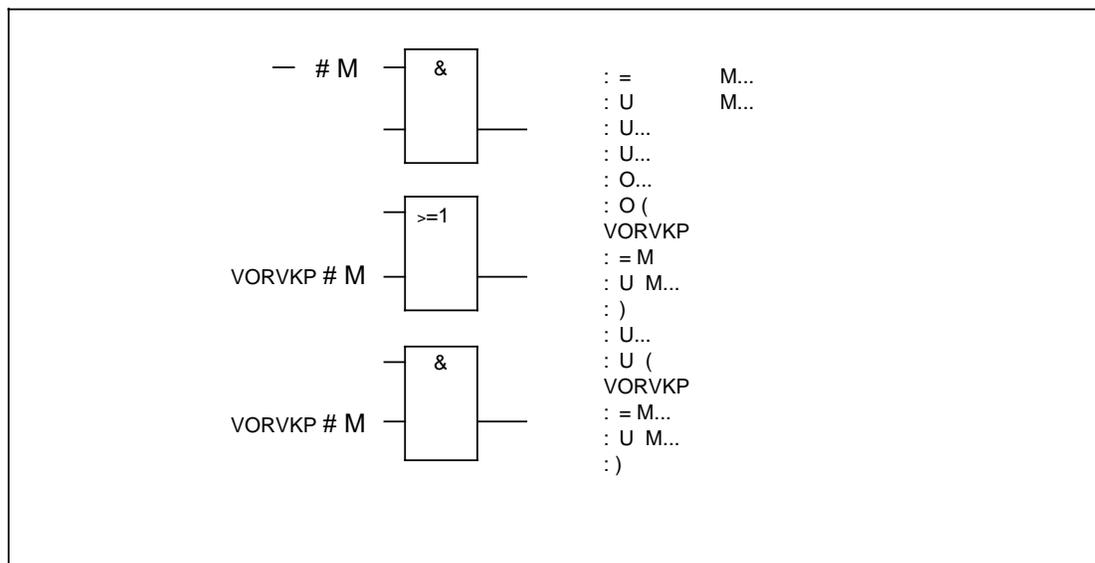


Bild 10.20 Konnektor-Regler für FUP

**Beispiele zu den Konnektoren:**

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

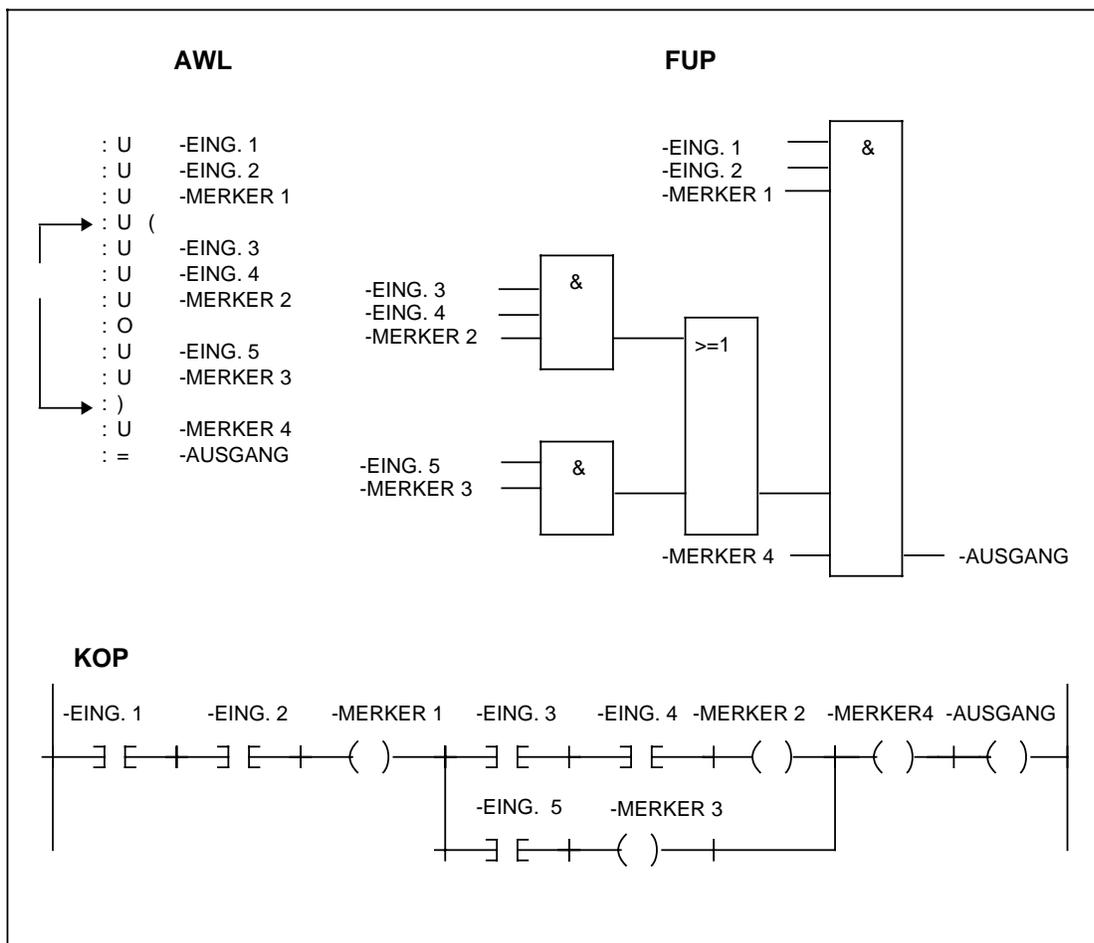


Bild 10.21 Beispiel ohne Konnektoren

An  
SIEMENS AG

AUT V280  
Postfach 3180  
D-91050 Erlangen

**Vorschläge**

**Korrekturen**

für Druckschrift:

SINUMERIK 810 GA3  
SINUMERIK 820 GA3  
Programmieren PLC

Hersteller-Dokumentation

**Absender**

Name \_\_\_\_\_

Anschrift Ihrer Firma/Dienststelle

Straße: \_\_\_\_\_

PLZ: \_\_\_\_\_ Ort: \_\_\_\_\_

Telefon: \_\_\_\_\_ / \_\_\_\_\_

Telefax: \_\_\_\_\_ / \_\_\_\_\_

Projektierungsanleitung

Bestell-Nr.: 6ZB5 410-0DW01-0AA3

Ausgabe: 07.94

Sollten Sie beim Lesen dieser Unterlage auf Druckfehler gestoßen sein, bitten wir Sie, uns diese mit diesem Vordruck mitzuteilen. Ebenso dankbar sind wir für Anregungen und Verbesserungsvorschläge.

Herausgegeben von Siemens AG  
Bereich Automatisierungstechnik  
Geschäftsgebiet Automatisierungssysteme  
für Werkzeugmaschinen, Roboter  
und Sondermaschinen  
Postfach 3180, D-91050 Erlangen

Siemens Aktiengesellschaft

© Siemens AG 1990 All Rights Reserved  
Änderungen vorbehalten

Bestell-Nr. 6ZB5410-0DW01-0AA3  
Printed in the Fed. Rep. of Germany

