

SIMATIC S5

AG S5-155U CPU 948

Programmieranleitung

Dieses Handbuch hat die Bestellnummer:
6ES5 998-3PR11

Inhaltsverzeichnis	
Einführung	1
Anwenderprogramm	2
Programmbearbeitung	3
Betriebszustände und Programmbearbeitungsebenen	4
Unterbrechungs- und Fehlerbehandlung	5
Integrierte Sonderfunktionen	6
Erweiterter Datenbaustein DX 0	7
Speicherbelegung und Speicherorganisation	8
Speicherzugriffe auf absolute Adressen	9
Mehrprozessorbetrieb und -kommunikation im AG S5-155U	10
PG-Schnittstellen und -Funktionen	11
Anhang	12
Verzeichnisse: Abkürzungsverzeichnis Stichwortverzeichnis	13
Das Tabellenheft 6ES5 997-3UA12 wird dem Handbuch beigelegt	

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährungsgradfolgendermaßen dargestellt:



Gefahr

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Warnung

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Vorsicht

bedeutet, daß eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Hinweis

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie folgendes:



Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Warenzeichen

SIMATIC®, SIMATIC HMI® und SIMATIC NET® sind eingetragene Warenzeichen der SIEMENS AG.

Die übrigen Bezeichnungen in dieser Schrift können Warenzeichen sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen können.

Copyright Siemens AG 1998 All rights reserved

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

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschäftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D- 90327 Nürnberg

Haftungsausschluß

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

© Siemens AG 1998
Technische Änderungen bleiben vorbehalten.

Inhaltsverzeichnis

1	Einführung	1 - 3
1.1	Anwendungsbereich S5-155U mit CPU 948	1 - 4
1.2	Typische Arbeitsweise einer CPU	1 - 5
1.3	Die Programme in einer CPU	1 - 7
1.4	Welche Operanden stehen dem Anwenderprogramm zur Verfügung?.....	1 - 11
1.5	Wieviel Speicher steht dem Anwenderprogramm zur Verfügung?.....	1 - 14
1.6	Wie können Sie beim Programmieren vorgehen?.....	1 - 15
1.7	Programmierwerkzeuge	1 - 18
1.8	Umsetzen von Anwenderprogrammen der CPU 928B für CPU 948.....	1 - 19
2	Anwenderprogramm	2 - 3
2.1	Programmiersprache STEP 5	2 - 4
2.1.1	Darstellungsarten KOP, FUP und AWL.....	2 - 4
2.1.2	Strukturierte Programmierung	2 - 5
2.1.3	STEP-5-Operationen	2 - 6
2.1.4	Zahlendarstellung	2 - 8
2.1.5	STEP-5-Bausteine und deren Ablage im Speicher	2 - 12
2.2	Programm-, Organisations- und Schrittbausteine	2 - 16
2.2.1	Organisationsbausteine für Anwenderschnittstellen	2 - 18
2.2.2	Sonderfunktions-Organisationsbausteine.....	2 - 22
2.3	Funktionsbausteine.....	2 - 23
2.3.1	Aufbau von Funktionsbausteinen.....	2 - 24
2.3.2	Programmieren von Funktionsbausteinen	2 - 26
2.3.3	Aufrufen und Parametrieren von Funktionsbausteinen.....	2 - 28
2.3.4	Spezielle Funktionsbausteine.....	2 - 33
2.4	Datenbausteine	2 - 35
2.4.1	Erstellen von Datenbausteinen.....	2 - 37
2.4.2	Aufschlagen von Datenbausteinen.....	2 - 38
2.4.3	Spezielle Datenbausteine	2 - 41

3	Programmbearbeitung	3 - 3
3.1	Prinzip der Programmbearbeitung	3 - 4
3.2	Programmorganisation	3 - 5
3.3	Speicherung von Programm- und Datenbausteinen	3 - 10
3.4	Bearbeitung des Anwenderprogramms	3 - 11
3.4.1	Begriffsdefinitionen für die Programmbearbeitung	3 - 12
3.5	STEP-5-Operationen mit Beispielen	3 - 15
3.5.1	Grundoperationen	3 - 19
3.5.2	Programmierbeispiele in den Darstellungsarten AWL, KOP und FUP	3 - 34
3.5.3	Ergänzende Operationen	3 - 49
3.5.4	Organisatorische Operationen	3 - 59
3.5.5	Semaphor-Operationen	3 - 75
4	Betriebszustände und Programmbearbeitungsebenen	4 - 3
4.1	Programmbearbeitungsebenen	4 - 4
4.2	Betriebszustand STOP	4 - 9
4.2.1	WEICHER STOP	4 - 9
4.2.2	HARTER STOP	4 - 13
4.2.3	URLÖSCHEN	4 - 14
4.3	Betriebszustand ANLAUF	4 - 16
4.3.1	MANUELLER und AUTOMATISCHER NEUSTART	4 - 17
4.3.2	MANUELLER und AUTOMATISCHER WIEDERANLAUF	4 - 18
4.3.3	Gegenüberstellung NEUSTART und WIEDERANLAUF	4 - 21
4.3.4	NEUSTART MIT GEDÄCHTNIS	4 - 22
4.3.5	Gegenüberstellung NEUSTART und NEUSTART MIT GEDÄCHTNIS	4 - 23
4.3.6	Anwenderschnittstellen für den Anlauf	4 - 24
4.3.7	Erweiterter automatischer Wiederanlauf mit der CPU 948	4 - 27
4.3.8	Unterbrechungen im ANLAUF	4 - 28
4.4	Betriebszustand RUN	4 - 29
4.4.1	Zyklische Programmbearbeitung	4 - 30
4.4.2	Festlegung der zeit- und alarmgesteuerten Programmbearbeitung	4 - 32
4.4.3	Zeitgesteuerte Programmbearbeitung	4 - 33
4.4.4	Alarmgesteuerte Programmbearbeitung	4 - 41
5	Unterbrechungs- und Fehlerbehandlung	5 - 3
5.1	Häufige Fehler im Anwenderprogramm	5 - 4
5.2	Fehlerinformationen	5 - 5
5.3	Vorgehen bei der Fehleranalyse	5 - 8
5.4	Steuerbits und Unterbrechungsstack	5 - 9
5.4.1	Steuerbits	5 - 10
5.4.2	USTACK-Inhalt	5 - 14
5.4.3	Beispiel zur Fehlerdiagnose über USTACK	5 - 19

5.5	Fehlerbehandlung über Organisationsbausteine	5 - 20
5.6	Fehlerursachen und Fehlerreaktionen der CPU.....	5 - 23
5.6.1	OB 19: Aufrufen eines nicht geladenen Codebausteins (KB)	5 - 24
5.6.2	OB 19: Aufschlagen eines nicht geladenen Datenbausteins (KDB).....	5 - 24
5.6.3	OB 23/24, OB 28/29: Quittungsverzug (QVZ).....	5 - 25
5.6.4	OB 25: Adressierfehler (ADF).....	5 - 26
5.6.5	OB 26: Zykluszeitfehler (ZYK).....	5 - 27
5.6.6	OB 27: (Substitutionsfehler SUF)	5 - 28
5.6.7	OB 30: Parityfehler und QVZ beim Anwenderspeicher (PARE)	5 - 28
5.6.8	OB 32: Lade- und Transferfehler (TLAF).....	5 - 29
5.6.9	OB 33: Weckfehler (WEFES/WEFEH).....	5 - 30
5.6.10	OB 34: Fehler bei E DB/EX DX (FEDBX)	5 - 32
5.6.11	OB 35: Kommunikationsfehler	5 - 32
5.6.12	OB 36: Fehler bei Selbsttest	5 - 33
5.7	Selbsttest.....	5 - 34
5.7.1	Übersicht.....	5 - 34
5.7.2	Beschreibung der Testfunktionen.....	5 - 35
5.7.3	Einstellungen	5 - 37
5.7.4	Fehlerbehandlung	5 - 38
6	Integrierte Sonderfunktionen	6 - 3
6.1	Einführung	6 - 4
6.2	OB 121: Systemzeit stellen/lesen	6 - 8
6.3	OB 122: "Alarmer sperren" ein-/ausschalten	6 - 12
6.4	OB 124: STEP-5-Bausteine löschen	6 - 14
6.5	OB 125: STEP-5-Bausteine erzeugen	6 - 17
6.6	OB 126: Prozeßabbilder definieren, übertragen	6 - 20
6.7	OB 129: Batteriezustand ermitteln	6 - 25
6.8	OB 131: AKKU 1, 2, 3 und 4 löschen	6 - 26
6.9	OB 132/133: AKKU-Roll-Up/AKKU-Roll-Down	6 - 27
6.10	OB 141: "Zyklische Weckalarmer einzeln sperren" ein-/ausschalten	6 - 29
6.11	OB 142 "Alarmer gemeinsam verzögern" ein-/ausschalten.....	6 - 32
6.12	OB 143 "Zyklische Weckalarmer einzeln verzögern" ein-/ausschalten.....	6 - 35
6.13	OB 150: Systemzeit stellen/lesen	6 - 38
6.14	OB 151: Zeit für uhrzeitgesteuerten Weckalarm stellen/lesen	6 - 43
6.15	OB 153: Zeit für Verzögerungsalarm stellen/lesen	6 - 50
6.16	OB 180: Variabler Datenbaustein-Zugriff.....	6 - 53
6.17	OB 181: Datenbausteine (DB/DX) testen.....	6 - 57

6.18	OB 182: Datenbereich kopieren	6 - 59
6.19	OB 202 bis 205: Mehrprozessorkommunikation	6 - 62
6.20	OB 222: Zyklusüberwachungszeit neu starten	6 - 63
6.21	OB 223: Anlaufarten vergleichen	6 - 64
6.22	OB 254/255: Datenbausteine kopieren/duplizieren	6 - 65
7	Erweiterter Datenbaustein DX 0	7 - 3
7.1	Anwendung	7 - 4
7.2	Aufbau des DX 0	7 - 5
7.2.1	Beispiel für Eingabe des DX 0	7 - 7
7.3	Parameter für DX 0	7 - 8
7.4	Parametrierungsbeispiele	7 - 12
7.4.1	STEP-5-Programmierung	7 - 12
7.4.2	Parametrierung über PG-Maske	7 - 14
8	Speicherbelegung und Speicherorganisation	8 - 3
8.1	Struktur des Speicherbereiches	8 - 4
8.2	Adreßraumteilung der CPU 948	8 - 5
8.2.1	Adreßraumteilung des System-RAMs	8 - 6
8.2.2	Adreßraumteilung der Peripherie	8 - 8
8.3	Organisation des Anwenderspeichers in der CPU 948	8 - 10
8.3.1	Bausteinköpfe im Anwenderspeicher	8 - 12
8.3.2	Bausteinadreßlisten im Datenbaustein DB 0	8 - 13
8.3.3	BA-/BB-Bereich	8 - 14
8.3.4	BS-/BT-Bereich	8 - 15
8.3.5	Bitbelegung der Systemdatenwörter	8 - 18
8.3.6	Frei adressierbarer Systemdatenbereich	8 - 42
9	Speicherzugriffe über absolute Adressen	9 - 3
9.1	Einführung	9 - 4
9.2	Speicherzugriffe über Adresse in AKKU 1	9 - 8
9.2.1	LIR/TIR: 16-bit-Register indirekt laden/transferieren	9 - 9
9.2.2	Beispiele für Zugriffe auf DW > 255	9 - 15
9.2.3	LDI/TDI: 32-bit-Register indirekt laden/transferieren	9 - 17
9.3	Speicherblöcke transferieren	9 - 19
9.4	Operationen mit dem Basisadressregister (BR-Register)	9 - 22
9.4.1	Transferoperationen zwischen Registern	9 - 23
9.4.2	Zugriffe auf den lokalen Speicher	9 - 24
9.4.3	Zugriffe auf den globalen Speicher	9 - 25
9.4.4	Zugriffe auf den Kachelspeicher	9 - 29

10	Mehrprozessorbetrieb und -kommunikation im AG S5-155U	10 - 3
10.1	Mehrprozessorbetrieb	10 - 4
10.1.1	Wann verwenden Sie den Mehrprozessorbetrieb?	10 - 4
10.1.2	Welche Kommunikationsmechanismen gibt es?	10 - 4
10.1.3	Daten über Koppelmerker austauschen	10 - 5
10.1.4	Daten über Hantierungsbausteine austauschen	10 - 8
10.1.5	Was müssen Sie für den Mehrprozessorbetrieb programmieren?	10 - 9
10.1.6	Wie erstellen Sie den Datenbaustein DB 1?	10 - 9
10.1.7	Was müssen Sie über den Anlauf im Mehrprozessorbetrieb wissen?	10 - 13
10.1.8	Was Sie über den Testbetrieb wissen sollten	10 - 14
10.2	Mehrprozessorkommunikation	10 - 15
10.2.1	Einführung	10 - 15
10.2.2	Wie Sender und Empfänger identifiziert werden	10 - 16
10.2.3	Warum Daten zwischengespeichert werden	10 - 17
10.2.4	Wie der Zwischenspeicher abgearbeitet und verwaltet wird	10 - 18
10.2.5	Was Sie beim System-Anlauf beachten müssen	10 - 21
10.2.6	Was Sie beim Aufrufen der Kommunikations-OBs beachten müssen	10 - 22
10.2.7	Wie parametrieren Sie die Kommunikations-OBs?	10 - 23
10.2.8	Wie können Sie die Ausgangsparameter auswerten?	10 - 24
10.3	Laufzeiten der Kommunikations-OBs	10 - 31
10.4	Funktion INITIALISIEREN (OB 200)	10 - 33
10.4.1	Funktion	10 - 33
10.4.2	Aufrufparameter	10 - 35
10.4.3	Eingangsparameter	10 - 35
10.4.4	Ausgangsparameter	10 - 38
10.5	Funktion SENDEN (OB 202)	10 - 40
10.5.1	Funktion	10 - 40
10.5.2	Aufrufparameter	10 - 40
10.5.3	Eingangsparameter	10 - 40
10.5.4	Ausgangsparameter	10 - 42
10.6	Funktion SENDE-TEST (OB 203)	10 - 45
10.6.1	Funktion	10 - 45
10.6.2	Aufrufparameter	10 - 45
10.6.3	Eingangsparameter	10 - 45
10.6.4	Ausgangsparameter	10 - 45
10.7	Funktion EMPFANGEN (OB 204)	10 - 47
10.7.1	Funktion	10 - 47
10.7.2	Aufrufparameter	10 - 47
10.7.3	Eingangsparameter	10 - 47
10.7.4	Ausgangsparameter	10 - 48
10.8	Funktion EMPFANGS-TEST (OB 205)	10 - 51
10.8.1	Funktion	10 - 51
10.8.2	Aufrufparameter	10 - 51
10.8.3	Eingangsparameter	10 - 51
10.8.4	Ausgangsparameter	10 - 51

10.9	Anwendungen	10 - 53
10.9.1	Aufruf der Sonderfunktions-OB über Funktionsbausteine	10 - 53
10.9.2	Übertragen von Datenbausteinen	10 - 60
10.9.3	Erweiterung des Koppelmerkerbereichs	10 - 66
11	PG-Schnittstellen und -Funktionen	11 - 3
11.1	Übersicht	11 - 4
11.2	PG-Funktionen	11 - 5
11.2.1	Auskunft	11 - 6
11.2.2	Inbetriebnahme	11 - 7
11.2.3	Programmtest	11 - 8
11.3	Serielle Kopplung PG – AG über 1. oder 2. serielle Schnittstelle	11 - 16
11.4	Parallelbetrieb von zwei seriellen PG-Schnittstellen	11 - 17
11.4.1	Inbetriebnahme	11 - 19
11.4.2	Betrieb	11 - 19
11.4.3	Ablauf bei bestimmten Betriebsfällen	11 - 21
11.5	PG-Funktionen über S5-Bus	11 - 27
11.5.1	Anwendung	11 - 27
11.5.2	Arbeitsweise der PG-Funktionen über S5-Bus	11 - 29
11.5.3	Inbetriebnahme	11 - 31
11.5.4	Anzeige von Störungen	11 - 35
12	Anhang	12 - 3
13	Verzeichnisse	13 - 1
	Abkürzungsverzeichnis	A - 1
	Stichwortverzeichnis	Index - 1

Einführung

1

Inhalt von Kapitel 1

1.1	Anwendungsbereich S5-155U mit CPU 948.	1 - 4
1.2	Typische Arbeitsweise einer CPU	1 - 5
1.3	Die Programme in einer CPU	1 - 7
1.4	Welche Operanden stehen dem Anwenderprogramm zur Verfügung?	1 - 11
1.5	Wieviel Speicher steht dem Anwenderprogramm zur Verfügung?	1 - 14
1.6	Wie können Sie beim Programmieren vorgehen?	1 - 15
1.7	Programmierwerkzeuge	1 - 18
1.8	Umsetzen von Anwenderprogrammen der CPU 928B für CPU 948	1 - 19

Einführung

1

Zielsetzung des Handbuchs

Dieses Handbuch soll Anwendern, die bereits Grundkenntnisse im Programmieren von Automatisierungsgeräten besitzen und die CPU 948 im Automatisierungsgerät S5-155U einsetzen wollen, spezielle Kenntnisse für das Programmieren der CPU 948 vermitteln.

Lesern, die diese Grundkenntnisse noch nicht besitzen, sei die Lektüre eines Einführungsbuches in die Programmiersprache STEP 5 /3/ oder die Teilnahme an einem Kurs in einem unserer Trainingscenter empfohlen. SIEMENS bietet Ihnen umfangreiche Schulungsmöglichkeiten zu SIMATIC S5. Nähere Informationen dazu erhalten Sie bei Ihrer SIEMENS-Geschäftsstelle.

Inhalt von Kapitel 1

Kapitel 1 gibt Ihnen Hinweise zur Benutzung des Handbuchs und informiert Sie über den Anwendungsbereich des Automatisierungsgerätes S5-155U mit der CPU 948 und seine Gerätestruktur.

Es erklärt die typische Arbeitsweise einer CPU und den Aufbau eines CPU-Programms.

Sie erhalten ferner einige Vorschläge, wie Sie beim Programmieren vorgehen können und erfahren, welche für die Programmierung wichtigen Kenndaten die CPU 948 hat.

Wenn Sie bereits mit der CPU 946/947 gearbeitet haben und wissen möchten, welche Unterschiede die CPU 948 dazu hat, können Sie dies in Abschnitt 1.8 nachlesen.

Kapitel 1 informiert Sie ferner über Unterschiede der Ausgaben A01 und A02 der CPU 948 und gibt Ihnen Hinweise, was Sie beim Umsetzen von "928B"-Programmen für die CPU 948 beachten müssen.

1.1 Anwendungsbereich S5-155U mit CPU 948

Einordnung in die Familie

Das Automatisierungsgerät S5-155U gehört zur Familie der speicherprogrammierbaren Steuerungen SIMATIC S5. Es ist mit der CPU 948 das leistungsfähigste Mehrprozessorgehäuse zur Prozeßautomatisierung (Steuern, Melden, Überwachen, Regeln, Protokollieren). Aufgrund seiner Modularität und sehr großen Leistungsfähigkeit kann es sowohl für Steuerungen mittleren bis sehr großen Umfangs als auch für komplexe Automatisierungsaufgaben in der Dispositions- und Prozeßleitebene eingesetzt werden.

Eignung

Das AG S5-155U mit der CPU 948 eignet sich besonders gut für:

- Aufgaben, die sehr schnelle Bit- und Wortverarbeitung sowie kurze Reaktionszeiten erfordern, d. h. bei sehr schnellem Steuern und Regeln.
Beispiele dafür sind schnelle Vorgänge im Maschinenbau (Flaschenabfüllanlage, Verpackungsmaschinen oder ähnliche Anlagen) und in der Automobilindustrie.
- Aufgaben, die ein sehr großes Speichervolumen und schnelle Zugriffszeiten erfordern z. B. in der Automobilindustrie, in der Verfahrenstechnik und im Anlagenbau.
- Aufgaben, die eine schnelle Kommunikation mit weiteren im AG eingesetzten CPUs im Mehrprozessorbetrieb sowie mit CP-Baugruppen erfordern (z. B. bei Anbindung an Bussysteme, an Leit-rechner, bei Visualisierungen, Bedienen und Beobachten).
- Komplexe Aufgaben, die effizient und übersichtlich mit den Hochsprachen C und SCL gelöst werden können.

1.2 Typische Arbeitsweise einer CPU

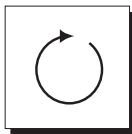
Arbeitsweise einer CPU

In einer CPU gibt es folgende Arbeitsweisen :



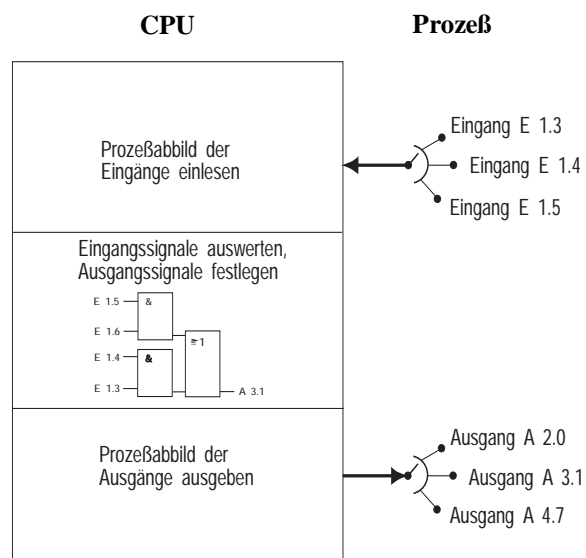
zyklische Bearbeitung

Sie stellt den Hauptanteil aller Vorgänge in der CPU. Wie ihr Name schon ausdrückt, wiederholt sich in einem endlosen Zyklus ständig die gleiche Bearbeitungsfolge.



Die zyklische Bearbeitung unterteilt sich in drei Hauptphasen :

Phase	Ablauf
1	Alle der CPU zugeordneten Eingabebaugruppen werden vom Systemprogramm abgefragt und die eingelesenen Werte im Prozeßabbild der Eingänge (PAE) zwischengespeichert.
2	Die im PAE enthaltenen Werte werden durch das Anwenderprogramm verarbeitet und die auszugebenden Werte in das Prozeßabbild der Ausgänge (PAA) eingetragen.
3	Die im Prozeßabbild der Ausgänge enthaltenen Werte werden vom Systemprogramm an die der CPU zugeordneten Ausgabebaugruppen ausgegeben.



zeitgesteuerte Bearbeitung



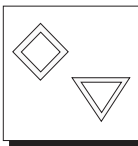
Für Prozesse, die in konstanten Zeitabschnitten Steuersignale benötigen, können Sie neben dem zyklischen Ablauf **zeitgesteuert** bestimmte Aufgaben bearbeiten z. B. zeitunkritische Überwachungsfunktionen im Sekundenraster.

alarmgesteuerte Bearbeitung



Einem Prozeßsignal, auf das besonders schnell reagiert werden muß, ordnen Sie einen **alarmgesteuerten** Bearbeitungsabschnitt zu. Sie können z. B. mit einem Systeminterrupt, der über eine alarmerzeugende Baugruppe ausgelöst wird, eine spezielle Bearbeitungsfolge in Ihrem Programm aktivieren.

Bearbeitung nach Priorität



Die o.g. Bearbeitungsarten werden von der CPU nach Wichtigkeitsgrad behandelt. Dieser Wichtigkeitsgrad heißt **Priorität**.

Da auf ein Zeit- oder Alarmereignis schnell reagiert werden muß, unterbricht die CPU die zyklische Bearbeitung, um ein Zeit- oder Alarmereignis zu behandeln. Die zyklische Bearbeitung hat daher die niedrigste Priorität

Ob die zeitgesteuerte Bearbeitung wichtiger ist als die alarmgesteuerte Bearbeitung, hängt u. U. von der jeweiligen Aufgabenstellung ab. Daher können die Prioritäten der zeit- und alarmgesteuerten Bearbeitung bei der CPU 948 eingestellt werden.

1.3 Die Programme in einer CPU

Das in jeder CPU vorhandene Programm unterteilt sich in

- das **Systemprogramm**
- und
- das **Anwenderprogramm.**

Systemprogramm

Das Systemprogramm organisiert alle Funktionen und Abläufe der CPU, die nicht mit einer spezifischen Steuerungsaufgabe verbunden sind (siehe Bild 1-2).

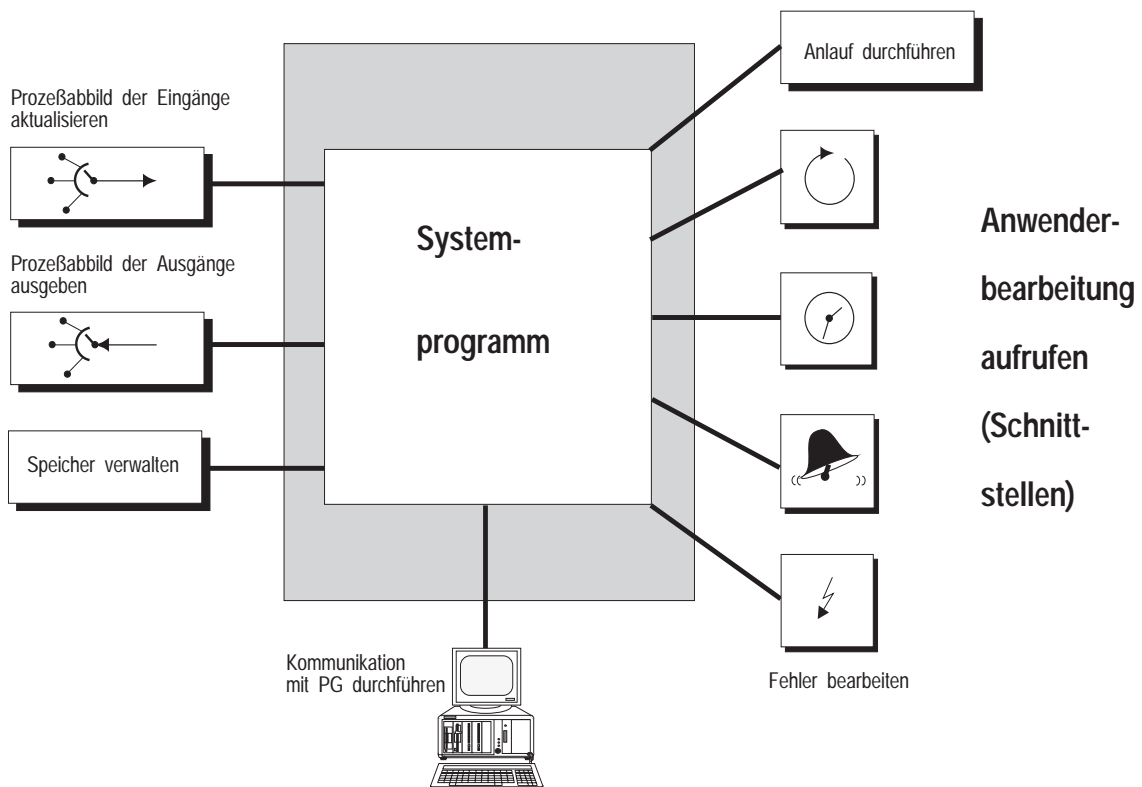


Bild 1-1 Aufgaben des Systemprogramms

Aufgaben

Zu den Aufgaben gehören : ¹⁾

- Neustart und Wiederanlauf,
- das Aktualisieren des Prozeßabbildes der Eingänge und die Ausgabe des Prozeßabbildes der Ausgänge,
- die Aufrufe der zyklischen, zeit- und alarmgesteuerten Programme,
- das Erkennen und Behandeln von Fehlern,
- die Speicherverwaltung,
- die Kommunikation mit dem Programmiergerät.

Anwenderschnittstellen

Als Anwender haben Sie die Möglichkeit, über spezielle Schnittstellen zum Systemprogramm das Verhalten der CPU in bestimmten Betriebs- und Fehlerfällen zu beeinflussen.

Speicherung des Systemprogramms

Das Systemprogramm wird nach Einschalten der AG-Netzspannung (NETZ EIN) aus dem EPROM-Speicher in das interne Betriebssystem-RAM geladen.

Voreinstellung des Systemprogramms

Die nachfolgenden Kapitel außer Kapitel 7 beschreiben das **voreingestellte Systemverhalten** bei Reaktionen auf Prozeßereignisse oder Fehler. So geht z. B. entsprechend der Voreinstellung die CPU in den Stoppzustand, wenn bei einem Befehlscode-Fehler der zugeordnete Fehler-Organisationsbaustein nicht geladen ist.

Voreinstellung modifizieren

Sie können dieses Systemverhalten durch Parametrieren des Datenbausteins DX 0 modifizieren.
Wie sich **nach** einer solchen **Modifizierung** das System verhält, wird **in Kapitel 7** beschrieben!

¹⁾ Im Betrieb mit mehreren CPUs (Mehrprozessorbetrieb) kommen noch weitere Aufgaben hinzu.

Anwenderprogramm**Aufgaben**

Das Anwenderprogramm enthält alle Funktionen, die zur Bearbeitung einer **spezifischen Steuerungsaufgabe** erforderlich sind. Diese Funktionen lassen sich bei einer groben Aufteilung direkt den Schnittstellen zuordnen, die das Systemprogramm für die unterschiedlichen Bearbeitungsarten zur Verfügung stellt:

Bearbeitungsart	Aufgabe
Neustart und Wiederanlauf	Voraussetzungen schaffen, daß die übrige Bearbeitung bei einem Neubeginn der Steuerung oder nach einem Wiederanlauf auf einem definierten Zustand aufsetzen kann (z. B. Signale mit einem bestimmten Wert vorbesetzen)
zyklische Bearbeitung	sich ständig wiederholende Signalverarbeitung (z. B. Binärsignale verknüpfen, Analogwerten einlesen und auswerten, Binärsignale für die Ausgabe festlegen, Analogwerte ausgeben).
zeitgesteuerte Bearbeitung	spezielle, zeitabhängige Bearbeitungen mit folgenden Zeitbedingungen: <ul style="list-style-type: none"> - schneller als der durchschnittliche Zyklus), - in einem Zeitraster, das größer ist als die durchschnittliche Zykluszeit ist, - zu einem bestimmten, einstellbaren Zeitpunkt.
alarmgesteuerte Bearbeitung	spezielle, schnelle Reaktionen auf bestimmte Prozeßsignale
Fehlerreaktion	Störungen des normalen Programmablaufs bearbeiten

1.4 Welche Operanden stehen dem Anwenderprogramm zur Verfügung?

Die CPU 948 stellt Ihnen für das Programmieren folgende Operandenbereiche zur Verfügung:

- Prozeßabbild und Peripherie
- Merker (M-Merker und S-Merker)
- Zeiten/Zähler
- Datenbausteine

Prozeßabbild der Ein- und Ausgänge PAE/PAA

Kennzeichen	Größe
Auf das Prozeßabbild kann das Anwenderprogramm sehr schnell zugreifen und zwar auf folgende Datentypen: - Einzelbits, - Bytes, - Wörter, - Doppelwörter.	je 128 byte für Ein- und Ausgänge

Peripheriebereich (P-Bereich)

Kennzeichen	Größe
Das Anwenderprogramm kann direkt über den S5-Bus auf die Peripheriebaugruppen zugreifen. Folgende Datentypen sind möglich: - Bytes, - Wörter.	je 256 byte für Ein- und Ausgänge

Erweiterter Peripheriebereich (Q-Bereich)

Kennzeichen	Größe
Das Anwenderprogramm kann direkt über den S5-Bus auf die Peripheriebaugruppen zugreifen. Folgende Datentypen sind möglich: - Bytes, - Wörter.	je 256 byte für Ein- und Ausgänge

M-Merker

Kennzeichen	Größe
<p>Der Merkerbereich ist ein Speicherbereich, auf den das Anwenderprogramm mit entsprechenden Operationen sehr schnell zugreifen kann. Der Merkerbereich sollte bevorzugt für oft benötigte Arbeitsdaten verwendet werden.</p> <p>Auf folgende Datentypen kann zugegriffen werden:</p> <ul style="list-style-type: none"> - Einzelbits, - Bytes, - Wörter, - Doppelwörter. <p>Einzelne Merkerbytes können als Koppelmerker zum Datenaustausch zwischen den CPUs im Mehrprozessorbetrieb (siehe Kapitel 10) genutzt werden. Koppelmerker werden vom Systemprogramm am Zyklusende über einen Zwischenspeicher im Koordinator bzw. CP/IP aktualisiert.</p>	<p>2048 bit</p>

S-Merker (erweiterter Merkerbereich)

Kennzeichen	Größe
<p>Die CPU 948 enthält einen zusätzlichen Merkerbereich, den S-Merkerbereich. Hierauf kann das Anwenderprogramm wie auf die M-Merker sehr schnell zugreifen.</p> <p>S-Merker lassen sich jedoch nicht als Aktualoperanden bei Funktionsbaustein-Aufrufen und auch nicht als Koppelmerker für Datenaustausch zwischen den CPUs benutzen. Auf die S-Merker können auch nicht die Bitttestoperationen der CPU 948 angewendet werden.</p> <p>Voraussetzung zu ihrer Benutzung ist die PG-Systemsoftware "S5-DOS" ab Version 3.0 oder "S5-DOS/MT" ab Version 1.0.</p>	<p>32 768 bit</p>

Zeiten T

Kennzeichen	Größe
Zeitzellen werden vom Anwenderprogramm mit einem Zeitwert zwischen 10ms und 9990s geladen und durch eine Startoperation von diesem Vorgabewert aus so lange im vorgewählten Zeitraster dekrementiert, bis sie den Wert Null erreicht haben.	256 Zeitzellen

Zähler Z

Kennzeichen	Größe
Zählzellen werden vom Anwenderprogramm mit einem Anfangswert (max. 999) geladen und hinauf- bzw. heruntergezählt.	256 Zähler

Datenwörter im aktuellen Datenbaustein

Kennzeichen	Größe
Ein Datenbaustein enthält Konstanten und/oder Variablen im Byte-, Wort oder Doppelwortformat. Mit STEP-5-Operationen können Sie immer auf den "aktuellen" Datenbaustein zugreifen (siehe Abschnitt 2.4.2) Auf folgende Datentypen kann zugegriffen werden: <ul style="list-style-type: none"> - Einzelbits, - Bytes, - Wörter, - Doppelwörter. 	256 Wörter 1)

1) bei Datenbausteinen, die größer als 256 Wörter lang sind, können Sie auf die Datenwörter mit der Nummer > 255 nur mit Operationen für absolute Speicherzugriffe (siehe Kapitel 9) zugreifen.

1.5 Wieviel Speicher steht dem Anwenderprogramm zur Verfügung?

Die CPU 948 besitzt zur Speicherung von Code- und Datenbausteinen ausschließlich den Anwenderspeicher im internen RAM.

Sie können die CPU 948 in zwei Versionen des Anwenderspeichers beziehen:

- Version 1: mit 640 kbyte,
- Version 2: mit 1 664 kbyte.

1.6 Wie können Sie beim Programmieren vorgehen?

Wenn Sie ein erfahrener Anwender sind, haben Sie bestimmt schon Ihre eigene Methode der Programmerstellung gefunden; Sie brauchen dann diesen Abschnitt nicht zu lesen.

Weniger erfahrenen Lesern seien folgende Tips gegeben, wie sie beim Entwerfen, Programmieren, Testen und Inbetriebnehmen ihres STEP-5-Programms vorgehen können.

Realisierungsabschnitte

Die Realisierung eines STEP-5-Steuerprogramm läßt sich in drei Abschnitte einteilen:

Abschnitt	Tätigkeit
1	technologische Aufgabenstellung festlegen
2	Programm entwerfen
3	Programm erstellen, testen und in Betrieb nehmen

rekursives Verfahren

In der Praxis wird sich erweisen, daß Sie einzelne Schritte wiederholen müssen ("Rekursives Verfahren"), z. B. wenn Sie bei der Verfeinerung der Aufgabenstellung feststellen, daß noch mehr Signale benötigt werden.

Schritt 1

Technologische Aufgabenstellung festlegen:

Schritt	Tätigkeit
1	Erstellen Sie ein grobes Blockdiagramm über die Steuerungsaufgaben Ihres Prozesses.
2	Erstellen Sie eine Liste der für die Aufgabe benötigten Ein- und Ausgangssignale.
3	Verfeinern Sie das Blockdiagramm, indem Sie den einzelnen Blöcken die Signale und evtl. notwendige Zeitbedingungen und/oder Zählerstände zuordnen.

Schritt 2

Programm entwerfen:

Schritt	Tätigkeit
1	Entwerfen Sie mit Hilfe des verfeinerten Blockdiagramms die notwendigen Bearbeitungsarten Ihres Programms (zyklische Bearbeitung, zeitgesteuerte Bearbeitung usw.) und benennen Sie die dazu verwendeten OBs.
2	Teilen Sie die Bearbeitungsarten in technologische und/oder funktionale Blöcke ein.
3	Prüfen Sie, ob Sie die Blöcke einem Programm- oder Funktionsbaustein zuordnen können, und benennen Sie die zu verwendenden Bausteine (PB x, FB y usw.)
4	Klären Sie, welche Zeiten, Zähler und Daten bzw. Ergebnisspeicher Sie benötigen.
5	Legen Sie die Aufgaben für jeden vorgesehenen Codebaustein und die Daten für evtl. benötigte Merker und Datenbausteine fest. Zeichnen Sie für die Codebausteine Ablaufdiagramme.

Hinweise zum Umfang der zyklischen Bearbeitung

Beachten Sie bei der Festlegung der Bearbeitungsarten folgende Randbedingungen.

- Der Zyklus muß ausreichend schnell ablaufen. Die Prozeßzustände dürfen sich nicht schneller ändern, als die CPU darauf reagieren kann. Ansonsten könnte der Prozeß außer Kontrolle geraten.
- Als maximale Reaktionszeit müssen Sie die doppelte Zykluszeit berücksichtigen.
Die Zykluszeit wird bestimmt durch die zyklische Bearbeitung des Systemprogramms und durch Art und Umfang des Anwenderprogramms. Sie ist oftmals nicht konstant, da das zyklische Anwenderprogramm u. U. mehrfach unterbrochen wird durch aufgerufene zeit- und alarmgesteuerte Programme.

Schritt 3

Programm erstellen, testen und in Betrieb nehmen:

Schritt	Tätigkeit
1	Legen Sie die Darstellungsart für die Codebausteine fest (KOP, FUP oder AWL, lesen Sie dazu bitte in Kapitel 2). Beachten Sie, daß Funktionsbausteine nur in der Darstellungsart AWL erstellt werden können.
2	Programmieren Sie alle Code- und Datenbausteine (die Hantierung dazu entnehmen Sie bitte Ihrem STEP-5-Handbuch).
3	Nehmen Sie die Bausteine nacheinander in Betrieb (dazu müssen Sie für jeden Teilschritt u. U. einen anderen OB 1 programmieren, der die betreffenden Codebausteine aufruft): 1a: Laden Sie den/die Baustein(e) 1b: Testen Sie den/die Baustein(e) (Die erforderlichen Hantierungen entnehmen Sie bitte Ihrem STEP-5-Handbuch und dem Kapitel 11.)
4	Wenn Sie sicher sind, daß alle Codebausteine richtig ablaufen und alle Daten richtig berechnet und abgespeichert werden, können Sie Ihr gesamtes Programm in Betrieb nehmen.

Hinweis zur Teststrategie

Zu welchem Zeitpunkt Sie Ihr Programm das erste Mal im "heißen" Prozeßbetrieb laufen lassen, d. h. mit echten Ein- und vor allem Ausgangssignalen, müssen Sie selbst bzw. ein Team von Fachleuten entscheiden.

Je komplexer der Prozeß ist und je schwerer das Sicherheitsrisiko wiegt, um so größer muß die Sorgfalt bei der Inbetriebnahme sein.

1.7 Programmierwerkzeuge

Einsetzbare PGs

Für die Erstellung Ihres Anwenderprogramms stehen Ihnen die Programmiergeräte PG 685, PG 710, PG 730, PG 750 und PG 770 zur Verfügung. Leistung und Eigenschaften dieser Geräte können Sie dem Katalog ST 59 /9/ entnehmen.

Hinweis

Wenn Sie in Ihrer Automatisierungssoftware den **gesamten Leistungsumfang** der CPU 948 ausschöpfen wollen (insbesondere DX-0-Maske, Maske "Ausgabe USTACK", die Anzeige bei der PG-Funktion "Speicherausbau", sowie PG-Funktionen über S5-Rückwandbus), so benötigen Sie die PG-Systemsoftware "**STEP 5/ST**" ab Version 6.3 oder "**STEP 5/MT**" ab Version 6.0 + jeweils die "Deltadiskette CPU 948" und ein PG 7xx..

Einsetzbare Software

Anwenderprogramme für SIMATIC-S5-Automatisierungsgeräte können Sie erstellen

- in der Programmiersprache **STEP 5**,

Dazu benutzen Sie das Programmierpaket STEP 5 zusammen mit der Systemsoftware STEP 5/ST bzw. STEP 5/MT (Beschreibung siehe /3/ – Literaturhinweise),

oder

- in einer höheren Programmiersprache:.

Wenn Sie gewohnt sind, Programme in einer höheren Programmiersprache zu schreiben, können Sie bei der CPU 948 Ihr STEP-5-Programm auch in folgenden Sprachen formulieren:

- **SCL** (siehe /12/ – Literaturhinweise, der SCL-Compiler ist in der PG-Software "S5-DOS/MT" ab Version 6 enthalten.)

oder

- **C mit S5-C-Compiler** (siehe /13/ – Literaturhinweise).

Programme für Ablaufsteuerungen können Sie auch in einer grafischen Darstellung mit dem Programmierpaket **GRAPH 5** erstellen (Beschreibung siehe /4/ – Literaturhinweise).

Je nach Aufgabenstellung können Sie in Ihr Anwenderprogramm auch fertige Standard-Funktionsbausteine einbeziehen, deren Leistung und Eigenschaften im Katalog ST 57 /11/ aufgeführt sind.

1.8 Umsetzen von Anwenderprogrammen der CPU 928B für CPU 948

Im folgenden Abschnitt erhalten Sie einige Hinweise, was Sie beachten müssen, wenn Sie Anwenderprogramme, die für die CPU 928B erstellt wurden, für den Ablauf auf CPU 948 umsetzen wollen.

Operationen

Bei folgenden Operationen müssen Sie auf Unterschiede in der Ausführung bzw. Handhabung (u. a. wegen unterschiedlicher Speicherteilung) beachten:

Operationen	CPU 928B	CPU 948
AS/AF (Alarme sperren/freigeben)	Es werden alle Prozeßalarme gesperrt bzw. freigegeben.	Es werden nur die Prozeßalarme über Eingangsbyte EB 0 gesperrt bzw. freigegeben. Verwenden Sie statt der Operationen die Sonderfunktions-OBs OB 122 bzw. OB 142.
LIR/TIR	Es werden 16-bit-breite Adressen verwendet.	Es werden 20-bit-breite Adressen verwendet. Eine Anpassung ist erforderlich.
Blocktransferoperation TNB	Es werden 16-bit-breite Adressen verwendet.	Die Operation ist nicht vorhanden. Verwenden Sie TNW für den Blocktransfer vom 8-bit- in den 8-bit-Bereich.
Blocktransferoperation TNW	<ul style="list-style-type: none"> - Es werden 16-bit-breite Adressen verwendet. - Blocktransfers vom 8-bit- in den 16-bit-Bereich und umgekehrt sind möglich. 	<ul style="list-style-type: none"> - Es werden 20-bit-breite Adressen verwendet. Eine Anpassung ist erforderlich. - Es sind nur Blocktransfers vom 8-bit in den 8-bit-Bereich und vom 16-bit- in den 16-bit-Bereich mit TNW möglich. Verwenden Sie <ul style="list-style-type: none"> - für den Blocktransfer vom 8-bit- in den 16-bit-Bereich die Operation TXB, - für den Blocktransfer vom 16-bit- in den 8-bit-Bereich die Operation TXW. (TXB und TXW sind bei CPU 928B nicht vorhanden.)
Alle Operationen mit dem BR-Register	Das BR-Register ist 20 bit breit.	Das BR-Register ist 32 bit breit. Eine Anpassung ist erforderlich

Zeitzellenbearbeitung

CPU 928B	CPU 948
Die Zeitzellen werden bereits im ANLAUF aktualisiert.	Die Zeitzellen werden erst im RUN aktualisiert. (Grund: Kompatibilität zu CPU 946/947)

FB 0 als Zyklusbaustein

CPU 928B	CPU 948
Wenn kein Zyklusbaustein OB 1 vorhanden ist, ruft das Systemprogramm zyklisch den Baustein FB 0 auf, falls er geladen ist.	Für die zyklische Bearbeitung kann nur der Baustein OB 1 verwendet werden. Falls Sie den FB 0 programmiert haben, erstellen Sie einen OB 1, in dem der FB 0 aufgerufen wird.

Vorbereitung von Prioritäten

CPU 928B	CPU 948
Prozeßalarme sind höherprior als Weckalarme.	Weckalarme sind höherprior als Prozeßalarme über EB 0 bzw. Systeminterrupts. Die Priorität können Sie durch entsprechende DX-0-Parametrierung ändern.

Datenbaustein DB 0 (Bausteinadreßliste)

CPU 928B	CPU 948
Die Bausteinadreßliste enthält die direkten Anfangsadressen der Bausteine.	Die Bausteinadreßliste enthält die Segmentadressen der Bausteine. Um die Anfangsadresse eines Bausteins zu erhalten, muß dessen Segmentadresse um 4 Bitstellen nach links verschoben werden.

Datenbaustein DX 0

Sie müssen einen neuen Datenbaustein DX 0 erstellen (siehe Kapitel 7), da der DX 0 für CPU 928B einen anderen Aufbau und andere Einstellmöglichkeiten hat.

Benutzung des BT-Bereichs

Bei der CPU 928B wird der BT-Bereich vom Systemprogramm nicht benutzt, bei der CPU 948 wird er teilweise durch die Hantierungsbausteine benutzt.

Sie dürfen den BT-Bereich nur dann für Ihr Anwenderprogramm benutzen, wenn Sie keine Standard-FB und keine PG-Funktionen über SINEC-H1 und den S5-Bus verwenden.

Organisationsbausteine

Anzahl und Funktion der Fehler- und Sonderfunktions-OBs sind bei CPU 928B und CPU 948 unterschiedlich:

Fehler-OBs

Folgende Fehler-OBs der CPU 948 verhalten sich anders als die gleichnamigen der CPU 928B:

OB	Funktion	Fehlerkennungen
OB 19 OB 26 OB 27	gleich wie bei CPU 928	unterschiedlich zur CPU 928B
OB 28 OB 29 OB 30 OB 31	unterschiedlich zur Funktion bei CPU 928B	–

Sonderfunktions-OBs

OB	Bemerkung
OB 110 OB 152 OB 160 bis 163 OB 170 OB 190 bis 193 OB 216 bis 218 OB 220 und 221 OB 224 OB 226 bis 228 OB 240 bis 242 OB 250 und 251	diese OBs sind bei der CPU 948 nicht vorhanden
OB 111 OB 112 OB 113 OB 120 OB 121 OB 122 OB 123	Der OB wird bei der CPU 948 ersetzt durch: OB 131 OB 132 OB 133 OB 122 OB 141 OB 142 OB 143

OB	Bemerkung
OB 122	Die Parametrierung ist anders als die des OB 122 bei der CPU 928B (Grund: Kompatibilität zur CPU 946/947).
OB 180	Im Unterschied zur CPU 928B kann das Zugriffsfenster bei der CPU 948 nur um Vielfaches von 16 verschoben werden.
OB 200 OB 202 bis 205 (Mehrprozessorkommunikation)	Im Unterschied zur CPU 928B wird von diesen OBs der CPU 948 der AKKU 4 verändert!

R64-Regelungssoftware

Die R64-Regelungssoftware ist auf der CPU 948 nicht ablauffähig.

Standard-FBs

In der Regel müssen Sie die bei der CPU 928B verwendeten Standard-FBs (z. B. für IPs) gegen die für die CPU 948 austauschen. Eine Ausnahme bilden die **HTBs**. Diese können Sie von der CPU 928B übernehmen (siehe Abschnitt 1.8.1).

Anwenderprogramm

2

Inhalt von Kapitel 2

2.1	Programmiersprache STEP 5	2 - 4
2.1.1	Darstellungsarten KOP, FUP und AWL	2 - 4
2.1.2	Strukturierte Programmierung	2 - 5
2.1.3	STEP-5-Operationen	2 - 6
2.1.4	Zahlendarstellung	2 - 8
2.1.5	STEP-5-Bausteine und deren Ablage im Speicher	2 - 12
2.2	Programm-, Organisations- und Schrittbausteine	2 - 16
2.2.1	Organisationsbausteine für Anwenderschnittstellen	2 - 18
2.2.2	Sonderfunktions-Organisationsbausteine	2 - 22
2.3	Funktionsbausteine	2 - 23
2.3.1	Aufbau von Funktionsbausteinen	2 - 24
2.3.2	Programmieren von Funktionsbausteinen	2 - 26
2.3.3	Aufrufen und Parametrieren von Funktionsbausteinen	2 - 28
2.3.4	Spezielle Funktionsbausteine	2 - 33
2.4	Datenbausteine	2 - 35
2.4.1	Erstellen von Datenbausteinen	2 - 37
2.4.2	Aufschlagen von Datenbausteinen	2 - 38
2.4.3	Spezielle Datenbausteine	2 - 41

Anwenderprogramm

2

Im nachfolgenden Kapitel erfahren Sie, aus welchen Komponenten sich ein STEP-5-Anwenderprogramm für die CPU 948 zusammensetzt und wie es strukturiert werden kann.

2.1 Programmiersprache STEP 5

Mit der Programmiersprache STEP 5 setzen Sie die Ihnen vorliegenden Automatisierungsaufgaben in Programme um, die in den SIMATIC-S5-Automatisierungsgeräten ablaufen. In STEP 5 können Sie sowohl einfache binäre Funktionen als auch komplexe digitale Funktionen und arithmetische Operationen einschließlich Gleitpunktarithmetik programmieren.

Operationsarten

Der **Befehlsumfang** der Programmiersprache STEP 5 gliedert sich in

Grundoperationen:

- in allen Code-Bausteinen anwendbar,
- Darstellungsarten Kontaktplan (KOP), Funktionsplan (FUP), Anweisungsliste (AWL).

Ergänzende Operationen und Systemoperationen:

- nur in Funktionsbausteinen anwendbar,
- Darstellungsart nur Anweisungsliste (AWL),
- Systemoperationen: nur für Anwender mit sehr guten Systemkenntnissen.

2.1.1 Darstellungsarten KOP, FUP und AWL

Beim Programmieren in STEP 5 können Sie für jeden einzelnen Code-Baustein zwischen den drei Darstellungsarten Kontaktplan (KOP), Funktionsplan (FUP) und Anweisungsliste (AWL) wählen, so daß die Programmiermethode dem jeweiligen Anwendungsfall angepaßt werden kann.

Der von den Programmiergeräten (PG) erzeugte Maschinencode MC 5 ist bei den drei Darstellungsarten identisch.

Wenn Sie beim Programmieren in STEP 5 bestimmte Regeln berücksichtigen (siehe /3/), kann das PG Ihr Anwenderprogramm von einer Darstellungsart in jede andere übersetzen!

Grafische Darstellung oder Liste mit Anweisungen

Während Sie mit den Darstellungsarten FUP und KOP die Möglichkeit haben, Ihr STEP-5-Programm grafisch darzustellen, werden in der Anweisungsliste die einzelnen STEP-5-Befehle aufgelistet.

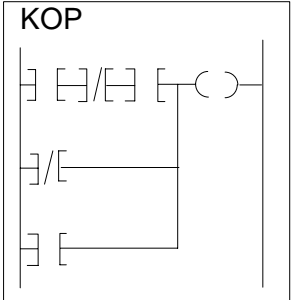
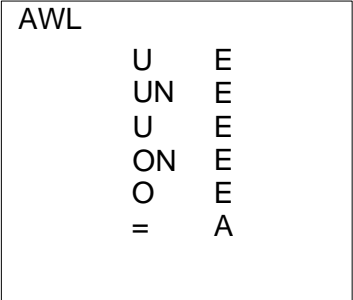
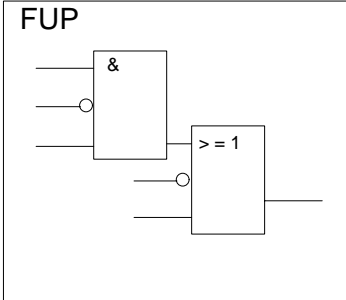
KONTAKTPLAN	ANWEISUNGSLISTE	FUNKTIONSPLAN
<p>Programmieren mit grafischen Symbolen wie Stromlaufplan</p> <p>entspricht DIN 19239</p> 	<p>Programmieren mit mnemotechnischen Abkürzungen der Funktionsbezeichnungen</p> <p>entspricht DIN 19239</p> 	<p>Programmieren mit grafischen Symbolen</p> <p>entspricht IEC 117-15 DIN 40700 DIN 40719 DIN 19239</p> 

Bild 2-1 Darstellungsarten der Programmiersprache STEP 5

Grafische Darstellung von Ablaufsteuerungen

GRAPH 5 /4/ ist eine Programmiersprache zur graphischen Darstellung von Ablaufsteuerungen. Sie ist den Darstellungsarten KOP, FUP und AWL übergeordnet. Das mit GRAPH 5 geschriebene Programm in grafischer Darstellung wird vom PG automatisch in ein STEP-5-Programm umgesetzt.

2.1.2 Strukturierte Programmierung

Mit STEP 5 kann das Anwenderprogramm strukturiert werden. Es wird dabei in einzelne, in sich abgeschlossene Programmabschnitte (Bausteine) aufgeteilt. Die Gliederung Ihres Anwenderprogramms verdeutlicht somit auf den ersten Blick die wesentlichen Programmstrukturen oder hebt programmtechnisch zusammenhängende Anlagenteile hervor.

Dieses Verfahren des "**strukturierten Programmierens**" bietet Ihnen folgende Vorteile:

- einfache und übersichtliche Programmierung auch umfangreicher Programme,
- Möglichkeit zum Standardisieren von Programmteilen,
- einfache Programmorganisation,
- leichte Änderungsmöglichkeiten,
- einfacher, abschnittsweiser Programmtest,
- einfache Inbetriebnahme.

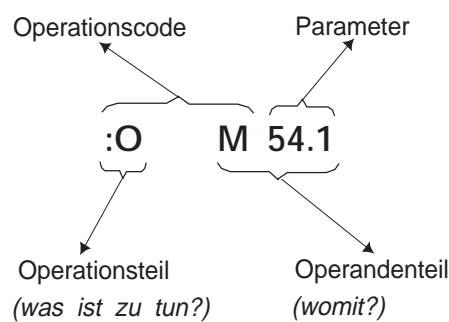
Was ist ein Baustein ?

Ein Baustein ist ein durch Funktion, Struktur oder Verwendungszweck abgegrenzter Teil des Anwenderprogramms. Man unterscheidet Bausteine, die **Anweisungen** (Code) enthalten (Organisationsbausteine, Programmbausteine, Funktionsbausteine, Schrittbausteine), und Bausteine, die **Daten** enthalten (Datenbausteine).

2.1.3 STEP-5-Operationen

Eine STEP-5-Operation ist die kleinste selbständige Einheit des Anwenderprogramms. Sie ist die Arbeitsvorschrift für die CPU. Eine STEP-5-Operation setzt sich zusammen aus einem Operationsteil und einem Operandenteil.

Beispiel



Absolute und symbolische Operanden

Den Operandenteil können Sie entweder **absolut** oder **symbolisch** (über die Zuordnungsliste) eingeben.

Beispiel für die absolute Darstellung: :U E 1.4

Beispiel für die symbolische Darstellung: :U -Motor1

Beachten Sie zur absoluten und symbolischen Programmierung die Bedienungsanleitung zu Ihrem PG.

Anwendung der STEP-5-Operationen

Der Operationsumfang von STEP 5 ermöglicht Ihnen

- binäre Werte miteinander zu verknüpfen, zu setzen oder rückzusetzen,
 - Werte zu laden und zu transferieren,
 - Werte miteinander zu vergleichen und arithmetisch zu bearbeiten,
 - Zeit- und Zählwerte vorzugeben,
 - Zahlendarstellungen umzuwandeln,
 - Bausteine aufzurufen und Sprünge innerhalb eines Bausteins auszuführen
- und
- die Programmbearbeitung zu beeinflussen.

Verknüpfungsergebnis VKE

Das zentrale Bit für die Programmsteuerung ist das Verknüpfungsergebnis VKE. Es wird durch binäre Verknüpfungen gebildet sowie von einigen Operationen beeinflusst.

Der gesamte STEP-5-Operationsvorrat sowie Einzelheiten zur VKE-Bildung werden ausführlich in Abschnitt 3.5 beschrieben. Sie finden dort Programmierbeispiele zu den einzelnen STEP-5-Befehlen.

2.1.4

Zahlendarstellung

Damit die CPU Zahlenwerte miteinander verknüpfen, verändern oder vergleichen kann, müssen diese in einer binärcodierten Darstellung in die Akkumulatoren (Arbeitsregister der CPU) geladen werden.

Abhängig von der auszuführenden Operation sind in STEP 5 folgende Zahlendarstellungen zulässig:

Dualzahlen:	16-bit-Festpunktzahlen
	32-bit-Festpunktzahlen
	32-bit-Gleitpunktzahlen (mit 24-bit-Mantisse)
Dezimalzahlen:	BCD-codierte Zahlen (Vorzeichen und 3 Dezimalstellen)

Zahleneingabe am PG

Bei der Ein- und Ausgabe von Zahlenwerten stellen Sie am Programmiergerät das Datenformat (z. B. KF für Festpunkt) ein, in dem Sie den Zahlenwert eingeben bzw. angezeigt haben möchten. Auf diese Weise übernimmt das PG die Umrechnung aus der intern verwendeten Zahlendarstellung in die von Ihnen gewünschte Darstellungsart.

Zulässige Operationen

Mit den 16-bit-Festpunktzahlen und Gleitpunktzahlen können Sie **alle arithmetischen Operationen** wie Vergleichen, Addieren, Subtrahieren, Multiplizieren und Dividieren ausführen.

Hinweis

BCD-codierte Zahlen dürfen nicht für arithmetische Operationen verwendet werden, da sie zu falschen Ergebnissen führen.

Mit 32-bit-Festpunktzahlen werden Vergleichsoperationen durchgeführt. Außerdem werden sie bei der Umwandlung von BCD-codierten Zahlen in Gleitpunktzahlen als Zwischenstufe benötigt. Mit den Befehlen +D und -D können sie auch für Additionen und Subtraktionen verwendet werden.

Die STEP-5-Sprache enthält **Umwandlungsoperationen**, mit denen Sie Zahlen direkt in die wichtigsten Zahlendarstellungen umwandeln können.

**16- und
32-bit-Festpunktzahlen**

Festpunktzahlen sind ganze, mit einem Vorzeichen versehene Dualzahlen.

*Codierung der
Festpunktzahlen*

Sie sind in binärcodierter Darstellung 16 bit (= 1 Wort) bzw. 32 bit (= 2 Wörter) breit, wobei Bit-Nr. 15 bzw. Bit-Nr. 31 das Vorzeichen enthält:

- '0' = positive Zahl
- '1' = negative Zahl

Negative Zahlen werden im 2er-Komplement dargestellt.

PG-Eingabe

Eingabe des Datenformats für 16-bit-Festpunktzahlen am PG: KF

Eingabe des Datenformats für 32-bit-Festpunktzahlen am PG: DH

zulässiger Zahlenbereich

16-bit-Festpunktzahl:
-32768 bis +32767 (16 bit)

32-bit-Festpunktzahl:
-2147483648 bis +2147483647 (32 bit)
(8000 0000H bis 7FFF FFFFH)

*Anwendung der
Festpunktzahlen*

Festpunktzahlen werden bei einfachen Rechenaufgaben und beim Vergleich von Zahlenwerten verwendet. Da Festpunktzahlen immer ganze Zahlen sind, beachten Sie bitte, daß das Ergebnis einer Division von zwei Festpunktzahlen ebenfalls eine Festpunktzahl ohne Dezimalstellen ist.

Gleitpunktzahlen

Gleitpunktzahlen sind positive und negative gebrochene Zahlen. Sie belegen immer ein Doppelwort (32 bit). Eine Gleitpunktzahl wird als Exponentialzahl dargestellt. Die Mantisse ist 24 bit, der Exponent 8 bit lang.

Die CPU 948 rechnet beim Addieren, Subtrahieren, Multiplizieren und Dividieren mit einer 24 bit breiten Mantisse (Bit 0 bis 23).

Der Exponent gibt die Größenordnung der Gleitpunktzahl an. Am Vorzeichen des Exponenten erkennen Sie, ob der Betrag der Gleitpunktzahl größer oder kleiner als 0,1 ist.

Anwendung der Gleitpunktzahlen

Verwenden Sie Gleitpunktzahlen für die Lösung umfangreicherer Rechenaufgaben, insbesondere bei Multiplikationen und Divisionen, und dann, wenn Sie mit sehr großen oder sehr kleinen Zahlen arbeiten!

Genauigkeit

Die Mantisse gibt die Genauigkeit der Gleitpunktzahl an:

- Genauigkeit bei einer 24-bit-Mantisse:

$$2^{-24} = 0,000000059604 \text{ (entspricht 7 Nachkommastellen)}$$

Ist das Vorzeichen der Mantisse '0', ist die Zahl positiv; bei Vorzeichen '1' ist es eine negative Zahl in 2er-Komplement-Darstellung.

Der **Gleitpunktwert '0'** wird als dualer Wert **80000000H** (32 bit, siehe unten) dargestellt.

Codierung der Gleitpunktzahlen

Codierung einer Gleitpunktzahl:

	31	30		24	23	22		0	
	V	2 ⁶ 2 ⁰	V	2 ⁻¹ 2 ⁻²³		
	Exponent				Mantisse				

Angabe des Datenformats für Gleitpunktzahlen am PG: KG

zulässiger Zahlenbereich

$$\pm 0,1469368 \times 10^{-38} \text{ bis } \pm 0,1701412 \times 10^{39}$$

Ein-/Ausgabe am PG

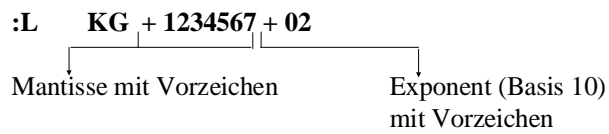
- a) im Code-Baustein:

Es soll die Zahl $Z = 12,34567$ als Gleitpunktkonstante geladen werden.

Eingabe:

:LKG1234567+2

Ausgabe durch PG nach Übernahme der Zeile:



Wert der eingegebenen Zahl: $+0,1234567 \times 10^{+2} = 12,34567$

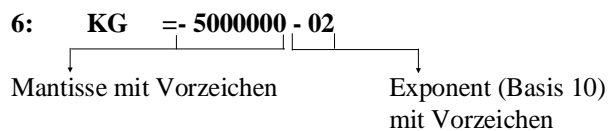
b) im Daten-Baustein:

Es soll die Zahl $Z = -0,005$ als Gleitpunktkonstante definiert werden.

Eingabe:

6: KG = - 5 - 2

Ausgabe durch PG nach Übernahme der Zeile:



Wert der eingegebenen Zahl: $-0,5 \times 10^{-2} = 0,005$

BCD-codierte Zahl

Dezimalzahlen werden als BCD-Zahlen dargestellt. Mit Vorzeichen und 3 Ziffern belegen sie im Akkumulator 16 bit (1 Wort):

15	12 11	8 7	4 3	0
V V V V	Hunderter	Zehner	Einer	

Die einzelnen Ziffern sind positive 4-bit-Dualzahlen zwischen 0000 und 1001 (0 und 9 dezimal).

Die linken Bit sind für das Vorzeichen reserviert.

Vorzeichen bei einer positiven Zahl: 0000

Vorzeichen bei einer negativen Zahl: 1111

zulässiger Zahlenbereich

-999 bis +999

2.1.5 STEP-5-Bausteine und deren Ablage im Speicher

Identifikation

Ein Baustein ist identifiziert durch:

- den Bausteintyp (OB, PB, SB, FB, FX, DB, DX)
- und
- die Bausteinnummer (Zahl zwischen 0 und 255).

Bausteintypen

Die Programmiersprache STEP 5 unterscheidet folgende Bausteintypen:

Organisationsbausteine (OB)

Die Organisationsbausteine sind die Schnittstelle zwischen dem Systemprogramm und dem Anwenderprogramm. Sie können in drei Gruppen unterteilt werden:

Mit den OB 1 bis 39 können Sie die Programmbearbeitung, das Anlaufverhalten der CPU und das Verhalten im Fehlerfall steuern, indem Sie diese Bausteine entsprechend den Ihnen vorliegenden Automatisierungsaufgaben programmieren. Diese OB werden vom Systemprogramm aufgerufen.

Die OB 40 bis 100 sind Bausteine des Betriebssystems. Diese dürfen Sie nicht aufrufen.

Die OB 121 bis 255 enthalten Sonderfunktionen des Systemprogramms. Sie können diese Organisationsbausteine bei Bedarf in Ihrem Anwenderprogramm aufrufen..

Programmbausteine (PB)

Programmbausteine werden zur Strukturierung des Anwenderprogramms verwendet und enthalten die nach technologischen oder funktionellen Gesichtspunkten gegliederten Teilprogramme. Die PB bilden den Kern des Anwenderprogramms.

Schrittbausteine (SB)

Schrittbausteine waren ursprünglich spezielle Programmbausteine zur "schritt"weisen Bearbeitung von Ablaufketten. Ablaufketten können jedoch inzwischen über GRAPH 5 /4/ programmiert werden. Daher haben Schrittbausteine in STEP 5 nicht mehr ihre ursprüngliche Bedeutung.

Schrittbausteine sind jetzt eine zahlenmäßige Erweiterung der Programmbausteine. Sie können wie diese eingesetzt werden.

Funktionsbausteine (FB/FX) Funktionsbausteine dienen zum Programmieren von häufig wiederkehrenden oder auch komplexen Funktionen (z.B. digitale Funktionen, Ablaufsteuerungen, Regelungen, Meldefunktionen).

Ein Funktionsbaustein kann von übergeordneten Bausteinen mehrfach aufgerufen werden und bei jedem Aufruf mit neuen Operanden versorgt ("parametriert") werden.

Durch die Verwendung der Bausteinart FX wird die Zahl der maximal möglichen Funktionsbausteine verdoppelt.

Datenbausteine (DB/DX) In Datenbausteinen stehen die (festen oder veränderbaren) Daten, mit denen das Anwenderprogramm arbeitet. Diese Bausteinart enthält keine STEP-5-Anweisungen und unterscheidet sich in ihrer Funktion grundsätzlich von den übrigen Bausteinen. Durch die Verwendung der Bausteinart DX wird die Zahl der maximal möglichen Datenbausteine verdoppelt.

Aufbau der Bausteine Alle Bausteintypen bestehen aus

- einem Bausteinkopf
- und
- einem Bausteinrumpf.

Bausteinkopf Der **Bausteinkopf** hat immer eine Länge von 5 Wörtern und enthält Informationen für die Bausteinverwaltung im PG und Daten für das Systemprogramm.

Bausteinrumpf Im **Bausteinrumpf** sind – abhängig vom Bausteintyp – enthalten:

- STEP-5-Befehle (bei OB, PB, SB, FB, FX),
- variable oder konstante Daten (bei DB, DX)
- und
- Formaloperandenliste (bei FB, FX).

Bausteinorkopf

Zu den Bausteinen vom Typ DB, DX, FB und FX erzeugt das Programmiergerät zusätzlich einen **Bausteinorkopf** (DV, DXV, FV, FXV). Diese Bausteinorköpfe enthalten Informationen über das Datenformat (bei DB und DX) bzw. über die Sprungmarken (bei FB und FX), die nur das Programmiergerät auswerten kann. Die Bausteinorköpfe werden deshalb nicht in den CPU-Speicher übertragen. Als Anwender haben Sie auf den Inhalt eines Bausteinorkopfes direkt keinen Einfluß.

Maximale Länge

Ein STEP-5-Baustein darf maximal 32 767 Wörter (1 Wort entspricht 16 bit) im Programmspeicher der CPU belegen.

Verfügbare Bausteine

Von den einzelnen Bausteintypen stehen Ihnen zum Programmieren zur Verfügung:

OB	1 bis 39	
FB	0 bis 255] insgesamt 512
FX	0 bis 255	
PB	0 bis 255	
SB	0 bis 255	
DB	2 bis 255] insgesamt 508
DX	3 bis 255	

Die Datenbausteine DB 0, DB 1, DX 0 und DX 1 enthalten Parameter. Sie sind für bestimmte Funktionen reserviert und deshalb nicht beliebig verwendbar.

Der Datenbaustein DX 2 ist für die 2. serielle Schnittstelle reserviert. Daher sollten Sie ihn auch nicht verwenden.

Ablage der Bausteine

Alle programmierbaren Bausteine werden vom PG in der Reihenfolge ihres Transfers im Anwenderspeicher hinterlegt (Bild 2-2). Mit der PG-Funktion "Übertragen Bausteine B" werden zunächst die Code- und dann die Datenbausteine zum AG übertragen.

Die Anfangsadressen der gespeicherten Bausteine werden im Datenbaustein DB 0 hinterlegt.

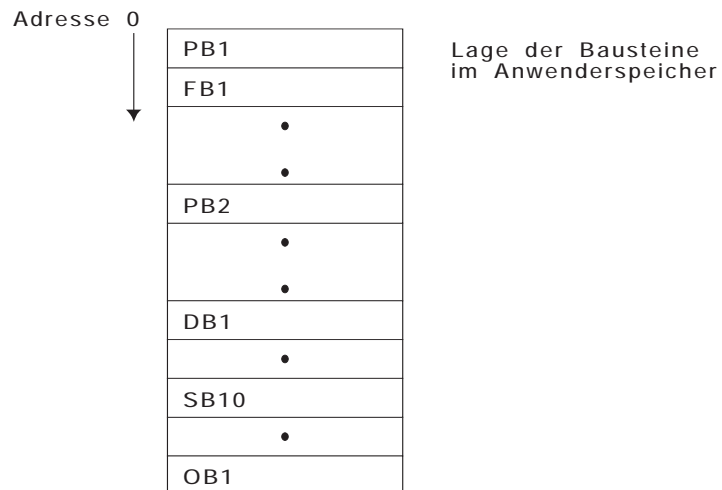


Bild 2-2 Beispiel für eine Ablage der Bausteine im Anwenderspeicher

Korrigieren und Löschen von Bausteinen

Beim **Korrigieren** von Bausteinen wird der "alte" Baustein im Speicher für ungültig erklärt und ein neuer Baustein im Speicher eingetragen.

Ebenso werden beim **Löschen** von Bausteinen die Bausteine nicht wirklich gelöscht, sondern nur für ungültig erklärt. Der von ihnen belegte Platz wird jedoch freigegeben und kann von später geladenen Bausteinen benutzt werden.

Hinweis

Mit der Online-Funktion SPEICHER KOMPRIMIEREN schaffen Sie Speicherplatz für neue Bausteine: Die Funktion optimiert die Speicherplatzbelegung, indem sie ungültig markierte Bausteine löscht und die gültigen Bausteine zusammenschiebt.

2.2 Programm-, Organisations- und Schrittbausteine

Programmbausteine (PB), Organisationsbausteine (OB) und Schrittbausteine (SB) unterscheiden sich hinsichtlich Programmierung und Aufruf nicht. Alle drei können Sie wahlweise in den Darstellungsarten KOP, FUP und AWL programmieren.

Programmieren

Gehen Sie beim Programmieren von Programm-, Organisations- und Schrittbausteinen folgendermaßen vor:

Schritt	Aktion
1	<p>Geben Sie zuerst den Bausteintyp, dann die Nummer des Bausteins an, den Sie programmieren wollen.</p> <p>Folgende Nummern stehen Ihnen jeweils zur Verfügung:</p> <ul style="list-style-type: none"> - Programmbausteine 0 bis 255, - Schrittbausteine 0 bis 255, - Organisationsbausteine 1 bis 39.
2	<p>Geben Sie Ihr Anwenderprogramm in STEP 5 ein.</p> <p>Beim Programmieren von PB, OB und SB dürfen Sie nur die STEP-5-Grundoperationen verwenden!</p> <p>Ein STEP-5-Baustein sollte immer einen in sich abgeschlossenen Programmabschnitt enthalten. Logische Verknüpfungen müssen innerhalb eines Bausteins abgeschlossen sein.</p>
3	<p>Schließen Sie die Programmeingabe mit der Anweisung "BE" (Bausteinende) ab.</p>

Aufrufen

Bausteine – außer OB 1 bis OB 39 – müssen zum Bearbeiten aufgerufen werden. Dies geschieht durch spezielle STEP-5-Operationen, die Bausteinaufrufe.

Diese Bausteinaufrufe können innerhalb eines Organisations-, Programm-, Funktions- oder Schrittbausteins programmiert werden. Sie sind vergleichbar mit Sprüngen in ein Unterprogramm. Jeder Sprung verursacht einen Bausteinwechsel. Die Rücksprungadresse im aufrufenden Baustein wird vom System zwischengespeichert.

Aufrufe können sowohl unbedingt als auch bedingt ausgeführt werden:

Unbedingter Aufruf

Die Anweisung SPA gehört zu den unbedingten Operationen. Sie hat selbst keinen Einfluß auf das VKE. Dieses wird beim Sprung in den neuen Baustein mitgenommen. Dort kann es zwar ausgewertet, jedoch nicht mehr weiter verknüpft werden.

Der angesprochene Baustein wird **unabhängig** vom Verknüpfungsergebnis (VKE – siehe Abschnitt 3.5) bearbeitet.

Beispiel: **SPA PB 100**

Bedingter Aufruf

Die Anweisung SPB gehört zu den bedingten Operationen, d. h. der angesprochene Baustein wird nur bearbeitet, wenn das Verknüpfungsergebnis **VKE = 1** ist. Bei VKE = 0 wird die Sprunganweisung nicht ausgeführt.

Beispiel: **SPB PB 100**

Hinweis

Nach Ausführen der bedingten Sprungoperation ist das VKE auf '1' gesetzt unabhängig davon, ob der Sprung in den aufgerufenen Baustein ausgeführt wird oder nicht.

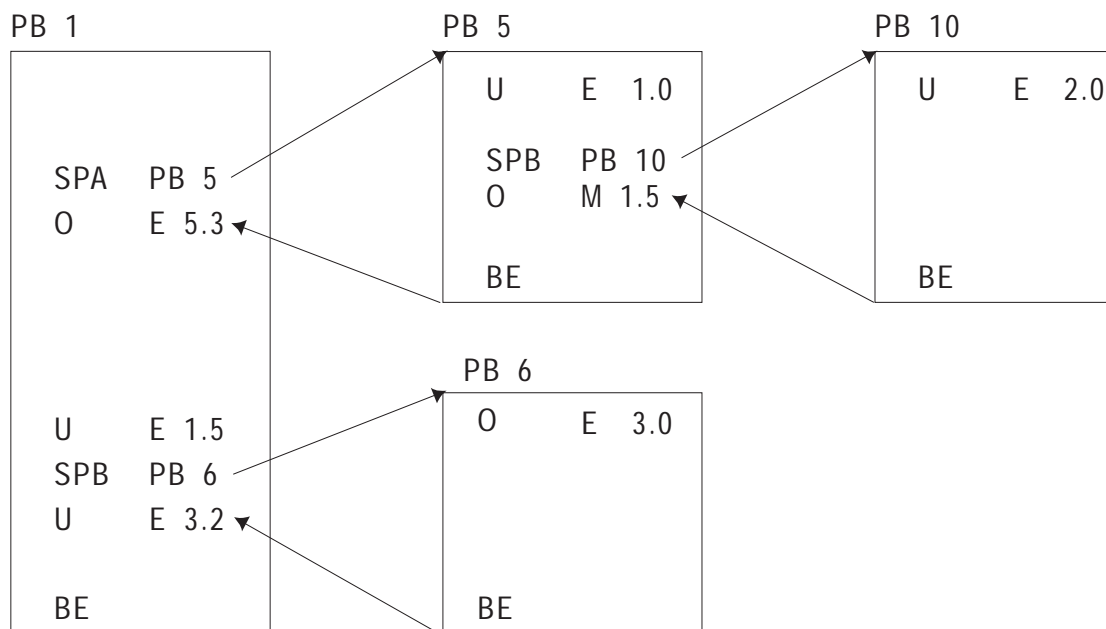


Bild 2-3 Bausteinaufrufe, die die Bearbeitung eines Programmbausteins freigeben

Was die BE-Anweisung bewirkt

Nach der Anweisung BE (Bausteinende) setzt die CPU das Anwenderprogramm in **dem** Baustein fort, in dem der **Bausteinaufruf** programmiert wurde: Die Programmbearbeitung wird bei der auf den Baustein aufruf folgenden STEP-5-Anweisung fortgesetzt.

Die Anweisung BE wird unabhängig vom Verknüpfungsergebnis bearbeitet. Nach BE kann das Verknüpfungsergebnis nicht mehr weiter verknüpft werden. Das bei Ausführung des BE-Befehls vorhandene Verknüpfungs-/Rechenergebnis wird jedoch an den aufrufenden Baustein übergeben und kann dort ausgewertet werden. Bei der Rückkehr aus dem aufgerufenen Baustein werden die Inhalte von AKKU 1, AKKU 2, AKKU 3 und AKKU 4, die Anzeigen ANZ 0 und ANZ 1 und das Verknüpfungsergebnis VKE nicht verändert (nähere Erläuterungen zu den AKKUs, ANZ 0/ANZ 1 und VKE siehe Abschnitt 3.5).

2.2.1 Organisationsbausteine für Anwenderschnittstellen

Die Schnittstelle zwischen dem Systemprogramm und dem Anwenderprogramm sind die Organisationsbausteine. Die Organisationsbausteine OB 1 bis 39 sind Teile des Anwenderprogramms, die Sie genauso wie Programmbausteine programmieren. Durch Programmierung dieser OB können Sie das Verhalten der CPU beim Anlauf, während der Programmbearbeitung und im Fehlerfall beeinflussen. Die Organisationsbausteine sind wirksam, sobald sie in den AG-Speicher geladen werden. **Dies ist auch im laufenden Betrieb möglich.**

Nachdem das Systemprogramm den betreffenden Organisationsbaustein aufgerufen hat, wird das darin enthaltene Anwenderprogramm bearbeitet.

Hinweis

Die Organisationsbausteine OB 1 bis OB 39 für Anwenderschnittstellen werden vom Anwender programmiert und vom Systemprogramm als Reaktion auf bestimmte Ereignisse automatisch aufgerufen!

Zu **Testzwecken** können diese Organisationsbausteine auch vom Anwenderprogramm aufgerufen werden (SPA/SPB OB xxx). Es ist jedoch nicht möglich, z. B. durch Aufruf von OB 20 einen NEUSTART auszulösen!

Die nachfolgenden Tabellen geben Ihnen eine Übersicht über die Anwenderschnittstellen (OBs).

Tabelle 2-1 Übersicht der Organisationsbausteine der CPU 948 für die Programmbearbeitung

Organisationsbausteine zum Steuern der Programmbearbeitung			
Baustein	Funktion und Aufrufkriterium		
OB 1	Organisation der zyklischen Programmbearbeitung: Erster Aufruf nach Ende einer Anlaufart, dann zyklischer Aufruf.		
OB 2 OB 3 OB 4 OB 5 OB 6 OB 7 OB 8 OB 9	<p>Bei DX-0-Einstellung "Prozeßalarmbearbeitung über Eingangsbyte EB 0 = ein": (Unterbrechbarkeit an Bausteingrenzen, im DX 0 einstellbar)</p> <p>Aufruf durch Signalzustandswechsel im Eingangsbyte EB 0 durch Bit:</p> <p>E 0.0 E 0.1 E 0.2 E 0.3 E 0.4 E 0.5 E 0.6 E 0.7</p>		
OB 2 OB 3 OB 4 OB 5 OB 6 OB 9	<p>Bei DX-0-Einstellung "Prozeßalarmbearbeitung über Eingangsbyte EB 0 = aus": (Unterbrechbarkeit an Befehls- oder an Bausteingrenzen, im DX 0 einstellbar)</p> <p>Aufruf über Interrupt-Leitungen des S5-Busses:</p> <p>Systeminterrupt INT X (INT A, B, C oder D, steckplatzabhängig) Systeminterrupt INT E Systeminterrupt INT F Systeminterrupt INT G</p> <p>Verzögerungsalarm Uhrzeitgesteuerter Weckalarm</p>		
OB 10 OB 11 OB 12 OB 13 OB 14 OB 15 OB 16 OB 17 OB 18	Organisation der zeitgesteuerten Programmbearbeitung (Weckalarm) mit Einstellbarkeit von Grundtakt (Grundeinstellung T = 100 ms) und Zeittaktverteiler (Grundeinstellung entspricht 150U) im Datenbaustein DX 0; Aufruf jeweils bei:		
	Grundeinstellung	150U-Zeittaktverteiler	2 ⁿ -Zeittaktverteiler
	0,1 s	T * 1	T * 1
	0,2 s	T * 2	T * 2
	0,5 s	T * 5	T * 4
	1,0 s	T * 10	T * 8
	2,0 s	T * 20	T * 16
	5,0 s	T * 50	T * 32
	10,0 s	T * 100	T * 64
	20,0 s	T * 200	T * 128
	50,0 s	T * 500	T * 256

Tabelle 2-2 Übersicht der Organisationsbausteine der CPU 948 für den Anlauf

Organisationsbausteine zum Steuern des Anlaufverhaltens	
Baustein	Funktion und Aufrufkriterium
OB 20	Aufruf bei NEUSTART (manuell und automatisch)
OB 21	Aufruf bei MANUELLEM WIEDERANLAUF/NEUSTART MIT GEDÄCHTNIS
OB 22	Aufruf bei AUTOMATISCHEM WIEDERANLAUF/NEUSTART MIT GEDÄCHTNIS

Tabelle 2-3 Organisationsbausteine der CPU 948 für Ablauf im WEICHEN STOP

Organisationsbausteine zum Steuern des Anlaufverhaltens	
Baustein	Funktion und Aufrufkriterium
OB 38	Organisation des Anlaufverhaltens bei Kommunikation im "weichen Stopp"
OB 39	Organisation des zyklischen Programms zur Kommunikation im "weichen Stopp"

Tabelle 2-4 Übersicht der Organisationsbausteine der CPU 948 für die Fehlerbearbeitung

Organisationsbausteine für Reaktionen auf Geräte- oder Programmfehler ¹⁾	
Baustein	Funktion und Aufrufkriterium
OB 19	Laufzeitfehler (LZF): Aufruf eines nicht geladenen Bausteins (PB, SB, FB, FX) bzw. Aufschlagen eines nicht geladenen Datenbausteins (DB, DX)
OB 23	Quittungsverzug (QVZ) im Anwenderprogramm (bei Direktzugriff auf Peripheriebaugruppen)
OB 24	Quittungsverzug (QVZ) beim Aktualisieren des Prozeßabbildes und bei Koppelmerkerübertragung
OB 25	Adressierfehler (ADF)
OB 26	Zykluszeitüberschreitung (ZYK)

Organisationsbausteine für Reaktionen auf Geräte- oder Programmfehler ¹⁾	
Baustein	Funktion und Aufrufkriterium
Fortsetzung der Tabelle 2-4:	
OB 27	Substitutionsfehler (SUF)
OB 28	Quittungsverzug Eingangsbyte EB 0 (Prozeßalarme,)
OB 29	Quittungsverzug dezentrale Peripherie, erweitertes Adreßvolumen
OB 30	Quittungsverzug und Parity-Fehler (PARE) bei Zugriff auf Anwenderspeicher
(OB 31)	(Zyklusüberwachungszeit einstellen) ²⁾
OB 32	Lade- und Transferfehler bei Zugriff auf Datenbausteine (TRAF)
OB 33	Weckfehler (WEFES/WEFEH)
OB 34	Fehler beim Einrichten eines Datenbausteins (E DB/EX DX)
OB 36	Fehler beim Ablauf einer Selbsttestfunktion

¹⁾ Ist im Fehlerfall der OB nicht programmiert, geht die CPU in den Stoppzustand. AUSNAHME: Bei nicht vorhandenem OB 19 (Codebaustein nicht geladen), OB 23, OB 24, OB 29 (Quittungsverzug) oder OB 33 (Weckfehler) erfolgt keine Reaktion!

²⁾ Den OB 31 gibt es in der CPU 948 nur aus Kompatibilitätsgründen.
Zum Einstellen der Zyklusüberwachungszeit sollten Sie den Datenbaustein DX 0 benutzen (siehe Kapitel 7).
Der OB 31 wird im Anlauf einmal aufgerufen, falls er geladen ist. Mit ihm können Sie auch die Zyklusüberwachungszeit einstellen, indem Sie in ihm folgende STEP-5-Operationen programmieren:

```
:L KF +nnn  
:BE
```

nnn ist eine Dezimalzahl. Die Zyklusüberwachungszeit ergibt sich aus "nnn * 10 ms".

Betriebssystem-Organisationsbausteine der CPU 948	
Baustein	Funktion
OB 0	interner Baustein des Betriebssystems

2.2.2

Sonderfunktions-Organisationsbausteine

Die folgenden Organisationsbausteine enthalten Sonderfunktionen des Systemprogramms. Sie können von Ihnen **nicht** programmiert (dies gilt für alle OB mit Nummern zwischen 121 und 255!), sondern lediglich aufgerufen werden. Sie enthalten kein STEP-5-Programm. Sonderfunktions-OB können in allen Code-Bausteinen aufgerufen werden.

Tabelle 2-5 Übersicht der Sonderfunktions-Organisationsbausteine der CPU 948

Integrierte Organisationsbausteine mit Sonderfunktionen	
Baustein:	Bausteinfunktion:
OB 121	Uhrzeit stellen/lesen (kompatibel zu CPU 946/947)
OB 122	"Alarmer sperren" ein-/ausschalten
OB 124	STEP-5-Bausteine löschen
OB 125	STEP-5-Bausteine erzeugen
OB 126	Prozeßabbilder definieren, übertragen
OB 129	Batteriezustand ermitteln
OB 131	AKKU 1 bis AKKU 4 löschen
OB 132	AKKU-Roll-Up
OB 133	AKKU-Roll-Down
OB 141	"Weckalarmer einzeln sperren" ein-/ausschalten
OB 142	"Alarmer gemeinsam verzögern" ein-/ausschalten
OB 143	"Weckalarmer einzeln verzögern" ein-/ausschalten
OB 150	Systemzeit stellen/lesen (kompatibel zu CPU 928B)
OB 151	Uhrzeitgesteuerte Weckalarmzeit stellen/lesen
OB 153	Zeit für Verzögerungsalarmer stellen/lesen
OB 180	Variabler Datenbaustein-Zugriff
OB 181	Datenbausteine (DB/DX) testen
OB 182	Datenbereich kopieren
OB 200, 202 bis 205	Funktionen zur Mehrprozessor-Kommunikation
OB 222	Zyklusüberwachungszeit neu starten
OB 223	Anlaufarten der CPUs im Mehrprozessorbetrieb vergleichen
OB 254, 255	DB- und DX-Datenbaustein von Memory Card in den Anwenderspeicher kopieren/duplizieren

Die ausführliche Beschreibung dieser Sonderfunktionen finden Sie in Kapitel 6.

2.3 Funktionsbausteine

Funktionsbausteine (FB/FX) sind ebenso Teile des Anwenderprogramms wie z. B. Programmbausteine. FX-Funktionsbausteine haben den gleichen Aufbau wie FB-Funktionsbausteine und werden ebenso programmiert.

Mit Funktionsbausteinen werden häufig wiederkehrende oder sehr komplexe Funktionen realisiert. Jeder Funktionsbaustein stellt innerhalb des Anwenderprogramms eine abgeschlossene Funktion dar. Sie können Funktionsbausteine

- als Softwareprodukt von SIEMENS beziehen (Standard-Funktionsbausteine auf Diskette – siehe /11/); mit diesen Standard-Funktionsbausteinen können Sie Ihre Programme für Steuern, Melden, Regeln und Protokollieren schnell und sicher erstellen)

oder

- selbst programmieren.

Funktionsbausteine weisen gegenüber den Organisations-, Programm- und Schrittbausteinen vier wesentliche **Unterschiede** auf:

	OB, PB, SB	FB/FX
1.	Operationsumfang	
	nur Grundoperationen	- Grundoperationen, - Ergänzende Operationen - Systemoperationen
2.	Darstellungsart	
	Programmieren und Aufrufen in AWL, KOP, FUP	+
3.	Namen	
	Namensgebung nicht möglich (nur Nummer)	zusätzlich zur Nummer kann ein Name mit max. 8 Zeichen zugewiesen werden
4.	Operanden	
	keine	Formaloperanden (Bausteinparameter). Beim Aufruf werden den Formaloperanden Aktualoperanden zugewiesen.

2.3.1

Aufbau von Funktionsbausteinen

Der **Baustein Kopf** (5 Wörter) eines Funktionsbausteins ist gleich aufgebaut wie die Baustein Köpfe der übrigen STEP-5-Bausteine.

Der **Bausteinrumpf** hingegen unterscheidet sich in seinem Aufbau von dem der anderen Bausteinararten. Die auszuführende Funktion ist darin in Form einer Anweisungsliste in der Programmiersprache STEP 5 geschrieben. Zwischen dem Baustein Kopf und den STEP-5-Anweisungen benötigt ein Funktionsbaustein weiteren Speicherplatz für die Angabe seines Namens und für die Liste der Formaloperanden. Da diese Liste keine Anweisungen für die CPU enthält, wird sie mit einem unbedingten Sprung übersprungen, den das PG automatisch erzeugt. Diese Sprunganweisung wird bei der Ausgabe am PG nicht angezeigt!

Bei einem Aufruf des Funktionsbausteins wird nur der Bausteinrumpf bearbeitet.

absolute oder symbolische Operanden

Operanden können in einem Funktionsbaustein absolut (z. B. M 2.5) oder symbolisch (z. B. -MOTOR1) eingegeben werden. Die Zuordnung der symbolischen Operanden müssen Sie zuvor in einer **Zuordnungsliste** (siehe /3/) ablegen.

So sieht ein Funktionsbaustein im AG-Speicher aus:

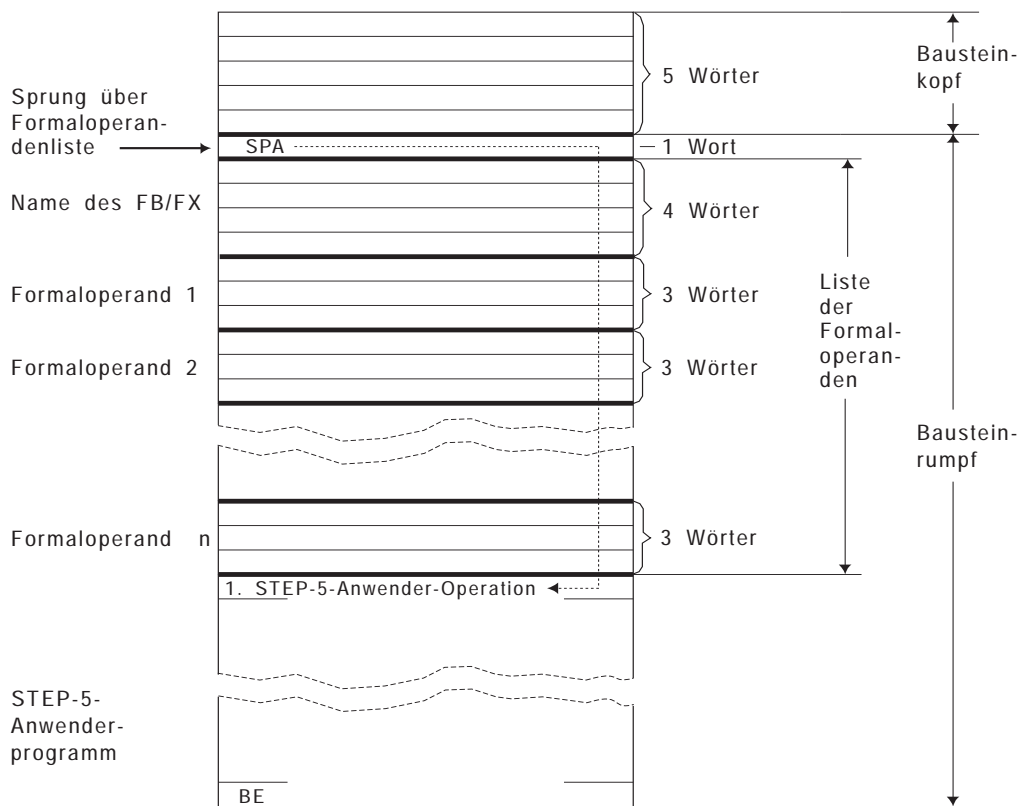


Bild 2-4 Aufbau eines Funktionsbausteins (FB/FX)

Im Speicher stehen somit alle Angaben, die das Programmiergerät benötigt, um den Funktionsbaustein beim Aufruf grafisch darstellen zu können und um die Operanden bei der Parametrierung und Programmierung des Funktionsbausteins zu überprüfen. Eine fehlerhafte Eingabe wird vom Programmiergerät abgelehnt.

Unterscheiden Sie bei der Behandlung von Funktionsbausteinen zwischen

- **FB/FX programmieren**

und

- **FB/FX aufrufen** und anschließend **parametrieren**.

*Unterschied:
"Programmieren" – "Aufrufen
und Parametrieren"*

Beim **Programmieren** legen Sie die Funktion des Bausteins fest. Hierbei müssen Sie überlegen, welche Eingangsoperanden die Funktion benötigt und welche Ausgangsergebnisse sie an das aufrufende Programm übergeben soll. Alle Eingangsoperanden und Ausgangsergebnisse definieren Sie dann als Formaloperanden. Diese üben eine Platzhalterfunktion aus.

Beim **Aufrufen** eines Bausteins durch einen übergeordneten Baustein (OB, PB, SB, FB, FX) werden die Formaloperanden (Bausteinparameter) durch Aktualoperanden ersetzt: der Funktionsbaustein wird **parametriert**.

*Vorgehen beim
Programmieren*

WENN...	DANN...
Sie einen Funktionsbaustein "direkt", d. h. ohne Formaloperanden programmieren wollen,	führen Sie dies durch wie bei Programm- oder Schrittbausteinen.
Sie in einem Funktionsbaustein Formaloperanden verwenden wollen,	gehen Sie so vor, wie auf den nächsten Seiten beschrieben. Beachten Sie dabei die erforderliche Reihenfolge: <ol style="list-style-type: none"> 1. den FB/FX mit den Formaloperanden programmieren und im PG (offline) oder im CPU-Speicher (online) halten, 2. den/die aufrufenden Baustein(e) mit den Aktualoperanden programmieren.

2.3.2

Programmieren von Funktionsbausteinen

Die Programmierung eines Funktionsbausteins können Sie nur in der Darstellungsart "**Anweisungsliste**" durchführen. Bei der Eingabe eines Funktionsbausteins am PG gehen Sie so vor:

Schritt	Aktion
1	<p>Geben Sie den Bausteintyp (FB/FX) und die Nummer des Funktionsbausteins ein.</p> <p>Anwender-Funktionsbausteine sollten von FB 255 an fallend nummeriert werden, um nicht mit den Standard-Funktionsbausteinen zu kollidieren, die von FB 1 bis FB 199 nummeriert sind.</p>
2	<p>Geben Sie den Namen des Funktionsbausteins ein.</p> <p>Der Name kann bis zu 8 Zeichen lang sein und muß mit einem Buchstaben beginnen.</p>
3	<p>Wenn der Funktionsbaustein Formaloperanden bearbeiten soll: Geben Sie als Bausteinparameter die Formaloperanden ein, die Sie im Baustein verwenden.</p> <p>Für jeden Formaloperanden müssen Sie angeben:</p> <ul style="list-style-type: none"> - den Namen des Bausteinparameters (maximal 4 Zeichen), - die Art des Bausteinparameters und evtl. den Typ des Bausteinparameters <p>Sie können maximal 40 Formaloperanden definieren.</p>
4	<p>Geben Sie Ihr STEP-5-Programm in der Darstellungsart AWL ein. Die Formaloperanden werden dabei durch ein vorangestelltes Gleichheitszeichen gekennzeichnet (z. B. U =X1). Sie können auch mehrmals an verschiedenen Stellen im Funktionsbaustein angesprochen werden.</p>
5	<p>Schließen Sie die Programmeingabe mit "BE" (Bausteinende) ab.</p>

Hinweis

Wenn Sie die **Reihenfolge** oder die **Anzahl** der Formaloperanden in der Formaloperandenliste ändern, müssen alle STEP-5-Anweisungen im Funktionsbaustein, die einen **Formaloperanden** ansprechen, und die Bausteinparameterliste im aufrufenden Baustein entsprechend nachgeführt werden!

Programmieren und ändern Sie Funktionsbausteine grundsätzlich auf Diskette oder Festplatte und übertragen Sie sie anschließend in die CPU!

Formaloperanden

Als Formaloperanden eines Funktionsbausteins (auch **Bausteinparameter** genannt) sind folgende Art- und Typbezeichnungen zugelassen:

Tabelle 2-6 Zulässige Formaloperanden für Funktionsbausteine

Parameterart	Parametertyp
E = Eingang A = Ausgang	BI/BY/W/D
D = Datum	KM/KH/KY/KC/KF/ KT/KZ/KG
B = Befehl T = Zeit (Timer) Z = Zähler	keine (Typangabe unzulässig)

E, D, B, T oder **Z** sind Parameterarten, die bei der graphischen Darstellung eines FB-Aufrufs auf der **linken** Seite des Funktionssymbols gezeichnet werden.

Mit **A** gekennzeichnete Parameter werden bei der grafischen Darstellung eines FB-Aufrufs auf der **rechten** Seite des Funktionssymbols gezeichnet.

Der Parametertyp gibt an, ob es sich bei E- und A-Parametern um Bit-, Byte-, Wort- oder Doppelwortgrößen handelt und welches Datenformat (z. B. Bitmuster oder Hexadezimalmuster) für D-Parameter gilt.

2.3.3

Aufrufen und Parametrieren von Funktionsbausteinen

Jeder Funktionsbaustein kann beliebig oft und an beliebigen Stellen im STEP-5-Anwenderprogramm aufgerufen werden. Aufrufe von Funktionsbausteinen können sowohl in einer Anweisungsliste als auch in einer grafischen Darstellung erfolgen (FUP oder KOP).

Gehen Sie beim Aufrufen und Parametrieren folgendermaßen vor:

Schritt	Aktion	Reaktion am PG
1	Stellen Sie sicher, daß der aufgerufene Funktionsbaustein entweder im PG-Speicher (offline) oder im CPU-Speicher (online) vorhanden ist.	keine
2	Geben Sie im aufrufenden Baustein die Aufrufanweisung für den Funktionsbaustein ein. Der Aufruf eines Funktionsbausteins kann innerhalb eines Organisations-, Programm- oder Schrittbausteins oder innerhalb eines anderen Funktionsbausteins programmiert werden.	Nachdem Sie die Aufrufanweisung (z. B. SPA FB 200) eingegeben haben, erscheinen automatisch der Name und die Formaloperandenliste des betreffenden Funktionsbausteins.
3	Ordnen Sie jedem einzelnen Formaloperanden den für diesen Aufruf gültigen Aktualoperanden zu . (Sie parametrieren dadurch den Funktionsbaustein.) Die Aktualoperanden können bei den einzelnen Aufrufen unterschiedlich sein: beim ersten Aufruf des FB 200 z. B. Ein- und Ausgänge, beim zweiten Aufruf Merker. Entsprechend der Formaloperandenliste müssen Sie bei jedem Aufruf eines Funktionsbausteins die erforderlichen Aktualoperanden zuordnen.	keine

Aufruf unbedingt/bedingt

unbedingter Aufruf	bedingter Aufruf
"SPA FBn" für Funktionsbausteine FB oder "BA FXn" für erweiterte Funktionsbausteine FX: Der angesprochene Funktionsbaustein wird unabhängig vom Verknüpfungsergebnis (VKE) bearbeitet.	"SPB FBn" für Funktionsbausteine FB oder "BAB FXn" für erweiterte Funktionsbausteine FX: Der angesprochene Funktionsbaustein wird nur dann bearbeitet, wenn das Verknüpfungsergebnis VKE = 1 ist. Bei VKE = 0 wird der Bausteinaufruf nicht ausgeführt. Unabhängig davon, ob der Bausteinaufruf ausgeführt wird oder nicht, wird anschließend das VKE immer auf '1' gesetzt.
Nach dem unbedingten und bedingten Aufruf kann das Verknüpfungsergebnis nicht weiter verknüpft werden. Es bleibt beim Sprung in den FB erhalten und kann dort ausgewertet werden.	

Zulässige Aktualoperanden Welche Operanden Sie einem Funktionsbaustein als **Aktualoperanden** zuordnen dürfen, entnehmen Sie bitte der nachfolgenden Tabelle.

Tabelle 2-7 Zulässige Aktualoperanden für Funktionsbausteine

Art des Parameters	Typ des Parameters	Zugelassene Aktualoperanden
E, A	BI für einen Operanden mit Bitadresse	E n.m Eingang A n.m Ausgang M n.m Merker
	BY für einen Operanden mit Byteadresse	EB n Eingangsbyte AB n Ausgangsbyte MB n Merkerbyte DL n Datenbyte links DR n Datenbyte rechts PY n Peripheriebyte QY n Byte der erweiterten Peripherie
	W für einen Operanden mit Wortadresse	EW n Eingangswort AW n Ausgangswort MW n Merkerwort DW n Datenwort PW n Peripheriewort QW n Wort der erweiterten Pripherie
	D für einen Operanden mit Doppelwortadresse	ED n Eingangs-Doppelwort AD n Ausgangs-Doppelwort MD n Merker-Doppelwort DD n Daten-Doppelwort
D	KM für ein Binärmuster (16 Stellen)	Konstante
	KY für zwei byteweise Betragswahlen im Bereich von jeweils 0 bis 255	
	KH für ein Hexadezimalmuster bis 4 Stellen	
	KC für zwei alphanumerische Zeichen	
	KT für einen Zeitwert (BCD-codiert) mit Zeitraster .0 bis .3 und Zeitwert 0 bis 999	
	KZ für einen Zählwert 0 bis 999	

Art des Parameters	Typ des Parameters	Zugelassene Aktualoperanden
Fortsetzung der Tabelle 2-6:		
D (Forts.)	KF für eine Festpunktzahl -32768 bis +32767	Konstante
	KG für eine Gleitpunktzahl ¹⁾	
B	keine Typangabe zulässig	DB n Datenbaustein; ausgeführt wird der Befehl A DB n
		FB n Funktionsbaustein (nur ohne Parameter zulässig) wird unbedingt (SPA . .n) aufgerufen
		OB n Organisationsbaustein wird unbedingt (SPA . .n) aufgerufen
		PB n Programmbaustein wird unbedingt (SPA . .n) aufgerufen
		SB n Schrittbaustein wird unbedingt (SPA . .n) aufgerufen
T	keine Typangabe zulässig	T 0 bis 255 Zeit
Z	keine Typangabe zulässig	Z 0 bis 255 Zähler

¹⁾ $\pm 0,1469368 \times 10^{-38}$ bis $\pm 0,1701412 \times 10^{39}$

Hinweis

S-Merker sind als Aktualoperanden für Funktionsbausteine **nicht** zugelassen.

Nach dem Sprung in den Funktionsbaustein werden bei der Bearbeitung des Funktionsbaustein-Programms anstelle der Formaloperanden die Aktualoperanden aus dem aufrufenden Baustein verwendet.

Aufgrund dieser Eigenschaft der parametrierbaren Funktionsbausteine können Sie diese in Ihrem Anwenderprogramm vielseitig verwenden.

Beim Rücksprung aus dem aufgerufenen Funktionsbaustein wird die Liste der Aktualoperanden im aufrufenden Baustein durch eine von STEP 5 implizit im MC-5-Code abgesetzte Sprungoperation übersprungen.

Beispiele

Beispiel 1: Folgendes (vollständiges) Beispiel soll Ihnen sowohl das Programmieren als auch das Aufrufen und Parametrieren eines Funktionsbausteins verdeutlichen. Sie können es selbst leicht nachvollziehen.

Der Funktionsbaustein FB 202 wird programmiert:

FB 202

NETZWERK 1

NAME **BEISPIEL**

BEZ : **MONI** E/A/D/B/T/Z: **E** BI/BY/W/D: **BI**
 BEZ : **BERT** E/A/D/B/T/Z: **E** BI/BY/W/D: **BI**
 BEZ : **HANS** E/A/D/B/T/Z: **A** BI/BY/W/D: **BI**

Formal-
operan-
denliste

:U= **MONI**
 :U= **BERT**
 :== **HANS**

STEP-5-
Anwei-
sungen

:
: BE



Der Funktionsbaustein FB 202 wird im Programmbaustein PB 25 aufgerufen und parametriert:

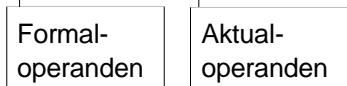
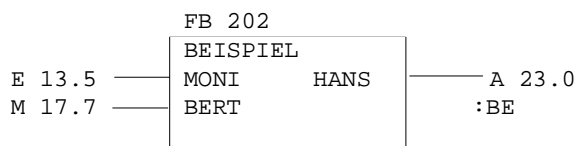
Darstellungsart AWL

Darstellungsart KOP/FUP

PB 25

NETZWERK 1

: SPA FB 202
 NAME : BEISPIEL
 MONI : E 13.5
 BERT : M 17.7
 HANS : A 23.0
 : BE



Folgende Operationen werden nach Sprung in den FB 202 ausgeführt:

: U E 13.5
 : U M 17.7

2.3.4 Spezielle Funktions- bausteine

Neben den Funktionsbausteinen, die der Anwender selbst programmiert, gibt es die **Standard-Funktionsbausteine**, die als fertiges Softwareprodukt zu beziehen sind. Sie enthalten allgemein verwendbare Standardfunktionen (z. B. Meldefunktionen, Ablaufsteuerungen usw.).

Standard-Funktionsbausteine belegen die Nummern FB 1 bis FB 199.

Wenn Sie Standard-Funktionsbausteine beziehen, beachten Sie die speziellen Hinweise in der dazugehörigen Beschreibung (belegte Bereiche, Konventionen usw.).

Die Standard-Funktionsbausteine für das AG S5-155U sind im Katalog ST 57 /11/ aufgeführt.

Beispiel

Gleitpunktradizierer RAD:GP FB 6

Der Funktionsbaustein RAD:GP radiziert eine Gleitpunktzahl (8-bit-Exponent und 24-bit-Mantisse), d. h. er bildet die Quadratwurzel. Das Ergebnis ist ebenfalls eine Gleitpunktzahl (8-bit-Exponent und 24-bit-Mantisse), wobei das niederwertigste Bit der Mantisse nicht gerundet wird.

Der Funktionsbaustein setzt für die weitere Verarbeitung gegebenenfalls die Kennung "Radikand negativ".

Zahlenbereich:

Radikand - 0,1469368 Exp. -38 bis +0,1701412 Exp. +39

Wurzel +0,3833434 Exp. -19 bis +0,1304384 Exp. +20

Funktion: $Y = \sqrt{A}$
Y = SQRT; A = RAD

Aufruf des Funktionsbausteins FB 6:

Im aufgeführten Beispiel wird eine Gleitpunktzahl, die im DD 5 des DB 17 mit 8-bit-Exponent und 24-bit-Mantisse bereitgestellt ist, radiziert. Das Ergebnis, wieder eine 32-bit-Gleitpunktzahl, wird im DD 10 abgelegt. Vorher muß der entsprechende Datenbaustein aufgeschlagen werden. Der Parameter VZ (Parameterart: A, Parametertyp: BI) gibt das Vorzeichen des Radikanden an: VZ = 1 bei negativem Radikanden.

Belegte Merkerwörter: MW 238 bis 254.

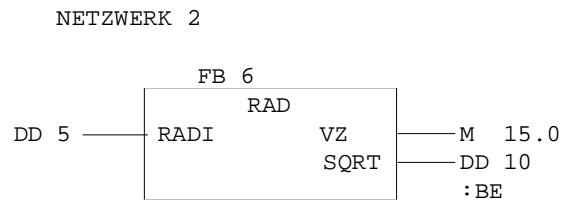
Fortsetzung "Gleitpunktradizierer":

Darstellungsart AWL

```

Netzwerk 1 [ : A DB 17
            :
            : ***
            : SPA FB 6
Netzwerk 2 [ NAME : RAD : GP
            : RADI : DD 5
            : VZ   : M 15.0
            *) [ SQRT : DD 10
    
```

Darstellungsart KOP



DD= Datendoppelwort

*) Muß in getrennten Netzwerken stehen, da der Befehl "A DB 17" in Netzwerk 1 nicht umsetzbar in KOP/FUP ist.

2.4 Datenbausteine

In Datenbausteinen (DB) oder erweiterten Datenbausteinen (DX) sind die festen oder variablen Daten abgelegt, mit denen das Anwenderprogramm arbeitet. In Datenbausteinen werden **keine** STEP-5-Operationen bearbeitet.

Die Daten eines Datenbausteins können sein:

- beliebige Bitmuster, z. B. für Anlagenzustände,
- Zahlen (hexadezimal, dual, dezimal) für Zeitwerte, Rechenergebnisse,
- alphanumerische Zeichen, z. B. für Meldetexte.

Aufbau eines Datenbausteins

Ein Datenbaustein (DB/DX) besteht aus den Teilen

- Bausteinorkopf (DV, DXV),
- Bausteinkopf und
- Bausteinrumpf.

Bausteinorkopf

Der **Bausteinorkopf** wird automatisch auf der Festplatte oder der Diskette des PG angelegt und nicht in die CPU übertragen. Er enthält die Datenformate der im Bausteinrumpf eingegebenen Datenwörter. Als Anwender haben Sie keinen Einfluß auf das Anlegen des Bausteinorkopfes.

Hinweis

Wenn Sie einen Datenbaustein vom AG auf Diskette oder auf Festplatte übertragen, wird der dazugehörige Bausteinorkopf u. U. gelöscht. Aus diesem Grund dürfen Sie einen Datenbaustein mit unterschiedlichen Datenformaten nie im AG ändern und ihn anschließend auf die Diskette zurückübertragen, sonst wird allen Datenwörtern dieses DB automatisch das Datenformat zugeordnet, das Sie in der Voreinstellungsmaske gewählt haben.

Bausteinkopf

Der **Bausteinkopf** belegt 5 Wörter im Speicher und enthält

- die Bausteinkennung,
- die Kennung des Programmiergerätes,
- den Bausteintyp und die Bausteinnummer,
- die Bibliotheksnummer und
- die Bausteinlänge (inkl. Länge des Baustein Kopfes).

Bausteinrumpf

Der **Bausteinrumpf** enthält in aufsteigender Reihenfolge, beginnend mit Datenwort DW 0, die Datenwörter, mit denen das Anwenderprogramm arbeitet. Jedes Datenwort belegt im Speicher 1 Wort (16 bit).

Maximale Länge

Datenbausteine dürfen insgesamt bis zu 32 767 Wörter (inkl. Kopf) im Speicher der CPU belegen. Berücksichtigen Sie beim Eingeben und Übertragen von Datenbausteinen mit dem PG den Speicherausbau Ihrer CPU!

2.4.1 Erstellen von Datenbausteinen

So erstellen Sie einen Datenbaustein:

Schritt	Aktion
1	Geben Sie den Bausteintyp (DB/DX) und eine Datenbaustein- Nummer zwischen 2 bzw. 3 und 255 ein.
2	Geben Sie die einzelnen Datenwörter im gewünschten Datenformat ein. (Die Eingabe der Datenwörter wird nicht mit einer BE-Anweisung abgeschlossen!)

Hinweis

Die Datenbausteine DB 0, DB 1, DX 0, DX 1 und DX 2 sind für bestimmte Funktionen reserviert und damit nicht frei verwendbar (siehe Abschnitt 2.4.3)!

Tabelle 2-8 In einem Datenbaustein zulässige Datenformate

Bezeichnung	Datenformat	Beispiel
KM	Bitmuster	00100110 00111111
KH	Hexadezimalzahl	263F
KY	2 Bytes	038,063
KF	Festpunktzahl	09791
KG	Gleitpunktzahl	+1356123+12
KC	Zeichen	?!ABCD123-+.,%
KT	Zeitwert eines Zeitglieds	055.2
KZ	Zählerwert	234

2.4.2

Aufschlagen von Datenbausteinen

Ein Datenbaustein (DB/DX) kann nur **unbedingt** aufgeschlagen werden. Dies ist möglich innerhalb eines Organisations-, Programm-, Schritt- oder Funktionsbausteins. Ein bestimmter Datenbaustein kann mehrfach im Programm aufgeschlagen werden.

So schlagen Sie Datenbausteine auf:

WENN ...	DANN ...
Sie einen DB -Datenbaustein aufschlagen wollen	geben Sie die STEP-5-Operation " A DB.. " ein
Sie einen DX -Datenbaustein aufschlagen wollen	geben Sie die STEP-5-Operation " AX DX.. " ein

Gültigkeitsbereich

Nach Aufschlagen eines Datenbausteins beziehen sich alle folgenden Anweisungen mit dem Operandenbereich '**D**' auf den aufgeschlagenen Baustein.

Der aufgeschlagene Datenbaustein bleibt auch dann gültig, wenn durch einen Bausteinaufruf die Programmbearbeitung in einem anderen Baustein fortgesetzt wird.

Wenn in diesem Baustein nun ein anderer Datenbaustein aufgeschlagen wird, ist dieser **nur** im aufgerufenen Baustein ab der Aufrufstelle gültig. Nach Rücksprung in den aufrufenden Baustein gilt wieder der alte Datenbaustein.

Zugriff

Der **Zugriff** auf die in dem aufgeschlagenen Datenbaustein gespeicherten Daten erfolgt bei der Programmbearbeitung durch **binäre Verknüpfungen, Speicheroperationen, Lade- oder Transferoperationen** (Einzelheiten dazu lesen Sie bitte in Kapitel 3).

Mit einer **binären Verknüpfung** wird das adressierte Datenwort-Bit zur Bildung des VKE herangezogen. Der Inhalt des Datenwortes wird nicht verändert.

Bei einer **Speicheroperation** wird dem adressierten Datenwort-Bit der Wert des VKE zugewiesen. Der Inhalt des Datenwortes kann dabei verändert werden.

Mit einer **Ladeoperation** wird der Inhalt des adressierten Datenwortes in den AKKU 1 übertragen. Der Inhalt eines Datenwortes nicht verändert.

Mit einer **Transferoperation** werden Daten aus dem AKKU 1 in das adressierte Datenwort übertragen. Der alte Inhalt eines Datenwortes wird überschrieben.

Hinweis

Vor dem Zugriff auf ein Datenwort müssen Sie im Anwenderprogramm den gewünschten Datenbaustein aufschlagen, da die CPU nur so das richtige Datenwort findet. Das adressierte Datenwort muß im aufgeschlagenen Baustein enthalten sein, sonst erkennt das Systemprogramm bei einem Zugriff einen Lade- bzw. Transferfehler.

Mit Lade- und Transferoperationen können Sie nur bis zu Datenwortnummer 255 zugreifen!

Ein aufgeschlagener Datenbaustein bleibt gültig bis

a) ein anderer Datenbaustein aufgeschlagen wird

oder

b) der Baustein, in dem der Datenbaustein aufgeschlagen wurde, mit 'BE', 'BEB' oder 'BEA' beendet wird.

Beispiele**Beispiel 1:** Transferieren von Datenwörtern

Es soll der Inhalt des Datenwortes DW 1 vom Datenbaustein DB 10 in das Datenwort DW 1 des Datenbausteins DB 20 transferiert werden.

Dazu geben Sie folgende Anweisungen ein:

```
:A  DB 10  (DB 10 aufschlagen)
:L  DW 1   (Inhalt DW 1 in den AKKU 1
:      übertragen)
:A  DB 20  (DB 20 aufschlagen)
:T  DW 1   (Inhalt AKKU 1 nach DW 1
:      übertragen)
:
```

Beispiel 2: Gültigkeitsbereich bei Datenbausteinen
(Bild 2-5)

Im Programmbaustein PB 7 wird der Datenbaustein DB 10 aufgeschlagen (A DB 10). In der folgenden Programmbearbeitung werden die Daten dieses Datenbausteins bearbeitet.

Nach dem Aufruf (SPA PB 20) wird der Programm-
baustein PB 20 bearbeitet. Der Datenbaustein DB 10 ist jedoch nach wie vor
gültig. Erst mit dem Aufschlagen des Datenbausteins DB 11 (A DB 11) wird der
Datenbereich gewechselt. Bis zum Ende des Programmbausteins PB 20 (BE) ist
nun der Datenbaustein DB 11 gültig.

Nach Sprung zurück in den Programmbaustein PB 7 ist wieder der Daten-
baustein DB 10 gültig.

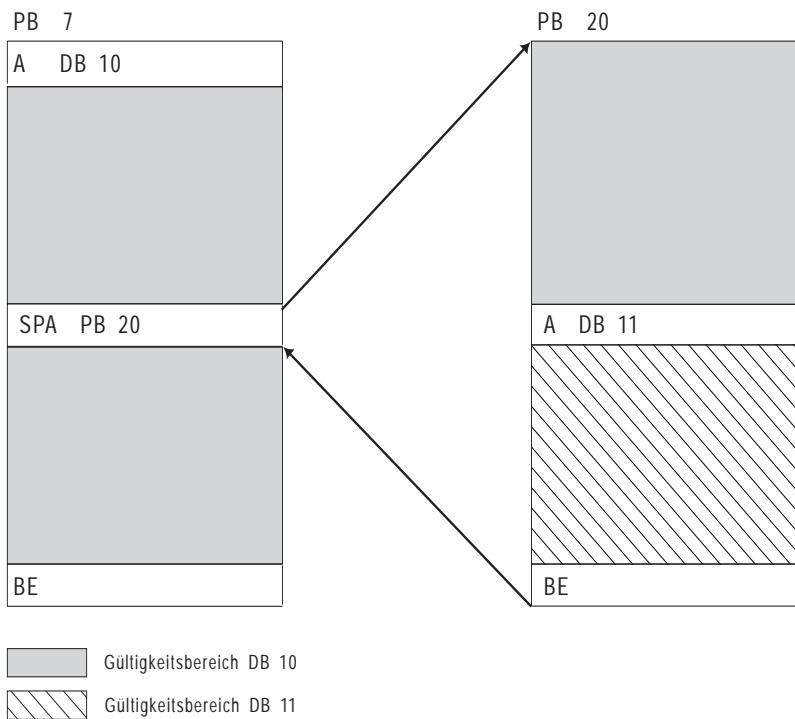


Bild 2-5 Gültigkeitsbereich eines aufgeschlagenen Datenbausteins

2.4.3

Spezielle Datenbausteine

Die Datenbausteine DB 0, DB 1, DX 0, DX 1 und DX 2 sind bei der CPU 948 für bestimmte Funktionen reserviert. Sie werden vom Systemprogramm verwaltet und sind für den Anwender nicht beliebig verwendbar.

DB 0

- **Datenbaustein DB 0** (siehe Abschnitt 8.3.2)

Der Datenbaustein DB 0 enthält die Adreßliste mit den Anfangsadressen aller Bausteine, die sich im Anwenderspeicher der CPU befinden. Diese Adreßliste wird vom Systemprogramm bei der Initialisierung (nach jedem NETZ EIN und nach URLÖSCHEN) erzeugt und bei der Eingabe oder Änderung von Bausteinen mit dem PG automatisch aktualisiert.

DB 1

- **Datenbaustein DB 1** (siehe Abschnitt 10.1.6)

Der Datenbaustein DB 1 enthält die Liste der digitalen Ein- und Ausgänge (P-Peripherie mit relativen Byteadressen von 0 bis 127) sowie der Koppelmerkerein- und -ausgänge, die der CPU zugeordnet sind, und gegebenenfalls eine Zeitenblocklänge.

DB 1 **kann** parametrieren und geladen werden:

Um die Zykluszeit im Einzelprozessorbetrieb zu verringern, da nur die im DB 1 eingetragenen Ein- und Ausgänge oder Zeiten aktualisiert werden.

DB 1 **muß** parametrieren und geladen werden:

- a) bei Mehrprozessorbetrieb.
- b) wenn Koppelmerker verwendet werden, die sich auf CPs befinden.

DX 0

- **Datenbaustein DX 0** (siehe Kapitel 7)

Durch Parametrieren und Laden des Datenbausteins DX 0 können Sie die Voreinstellungen bestimmter Systemprogrammfunktionen (z. B. bei der Bearbeitung des Anlaufs) ändern und damit die Leistungen des Systemprogramms Ihren Erfordernissen anpassen.

DX 1

- **Datenbaustein DX 1**

Reserviert.

DX 2

- **Datenbaustein DX 2**

Reserviert für die 2. serielle Schnittstelle.

Programmbearbeitung

3

Inhalt von Kapitel 3

3.1	Prinzip der Programmbearbeitung	3 - 4
3.2	Programmorganisation	3 - 5
3.3	Speicherung von Programm- und Datenbausteinen	3 - 10
3.4	Bearbeitung des Anwenderprogramms	3 - 11
3.4.1	Begriffsdefinitionen für die Programmbearbeitung	3 - 12
3.5	STEP-5-Operationen mit Beispielen	3 - 15
3.5.1	Grundoperationen	3 - 19
3.5.2	Programmierbeispiele in den Darstellungsarten AWL, KOP und FUP	3 - 34
3.5.3	Ergänzende Operationen	3 - 49
3.5.4	Organisatorische Operationen	3 - 59
3.5.5	Semaphor-Operationen	3 - 75

Programmbearbeitung

3

Dieses Kapitel wendet sich an Leser, die in der Anwendung der Programmiersprache noch keine große Erfahrung haben. Es führt daher in die Grundlagen der STEP-5-Programmierung ein und erläutert im weiteren ausführlich (mit Beispielen) die STEP-5-Operationen der CPU 948.

Erfahrenen Lesern, denen die Information zu einer speziellen STEP-5-Operation im Tabellenheft /1/ nicht ausreicht, kann der Abschnitt 3.5 als Nachschlageteil dienen.

3.1 Prinzip der Programmbearbeitung

Das STEP-5-Anwenderprogramm kann auf verschiedene Art und Weise bearbeitet werden.

Typischerweise herrscht bei speicherprogrammierbaren Steuerungen (SPS) die zyklische Programmbearbeitung vor: Das Systemprogramm läuft in einer Programmschleife (dem Zyklus, siehe Abschnitt 3.4) und ruft dabei in jeder Schleife einmal den Organisationsbaustein OB 1 auf (siehe Bild 3-1).

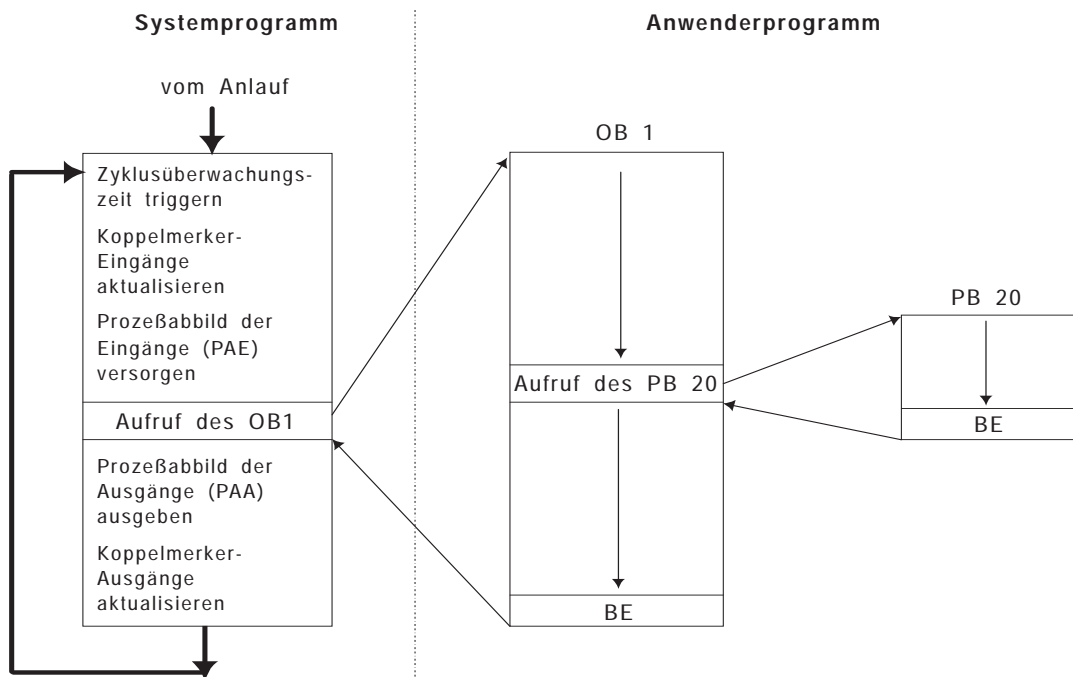


Bild 3-1 Prinzip der zyklischen Programmbearbeitung

3.2 Programmorganisation

Mit der Programmorganisation legen Sie fest, nach welchen Bedingungen und in welcher Reihenfolge die von Ihnen erstellten Bausteine bearbeitet werden sollen. Dazu programmieren Sie in den Organisationsbausteinen bedingte oder unbedingte Aufrufe der gewünschten Bausteine.

In den Programmteilen der einzelnen Organisations-, Programm-, Funktions- und Schrittbausteine können weitere Programm-, Funktions- und Schrittbausteine in beliebiger Kombination (nacheinander oder ineinander verschachtelt) aufgerufen werden.

Das Anwenderprogramm sollte zweckmäßigerweise so organisiert sein, daß es die wesentlichen Programmstrukturen oder programmtechnisch zusammenhängende Anlagenteile hervorhebt.

Die Bilder 3-2 und 3-3 zeigen Ihnen zwei Beispiele einer Programmstruktur.

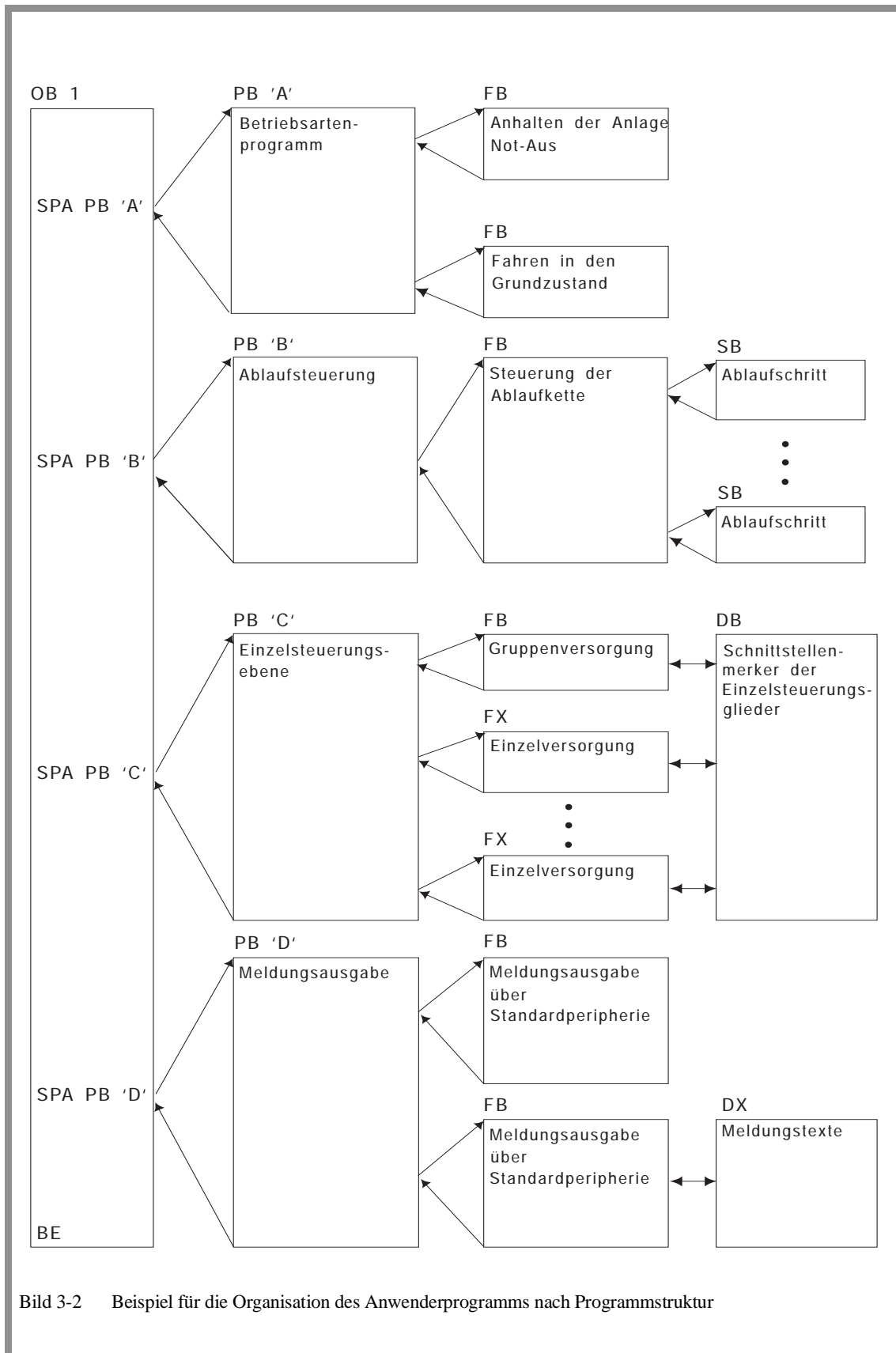


Bild 3-2 Beispiel für die Organisation des Anwenderprogramms nach Programmstruktur

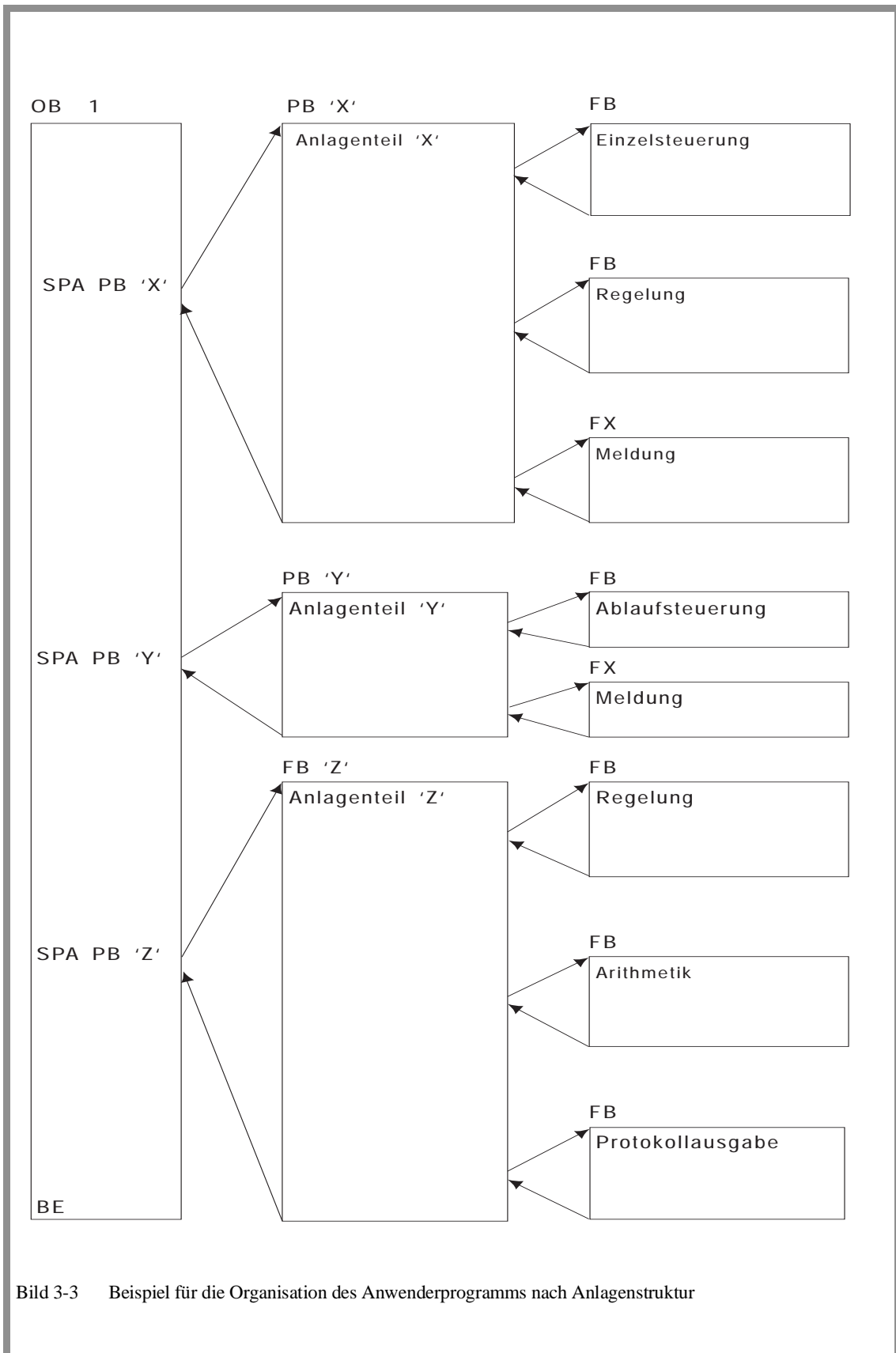
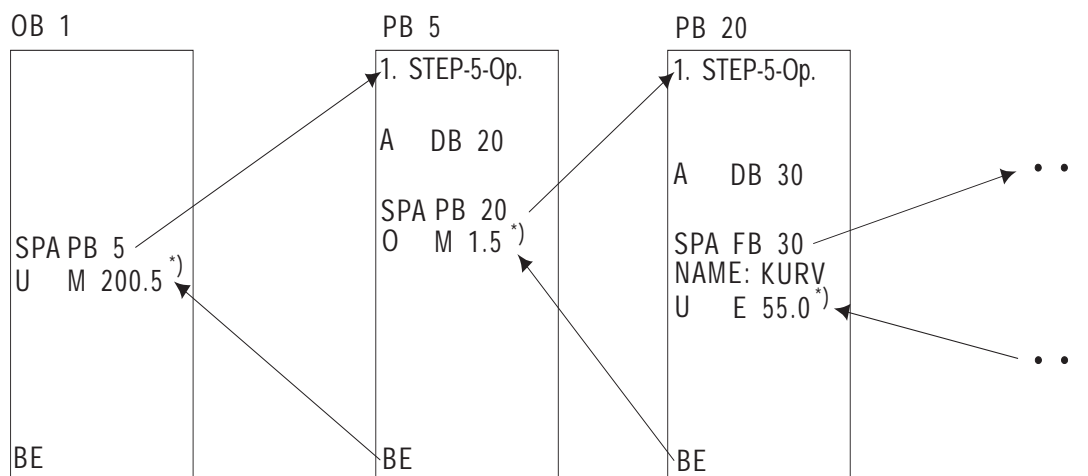


Bild 3-3 Beispiel für die Organisation des Anwenderprogramms nach Anlagenstruktur

Bausteinschachtelung

Bild 3-4 zeigt Ihnen das Prinzip der verschachtelten Bausteinaufrufe.



*) Operation, zu der zurückgesprungen wird

Bild 3-4 Verschachteltes Aufrufen von Codebausteinen

Bausteinadressen

Die Lage eines Bausteins im Anwenderspeicher ist festgelegt durch seine Baustein-Anfangsadresse: Dies ist bei Code-Bausteinen die Adresse derjenigen Zelle im Speicher, in der sich der erste STEP-5-Befehl des Bausteins befindet (bei FB und FX der SPA-Befehl über die Formaloperandenliste); bei Datenbausteinen ist es die Adresse des ersten Datenwortes.

Damit die CPU bei einem Bausteinaufruf den aufgerufenen Baustein im Speicher findet, sind die Anfangsadressen aller gültigen Bausteine in der Bausteinadreßliste im Datenbaustein DB 0 eingetragen. Der DB 0 wird vom Systemprogramm verwaltet, als Anwender können Sie ihn nicht aufschlagen!

Um nach Abarbeitung des aufgerufenen Bausteins den Rückweg in den aufrufenden Baustein zu finden, speichert die CPU bei jedem Aufruf eines neuen Bausteins die **Rücksprungadresse**: Die Rücksprungadresse ist die Adresse derjenigen Zelle im Speicher, in der die dem Bausteinaufruf folgende STEP-5-Anweisung steht. Außerdem wird von der CPU die **Anfangsadresse und Länge des Datenbausteins** gespeichert, der an dieser Stelle gültig ist.

Schachteltiefe

Sie können maximal 40 Bausteine ineinander schachteln. Werden mehr als 40 Bausteine aufgerufen, meldet die CPU einen Fehler und geht in den Stoppzustand.

Beispiel für Schachteltiefe

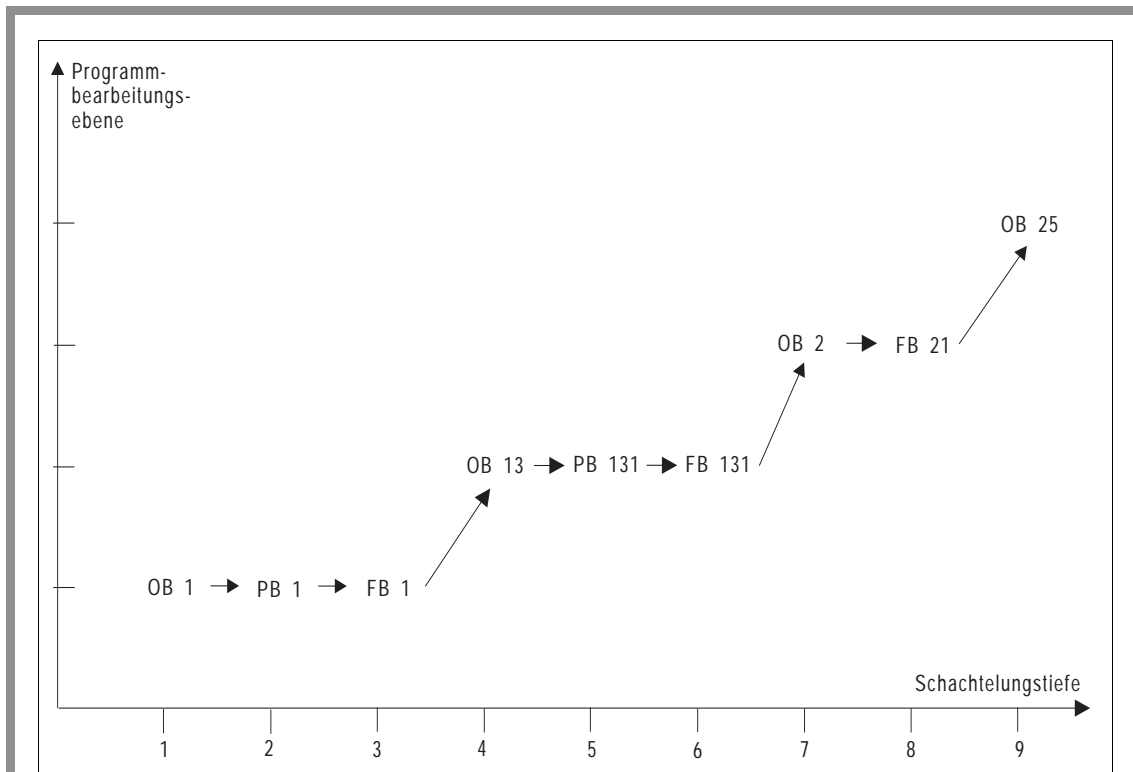


Bild 3-5 Beispiel für Bausteinschachtelungstiefe

So ermitteln Sie die Schachtelungstiefe Ihres Programms :

- Addieren Sie alle von Ihnen programmierten Organisationsbausteine (im Beispiel: 4 OB).
- Addieren Sie die Schachtelungstiefen der einzelnen Organisationsbausteine (im Beispiel: $2 + 2 + 1 + 0 = 5$).
- Beide Beträge zusammen ergeben die Programm-Schachtelungstiefe (im Beispiel: $4 + 5 =$ Schachtelungstiefe 9). Sie darf den Wert 40 nicht überschreiten

3.3 Speicherung von Programm- und Datenbausteinen

Bei der CPU 948 läuft das Anwenderprogramm ausschließlich im internen RAM ab. Deshalb muß es – einschließlich vorhandener Datenbausteine – in den Anwenderspeicher der CPU 948 geladen werden.

Wie werden Programme und Datenbausteine in das interne RAM geladen?

Sie können folgende Methoden benutzen:

- Sie laden die einzelnen Code- und Datenbausteine mit Ihrem PG in das RAM.
- Sie stecken eine Memory Card (Flash EPROM!), die Sie zuvor am PG mit Ihrem gesamten Programm einschließlich aller Datenbausteine programmiert haben, in den dafür vorgesehenen Schacht der CPU.
Beim URLÖSCHEN der CPU (siehe Kapitel 4) wird automatisch der gesamte Inhalt der Memory Card "1:1" in das interne RAM geladen.
- Sie haben Ihr Programm mit dem PG geladen oder die Memory Card per URLÖSCHEN in das interne RAM geladen. Anschließend können Sie über Ihr PG entweder zusätzliche Bausteine in das interne RAM laden oder vorhandene Bausteine ersetzen.

Hinweis

Die Memory Card kann **nur am PG programmiert werden**. Dazu können Sie die PG-Software ab Version 6 verwenden. Beim Programmieren müssen Sie am PG die Betriebsart "WORD BLOCK" wählen (siehe Beschreibung S5-DOS /3/).



Vorsicht

Haben Sie nach dem Laden Ihres Programms von der Memory Card mit dem PG Bausteine verändert oder hinzugefügt, so werden diese Änderungen **mit dem nächsten URLÖSCHEN rückgängig gemacht**, da der Anwenderspeicher erneut mit dem Inhalt der Memory Card **überschrieben wird!**

3.4 Bearbeitung des Anwenderprogramms

Die gesamte Software auf der CPU (diese besteht aus dem Systemprogramm und dem STEP-5-Anwenderprogramm) hat folgende Aufgaben zu bearbeiten:

- ANLAUF der CPU,
- Steuerung eines Automatisierungsprozesses durch ständig sich wiederholende Operationsfolgen (ZYKLUS),
- Steuerung eines Automatisierungsprozesses durch Reaktion auf Ereignisse, die sporadisch oder zu bestimmten Zeiten eintreffen (Alarmer) sowie Reaktion auf Fehler.

Bei allen drei Aufgaben haben Sie die Möglichkeit, über Anwenderschnittstellen (Organisationsbausteine OB 1 bis OB 35 – siehe Abschnitt 2.2.3) spezielle Teile Ihres Anwenderprogramms auf der CPU ablaufen zu lassen.

ANLAUF

Bevor die CPU in die zyklische Bearbeitung eintreten kann, ist es erforderlich, eine Initialisierung durchzuführen, um einen definierten Ausgangszustand für die zyklische Bearbeitung herzustellen und z. B. Zeitraster für die Ausführung bestimmter Funktionen vorzugeben. Welche Initialisierung dies ist, hängt von dem Ereignis ab, das zu einem ANLAUF führt, sowie von Einstellungen, die Sie an Ihrer CPU vornehmen. Nähere Erläuterungen hierzu finden Sie in Kapitel 4.

Sie können das Verhalten der CPU bei einem ANLAUF beeinflussen durch Programmieren der Organisationsbausteine OB 20, OB 21 und OB 22 und durch Parametrieren des Datenbausteins DX 0 (siehe Kapitel 7).

ZYKLUS

Nachdem ein ANLAUF durchgeführt worden ist, tritt das Systemprogramm in die zyklische Bearbeitung ein. Es übernimmt dabei Hintergrundfunktionen, die für Automatisierungsaufgaben erforderlich sind (siehe Bild 3-1 am Kapitelanfang).

Nach Ausführung der Systemfunktionen zu Beginn eines ZYKLUS wird vom Systemprogramm als zyklisches Anwenderprogramm der Organisationsbaustein OB 1 aufgerufen. In diesem Baustein programmieren Sie die STEP-5-Operationen für die zyklische Bearbeitung.

Reaktion bei Alarmen und Fehlern

Um auf einen Alarm oder Fehler speziell reagieren zu können, stehen Ihnen auf der CPU 948 spezielle Organisationsbausteine (OB 2 bis OB 18 für Alarmbearbeitung, OB 19 und OB 23 bis OB 34 für Reaktionen im Fehlerfall) zur Verfügung, in denen Sie das entsprechende STEP-5-Programm hinterlegen können.

Bei der Alarm- oder Fehlerbearbeitung wird vom Systemprogramm der zugehörige Organisationsbaustein in die zyklische Bearbeitung "eingeschachtelt". Dies bedeutet, daß die zyklische Bearbeitung durch die Bearbeitung eines Alarms oder Fehlers unterbrochen wird. Die Einschachtelung der entsprechenden Organisationsbausteine erfolgt nach einem festen Prioritätenschema (weitere Informationen hierzu finden Sie in den Kapiteln 4 und 5).

Außer durch die genannten Organisationsbausteine können Sie das Verhalten der CPU bei der Alarmbearbeitung durch Parametrieren des Datenbausteins DX 0 beeinflussen.

Die Organisationsbausteine OB 1 bis OB 39 können vom Systemprogramm aufgerufen werden, sobald sie in den Programmspeicher geladen worden sind (**auch im laufenden Betrieb**).

Werden sie nicht geladen, so erfolgt entweder keine Reaktion der CPU oder – in den meisten Fehlerfällen – sie geht in den Stoppzustand (siehe hierzu Abschnitt 5.4).

Wie die Organisationsbausteine läßt sich auch der Datenbaustein DX 0 im laufenden Betrieb in den Programmspeicher laden. **Er wird jedoch erst mit dem nächsten NEUSTART wirksam.** Wird der DX 0 nicht geladen, gelten die Standardeinstellungen (siehe Kapitel 7).

3.4.1 Begriffsdefinitionen für die Programmbearbeitung

Zykluszeit

Der Zyklus beginnt mit der Triggerung der sogenannten Zykluszeitüberwachung und endet bei der nächsten Triggerung. Die Zeit, die die CPU für die Programmbearbeitung zwischen zwei Triggerungen benötigt, wird Zykluszeit genannt. Sie setzt sich aus der Laufzeit des Systemprogramms und der Laufzeit des Anwenderprogramms zusammen.

In die Zykluszeit geht somit ein:

- die Bearbeitungszeit des zyklischen Programms (System- und Anwenderprogramm),
- die Bearbeitungszeit von Alarmen (z. B. zeitgesteuerter Alarm),
- die Bearbeitungszeit von Unterbrechungen (Fehler).

Zykluszeitüberwachung

Die Zykluszeit wird von der CPU auf einen Maximalwert überwacht. Standardmäßig ist dieser Maximalwert auf 200 ms eingestellt. Als Anwender haben Sie die Möglichkeit, die Zyklusüberwachungszeit selbst einzustellen bzw. sie während der Anwenderprogrammbearbeitung neu zu starten (siehe DX 0/Kapitel 7 und Sonderfunktions-OB OB 222/Abschnitt 6.16).

Prozeßabbild der Ein- und Ausgänge (PAE und PAA)

Das Prozeßabbild der Ein- und Ausgänge ist ein Speicherbereich im internen RAM. Vor Beginn der zyklischen Anwenderprogrammbearbeitung liest das Systemprogramm die Signalzustände der Eingabe-Peripheriebaugruppen in das Prozeßabbild der Eingänge ein. Das Anwenderprogramm wertet die Signalzustände im Prozeßabbild der Eingänge aus und setzt in Abhängigkeit dieser Auswertung die Signalzustände für die Ausgänge im Prozeßabbild der Ausgänge. Nach Bearbeitung des Anwenderprogramms überträgt das Systemprogramm die Signalzustände des Prozeßabbilds der Ausgänge zu den Ausgabe-Peripheriebaugruppen.

Durch die Zwischenspeicherung der Peripheriesignale im Prozeßabbild der Ein- und Ausgänge wird vermieden, daß ein Ändern des logischen Zustandes eines Bits innerhalb eines Programmzyklus zum "Flattern" des zugehörigen Peripherieausgangs führt.

Das Prozeßabbild ist somit ein Speicherbereich, dessen Inhalt nur **einmal pro Zyklus** an die Peripherie ausgegeben bzw. von der Peripherie eingelesen wird.

Hinweis

Ein Prozeßabbild existiert nur für Ein- und Ausgabebytes der P-Peripherie mit Byteadressen von 0 bis 127!
Neben dem im System integrierten Prozeßabbild können Sie über den OB 126 (siehe Abschnitt 6.6) weitere Prozeßabbilder definieren und übertragen.

Koppelmerker

Koppelmerker dienen zum Datenaustausch zwischen den einzelnen CPUs im Mehrprozessorbetrieb bzw. zwischen der CPU und einigen Kommunikationsprozessoren.

Das Systemprogramm liest vor Beginn der zyklischen Anwenderprogrammbearbeitung die Eingangskoppelmerker der CPU ein und überträgt nach Bearbeitung des Anwenderprogramms die Ausgangskoppelmerker zum Koordinator bzw. zu den Kommunikationsprozessoren.

Die Definition der Ein- und Ausgangskoppelmerker erfolgt durch Erstellen des Datenbausteins DB 1 (siehe Abschnitt 10.1.6).

Unterbrechungseignisse

Die zyklische Programmbearbeitung kann unterbrochen werden durch

- eine zeitgesteuerte Programmbearbeitung (Verzögerungsalarm, zyklische Weckalarne, uhrzeitgesteuerter Weckalarm),
- eine alarmgesteuerte Programmbearbeitung (Prozeßalarm, Systeminterrupt).

Sie kann unterbrochen bzw. ganz abgebrochen werden

- beim Auftreten eines Geräte- oder Programmfehlers,
- durch Bedienung (PG-Funktion, Stoppschalter, Mehrprozessor-Stopp MP-STP),
- durch eine Stopp-Operation.

3.5 STEP-5-Operationen mit Beispielen

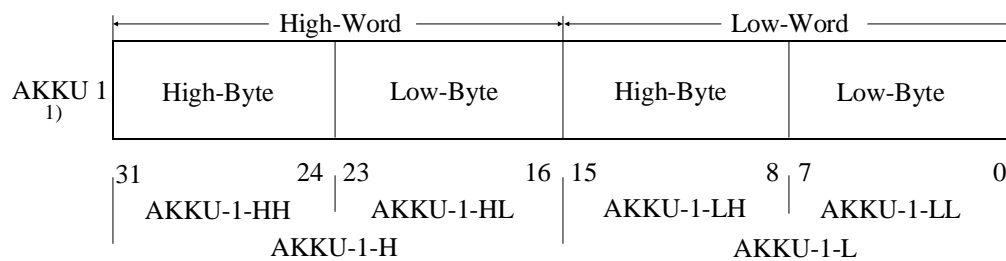
Eine STEP-5-Operation setzt sich zusammen aus einem Operationsteil und einem Operanden. Im Operationsteil wird festgelegt, **was** durch die CPU getan werden soll (Operation). Der Operandenteil gibt an, **womit** eine Operation durchgeführt werden soll.

Die STEP-5-Operationen lassen sich in verschiedene Gruppen einteilen:

- **Grundoperationen** (in **allen** Codebausteinen anwendbar),
- **Ergänzende Operationen**
- **Organisatorische Operationen** (nur in Funktionsbausteinen FB/FX anwendbar),
- Semaphore-Operationen (nur in Funktionsbausteinen FB/FX anwendbar).

Akkumulatoren als Hilfsregister

Die CPU 948 besitzt vier Akkumulatoren, AKKU 1 bis AKKU 4. Ein überwiegender Teil der STEP-5-Operationen verwendet als Quelle für die Operanden und als Ziel für die Ergebnisse zwei Register (32 bit): AKKU 1 und AKKU 2.



Die Akkumulatoren werden abhängig vom auszuführenden STEP-5-Befehl beeinflusst, z. B.:

- bei den Ladeoperationen wird als Ziel immer der AKKU 1 verwendet; der alte Inhalt von AKKU 1 wird in den AKKU 2 (Stack Lift) geschoben; die Akkumulatoren 3 und 4 werden bei allen Ladeoperationen nicht verändert,

¹⁾ bei AKKU 2 bis AKKU 4 analog

- arithmetische Operationen verknüpfen den Inhalt von AKKU 1 und AKKU 2, schreiben das Ergebnis in den AKKU 1 und übertragen den Inhalt des AKKU 3 nach AKKU 2 und den Inhalt des AKKU 4 nach AKKU 3 (Stack Drop); bei 16-bit-Festpunktarithmetik werden nur das Low-Word von AKKU 3 in das Low-Word von AKKU 2 und das Low-Word von AKKU 4 in das Low-Word von AKKU 3 übertragen,
- beim Addieren einer Konstanten (ADD BF/KF/DH) zum Inhalt des AKKU 1 werden die Akkumulatoren 2, 3 und 4 nicht verändert.

Ergebnisanzeigen

STEP-5-Operationen sind entweder anzeigenbildend oder anzeigenabfragend. Die Anzeigen werden in einem Anzeigenbyte hinterlegt. Bei den Anzeigen sind zwei Gruppen zu unterscheiden: Anzeigen von digitalen Operationen (Wortanzeigen – Bit 4 bis 7 im Anzeigenbyte) und Anzeigen von binären und organisatorischen Operationen (Bit-Anzeigen – Bit 0 bis 3 im Anzeigenbyte). Wie die verschiedenen Anzeigen durch STEP-5-Operationen beeinflusst bzw. ausgewertet werden können, entnehmen Sie bitte der Operationsliste /1/.

Das Anzeigenbyte wird bei der PG-Online-Funktion "Status" (siehe Abschnitt 11.2.3) angezeigt und hat folgenden Aufbau:

Wort-Anzeigen				Bit-Anzeigen			
ANZ 1	ANZ 0	OV	OS	OR	STA	VKE	$\overline{\text{ERAB}}$
Bit 7	6	5	4	3	2	1	0

Bit-Anzeigen

- $\overline{\text{ERAB}}$ >ERAB<D> (Erstabfrage);ErgebnisanzeigenErstabfrage

>ERAB<D>Eine logische Verknüpfungskette mit binären Verknüpfungen **beginnt** immer mit einer sogenannten **Erstabfrage**, bei der das VKE **neu** gebildet wird. Mit ihr wird die Bit-Anzeige ERAB = 1 gesetzt. Im Laufe weiterer logischer Verknüpfungen in der begonnenen Kette bleibt ERAB = 1, und das VKE kann durch diese logischen Verknüpfungen verändert werden.

Die begonnene Verknüpfungskette wird **beendet** durch eine binäre Speicheroperation (z. B. S A 5.0). Mit der Speicheroperation wird ERAB = 0 gesetzt; das VKE kann jetzt nur noch ausgewertet (z. B. durch VKE-abhängige Befehle), aber nicht mehr weiter verknüpft werden. Die nächste binäre logische Verknüpfung nach einer binären Speicheroperation ist wieder eine "Erstabfrage".

Beispiel zu \overline{ERAB}

```

:U   E  1.0    $\overline{ERAB}$  wird auf '1' gesetzt,
:
:
:O   E  6.3   VKE wird durch UND-Ver-
:           knüpfung neu gebildet
:UN  E  2.1   VKE wird durch ODER-Verknüp-
:           fung beeinflusst
:S   A  2.4   VKE wird durch UND-NICHT-
:           Verknüpfung beeinflusst
:SPB FB 150    $\overline{ERAB}$  wird auf '0' gesetzt,
:           Verknüpfungskette ist beendet
:           Funktionsbaustein in Abhängig-
:           keit vom VKE aufrufen
:
:
:
```

weitere Bit-Anzeigen

- **VKE** Verknüpfungsergebnis

Ergebnis bit-breiter Verknüpfungen. Wahrheitsaussage bei den Vergleichsbefehlen (siehe Operationsliste, binäre Verknüpfungsoperationen bzw. Vergleichsoperationen).

- **STA** Status

Gibt bei Bit-Befehlen den logischen Zustand des gerade abgefragten oder gesetzten Bits an. Der Status wird bei binären Verknüpfungsoperationen – ausgenommen U(, O(,), O – und bei Speicheroperationen aktualisiert.

- **OR** Oder

Interne Anzeige der CPU für die Behandlung von "UND-vor-ODER-Verknüpfungen".

Wort-Anzeigen

- **OV** Overflow

Gibt an, ob bei der eben abgeschlossenen arithmetischen Operation der zulässige Zahlenbereich überschritten worden ist.

- **OS** Overflow speichernd

Das Overflow-Bit ist gespeichert. OS dient dazu, im Verlaufe mehrerer arithmetischer Operationen zu erkennen, ob irgendwann ein Überlauf (Overflow) aufgetreten ist.

- ANZ 1 und ANZ 0

Codierte Ergebnisanzeigen, deren Interpretation aus der folgenden Tabelle ersichtlich wird.

Hinweis

Zur unmittelbaren Auswertung der Anzeigen stehen Vergleichs- und Sprungoperationen zur Verfügung (siehe Abschnitte 3.5.1 und 3.5.3).

Tabelle 3-1 Ergebnisanzeigen von STEP-5-Operationen

Wort-Anzeigen		Arithmetische Operationen	Digitale Verknüpfungsoperationen	Vergleichsoperationen	Schiebeoperationen	Bei SES, SEF	Ausgeführte Sprungoperationen
ANZ 1	ANZ 0						
0	0	Ergebnis = 0	Ergebnis = 0	AKKU 2 = AKKU 1	gescho-benes Bit = 0	Semaphor ist gesetzt	SPZ
0	1	Ergebnis < 0	–	AKKU 2 < AKKU 1	–	–	SPM SPN
1	0	Ergebnis > 0	Ergebnis ≠ 0	AKKU 2 > AKKU 1	gescho-benes Bit = 1	Semaphor wird gesetzt bzw. freigegeben	SPP SPN
1	1	Division durch Null	–	–	–	–	–

Hinweis

Bei einem Ebenenwechsel, z. B. bei der Bearbeitung eines Weckalarms, werden alle Akkumulatoren sowie die Bit- und Wortanzeigen (VKE usw.) gerettet und nach Fortsetzung der unterbrochenen Ebene wieder geladen.

3.5.1 Grundoperationen

Die Grundoperationen können Sie in **allen** Codebausteinen und in allen Darstellungsarten (KOP, FUP und AWL) benutzen

Binäre Verknüpfungsoperationen

Tabelle 3-2 Binäre Verknüpfungsoperationen

Operation	Operand	Funktion
U O	E 0.0 bis 127.7 A 0.0 bis 127.7 M 0.0 bis 255.7 S 0.0 bis 4095.7 D 0.0 bis 255.15 T 0 bis 255 Z 0 bis 255	UND-Verknüpfung mit Abfrage auf Signalzustand "1" ODER-Verknüpfung mit Abfrage auf Signalzustand "1" eines Eingangs im PAE eines Ausgangs im PAA eines Merkerbits eines S-Merkerbits eines Datenwortbits einer Zeit eines Zählers
UN ON	E 0.0 bis 127.7 A 0.0 bis 127.7 M 0.0 bis 255.7 S 0.0 bis 4095.7 D 0.0 bis 255.15 T 0 bis 255 Z 0 bis 255	UND-Verknüpfung mit Abfrage auf Signalzustand "0" ODER-Verknüpfung mit Abfrage auf Signalzustand "0" eines Eingangs im PAE eines Ausgangs im PAA eines Merkerbits eines S-Merkerbits eines Datenwortbits einer Zeit eines Zählers
O	–	ODER-Verknüpfung von UND-Funktionen
U(O()	–	UND-Verknüpfung von Klammerausdrücken ODER-Verknüpfungen von Klammerausdrücken Klammer zu (Abschluß eines Klammerausdrucks) Es sind maximal 8 Ebenen zulässig, d. h. 7 geöffnete Klammern.

VKE-Bildung

Die binären Verknüpfungsoperationen erzeugen das Verknüpfungsergebnis (VKE).

Am Anfang einer Verknüpfungskette hängt die Bildung des VKE (Erstabfrage) nur vom abgefragten Signalzustand (Status) ab, jedoch nicht von der Verknüpfungsart (O = ODER, U = UND).

Innerhalb einer Verknüpfungskette wird das VKE aus Verknüpfungsart, bisherigem VKE und dem abgefragten Signalzustand gebildet. Eine Verknüpfungskette wird durch einen VKE-begrenzenden (ERAB >ERAB<D>= 0) Befehl (z. B. Speicheroperationen) abgeschlossen. Danach kann das VKE zwar ausgewertet, jedoch nicht weiter verknüpft werden.

Beispiel

Programm	STA	VKE	ERAB
:			
= A 0.0	0	0	0 ← VKE-begrenzt
U E 1.0	1	1	1 ← Erstabfrage
U E 1.1	1	1	1
U E 1.2	0	0	1
= A 0.1	0	0	0 ← VKE-begrenzt, Ende der Verknüpfungskette

Speicheroperationen

Tabelle 3-3 Speicheroperationen

Operation	Operand	Funktion
S R	E 0.0 bis 127.7 A 0.0 bis 127.7 M 0.0 bis 255.7 S 0.0 bis 4095.7 D 0.0 bis 255.15	Bei Zustand '1' des VKE: Setzen Bei Zustand '1' des VKE: Zurücksetzen eines Eingangs im PAE eines Ausgangs im PAA eines Merkers eines S-Merkers eines Bits im Datenwort
=	E 0.0 bis 127.7 A 0.0 bis 127.7 M 0.0 bis 255.7 S 0.0 bis 4095.7 D 0.0 bis 255.15	Zuweisung des VKE zu einem Eingang im PAE Ausgang im PAA Merker S-Merker Bit im Datenwort

Lade- und Transferoperationen

Tabelle 3-4 Lade- und Transferoperationen/Teil 1

Operation	Operand	Funktion
L T		Laden Transferieren
	EB 0 bis 127	eines Eingabebytes vom/zum PAE
	EW 0 bis 126	eines Eingabewortes vom/zum PAE
	ED 0 bis 124	eines Eingabe-Doppelwortes vom/zum PAE
	AB 0 bis 127	eines Ausgabebytes vom/zum PAA
	AW 0 bis 126	eines Ausgabewortes vom/zum PAA
	AD 0 bis 124	eines Ausgabe-Doppelwortes vom/zum PAA
	MB 0 bis 255	eines Merkerbytes
	MW 0 bis 254	eines Merkerwortes
	MD 0 bis 252	eines Merker-Doppelwortes
	SY 0 bis 4095	eines S-Merkerbytes
	SW 0 bis 4094	eines S-Merkerwortes
	SD 0 bis 4092	eines S-Merker-Doppelwortes
	DR 0 bis 255	des rechten Bytes eines Datenwortes aus bzw. nach DB, DX
	DL 0 bis 255	des linken Bytes eines Datenwortes aus bzw. nach DB, DX
	DW 0 bis 255	eines Datenwortes aus bzw. nach DB, DX
	DD 0 bis 254	eines Daten-Doppelwortes aus bzw. nach DB, DX
	PY 0 bis 127	eines Peripheriebytes der Digitaleingaben bzw. -ausgaben (P-Bereich)
	PY 128 bis 255	eines Peripheriebytes der Analog- oder Digitaleingaben bzw. -ausgaben (P-Bereich)
	PW 0 bis 126	eines Peripheriewortes der Digitaleingaben bzw. -ausgaben (P-Bereich)
	PW 128 bis 254	eines Peripheriewortes der Analog- oder Digitaleingaben bzw. -ausgaben (P-Bereich)
	QB 0 bis 255	eines Bytes der erweiterten Peripherie (Q-Bereich)
	QW 0 bis 254	eines Wortes der erweiterten Peripherie (Q-Bereich)

Tabelle 3-5 Lade- und Transferoperationen/Teil 2

Operation	Operand	Funktion
L		Laden
	KB 0 bis 255	einer Konstanten, Bytewert
	KC 2 ASCII-Zeichen	einer Konstanten als 2 ASCII-Zeichen
	KF -32768 bis +32767 ¹⁾	einer Konstanten als Festpunktzahl
	KG	einer Konstanten als Gleitpunktzahl
	KH 0 bis FFFF	einer Konstanten als Hexadezimalzahl
	DH 0 bis FFFF FFFF	einer Doppelwort-Konstanten als Hexadezimalzahl
	KM 16-bit-Muster	einer Konstanten als Bitmuster
	KY 0 bis 255 für jedes Byte	einer Konstanten als 2-Byte-Zahl
	KT 0.0 bis 999.3	einer Konstanten als Zeitwert (BCD-codiert)
	KZ 0 bis 999	einer Konstanten als Zählwert
T 0 bis 255	eines Zeitwertes, dual-codiert	
Z 0 bis 255	eines Zählwertes, dual-codiert	
LC		Laden eines
	T 0 bis 255	Zeitwertes
	Z 0 bis 255	Zählwertes
		im BCD-Code

¹⁾ $\pm 0,1469368 \times 10^{-38}$ bis $\pm 0,1701412 \times 10^{39}$

Ladeoperationen

Ladeoperationen schreiben den adressierten Wert in den AKKU 1, dessen vorheriger Inhalt in den AKKU 2 gerettet wird (Stack Lift).

Transferoperationen

Transferoperationen schreiben den Inhalt des AKKU 1 in die adressierte Speicherzelle.

Beispiele zu den Lade- und Transferoperationen

Beispiel 1:

Bild 3-6 zeigt das Laden/Transferieren eines Bytes, Wortes oder Doppelwortes aus einem/in einen **byteweise** organisierten Speicherbereich (PAE, PAA, Merker, Peripherie):

- :L EB i geladen wird Byte i des PAE in den AKKU-1-LL
- :L EW j geladen werden Byte j und j+1 des PAE in den AKKU-1-L
- :L MD k geladen werden die Merkerbytes k bis k+3 in den AKKU 1

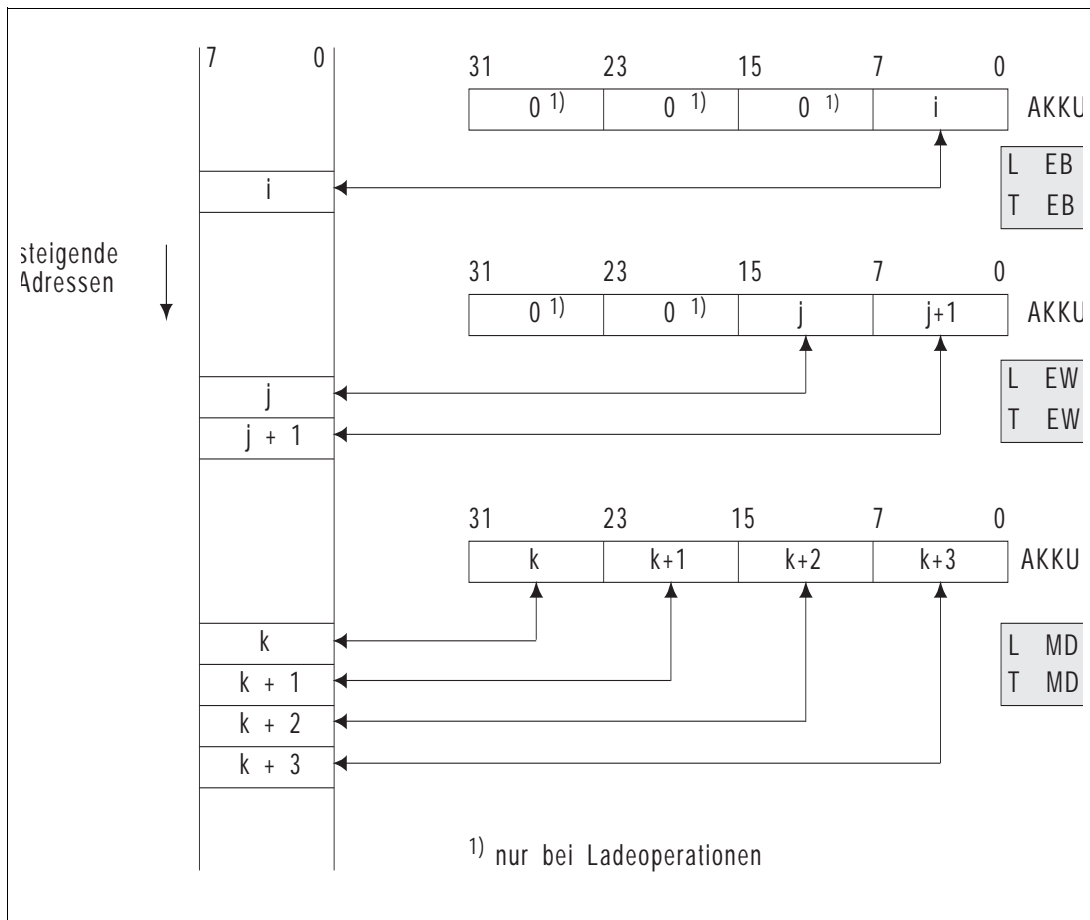


Bild 3-6 Lade- und Transferoperationen in einem byteweise orientierten Speicherbereich

Beispiel 2:

Bild 3-7 zeigt das Laden/Transferieren eines Bytes, Wortes oder Doppelwortes aus einem/in einen **wortweise** organisierten Speicherbereich.

- :L DR i geladen wird das rechte Byte aus dem Datenwort i in den AKKU-1-LL
- :L DL j geladen wird das linke Byte aus dem Datenwort j in den AKKU-1-LL
- :L DW k geladen wird das Datenwort k in den AKKU-1-L
- :L DD l geladen werden die Datenwörter l und l+1 in den AKKU 1

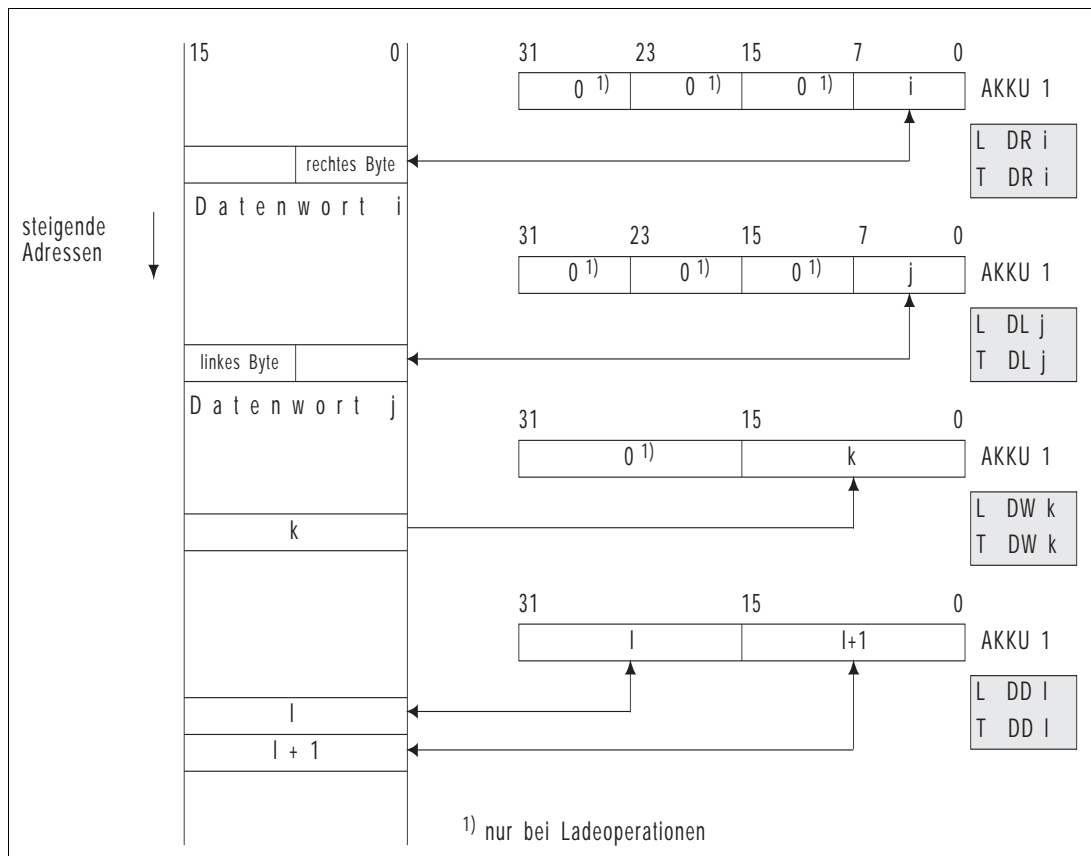


Bild 3-7 Lade- und Transferoperationen in einem wortweise orientierten Speicherbereich

Hinweis

Ladeoperationen beeinflussen die **Anzeigen nicht**.
Transferoperationen löschen das **OS-Bit**.

Beim **Laden** eines **Bytes** oder **Wortes** werden die **höherwertigen Bytes im AKKU 1 gelöscht**.

Ansprechen der Peripherie

Die Peripherie kann durch Lade- und Transferoperationen angesprochen werden:

- **direkt:**

mit L./T...PY, ..PW, ..QB, ..QW

oder

- **über das Prozeßabbild:**

mit L./T...EB, ..EW, ..ED, .AB, ..AW, ..AD

und mit Verknüpfungs- und Speicheroperationen.

Hinweis

Bei den Transferoperationen T PY 0 bis 127 und T PW 0 bis 126 wird parallel das Prozeßabbild der Ausgänge nachgeführt.

Ausnahme: die Befehlsausgabe ist durch die STEP-5-Operation BAS gesperrt worden (siehe Abschnitt 3.5.4).

Beachten Sie zur Peripherie folgende Punkte:

- Ein Prozeßabbild der Ein- und Ausgänge existiert für je 128 Ein- und Ausgabebytes der P-Peripherie mit Byteadressen von 0 bis 127.
- Für den Bereich der P-Peripherie mit den relativen Byteadressen von 128 bis 255 und den gesamten Bereich der Q-Peripherie existiert kein Prozeßabbild! (Zur Adreßraumteilung der Peripherie siehe Abschnitt 8.2.2.)
- Ein-/Ausgabebaugruppen mit Adressen der Q-Peripherie dürfen nur in Erweiterungsgeräten stecken (nicht im Zentralgerät).
- In **einem** Erweiterungsgerät kann man entweder nur P-Peripherie oder nur Q-Peripherie verwenden.



Vorsicht

Falls in einem Erweiterungsgerät Relativadressen der Q-Peripherie verwendet werden, sind diese Adressen für Peripheriebaugruppen im Zentralgerät nicht mehr zulässig (Doppeladressierung!).

Zeit- und Zähloperationen

Um eine Zeit durch eine Startoperation oder einen Zähler durch eine Setzoperation zu laden, muß der Wert vorher in den AKKU 1 geladen werden.

Zu bevorzugen sind folgende Ladeoperationen:

Für Zeiten: L KT, L EW, L AW, L MW, L DW, L SW.

Für Zähler: L KZ, L EW, L AW, L MW, L DW, L SW.

Damit eine **Zeit** mit dem vorgegebenen Zeitwert gestartet wird, ist ein Flankenwechsel des VKE notwendig.

Ein **Zähler** wird mit dem vorgegebenen Zählwert gesetzt oder gezählt, wenn eine positive Flanke des VKE erkannt wird.

In der nachfolgenden Tabelle sind die Flankenwechsel in der Spalte VKE mit entsprechenden Pfeilen gekennzeichnet.

Tabelle 3-6 Zeit- und Zähloperationen

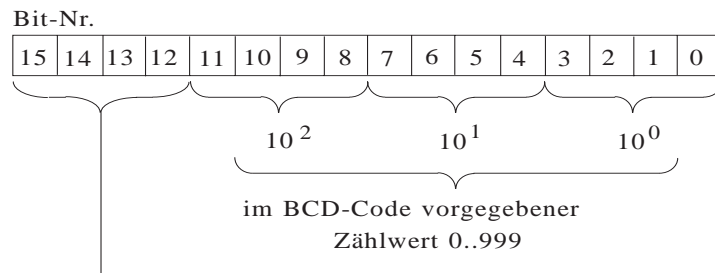
Operation	Operand	VKE 1)	Funktion
SI	T 0 bis 255	↑	Starten einer Zeit als Impuls
SV	T 0 bis 255	↑	Starten einer Zeit als verlängerter Impuls
SE	T 0 bis 255	↑	Starten einer Zeit als Einschaltverzögerung
SS	T 0 bis 255	↑	Starten einer Zeit als speichernde Einschaltverzögerung
SA	T 0 bis 255	↓	Starten einer Zeit als Ausschaltverzögerung
R	T 0 bis 255	1	Rücksetzen einer Zeit
S	Z 0 bis 255	↑	Setzen eines Zählers (BCD-Wert von 0 bis 999)
R	Z 0 bis 255	1	Rücksetzen eines Zählers
ZV	Z 0 bis 255	↑	Vorwärtszählen eines Zählers
ZR	Z 0 bis 255	↑	Rückwärtszählen eines Zählers

1) positive Flanke (↑): Signalzustandsänderung von Zustand '0' nach Zustand '1'
negative Flanke (↓): Signalzustandsänderung von Zustand '1' nach Zustand '0'

Bei der Ausführung der Zeit- bzw. Zähloperationen SI T, SE T, SV T, SS T, SA T und S Z wird der im AKKU 1 stehende Wert in die Zeit- bzw. Zählzelle gebracht (entspricht dem Transferbefehl) und die entsprechende Operation veranlaßt.

Zählwerte

Ein **Zählwert** kann mit der Operation L KZ direkt oder aus einem Merker- oder Datenwort indirekt in den AKKU 1 geladen werden. Er muß folgenden Aufbau haben:



diese Bits sind irrelevant,
 d. h. sie werden beim
 Starten des Zählwertes nicht beachtet

Beispiel

Es soll ein Zählwert von 127 vorgegeben werden:
 Belegung der Bits:

x	x	x	x	0	0	0	1	0	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

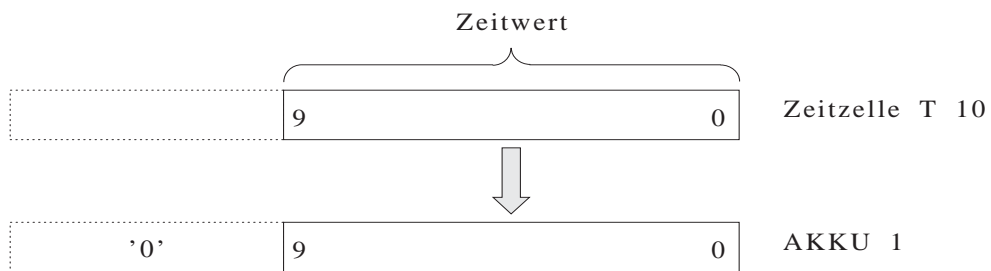
1 2 7
 Zählwert: 127

wird nicht beachtet

In der Zeit- bzw. Zählzelle selbst liegt der Zeit- bzw. Zählwert dualcodiert vor. Zur Abfrage der Zeit bzw. des Zählers kann der Wert der Zeit- bzw. Zählzelle **direkt** oder **BCD-codiert** in den AKKU 1 geladen werden.

Weitere Beispiele zu Zeit- und
Zählwerten

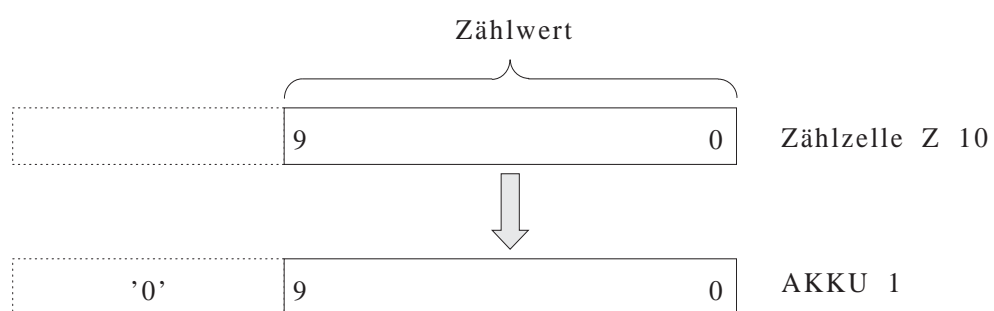
Direktes Laden von **Zeitwerten:**



"L T 10": Direktes Laden des dualen Zeitwertes der Zeit T 10
in den AKKU 1

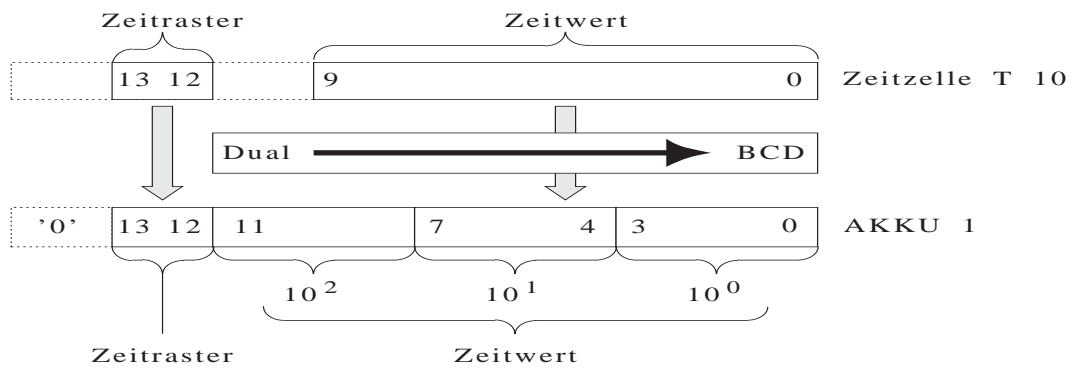
Das Zeitraster wird nicht mitgeladen.

Direktes Laden von **Zählwerten:**



"L Z 10": Direktes Laden des dualen Zählwertes des Zählers Z 10
in den AKKU 1

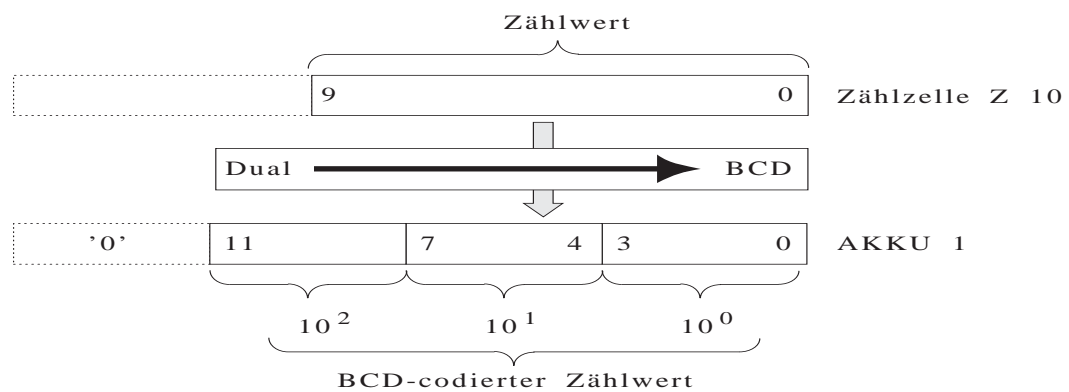
Codiertes Laden von **Zeitwerten**:



"LC T 10": Codiertes Laden des Zeitwertes und des Zeitrasters der Zeit T 10 in den AKKU 1

Das Zeitraster wird mitgeladen.

Codiertes Laden von **Zählwerten**:



"LC Z 10": Codiertes Laden des Zählwertes des Zählers Z 10 in den AKKU 1

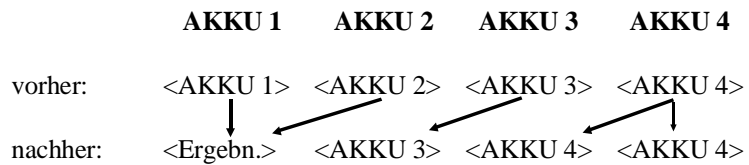
Beim codierten Laden werden die Zustandsbits 14 und 15 der Zeitzellen bzw. 12 bis 15 der Zählzellen nicht geladen. An ihrer Stelle steht 0 in AKKU 1. Der nun im Akku stehende Wert kann weiterverarbeitet werden.

Arithmetische Operationen

Tabelle 3-7 Arithmetische Operationen

Operation	Operand	Funktion
+ F	-	Addition zweier Festpunktzahlen (16 bit)
- F		Subtraktion zweier Festpunktzahlen (16 bit)
x F		Multiplikation zweier Festpunktzahlen (16 bit)
: F		Division zweier Festpunktzahlen (16 bit): Quotient in AKKU-1-L, Rest in AKKU-1-H
+ G		Addition zweier Gleitpunktzahlen (32 bit)
- G		Subtraktion zweier Gleitpunktzahlen (32 bit)
x G		Multiplikation zweier Gleitpunktzahlen (32 bit)
: G		Division zweier Gleitpunktzahlen (32 bit)

Die arithmetischen Operationen verknüpfen den Inhalt von AKKU 1 und AKKU 2 (z. B. "AKKU 2 - AKKU 1"). Das Ergebnis steht anschließend im AKKU 1. Die Rechenregister werden durch eine arithmetische Operation wie folgt verändert (bei Festpunkt-Operationen nur das Low-Word):



Hinweis

Innerhalb der **Ergänzenden Operationen** stehen Ihnen Operationen zur **Subtraktion** und **Addition** von **Doppelwort-Festpunktzahlen** zur Verfügung.

Vergleichsoperationen

Tabelle 3-8 Vergleichsoperationen

Operation	Operand	Funktion
!= >< > >= < <=	– [F D G]	Vergleich auf gleich Vergleich auf ungleich Vergleich auf größer Vergleich auf größer/gleich Vergleich auf kleiner Vergleich auf kleiner/gleich ...F: Vergleich zweier Festpunktzahlen (16 bit) ...D: Vergleich zweier Festpunktzahlen (32 bit) ...G: Vergleich zweier Gleitpunktzahlen (32 bit)

Bausteinoperationen

Tabelle 3-9 Bausteinoperationen

Operation	Operand	Funktion
S P A S P B	OB 1 bis 39 ¹⁾ OB 121 bis 255 PB 0 bis 255 FB 0 bis 255 SB 0 bis 255	Sprung unbedingt Sprung bedingt (nur wenn VKE = 1) zu einem Organisationsbaustein zu einer Systemprogramm- Sonderfunktion zu einem Programmbaustein zu einem FB-Funktionsbaustein zu einem Schrittbaustein
B A B A B	FX 0 bis 255	Sprung unbedingt Sprung bedingt (nur wenn VKE = 1) zu einem FX-Funktionsbaustein
B E B E B B E A	–	Bausteinende Bausteinende bedingt (nur wenn VKE = 1) Bausteinende absolut
A A X	DB 2 bis 255 DX 3 bis 255	Aufschlagen eines DB-Datenbausteins Aufschlagen eines DX-Datenbausteins
E EX	DB 2 bis 255 DX 3 bis 255	Datenbaustein DB erzeugen Datenbaustein DX erzeugen (AKKU 1 muß Anzahl Datenwörter – max. 4091– enthalten, die der neue Baustein haben soll)

¹⁾ nur für Testzwecke!

E DB/EX DX

Erzeuge Datenbaustein

Die Operation E DBx erzeugt einen DB-Datenbaustein mit der Nummer x ($2 \leq x \leq 255$) im Anwenderspeicher der CPU. Der Inhalt des Datenbausteins wird dabei **nicht** mit Null besetzt, d.h. die Datenwörter haben beliebige Inhalte.

Vor dem Programmieren der Anweisung müssen Sie die Anzahl der Datenwörter, die der neue DB haben soll, im AKKU-1-L hinterlegen. Der dazugehörige Bausteinkopf wird von der Operation "E DB" bzw. "EX DX" erzeugt. Ein so erzeugter Datenbaustein darf (**ohne** Bausteinkopf) maximal 4091 Wörter lang sein. Längere Datenbausteine können Sie über OB 125 (siehe Abschnitt 6.5) erzeugen.

Falls der entsprechende Datenbaustein schon existiert, die Länge des DB unzulässig ist oder der Platz im Anwenderspeicher nicht ausreicht, ruft das Systemprogramm den **OB 34** auf. Wenn dieser nicht geladen ist, geht die CPU in den Stoppzustand.

Die Operation EX DXx erzeugt im Anwenderspeicher einen DX-Datenbaustein und arbeitet wie E DBx .

Null-/Bildaufbau-/Stopp-Operationen

Tabelle 3-10 Null-/Bildaufbau-/Stop-Operationen

Operation	Operand	Funktion
N O P 0 N O P 1	–	Nulloperation Nulloperation
B L D	0 bis 255 130 131/131/133 255	Bildaufbauanweisung für das PG: wird von der CPU wie eine Nulloperation behandelt Leerzeile durch Carriage Return erzeugen auf AWL/FUP/KOP umschalten Netzwerk beenden
S T P	–	CPU geht am Zyklusende bzw. am Ende des OB 1 in den weichen STOP.

Hinweis

Da die Operation STP erst am Zyklusende wirksam wird, kommt es dabei zu keinem USTACK-Eintrag. Die Ursache für den Stoppzustand ist daher nachträglich über Diagnosehilfsmittel schwer zu finden.

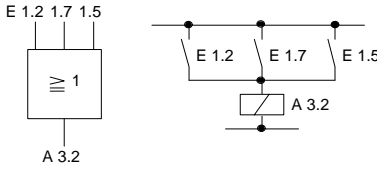
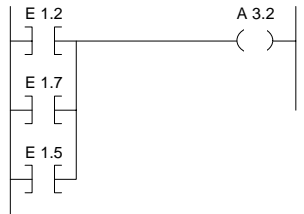
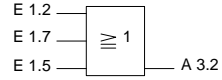
Um die Diagnose zu erleichtern, sollten Sie daher vor Aufruf der STP-Operation eine Kennung setzen, z. B. ein spezielles Bitmuster in einem Diagnose-DB, oder die STEP-5-Operation STS benutzen – siehe Abschnitt 3.5.4.

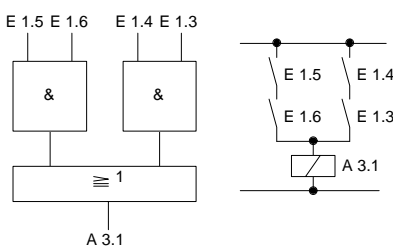
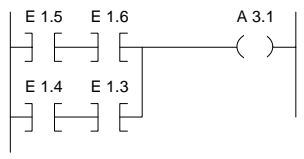
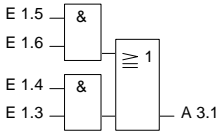
3.5.2 Programmierbeispiele in den Darstellungsarten AWL, KOP und FUP

Verknüpfungsoperationen

UND-Verknüpfung			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 1.1 U E 1.3 U E 1.7 = A 3.5 </pre>		
<p>Am Ausgang A 3.5 erscheint Signalzustand "1", wenn alle Eingänge gleichzeitig den Signalzustand "1" aufweisen.</p> <p>Am Ausgang A 3.5 erscheint Signalzustand "0", wenn mindestens einer der Eingänge den Signalzustand "0" aufweist.</p> <p>Die Anzahl der Abfragen und die Reihenfolge der Programmierung ist beliebig.</p>			

Verknüpfungsoperationen
(Fortsetzung)

ODER-Verknüpfung			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<ul style="list-style-type: none"> O E 1.2 O E 1.7 O E 1.5 = A 3.2 		
<p>Am Ausgang A 3.2 erscheint Signalzustand "1", wenn mindestens einer der Eingänge den Signalzustand "1" aufweist.</p> <p>Am Ausgang A 3.2 erscheint Signalzustand "0", wenn alle Eingänge gleichzeitig den Signalzustand "0" aufweisen.</p> <p>Die Anzahl der Abfragen und die Reihenfolge der Programmierung ist beliebig.</p>			

UND-vor-ODER-Verknüpfung			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<ul style="list-style-type: none"> U E 1.5 U E 1.6 O U E 1.4 U E 1.3 = A 3.1 		
<p>Am Ausgang A 3.1 erscheint Signalzustand "1", wenn mindestens eine UND-Verknüpfung erfüllt ist.</p> <p>Am Ausgang A 3.1 erscheint Signalzustand "0", wenn keine UND-Verknüpfung erfüllt ist.</p>			

Verknüpfungsoperationen
(Fortsetzung)

ODER-vor-UND-Verknüpfung		1. Beispiel	
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	<pre> U E 6.0 O U E 6.1 U (O E 6.2 O E 6.3) = A 2.1 </pre>		
<p>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.</p> <p>Am Ausgang A 2.1 erscheint Signalzustand "0", wenn Eingang E 6.0 Signal "0" führt und die UND-Verknüpfung nicht erfüllt ist.</p>			

ODER-vor-UND-Verknüpfung		2. Beispiel	
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	<pre> U (O E 1.4 O E 1.5) U (O E 2.0 O E 2.1) = A 3.0 </pre>		
<p>Am Ausgang A 3.0 erscheint Signalzustand "1", wenn beide ODER-Verknüpfungen erfüllt sind.</p> <p>Am Ausgang A 3.0 erscheint Signalzustand "0", wenn mindestens eine ODER-Verknüpfung nicht erfüllt ist.</p>			

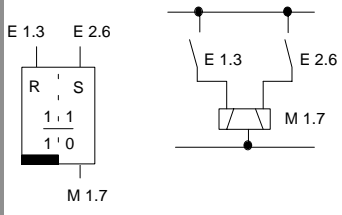
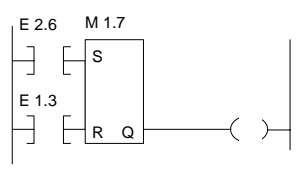
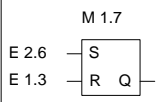
Verknüpfungsoperationen
(Fortsetzung)

Abfrage auf Signalzustand "0"			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	<pre> U E 1.5 UN E 1.6 = A 3.0 </pre>		
<p>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 nicht betätigt) führt.</p>			

Speicherooperationen

RS-Speicherglied für speichernde Signalausgabe			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	<pre> U E 2.7 S A 3.5 U E 1.4 R A 3.5 </pre>		
<p>Signalzustand "1" am Eingang E 2.7 bewirkt das Setzen des Speicherglieds (Signalzustand "1" am Ausgang A 3.5). Wechselt der Signalzustand am Eingang E 2.7 nach "0", so bleibt dieser Zustand erhalten, d. h., das Signal wird gespeichert.</p> <p>Signalzustand "1" am Eingang E 1.4 bewirkt das Rücksetzen des Speichergliedes (Signalzustand "0" am Ausgang A 3.5). Wechselt der Signalzustand am Eingang E 1.4 nach "0", so bleibt dieser Zustand erhalten.</p> <p>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 (Rücksetzen vorrangig).</p>			

Speicherooperationen
(Fortsetzung)

RS-Speicherglied mit Merkern			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 2.6 S M 1.7 U E 1.3 R M 1.7 </pre>		
<p>Signalzustand "1" am Eingang E 2.6 bewirkt das Setzen des Speichergliedes.</p> <p>Wechselt der Signalzustand am Eingang E 2.6 nach "0", so bleibt dieser Zustand erhalten, d. h. das Signal wird gespeichert.</p> <p>Signalzustand "1" am Eingang E 1.3 bewirkt das Rücksetzen des Speichergliedes.</p> <p>Wechselt der Signalzustand am Eingang E 1.3 nach "0", so bleibt dieser Zustand erhalten.</p> <p>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 (Rücksetzen vorrangig).</p>			

Speicherooperationen
(Fortsetzung)

Nachbildung eines Wischrelais			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungsliste	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>		
<p>Bei jeder ansteigenden Flanke des Einganges 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 ("Impulsmerker") gesetzt.</p> <p>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.</p> <p>Der Merker M 2.0 wird rückgesetzt.</p> <p>Der Merker M 2.0 führt also während eines einzigen Programmdurchlaufs Signalzustand "1".</p>			

Binäruntersetzer (T-Kippglied)			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungsliste	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>		
<p>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.</p>			

Zeitoperationen

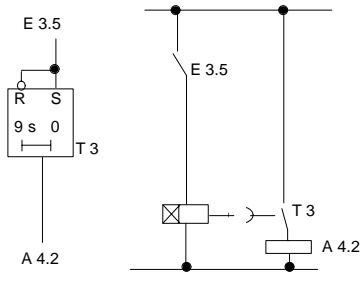
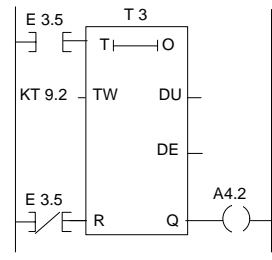
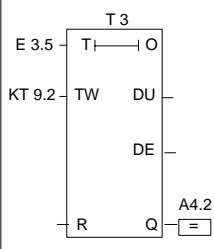
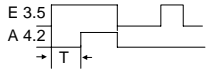
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 3.0 L KT 10.2 SI T 1 UN E 3.0 R T 1 L T 1 T AW 0 T AW 2 LC T 1 T AW 2 U T 1 = A 4.0 </pre>		
<p>Bei Signalzustandswechsel von "0" nach "1" am Eingang E 3.0 wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis "1" bleibt das Zeitglied unbeeinflusst.</p> <p>Bei Signalzustand "0" am Eingang E 3.0 wird das Zeitglied auf Null gesetzt (gelöscht).</p> <p>Die Abfragen U T bzw. O T liefern Signalzustand "1", solange die Zeit läuft.</p> <p>KT 10.2: Das Zeitglied wird mit dem angegebenen Wert (10) geladen. Die Zahl rechts vom Punkt gibt das Zeitraster an: 0 = 001 s 2 = 1 s 1 = 0.1 s 3 = 10 s</p> <div style="display: flex; align-items: center;"> </div> <p>DU und DE sind digitale Ausgänge der Zeitzelle. Am Ausgang DU steht der Zeitwert dualcodiert, am Ausgang DE BCD-codiert mit Zeitraster an.</p>			

Zeitoperationen (Fortsetzung)

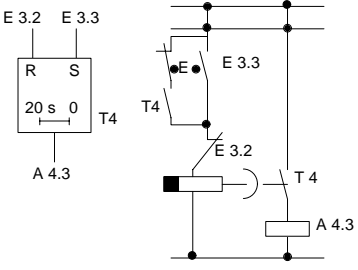
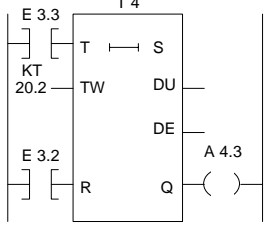
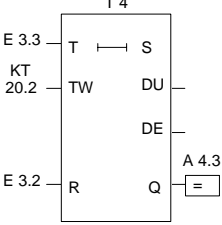
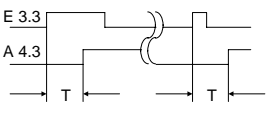
Verlängerter Impuls

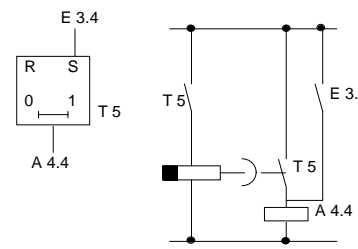
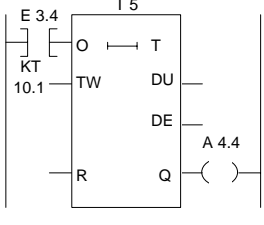
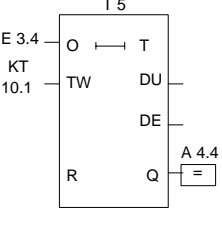
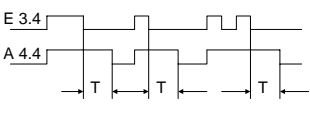
Aufgabenstellung	STEP-5-Darstellung												
	Anweisungs- liste	Kontaktplan	Funktionsplan										
	<pre> U E 3.1 L EW 15 SV T 2 U T 2 = A 4.1 </pre>												
<p>Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied gestartet.</p> <p>Bei Verknüpfungsergebnis "0" bleibt das Zeitglied unbeeinflusst</p> <p>Die Abfragen U T oder OT liefern Signalzustand "1", solange die Zeit läuft.</p>													
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">(EB 15)</td> <td style="text-align: center;">(EB 16)</td> </tr> <tr> <td style="text-align: center;">5 4 3 0 7</td> <td style="text-align: center;">4 3 0</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;"> </td> </tr> <tr> <td style="text-align: center;">10²</td> <td style="text-align: center;">10¹ 10⁰</td> </tr> <tr> <td style="text-align: center;">Zeit- raster</td> <td style="text-align: center;">Zeitwert</td> </tr> </table>				(EB 15)	(EB 16)	5 4 3 0 7	4 3 0			10 ²	10 ¹ 10 ⁰	Zeit- raster	Zeitwert
(EB 15)	(EB 16)												
5 4 3 0 7	4 3 0												
10 ²	10 ¹ 10 ⁰												
Zeit- raster	Zeitwert												
<p>EW 15: Setzen des Zeitwerts mit dem im BCD-Code vorliegenden Wert der Operanden E, A, M oder D (im Beispiel Eingangswort 15)</p>													

Zeitoperationen (Fortsetzung)

Einschaltverzögerung							
Aufgabenstellung	STEP-5-Darstellung						
	Anweisungsliste	Kontaktplan	Funktionsplan				
	<pre> U E 3.5 L KT 9.2 SE T 3 UN E 3.5 R T 3 U T 3 = A 4.2 </pre>						
<p>Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis "1" bleibt das Zeitglied unbeeinflusst.</p> <p>Bei Signalzustand "0" am Eingang E 3.5 wird das Zeitglied auf Null gesetzt (gelöscht).</p> <p>Die Abfragen U T bzw. O T liefern Signalzustand "1", wenn die Zeit abgelaufen ist und der Signalzustand "1" am Eingang E 3.5 noch ansteht.</p> <p>KT 9.2: Das Zeitglied wird mit dem angegebenen Wert (9) geladen. Die Zahl rechts vom Punkt gibt das Zeitraster an:</p> <table style="margin-left: 20px;"> <tr> <td>0 = 0.01 s</td> <td>2 = 1 s</td> </tr> <tr> <td>1 = 0.1 s</td> <td>3 = 10 s</td> </tr> </table>				0 = 0.01 s	2 = 1 s	1 = 0.1 s	3 = 10 s
0 = 0.01 s	2 = 1 s						
1 = 0.1 s	3 = 10 s						
							

Zeitoperationen (Fortsetzung)

Speichernde Einschaltverzögerung			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 3.3 L KT 20.2 SS T 4 U E 3.2 R T 4 U T 4 = A 4.3 </pre>		
<p>Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied gestartet.</p> <p>Bei Verknüpfungsergebnis "0" bleibt das Zeitglied unbeeinflusst.</p> <p>Die Abfragen UT bzw. OT liefern Signalzustand "1", wenn die Zeit abgelaufen ist. Der Signalzustand wird erst dann "0", wenn das Zeitglied mit der Funktion RT zurückgesetzt worden ist.</p>			
			

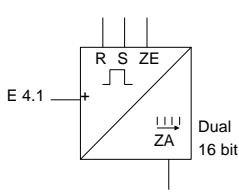
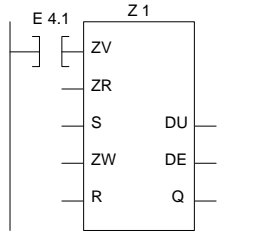
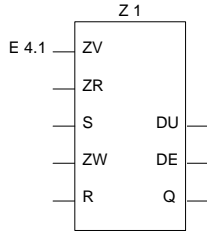
Ausschaltverzögerung			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 3.4 L KT 10.1 SA T 5 U T 5 = A 4.4 </pre>		
<p>Wenn das Verknüpfungsergebnis am Starteingang von "1" nach "0" wechselt, wird die Zeit gestartet. Sie läuft mit der programmierten Zeitdauer ab.</p> <p>Bei Verknüpfungsergebnis "1" wird das Zeitglied auf Null gesetzt (gelöscht).</p> <p>Die Abfragen UT bzw. OT liefern Signalzustand "1", wenn die Zeit läuft <u>oder</u> das Verknüpfungsergebnis am Eingang "1" ist.</p>			
			

Zähloperationen

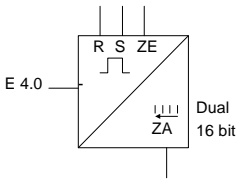
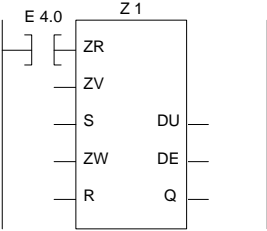
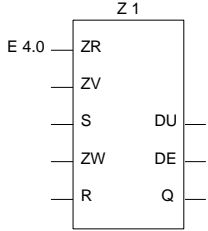
Zähler setzen			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 4.0 ZV Z 1 U E 4.1 L KZ 150 S Z 1 </pre>		
<p>Wenn das Verknüpfungsergebnis am Starteingang (E 4.1) von "0" nach "1" wechselt, wird der Zähler mit dem angegebenen Wert (150) geladen.</p> <p>Der für die Flankenauswertung des Setzeingangs erforderliche Merker ist im Zählwort mitgeführt. DU und DE sind digitale Ausgänge der Zählerzelle. Am Ausgang DU steht der Zählwert dualcodiert, am Ausgang DE BCD-codiert an.</p>			

Zähler rücksetzen			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 4.0 ZR Z 2 U E 4.2 R Z 2 U Z 2 = A 2.4 </pre>		
<p>Bei Verknüpfungsergebnis "1" (E 4.2) wird der Zähler auf Null gesetzt (rückgesetzt).</p> <p>Bei Verknüpfungsergebnis "0" bleibt der Zähler unbeeinflusst.</p>			

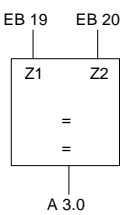
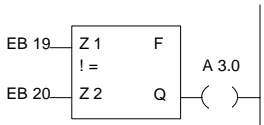
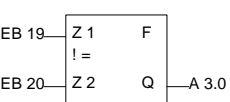
Zähloperationen (Fortsetzung)

Vorwärts zählen			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
 <p style="font-size: small;">Dual 16 bit</p>	<pre> U E 4.1 ZV Z 1 </pre>		
<p>Der Wert des adressierten Zählers wird um 1 erhöht, maximal bis zum Zählwert 999. Die Funktion ZV wird nur bei einer positiven Flanke (von "0" nach "1") der vor ZV programmierten Verknüpfung ausgeführt. Die für die Flankenauswertung der Zählwege erforderlichen Merker sind im Zählwort mitgeführt.</p> <p>Durch die zwei getrennten Flankenmerker für ZV und ZR kann ein Zähler mit zwei verschiedenen Eingängen als Vorwärts-/Rückwärtszähler verwendet werden.</p>			

Zähloperationen (Fortsetzung)

Rückwärts zählen			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
 <p style="font-size: small;">Dual 16 bit</p>	<pre> U E 4.0 ZR Z 1 </pre>		
<p>Der Wert des adressierten Zählers wird um 1 erniedrigt, maximal bis zum Zählwert 0. Die Funktion wird nur bei einer positiven Flanke (von "0" nach "1") der vor ZR programmierten Verknüpfung wirksam. Die für die Flankenauswertung der Zählwege erforderlichen Merker sind im Zählwort mitgeführt.</p> <p>Durch die zwei getrennten Flankenmerker für ZV und ZR kann ein Zähler mit zwei verschiedenen Eingängen als Vorwärts-/Rückwärtszähler verwendet werden.</p>			

Vergleichsoperationen

Vergleich auf gleich			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> L EB 19 L EB 20 ! = F = A 3.0 </pre>		
<p>Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Der Vergleich ergibt ein binäres Verknüpfungsergebnis. VKE = "1": Vergleich ist erfüllt, wenn AKKU-1-L gleich AKKU-2-L VKE = "0": Vergleich ist nicht erfüllt, wenn AKKU-1-L ungleich AKKU-2-L Die Anzeigen ANZ 1 und ANZ 0 werden, wie in der Operationsliste beschrieben, gesetzt. AKKU-2-H und AKKU-1-H bleiben beim 16-bit-Festpunktvergleich an der Operation unbeteiligt. Beim 32-bit-Festpunktvergleich (!=D) und Gleitpunktvergleich (!=G) werden die gesamten Inhalte von AKKU 1 und AKKU 2 (32 bit) miteinander verglichen. Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d. h., der Inhalt von AKKU-1-L und AKKU-2-L wird hier als Festpunktzahl interpretiert.</p>			

Vergleichsoperationen
(Fortsetzung)

Vergleich auf ungleich			
Aufgabenstellung	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	<pre> L EB 21 L DW 3 >< F = A 3.1 </pre>		
<p>Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Der Vergleich ergibt ein binäres Verknüpfungsergebnis. VKE = "1": Vergleich ist erfüllt, wenn AKKU-1-L ungleich AKKU-2-L VKE = "0": Vergleich ist nicht erfüllt, wenn AKKU-1-L gleich AKKU-2-L Die Anzeigen ANZ 1 und ANZ 0 werden, wie am Anfang des Kapitels 3.5 beschrieben, gesetzt. AKKU-2-H und AKKU-1-H bleiben beim 16-bit-Festpunktvergleich an der Operation unbeteiligt. Beim 32-bit-Festpunktvergleich und Gleitpunktvergleich sind auch AKKU-2-H und AKKU-1-H beim Vergleich beteiligt. Entsprechendes gilt für die Vergleiche auf größer, größer gleich, kleiner und kleiner gleich (siehe Operationsliste). Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d. h., der Inhalt von AKKU-1-L und AKKU-2-L wird hier als Festpunktzahl interpretiert.</p>			

3.5.3 Ergänzende Operationen

Den ergänzenden Operationsvorrat können Sie auf dem Programmiergerät nur für Funktionsbausteine (FB und FX) verwenden. Der Gesamtoperationsvorrat für Funktionsbausteine besteht daher aus den Grundoperationen und den ergänzenden Operationen.

Zu den ergänzenden Funktionen gehören auch Systemoperationen: Mit den Systemoperationen können Sie z. B. den Speicher an beliebiger Stelle überschreiben oder den Inhalt der Arbeitsregister der CPU verändern. Systemoperationen können nur programmiert werden, wenn sie in der Voreinstellungsmaske des Programmiergerätes freigegeben werden (ab S5-DOS, Version 2.0, nicht mehr notwendig).

Beachten Sie zum Thema "Systemoperationen" das Kapitel 9 "Speicherzugriffe".



Vorsicht

Systemoperationen sollten nur von erfahrenen Programmierern und Systemkennern und nur mit äußerster Vorsicht angewendet werden.

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

Im folgenden werden die ergänzenden Operationen beschrieben. Zusätzlich sind die Kombinationsmöglichkeiten der Substitutionsbefehle mit den Aktualoperanden angegeben.

Systemoperationen

Systemoperationen sind in der ersten Spalte der Tabellen mit



Binäre Verknüpfungen

Tabelle 3-11 Binäre Verknüpfungen mit Formaloperanden

Operation	Operand	Funktion
U =	<input type="checkbox"/>	UND-Funktion, Abfrage eines Formaloperanden auf Signalzustand '1'
UN =	<input type="checkbox"/>	UND-Funktion, Abfrage eines Formaloperanden auf Signalzustand '0'
O =	<input type="checkbox"/>	ODER-Funktion, Abfrage eines Formaloperanden auf Signalzustand '1'
ON =	<input type="checkbox"/> ↑ Formaloperand einsetzen	ODER-Funktion, Abfrage eines Formaloperanden auf Signalzustand '0'
<p>Als Aktualoperanden sind binär adressierte Eingänge, Ausgänge, Daten und Merker (Parameterart: E, A ; Parametertyp: BI) sowie Zeiten und Zähler (Parametertyp: T, Z) zugelassen.</p>		

Digitalverknüpfungen

Tabelle 3-12 Digitalverknüpfungen

Operation	Operand	Funktion
UW		UND-Verknüpfung von AKKU-1-L und AKKU-2-L
OW		ODER-Verknüpfung von AKKU-1-L und AKKU-2-L
XOW		Exklusiv-ODER-Verknüpfung von AKKU-1-L und AKKU-2-L

Die AKKUs 2, 3 und 4 werden nicht beeinflusst, jedoch die Anzeigen ANZ 1 und ANZ 0 (siehe Wort-Ergebnisanzeigen).

Bittestoperationen

Tabelle 3-13 Bittestoperationen

Operation	Operand	Funktion
P	E 0.0 bis 127.7 A 0.0 bis 127.7 M 0.0 bis 255.7 D 0.0 bis 255.15 T 0.0 bis 255.15 Z 0.0 bis 255.15 BA 0.0 bis 255.15 BB 0.0 bis 255.15 BS 0.0 bis 255.15 BT 0.0 bis 255.15	Abfrage auf Signalzustand "1" eines Eingangs (PAE) eines Ausgangs (PAA) eines Merkers eines Datenwort-Bits eines Zeitwort-Bits eines Zählerwort-Bits eines Bits im BA-Bereich eines Bits im BB-Bereich eines Bits im BS-Bereich eines Bits im BT-Bereich
PN	E 0.0 bis 127.7 A 0.0 bis 127.7 M 0.0 bis 255.7 D 0.0 bis 255.15 T 0.0 bis 255.15 Z 0.0 bis 255.15 BA 0.0 bis 255.15 BB 0.0 bis 255.15 BS 0.0 bis 255.15 BT 0.0 bis 255.15	Abfrage auf Signalzustand "0" eines Eingangs (PAE) eines Ausgangs (PAA) eines Merkers eines Datenwort-Bits eines Zeitwort-Bits eines Zählerwort-Bits eines Bits im BA-Bereich eines Bits im BB-Bereich eines Bits im BS-Bereich eines Bits im BT-Bereich

Die Bittestoperationen fragen den Zustand des Bits ab und zeigen ihn über das VKE an.

Speicheroperationen

Tabelle 3-14 Speicheroperationen mit Formaloperanden


Operation	Operand	Funktion
S =	<input type="text"/>	Setzen (binär) eines Formaloperanden
RB =	<input type="text"/>	Rücksetzen (binär) eines Formaloperanden
RD=	<input type="text"/>	Rücksetzen (digital) eines Formaloperanden für Zeiten und Zähler
= =	<input type="text"/> 	Zuweisen des Verknüpfungsergebnisses an einen Formaloperanden Formaloperand einsetzen
Als Aktualoperanden sind binär adressierte Eingänge, Ausgänge und M-Merker zugelassen (Parameterart: E, A; Parametertyp: BI).		

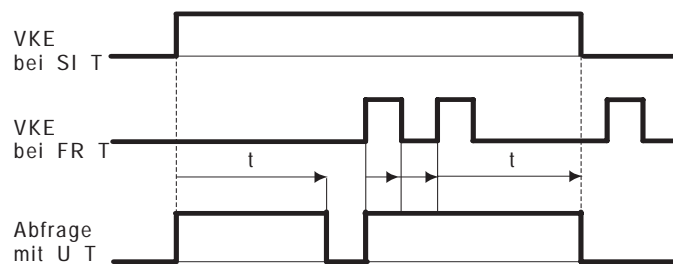
Tabelle 3-15 Setz- und Rücksetzoperationen

Operation	Operand	Funktion
SU	E 0.0 bis 127.7 A 0.0 bis 127.7 M 0.0 bis 255.7 D 0.0 bis 255.15 T 0.0 bis 255.15 Z 0.0 bis 255.15 BA 0.0 bis 255.15 BB 0.0 bis 255.15 BS 60.0 bis 63.15 BT 0.0 bis 255.15	unbedingtes Setzen eines Eingangs (PAE) eines Ausganges (PAA) eines Merkers eines Datenwort-Bits eines Zeitwort-Bits eines Zählerwort-Bits eines Bits im BA-Bereich eines Bits im BB-Bereich eines Bits im BS-Bereich eines Bits im BT-Bereich
RU	E 0.0 bis 127.7 A 0.0 bis 127.7 M 0.0 bis 255.7 D 0.0 bis 255.15 T 0.0 bis 255.15 Z 0.0 bis 255.15 BA 0.0 bis 255.15 BB 0.0 bis 255.15 BS 60.0 bis 63.15 BT 0.0 bis 255.15	unbedingtes Rücksetzen eines Eingangs (PAE) eines Ausganges (PAA) eines Merkers eines Datenwort-Bits eines Zeitwort-Bits eines Zählerwort-Bits eines Bits im BA-Bereich eines Bits im BB-Bereich eines Bits im BS-Bereich eines Bits im BT-Bereich

Zeit- und Zähloperationen

Tabelle 3-16 Zeit- und Zähloperationen mit Formaloperanden

Operation	Operand	Funktion
SI =	<input type="text"/>	Eine Zeit (Formaloperand) als Impuls starten. Der Zeitwert muß in AKKU-1-L hinterlegt sein (Parameterart T).
SE =	<input type="text"/>	Eine Zeit (Formaloperand) einschaltverzögernd starten. Der Zeitwert muß in AKKU-1-L hinterlegt sein (Parameterart T).
SVZ =	<input type="text"/>	Eine Zeit (Formaloperand) als verlängerten Impuls starten mit dem im AKKU-1-L hinterlegten Zeitwert/einen Zähler (Formaloperand) setzen mit dem im AKKU-1-L hinterlegten Zählwert; (Parameterart: T, Z).
SSV =	<input type="text"/>	Eine Zeit (Formaloperand) als speichernde Einschaltverzögerung starten mit dem im AKKU-1-L hinterlegten Wert bzw. Vorwärtszählen eines Zählers (Formaloperand); (Parameterart: T, Z).
SAR =	<input type="text"/>	Eine Zeit (Formaloperand) als Ausschaltverzögerung starten mit dem im AKKU-1-L hinterlegten Wert bzw. Rückwärtszählen eines Zählers (Formaloperand); (Parameterart: T, Z).
FR =	<input type="text"/>	Formaloperanden (Zeit/Zähler) für den Neustart freigeben (siehe "FR T", "FR Z"); (Parameterart: T, Z).
	↑ _____	Formaloperand einsetzen
FR	T 0 bis 255	Eine Zeit für den Neustart freigeben: Die Operation wird nur bei steigender Flanke des VKE ausgeführt (Wechsel von '0' auf '1'). Die Zeit wird neu gestartet, wenn bei der Startoperation das VKE = '1' ist. (siehe Skizze mit Zeitdiagramm unter der Tabelle)
	Z 0 bis 255	Einen Zähler für das Setzen bzw. Zählen freigeben: Die Operation wird nur bei steigender Flanke des VKE ausgeführt (Wechsel von '0' auf '1'). Der Zähler wird neu bearbeitet, wenn bei der Zähloperation das VKE = '1' ist.








Beispiele

Funktionsbausteinanruf	Programm im Funktionsbaustein	ausgeführtes Programm
a)		
<pre> :SPA FB 203 NAME :BEISP1 ANNA : E 10.3 BERT : T 17 HANS : A 18.4 </pre>	<pre> :U =ANNA :L KT 010.2 :SSV =BERT :U =BERT := =HANS </pre>	<pre> :U E 10.3 :L KT 010.2 :SS T 17 :U T 17 := A 18.4 </pre>
b)		
<pre> :SPA FB 204 NAME :BEISP2 MAXI : E 10.5 IRMA : E 10.6 EVA : E 10.7 DORA : Z 15 EMMA : M 58.3 </pre>	<pre> :U =MAXI :SSV =DORA :U =IRMA :SAR =DORA :U =EVA :L KZ 100 :SVZ =DORA :UN =DORA := =EMMA </pre>	<pre> :U E 10.5 :ZV Z 15 :U E 10.6 :ZR Z 15 :U E 10.7 :L KZ 100 :S Z 15 :UN Z 15 := M 58.3 </pre>
c)		
<pre> :SPA FB 205 NAME :BEISP3 KURT : E 10.4 KARL : T 18 EGON : EW 20 MAUS : M 100.7 </pre>	<pre> :U =KURT :L =EGON :SVZ =KARL :U =KARL := =MAUS </pre>	<pre> :U E 10.4 :L EW 20 :SV T 18 :U T 18 := M 100.7 </pre>

Lade- und Transferoperationen

Tabelle 3-17 Lade- und Transferoperationen mit Formaloperanden

Operation	Operand	Beschreibung
L =		Laden eines Formaloperanden: Der Wert des als Formaloperand vorgegebenen Operanden wird in den AKKU geladen (Parameterart: E, T, Z, A; Parametertyp: BY, W, D).
LC =		Codiertes Laden eines Formaloperanden: Der Wert der als Formaloperand vorgegebenen Zeit- oder Zählzelle wird BCD-codiert in den AKKU geladen (Parametertyp: T, Z).
LW =		Laden des Bitmusters eines Formaloperanden: Das Bitmuster des Formaloperanden wird in den AKKU geladen (Parameterart: D; Parametertyp: KF, KH, KM, KY, KC, KT, KZ).
LD =		Laden des Bitmusters eines Formaloperanden: Das Bitmuster des Formaloperanden wird in den AKKU geladen (Parameterart: D; Parametertyp: KG).
T =		Transferieren zu einem Formaloperanden: Der Akkumulatorinhalt wird zu dem als Formaloperand vorgegebenen Operanden transferiert (Parameterart: E, A; Parametertyp: BY, W, D).
		Formaloperand einsetzen

Als Aktualoperanden sind die den Grundoperationen entsprechenden Operanden – **ausgenommen S-Merker** – zugelassen. Bei 'LW=' sind ein Datum in Form eines Binär- (KM) oder eines Hexadezimalmusters (KH), 2 byteweise Betragzahlen (KY), Zeichen (KC), Festpunktzahl (KF), Zeitwert (KT) und Zählwert (KZ) zugelassen. Bei 'LD=' ist eine Gleitpunktzahl als Datum zugelassen.

Tabelle 3-18 Lade- und Transferoperationen mit speziellen Operanden

Operation		Operand	Funktion
L	BA	0 bis 255	Laden eines Wortes in den AKKU 1 aus dem Bereich "Anschaltung" (BA-Bereich)
	BB	0 bis 255	Laden eines Wortes in den AKKU 1 aus dem erweiterten Bereich "Anschaltung" (BB-Bereich)
L	BS	0 bis 255	Laden eines Wortes in den AKKU 1 aus dem Bereich "Systemdaten" (BS-Bereich)
	BT	0 bis 255	Laden eines Wortes in den AKKU 1 aus dem erweiterten Bereich "Systemdaten" (BT-Bereich)
T	BA	0 bis 255	Transferieren des Inhalts von AKKU 1 zu einem Wort des Bereichs "Anschaltung" (BA-Bereich)
	BB	0 bis 255	Transferieren des Inhalts von AKKU 1 zu einem Wort des erweiterten Bereichs "Anschaltung" (BB-Bereich)
T	BS	60 bis 63	Transferieren des Inhalts von AKKU 1 zu einem Wort des Bereichs "Systemdaten" (BS-Bereich)
	BT	0 bis 255	Transferieren des Inhalts von AKKU 1 zu einem Wort des erweiterten Bereichs "Systemdaten" (BT-Bereich)

Im Gegensatz zu den Bereichen BA, BB und BT dürfen vom BS-Bereich nur die Wörter BS 60 bis BS 63 für Anwenderzwecke frei genutzt werden. Beachten Sie dazu den Abschnitt 8.3.4 "BS-/BT-Bereich".

Den BT-Bereich können Sie in seiner gesamten Länge (BT 0 bis BT 255) benutzen, **vorausgesetzt**, Sie verwenden keine Standard-Funktionsbausteine.

Rechenoperationen

Tabelle 3-19 Rechenoperation ENT

Operation	Operand	Funktion
ENT	–	<p>Es findet ein Stack-Lift in die AKKUs 3 und 4 statt:</p> <p><AKKU 4> := <AKKU 3></p> <p><AKKU 3> := <AKKU 2></p> <p><AKKU 2> := <AKKU 2></p> <p><AKKU 1> := <AKKU 1></p> <p>Die AKKUs 1 und 2 werden nicht verändert. Der alte Inhalt des AKKUs 4 geht verloren.</p>

Beispiel

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

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

Tabelle 3-20 Ergänzende arithmetische Operationen

Operation		Operand	Funktion
S	ADD	BF -128 bis +127	Addieren einer Byte-Konstanten (Festpunkt) zum AKKU-1-L (vorzeichenexpandiert)/die Anzeigen in ANZ 0, ANZ 1, OV und OS werden nicht beeinflusst! – AKKU-1-H sowie die AKKUs 2 bis 4 bleiben unverändert.
S	ADD	KF -32 768 bis +32 767	Addieren einer Festpunktkonstanten (Wort) zum AKKU-1-L/ die Anzeigen in ANZ 0, ANZ 1, OV und OS werden nicht beeinflusst! – AKKU-1-H sowie die AKKUs 2 bis 4 bleiben unverändert.
S	ADD ¹⁾	DH 0000 0000 bis FFFF FFFF	Addieren einer Doppelwort-Festpunktkonstanten zum AKKU 1/die Anzeigen in ANZ 0, ANZ 1, OV und OS werden nicht beeinflusst! – Die AKKUs 2 bis 4 bleiben unverändert.
S	+D ¹⁾		Addieren zweier Doppelwort-Festpunktkonstanten (AKKU 2 + AKKU 1)/das Ergebnis ist über ANZ 0/ANZ 1 auswertbar. ²⁾
S	-D ¹⁾		Subtrahieren zweier Doppelwort-Festpunktkonstanten (AKKU 2 - AKKU 1); das Ergebnis ist über ANZ 0/ANZ 1 auswertbar. ²⁾
S	TAK		Tauschen der Inhalte von AKKU 1 und AKKU 2

¹⁾ Die Programmierung ist abhängig vom PG-Typ und vom Ausgabestand der PG-Systemsoftware.

²⁾ Veränderungen von AKKU 2 und AKKU 3: siehe Abschnitt 3.5.1 "Grundoperationen/arithmetische Operationen".

3.5.4 Organisatorische Operationen



Zu den organisatorischen Operationen gehören auch Systemoperationen.

Vorsicht

Systemoperationen sollten nur von erfahrenen Programmierern und Systemkennern und nur mit äußerster Vorsicht angewendet werden.

Systemoperationen sind in der ersten Spalte der Tabellen mit **S** gekennzeichnet!

Sprungoperationen

Das Sprungziel für unbedingte und bedingte Sprünge wird symbolisch angegeben (maximal 4 Zeichen, beginnend mit einem Buchstaben). Dabei ist der Symbolparameter des Sprungbefehls identisch mit der Symboladresse der anzuspringenden Anweisung. Bei der Programmierung muß berücksichtigt werden, daß die absolute Sprungdistanz nicht mehr als ± 127 Wörter umfaßt und eine STEP-5-Anweisung aus mehr als einem Wort bestehen kann. Sprünge dürfen nur innerhalb eines Bausteins durchgeführt werden; Sprünge über "Netzwerke" hinweg sind unzulässig ("Netzwerk" = Gliederungselement bei PB, SB, FB, FX und OB, siehe PG-Beschreibung).

Hinweis

Sprunganweisung und Sprungziel (Symboladresse) müssen im **gleichen** Netzwerk liegen. Pro Netzwerk darf der Name einer Symboladresse nur **einmal** vergeben werden.

Ausnahme: Dies gilt nicht für den Sprung SPR, bei dem als Parameter eine absolute Sprungdistanz angegeben wird.

Tabelle 3-21 Sprungoperationen

Operation	Operand	Funktion
SPA =	adr	Sprung unbedingt: Der unbedingte Sprung wird unabhängig von Bedingungen ausgeführt.
SPB =	(adr =Symbol- adresse mit maximal 4 Zeichen)	Sprung bedingt: Der bedingte Sprung wird ausgeführt, wenn VKE = 1 ist. Bei VKE = 0 wird die Anweisung nicht ausgeführt und das Verknüpfungsergebnis auf VKE = 1 gesetzt.
SPZ =		Sprung bei Ergebnis '0' : Der Sprung wird nur dann ausgeführt, wenn ANZ 1 = 0 und ANZ 0 = 0 ist. Das Verknüpfungsergebnis wird nicht verändert.

Operation	Operand	Funktion
Fortsetzung der Tabelle 3-21:		
SPN =	adr (adr = Symbol- adresse mit maximal 4 Zeichen)	Sprung bei Ergebnis $\neq 0$: Der Sprung wird nur dann ausgeführt, wenn ANZ 1 \neq ANZ 0 ist. Das Verknüpfungsergebnis wird nicht verändert.
SPP =		Sprung bei Ergebnis $> '0'$: Der Sprung wird nur dann ausgeführt, wenn ANZ 1 = 1 und ANZ 0 = 0 ist. Das Verknüpfungsergebnis wird nicht verändert.
SPM =		Sprung bei Ergebnis $< '0'$: Der Sprung wird nur dann ausgeführt, wenn ANZ 1 = 0 und ANZ 0 = 1 ist. Das Verknüpfungsergebnis wird nicht verändert.
SPO =		Sprung bei Überlauf (Overflow): Der Sprung wird ausgeführt, wenn die Anzeige OV = 1 ist. Wenn kein Überlauf vorliegt (OV = 0), wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert. Ein Überlauf entsteht, wenn bei gegebener Zahlendarstellung der zulässige Bereich durch eine arithmetische Operation überschritten wird.
SPS =		Sprung, wenn die Anzeige OS (Overflow speichernd) gesetzt ist : Der Sprung wird ausgeführt, wenn die Anzeige OS = 1 ist. Wenn kein Überlauf vorliegt (OS = 0), wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert. Ein Überlauf entsteht, wenn bei gegebener Zahlendarstellung der zulässige Bereich im Verlauf mehrerer arithmetischer Operationen überschritten wird.
S SPR	-32 768 bis +32 767	Beliebiger relativer Sprung innerhalb des Anwenderspeichers bzw. innerhalb eines Funktionsbausteins (z. B. um in ein anderes Netzwerk zu gelangen). Die Operation wird immer ausgeführt, unabhängig von Bedingungen. Als Operand muß die Adressen-Differenz "Sprungziel - aktuelle Operation" (Anzahl Wörter) angegeben werden. Dabei wird der Sprung zu einer höheren (positiver Operand) oder niedrigeren (negativer Operand) Adresse als die der aktuellen Operation ausgeführt.



Vorsicht

Bei unsachgemäßer Anwendung der Operation **SPR** können undefinierte Zustände der Anlage auftreten! Sie sollte daher nur von sehr erfahrenen Programmierern und Systemkennern benutzt werden.

Schiebeoperationen

Tabelle 3-22 Schiebeoperationen

Operation	Operand	Funktion (Operation mit AKKU 1)
SLW	0 bis 15	Schieben nach links (von rechts werden Nullen nachgezogen)
SRW	0 bis 15	Schieben nach rechts (von links werden Nullen nachgezogen)
SLD	0 bis 32	Schieben eines Doppelwortes nach links (von rechts werden Nullen nachgezogen)
SVW	0 bis 15	Schieben mit Vorzeichen nach rechts (von links wird Bit 15 nachgezogen)
SVD	0 bis 32	Schieben eines Doppelwortes mit Vorzeichen nach rechts (von links wird Bit 31 nachgezogen)
RLD	0 bis 32	Rotieren nach links
RRD	0 bis 32	Rotieren nach rechts

Bei den Schiebeoperationen ist nur der AKKU 1 an der Ausführung beteiligt. Der Parameterteil dieser Operationen gibt an, um wieviele Stellen der AKKU-Inhalt geschoben bzw. rotiert wird. Bei SLW, SRW und SVW ist nur das niederwertige Wort an den Schiebeoperationen beteiligt, bei SLD, SVD, RLD und RRD der gesamte Inhalt des AKKU 1 (32 bit).

Die Schiebeoperationen werden unabhängig von Bedingungen ausgeführt.

Der Wert des zuletzt hinausgeschobene Bits kann über ANZ 1/ANZ 0 mit Sprungoperationen abgefragt werden:

Schieben: letztes geschobenes Bit	ANZ 1	ANZ 0	Sprungoperation
0	0	0	SPZ=
1	1	0	SPN= SPP=

Beispiele

1. Der Inhalt des Datenwortes DW 52 soll gelesen, um 4 bit nach links verschoben und dann im Datenwort DW 53 abgelegt werden.

STEP-5-Programm: Inhalt der Datenwörter:

```
:L   DW 52           KH = 14AF
:SLW 4
:T   DW 53           KH = 4AF0
```

2. Das Eingangs-Doppelwort ED 0 soll gelesen, und der Inhalt von AKKU 1 so verschoben werden, daß die fettgedruckten Bitstellen des Eingangs-Doppelwortes (z. T. verschoben) erhalten bleiben und die übrigen Bitstellen mit definierten Werten (0 H bzw. 0F H) besetzt werden.

STEP-5-Programm: Inhalt von AKKU 1 (hexadezimal)

	AKKU-1-H:	AKKU-1-L:
:L ED 0	2348	ABCD
:SLW 4	2348	BCD0
:SRW 4	2348	0BCD
:SLD 4	3480	BCD0
:SVW 4	3480	FBCD
:SVD 4	0348	0FBC
:RLD 4	3480	FBC0
:RRD 4	0348	0FBC

3. Anwendung: Multiplikation mit 2er-Potenz, z. B. neuer Wert = alter Wert x 8

```
:L   MW 10
:SLW 3
:T   MW 10            Achtung: Positive Bereichsgrenze
                       nicht überschreiten!
```

4. Anwendung: Division durch 2er-Potenz, z. B. neuer Wert = alter Wert : 4

```
:A   DB 5
:L   DW 0
:SRW 2
:T   DW 0
```

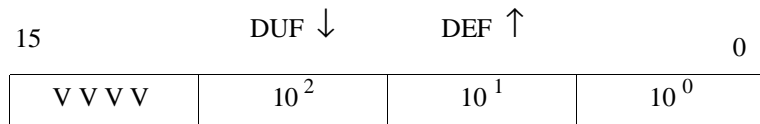
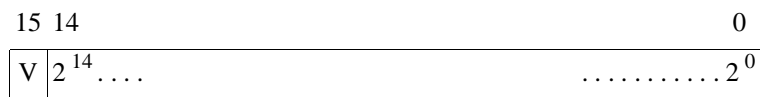
Umwandlungsoperationen

Tabelle 3-23 Umwandlungsoperationen

Operation	Funktion
KEW	Bildung des 1er-Komplements von AKKU-1-L (16 bit)
KZW	Bildung des 2er-Komplements von AKKU-1-L (16 bit)
KZD	Bildung des 2er-Komplements von AKKU 1 (32 bit)
DEF	Festpunkt wandlung (16 bit) von BCD in dual
DUF	Festpunkt wandlung (16 bit) von dual in BCD
DED	Doppelwort wandlung (32 bit) von BCD in dual
DUD	Doppelwort wandlung (32 bit) von dual in BCD
FDG	Wandlung einer Festpunktzahl (32 bit) in eine Gleitpunktzahl (32 bit)
GFD	Wandlung einer Gleitpunktzahl in eine Festpunktzahl (32 bit)

DEF Der im AKKU-1-L (Bit 0 bis Bit 15) stehende Wert wird als BCD-codierte Zahl interpretiert. Nach der Umwandlung steht im AKKU-1-L eine 16-bit-Festpunktzahl.

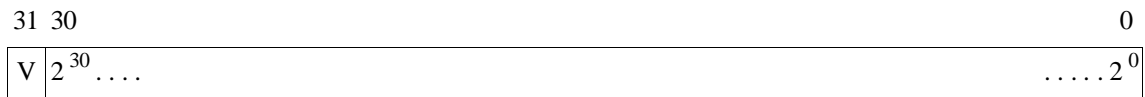
DUF Der im AKKU-1-L (Bit 0 bis Bit 15) stehende Wert wird als 16-bit-Festpunktzahl interpretiert. Nach der Umwandlung steht im AKKU-1-L eine BCD-codierte Zahl.



V (Vorzeichen): 0 = positiv
1 = negativ

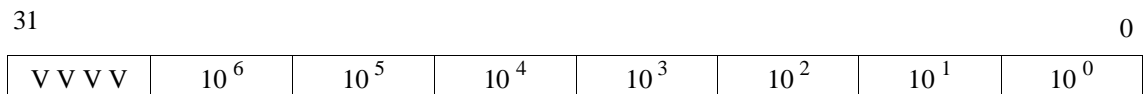
DED Der im AKKU 1 (Bit 0 bis Bit 31) stehende Wert wird als BCD-codierte Zahl interpretiert. Nach der Umwandlung steht im AKKU 1 eine 32-bit-Festpunktzahl.

DUD Der im AKKU 1 (Bit 0 bis Bit 31) stehende Wert wird als 32-bit-Festpunktzahl interpretiert. Nach der Umwandlung steht im AKKU 1 eine BCD-codierte Zahl.



DUD ↓

DED ↑



V (Vorzeichen): 0 = positiv
1 = negativ

FDG Der im AKKU 1 (Bit 0 bis Bit 31) stehende Wert wird als 32-bit-Festpunktzahl interpretiert. Nach der Umwandlung steht im AKKU 1 eine Gleitpunktzahl (Exponent und Mantisse).

GFD Der im AKKU 1 (Bit 0 bis Bit 31) stehende Wert wird als Gleitpunktzahl interpretiert. Nach der Umwandlung steht im AKKU 1 eine 32-bit-Festpunktzahl.



FDG ↓

GFD ↑



Exponent

Mantisse

Die Umwandlung geschieht durch Multiplikation der (dualen) Mantisse mit dem Wert des (dualen) Exponenten, indem der Mantissenwert über einen gedachten Dezimalpunkt um den Wert des Exponenten (zur Basis '2') zu höherwertigen Bitstellen verschoben wird. Nach der Multiplikation bleiben Reste der ursprünglichen Mantisse rechts vom gedachten Dezimalpunkt übrig. Diese Bitstellen werden vom ganzzahligen Ergebnis abgeschnitten.

Durch diesen Umwandlungsalgorithmus ergeben sich folgende Ergebnisklassen:

- Gleitpunktzahlen ≥ 0 oder ≤ -1 ergeben die **nächst kleinere ganze Zahl**.
- Gleitpunktzahlen < 0 und > -1 ergeben den **Wert '0'**.

Umwandlungsbeispiele

Gleitpunktzahl		32-bit-Festpunktzahl
	GFD	
+5,7	→	5
-2,3	→	-3
-0,6	→	0
+0,9	→	0

Beispiele zu KEW, KZW

1. Der Inhalt des Datenwortes 64 soll Bit für Bit invertiert ("umgekehrt") und in Datenwort 78 abgelegt werden.	
STEP-5-Programm:	Belegung der Datenwörter:
:L DW 64	KM = 0011111001011011
:KEW	
:T DW 78	KM = 1100000110100100
2. Der Inhalt des Datenwortes 207 ist als Festpunktzahl zu interpretieren und mit umgekehrtem Vorzeichen im Datenwort 51 abzulegen.	
STEP-5-Programm:	Belegung der Datenwörter:
:L DW 207	KF = +51
:KZW	
:T DW 51	KF = -51

**Dekrementieren/
Inkrementieren**

Tabelle 3-24 Dekrementier-/Inkrementieroperation

Operation	Operand	Funktion
D	1 bis 255	Dekrementieren des Low-Bytes (Bits 0 bis 7) von AKKU-1-L um den Operandenwert ¹⁾
I	1 bis 255	Inkrementieren des Low-Bytes (Bits 0 bis 7) von AKKU-1-L um den Operandenwert ¹⁾

¹⁾ Der Inhalt des Low-Bytes von AKKU-1-L wird um die als Operand angegebene Zahl ohne Übertrag dekrementiert (erniedrigt) bzw. inkrementiert (erhöht). Die Operationsausführung ist unabhängig von Bedingungen.

Beispiel

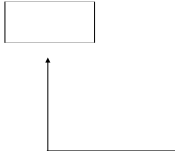
```

STEP-5-Programm:      Belegung der Datenwörter:

:L DW 7              KH = 1010
:I 16
:T DW 8              KH = 1020
:D 33
:T DW 9              KH = 10FF
    
```

Bearbeitungsoperationen

Tabelle 3-25 Bearbeitungsoperationen

Operation	Operand	Funktion
B	DW 0 bis 255	Bearbeite Datenwort: Die nachfolgend angegebene Operation wird mit dem im adressierten Datenwort angegebenen Parameter kombiniert und ausgeführt.
	MW 0 bis 254	Bearbeite Merkerwort: Die nachfolgend angegebene Operation wird mit dem im adressierten M-Merker angegebenen Parameter kombiniert und ausgeführt.
B =		Bearbeite Formaloperanden (Parameterart: B): Nur A DB, SPA PB, SPA OB, SPA FB, SPA SB können substituiert werden. Formaloperanden einsetzen

Operation		Operand	Funktion
Fortsetzung der Tabelle 3-25:			
S	BI	1)	Indirekte Bearbeitung eines Formaloperanden: Eine Operation ausführen, deren Operationscode in einem Formaloperanden abgelegt ist. Die Nummer des auszuführenden Formaloperanden steht im AKKU 1.
	B	BS 60 bis 63 1)	Eine Operation, die im Bereich Systemdaten (BS) steht, soll ausgeführt werden (freie Systemdaten: BS 60 bis 63). Bei 2-Wort-Befehlen muß das 2. Wort nach BS n+1 geladen werden.

1) Der Wert, der im Systemdatum oder im Formaloperanden steht, wird als Operationscode einer STEP-5-Operation interpretiert, die dann ausgeführt wird.

Hinweis

Mit **B DW**, **B MW**, **BI** oder **B BS** dürfen nur folgende Operationen kombiniert werden:

- U.. , UN.. , O.. , ON.. , S.. , R.. , =..
mit den Bereichen E, A, M, S,
- FR T, R T, SA T, SE T, SI T, SS T, SV T,
- FR Z, R Z, S Z, ZR Z, ZV Z,
- L.., T.. mit den Bereichen P, Q, E, A, M, S, D, BA, BB, BS, BT,
- L T, L Z,
- LC T, LC Z,
- SPA=, SPB=, SPZ=, SPN=, SPP=, SPM=, SPO=,
- SLW, SRW,
- D, I, SES, SEF,
- A DB, SPA.. , SPB..., E DB, EX DX, AX DX, BAB FX, BA FX.

Das PG prüft die Zulässigkeit der Kombinationen nicht!

**Beispiele zu den
Bearbeitungs-Operationen**

B DW/B MW

Operanden-Substitution

Mit den Anweisungen "B DW" und "B MW" können Sie, z. B. in einer Programmschleife, substituiert auf Daten zugreifen. Der substituierte Zugriff setzt sich zusammen aus der Anweisung B DW/B MW und einer unmittelbar nachfolgenden STEP-5-Operation aus dem o. g. Operationsspektrum.

"Substituiert" bedeutet, daß der Operand für die Operation nicht statisch beim Programmieren vorgegeben, sondern erst beim Ablauf Ihres STEP-5-Programms festgelegt wird.

Den Operandentyp wählen Sie beim Programmieren aus dem für die Operation zulässigen Spektrum aus, z. B. **PB** für die Operation "SPA PB nn".

Den Operandenwert (**nn** im Beispiel "SPA PB nn") müssen Sie vor einem substituierten Zugriff mit B DW/B MW in ein Daten- oder M-Merkerwort (Parameterwort) laden.

1. Prinzip der Substitution:

```

:L   KF +120
:T   MW 14      MW 14 mit dem Wert "KF +120" laden
:B   MW 14
:L   EB 0

```

→ vor Ausführung der Operation "L EB" wird der Operandenwert '0' durch den Wert '120' ersetzt;
Ausführung: **L EB 120**

2. Datenwort als Indexregister:

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.

```

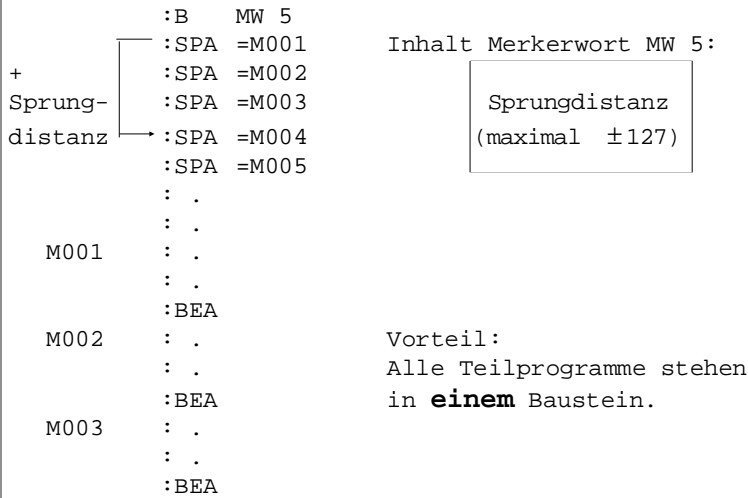
:L   KF +20      Versorgung des Indexregisters
:T   DW 1
M001 :L   KF +0      Rücksetzen
      :B   DW 1
      :T   DW 0
      :L   DW 1      Erhoehen des Indexregisters
      :L KF +1
      :+F
      :T   DW 1
      :L   KF +100
      :<=F
      :SPB =M001    Sprung, wenn Index im Bereich liegt
      ...          weiteres STEP-5-Programm

```

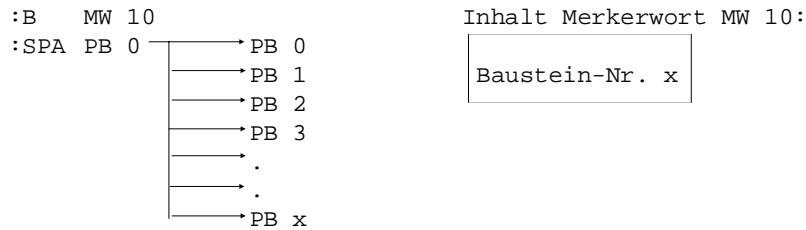
Fortsetzung auf der nächsten Seite

Fortsetzung der Beispiele zur Operanden-Substitution:

3. Sprungverteiler für Unterprogrammtechnik:



4. Sprungverteiler für Bausteinaufrufe:



Operanden-Substitution mit binären Operationen

Bei Operanden-Substitutionen mit **binären** Operationen können Sie folgende Operandentypen verwenden: Eingänge, Ausgänge, M-Merker, S-Merker, Zeiten und Zähler.

Der Aufbau des M-Merker- oder Datenwortes (Parameterwort) hängt bei dieser Substitution davon ab, welchen Operandentyp Sie verwenden.

Parameterwort für Ein- und Ausgänge

Bit-Nr.	15	11	10	8	7	6	0
	ohne Bedeutung		Bit-Adresse von 0 bis 7		0	Byteadresse von 0 bis 127	

Parameterwort für M-Merker

Bit-Nr.	15	11	10	8	7	0
	ohne Bedeutung		Bit-Adresse von 0 bis 7		Byteadresse von 0 bis 255	

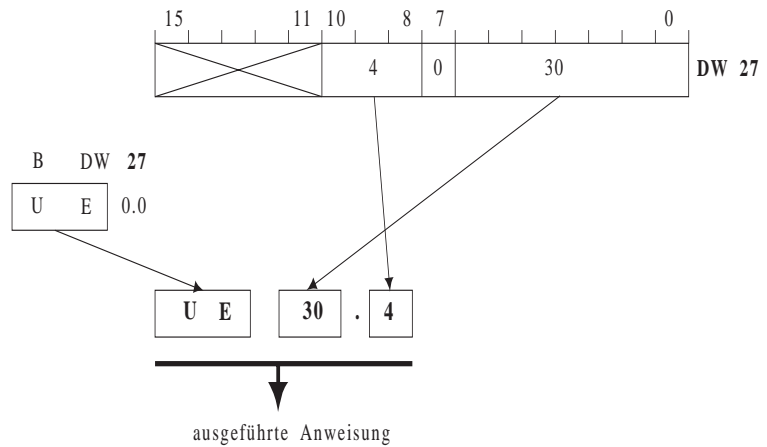
Parameterwort für S-Merker

Bit-Nr.	15	14	12	11	0
	0	Bit-Adresse von 0 bis 7		Byteadresse von 0 bis 4095	

Parameterwort für Zeiten und Zähler

Bit-Nr.	15	8	7	0
	ohne Bedeutung		Nummer von Zeit- oder Zählerzelle von 0 bis 255	

Prinzip der Substitution mit einer binären Operation



Beispiel zur BI-Operation

Im Funktionsbaustein FB 1 werden STEP-5-Operationen ausgeführt, deren Operations-Codes als Formaloperanden MW 10, MW 12 und MW 14 von einem aufrufenden Baustein übergeben werden.

Welcher der Operationscodes ausgeführt werden soll, wird vom aufrufenden Baustein als lfd. Nummer im Merkerwort **MW 16** hinterlegt.

Das Ergebnis der ausgeführten Operation befindet sich anschließend in AKKU 1 und wird in das Merkerwort MW 18 transferiert.

FB 1

NAME :TEST

BEZ :MW10 E/A/D/B/T/Z: D KM/KH/KY/KC/KF/KT/KZ/KG: KH
 BEZ :MW12 E/A/D/B/T/Z: D KM/KH/KY/KC/KF/KT/KZ/KG: KH
 BEZ :MW14 E/A/D/B/T/Z: D KM/KH/KY/KC/KF/KT/KZ/KG: KH

:L MW 16 lfd. Nr. des Formaloperanden mit dem
 : gewuenschten Operationscode

:BI uebergabener Operationscode wird ausgeuehrt

:T MW 16 Ergebnis aus AKKU 1

:BE

FB 2

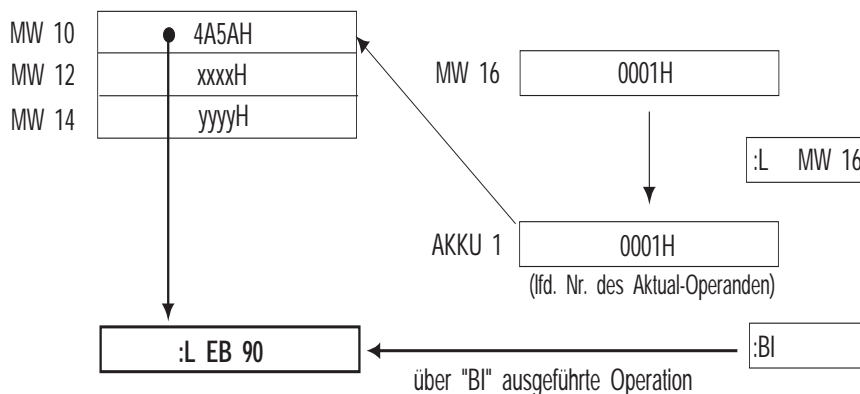
:
 :L KF +1
 :T MW 16 lfd. Nr. des Formaloperanden mit Operationscode
 :SPA =AUFR
 :

AUFR :
 :SPA FB 1 FB TEST aufrufen

NAME :TEST
 MW10 : KH 4A5A Operationscode "L EB 90", Formal-Operand 1
 MW12 : KH xxxx anderer Operationscode, Formal-Operand 2
 MW14 : KH yyyy anderer Operationscode, Formal-Operand 3
 :T MW 18 AKKU 1 → MW 18
 :BE

Liste der Aktual-Operanden im FB 2

Ablaufprinzip im FB 1



Peripherie-Operationen

Tabelle 3-26 Peripherie-Operationen

Operation	Operand	Funktion
AS		Prozeßalarmbearbeitung (über EB 0) sperren (nicht für System-interrupts!)
AF		Prozeßalarmbearbeitung (über EB 0) freigeben (nicht für System-interrupts!)
AFS		Adressierfehler sperren
AFF		Adressierfehler freigeben
BAS		Befehlsausgabe sperren: PAA wird durch die Operationen S A, R A, = A, T PY und T PW nicht mehr beeinflußt.
BAF		Befehlsausgabe freigeben

"Prozeßalarme sperren/freigeben" kann z. B. angewendet werden, wenn bei einer zeitgesteuerten Bearbeitung die prozeßalarmgesteuerte Bearbeitung unterdrückt werden soll. In dem Programmteil, der zwischen den Anweisungen AS und AF steht, ist dann die prozeßalarmgesteuerte Bearbeitung nicht mehr möglich.
Beachten Sie hierzu die Sonderfunktion OB 122 "Alarme sperren", Abschnitt 6.3.

Sonstige Operationen

Tabelle 3-27 Sonstige Operationen

Operation	Operand	Funktion
S STS		Stoppbefehl, der direkt in den weichen STOP führt.
STW		Stoppbefehl, der direkt in den harten STOP führt (nur über NETZ AUS/NETZ EIN zu verlassen).
SIM		Interrupt-Maske (UAMW) setzen (32 bit): Vor Aufruf der Operation muß das Bitmuster für die Maske in den AKKU 1 (32 bit) geladen werden.
LIM		Interrupt-Maske (UAMW) lesen: Bitmuster der Interrupt-maske (32 bit) wird in den AKKU 1 geladen.

*SIM/LIM – Unterbrechungs-
anzeigenmaske (UAMW)
setzen/lesen*

Die Unterbrechungsanzeigen-Maske "maskiert" Unterbrechungen und Alarime im Unterbrechungsanzeigenwort bis zum Zyklusende, d. h. alle Unterbrechungen und Alarime stehen zwar weiterhin an, das ablaufende Programm wird jedoch von ihnen nicht unterbrochen.

Bit in der Unterbrechungsanzeigenmaske = 0: Interrupt ist gesperrt
Bit in der Unterbrechungsanzeigenmaske = 1: Interrupt ist freigegeben

Bedeutung der Bits in UAMW-H bzw. AKKU-1-H:

15																0
	INTX	INTE	INTF	INTG	WEFE	WA	PA	BULE	PEU	HALT	ES	AV	INTAS	TAU	DARY	KZU

Bedeutung der Bits in UAMW-L bzw. AKKU-1-L:

15																0
	-	KB	KDB	STS	TLAF	SUF	STUEB	STUEU	NAU	ZA	QVZ	ADF	PARE	ZYK	STOP	HOLD

Tabelle 3-28 Bedeutung der Abkürzungen in UAMW

Abkürzung	Bedeutung
High-Wort	
INTX	S5-Bus-/System-Interrupt A, B, C oder D (steckplatzabhängig)
INTE	S5-Bus-/System-Interrupt E
INTF	S5-Bus-/System-Interrupt F
INTG	S5-Bus-/System-Interrupt G
WEFE	Weckfehler
WA	Weckalarm
PA	Prozeßalarm
BULE	Buslock Error
PEU	Peripherie unklar
HALT	Stoppanweisung vom Koordinator KOR
ES	Einzelschrittbetrieb
AV	Adreßvergleich aktiv
INTAS	Interrupt vom SPU--Prozessor
TAU	Taktausfall des SPU-Prozessors
DARY	Dauer-Ready (Zugriff auf fehlerhaften Speicher)
KZU	Klammerzähler-Überlauf

Abkürzung	Bedeutung
Fortsetzung der Tabelle 3-28:	
Low-Wort	
KB	kein Baustein
KDB	kein Datenbaustein
STS	befehlsgenauer Stopp
TLAF	Transfer-/Ladefehler
SUF	Substitutionsfehler
STUEB	BSTACK-Überlauf
STUEU	USTACK-Überlauf
NAU	Netzausfall
ZA	Zeitalarm (Verzögerungsalarm, uhrzeitgesteuerter Weckalarm)
QVZ	Quittungsverzug
ADF	Adressierfehler
PARE	Parity-Fehler
ZYK	Zykluszeitfehler
STOP	Betriebsartenschalter nach STOP betätigt
HOLD	DMA-Anforderung vom SPU-Prozessor

3.5.5 Semaphor-Operationen

Benutzen zwei oder mehrere CPUs im Mehrprozessorbetrieb (siehe Kapitel 10) eines Automatisierungsgerätes bestimmte globale Speicherbereiche (Peripherie, CPs, IPs) gemeinsam, besteht die Gefahr, daß die CPUs einander Daten überschreiben oder daß ungültige Zwischenstände der Daten ausgelesen werden. Deshalb ist es erforderlich, den Zugriff der CPUs auf die gemeinsamen Speicherbereiche zu koordinieren.

Die Koordinierung der einzelnen CPUs ist mit Semaphoren durch die Befehle SES und SEF möglich:

Es kann z. B. folgende Koordinierung zweier CPUs programmiert werden: Nur nach erfolgreichem Setzen eines vereinbarten Semaphors (SES) greift jede der am Mehrprozessorbetrieb beteiligten CPUs auf den gemeinsamen Speicherbereich zu. Ein Semaphor xx kann dabei immer nur durch eine einzige CPU gesetzt werden. Gelingt einer CPU das Setzen des Semaphors nicht, so muß sie auf den Zugriff verzichten. Ebenso muß eine CPU auf einen weiteren Zugriff verzichten, nachdem sie den Semaphor wieder freigegeben hat (SEF).

SES/SEF Semaphor setzen/freigeben

(keine Systemoperationen)

Tabelle 3-29 Semaphor setzen/freigeben

Operation	Operand	Funktion
SES	0 bis 31	Setzen eines Semaphors
SEF	0 bis 31	Freigeben eines Semaphors
		Auswertung der Operationsergebnisse über ANZ 0/ANZ 1

Hinweis

Die Operationen SES xx und SEF xx müssen in **allen CPUs** programmiert werden, die synchronisiert auf einen **gemeinsamen** globalen Speicherbereich zugreifen sollen.

Standard-FB, Hantierungsbausteine und Bausteine für Mehrprozessor-Kommunikation verwalten die Koordinierung intern. Sie brauchen bei der Anwendung dieser Bausteine die Operationen SES xx und SEF xx nicht zu programmieren.

Wirkung von SES/SEF

Die Operation SES xx (Semaphor setzen) belegt für die befehlsausführende CPU ein bestimmtes Byte im Koordinator (**vorausgesetzt**, dieses ist nicht bereits durch eine andere CPU belegt). Solange sich die CPU dort eingetragen hat, dürfen die übrigen CPUs auf den mit dem Semaphor (Nummer 0 bis 31) geschützten Speicherbereich nicht mehr zugreifen. Der Bereich ist damit für alle anderen CPUs gesperrt.

Damit diese Koordination richtig funktioniert, müssen alle CPUs, die auf denselben Bereich auf dem globalen Speicher zugreifen wollen, denselben Semaphor benutzen.

Der Befehl SEF xx (Semaphor freigeben) setzt das Byte auf dem Koordinator wieder zurück. Dadurch wird der geschützte Speicherbereich für die anderen CPUs durch Semaphorsetzen wieder belegbar. Ein Semaphor kann nur von derjenigen CPU freigegeben werden, von der er gesetzt wurde.

Anwendung von SES/SEF

Bild 3-8 zeigt den prinzipiellen Ablauf einer Zugriffs koordinierung

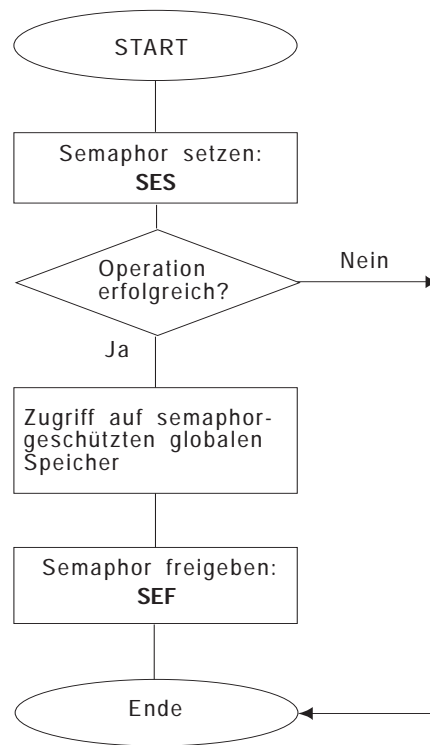


Bild 3-8 Koordination des Zugriffs auf den globalen Speicher

mit Hilfe eines Semaphors.

Vor jedem Setzen bzw. Freigeben eines bestimmten Semaphors fragen die Befehle SES und SEF den **Zustand** (= Status) dieses Semaphors ab. Die Anzeigen ANZ 0 und ANZ 1 werden dabei wie folgt beeinflusst:

ANZ 1	ANZ 0	Auswertung	Bedeutung
0	0	SPZ	Semaphor ist von anderer CPU gesetzt worden und kann nicht gesetzt/freigegeben werden
1	0	SPN, SPP	Semaphor wurde gesetzt/freigegeben

Hinweis

Der Vorgang des Abfragens eines bestimmten Semaphors (= Lesevorgang) und der Vorgang des Setzens bzw. des Freigebens des Semaphors (= Schreibvorgang) bilden eine **Einheit**. Keine andere CPU kann während dieser Vorgänge auf diesen Semaphor zugreifen!

Zur Anwendung der Semaphoren beachten Sie bitte die folgenden Punkte:

- Ein Semaphor ist eine globale Größe, d. h. der Semaphor mit der Nummer 16 ist auch bei Einsatz von z. B. drei CPUs im gesamten System nur **einmal** vorhanden.
- Die Befehle SES und SEF müssen von **allen** CPUs verwendet werden, deren Zugriff auf einen gemeinsamen Speicherbereich koordiniert erfolgen soll.
- Alle beteiligten CPUs müssen immer die **gleiche** Anlaufart durchführen. Bei NEUSTART werden alle Semaphore gelöscht, bei einem manuellen oder automatischen Wiederanlauf bleiben sie erhalten.
- Der ANLAUF Mehrprozessorbetrieb muß synchronisiert erfolgen. Aus diesem Grund ist **kein** Testbetrieb erlaubt.

Anwendungsbeispiel für
Semaphore

Aufgabenstellung:

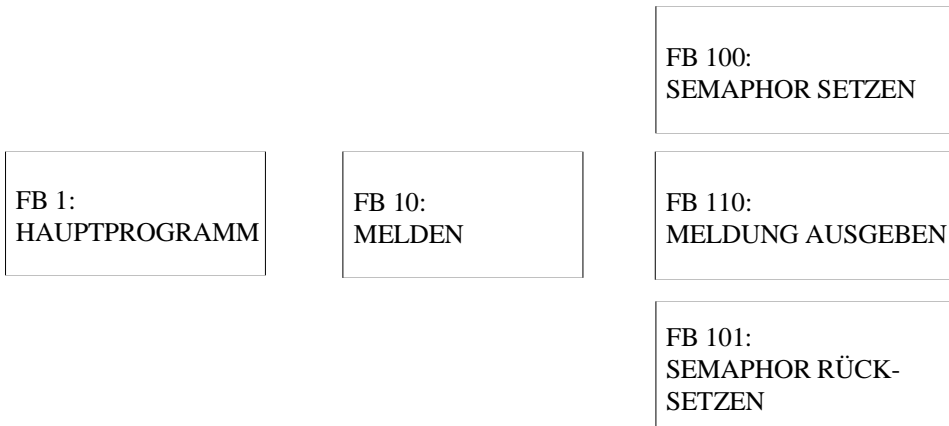
In einem AG S5-155U stecken vier CPUs, die über einen gemeinsamen Speicherbereich der Q-Peripherie (QW 6) Statusmeldungen an ein Statusmeldegerät abgeben. Jede Statusmeldung muß 10 Sekunden lang ausgegeben werden und darf erst dann von einer neuen Meldung der gleichen oder einer anderen CPU überschrieben werden.

Die Benutzung des Peripheriewortes QW 6 (erweiterte Peripherie, kein Prozeßabbild) wird über einen Semaphor gesteuert. Es darf nur diejenige CPU ihre Meldung auf das QW 6 schreiben, die durch erfolgreiches Setzen des zugeordneten Semaphors diesen Bereich für sich belegen konnte. Für die Dauer von jeweils 10 Sekunden bleibt der Semaphor gesetzt (TIMER T10). Erst nach Ablauf des Timers gibt die CPU den Semaphor und damit den belegten Bereich für die anderen CPUs wieder frei. Das QW 6 kann mit einer neuen Meldung beschrieben werden.

Ist beim Versuch einer CPU, den Semaphor zu setzen, dieser bereits durch eine andere CPU gesetzt worden, versucht die CPU im nächsten Zyklus erneut, den Semaphor zu setzen und ihre Meldung auszugeben.

Realisierung:

Das folgende Programm kann in allen vier CPUs - mit einer jeweils anderen Meldung - ablaufen. Folgende Bausteine werden geladen:



Es werden 5 Merker verwendet:

- M 10.0 = 1: Eine Meldung ist angefordert oder in Bearbeitung.
- M 10.1 = 1: Semaphor ist erfolgreich gesetzt worden.
- M 10.2 = 1: Der Timer ist gestartet.
- M 10.3 = 1: Die Meldung ist übertragen.
- M 10.4 = 1: Semaphor ist rückgesetzt.

Fortsetzung auf der nächsten Seite

Fortsetzung des Semaphor-Anwendungsbeispiels:

FB 1

```

:U   M 10.0
:SPB =M001      falls keine Meldung aktiv
:
:UN  E  0.0
:BEB
:
:L   KH 2222      Meldung erzeugen und
:T   MW 12
:UN  M 10.0
:S   M 10.0      Merker "MELDUNG" setzen.
:
M001 :SPA FB10      FB "MELDE" aufrufen
NAME :MELDE
:
:BE

```

FB 10

```

NAME :MELDE

:UN  M 10.1      Wenn kein Semaphor gesetzt,
:SPB FB 100      FB "Semaphor setzen" aufrufen.
NAME :SEMASET
:
:U   M 10.1      Wenn Semaphor gesetzt
:UN  M 10.2      und Zeit nicht gestartet,
:S   M 10.2
:L   KT010.2     Zeit starten.
:SV  T 10
:
:U   M 10.2      Wenn Zeit gestartet
:UN  M 10.3      und keine Meldung uebertragen wird,
:SPB FB 110      FB "Meldung ausgeben" aufrufen.
NAME :MELDAUSG
:
:U   M 10.2      Wenn Zeit gestartet
:UN  M 10.4      und Semaphor nicht zurueckgesetzt
:UN  T 10        und Zeit abgelaufen,
:SPB FB 101      FB "Semaphor ruecksetzen" aufrufen.
NAME :SEMARESE
:
:UN  M 10.4      Wenn Semaphor rueckgesetzt,
:BEB
:
:L   KH0000
:T   MB10        alle Merker ruecksetzen.
:BE

```

Fortsetzung auf der nächsten Seite

Fortsetzung des Semaphor-Anwendungsbeispiels:

FB 100

NAME :SEMASET

```
      :SES 10           Semaphor Nr. 10 setzen
      :SPZ =M001
      :UN  M 10.1       Falls Semaphor erfolgreich gesetzt,
      :S   M 10.1       Merker "SEMAPHOR-GESETZT" setzen
M001 :BE
```

FB 110

NAME: MELDAUSG

```
      :L  MW12          Meldung an die
      :T   QW 6          Peripherie uebertragen
      :UN  M 10.3
      :S   M 10.3       Merker "MELDUNG-UEBERTRAGEN"
      :
      :BE              setzen
```

FB 101

NAME :SEMARESE

```
      :SEF 10           Semaphor Nr. 10 freigeben
      :SPZ =M001
      :UN  M 10.4
      :S   M 10.4       Merker "SEMAPHOR-RUECKGESETZT"
      :
      :BE              setzen
M001 :BE
```


Betriebszustände und Programm- bearbeitungsebenen

4

Inhalt von Kapitel 4

4.1	Programmbearbeitungsebenen	4 - 4
4.2	Betriebszustand STOP	4 - 9
4.2.1	WEICHER STOP	4 - 9
4.2.2	HARTER STOP	4 - 13
4.2.3	URLÖSCHEN	4 - 14
4.3	Betriebszustand ANLAUF	4 - 16
4.3.1	MANUELLER und AUTOMATISCHER NEUSTART.	4 - 17
4.3.2	MANUELLER und AUTOMATISCHER WIEDERANLAUF	4 - 18
4.3.3	Gegenüberstellung NEUSTART und WIEDERANLAUF	4 - 21
4.3.4	NEUSTART MIT GEDÄCHTNIS.	4 - 22
4.3.5	Gegenüberstellung NEUSTART und NEUSTART MIT GEDÄCHTNIS.	4 - 23
4.3.6	Anwenderschnittstellen für den Anlauf	4 - 24
4.3.7	Erweiterter automatischer Wiederanlauf mit der CPU 948.	4 - 27
4.3.8	Unterbrechungen im ANLAUF	4 - 28
4.4	Betriebszustand RUN	4 - 29
4.4.1	Zyklische Programmbearbeitung	4 - 30
4.4.2	Festlegung der zeit- und alarmgesteuerten Programmbearbeitung	4 - 32
4.4.3	Zeitgesteuerte Programmbearbeitung	4 - 33
4.4.4	Alarmgesteuerte Programmbearbeitung	4 - 41

4

Betriebszustände und Programm- bearbeitungsebenen

Dieses Kapitel gibt Ihnen eine Übersicht über die Betriebszustände und die Programmbearbeitungsebenen der CPU 948. Es informiert Sie ausführlich über verschiedene Anlaufarten und damit verbundene Organisationsbausteine, in denen Sie Ihr spezifisches Programm für verschiedene Anlauffälle programmieren können.

Sie erfahren ferner, wodurch sich die Programmbearbeitungen "Zyklische Bearbeitung", "Zeitgesteuerte Bearbeitung" und "Alarmgesteuerte Bearbeitung" auszeichnen und welche Bausteine für Ihr Anwenderprogramm dabei zur Verfügung stehen.

4.1 Programmbearbeitungsebenen

Bild 4-1 gibt Ihnen eine Übersicht, welche Programmbearbeitungsebenen es in den einzelnen Betriebszuständen gibt. Die Erklärungen zu den Abkürzungen in den Fehlerebenen finden Sie auf der nächsten Seite.

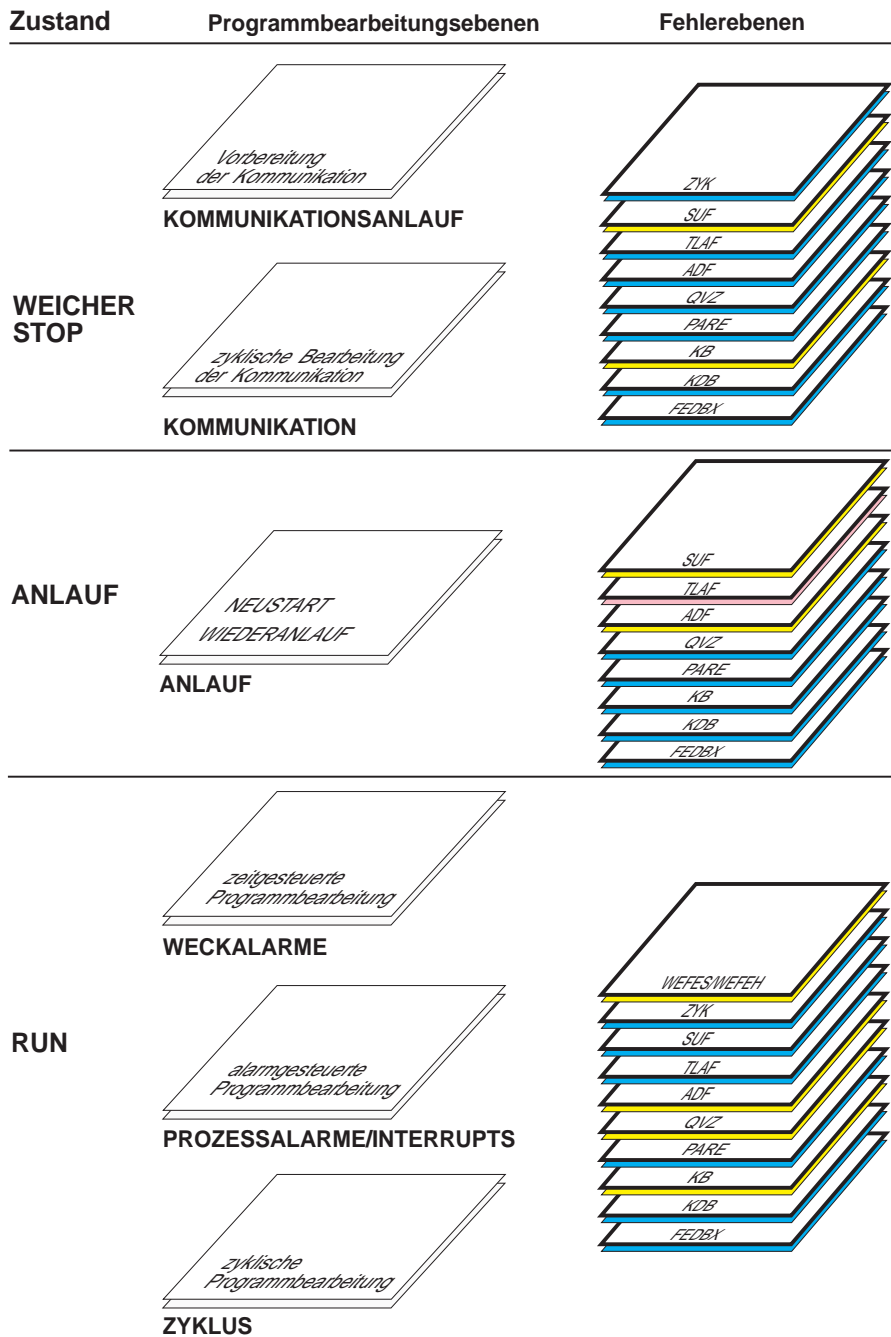


Bild 4-1 Programmbearbeitungsebenen

Tabelle 4-1 Programmbearbeitungsebenen

Ebene	Bedeutung	Priorität
Fehlerebenen		
WEFES/WEFEH ZYK SUF TLAF ADF QVZ PARE KB KDB FEDBX	Weckfehler Zyklusfehler Substitutionsfehler Transfer-/Ladefehler Adressierfehler Quittungsverzug Parityfehler aufgerufener Codebaustein nicht vorhanden aufgerufener Datenbaustein nicht vorhanden Fehler beim Erzeugen eines Datenbausteins DB oder DX	Jede Fehlerbehandlung hat die höchste Priorität. Beim Auftreten eines Fehlers wird die zugehörige Fehler-ebene sofort eingeschachtelt.
Programmbearbeitungsebenen im WEICHEN STOP		
KOMMUNIKATIONSANLAUF KOMMUNIKATION	Vorbereitung (Anlauf) der Kommunikation zyklische Bearbeitung der Kommunikation	–
Programmbearbeitungsebenen im ANLAUF		
NEUSTART WIEDERANLAUF	definierter Start des Anwenderprogramms Fortsetzen des Anwenderprogramms an der Unterbrechungsstelle	–
Programmbearbeitungsebenen im RUN		
WECKALARME PROZESSALARME/ INTERRUPTS ZYKLUS	zeitgesteuerte Programmbearbeitung alarmgesteuerte Programmbearbeitung zyklische Programmbearbeitung	steigende Priorität (Vorein- stellung) ¹⁾

¹⁾ Die Voreinstellung kann durch DX-0-Parametrierung geändert werden (siehe Kapitel 7).

Bearbeitung durch das Systemprogramm

Für jede Ebene ist ein spezifisches Systemprogramm zuständig.

Unterbrechungsstack (USTACK)

Im Unterbrechungsfall legt das Systemprogramm für jede Ebene einen Informationsblock im USTACK an, damit es nach dem Bearbeiten der Unterbrechung die Bearbeitung in der unterbrochenen Ebene fortsetzen kann.

*Einschachtelung von
anderen Ebenen*

Wenn ein Ereignis eintritt, das eine höherpriorie Bearbeitung erfordert, wird die aktuelle Ebene vom Systemprogramm unterbrochen und die höherpriorie Ebene eingeschachtelt.

Die Einschachtelung erfolgt

- bei Fehlerebenen: grundsätzlich an Befehlsgrenzen,
- bei allen anderen Ebenen: an Baustein- oder an Befehlsgrenzen (je nach DX-0-Einstellung – siehe Kapitel 7)

Hinweis

Es können nicht mehr als 5 Fehler-Organisationsbausteine ineinandergeschachtelt werden. Bei gleichzeitiger Aktivierung von 5 Fehlerebenen kommt es zu einem USTACK-Überlauf und die CPU geht in den **HARTEN STOP**.

Einem oder einer Gruppe der Organisationsbausteine, die nach einem Ereignis vom Systemprogramm aufgerufen werden, ist eine bestimmte **Programmbearbeitungsebene** zugeordnet: Wird z. B. zur Bearbeitung des uhrzeitgesteuerten Weckalarms der OB 9 aufgerufen, so wird damit die Programmbearbeitungsebene WECKALARME aktiviert.

Nach Aufruf eines Organisationsbausteins durch das Systemprogramm führt die CPU die darin enthaltenen STEP-5-Anweisungen aus. Dabei wird der aktuelle Registersatz im USTACK gerettet und ein **neuer Registersatz** angelegt (Register: AKKU 1 bis 4, Baustein-stack-Pointer, Baustein-Adreßregister, Datenbaustein-Anfangsadresse, Datenbaustein-Länge, STEP-Adreßzähler, Basisadreßregister und die Unterbrechungsanzeigen-Wörter UAMK und UALW).

Ist durch das Auftreten des Ereignisses die "normale" Programmbearbeitung unterbrochen worden, so setzt die CPU nach der Bearbeitung des OB – inklusive aller dort eingeschachtelten Bausteine – die unterbrochene Programmbearbeitung an der Unterbrechungsstelle fort, sofern innerhalb des OB kein Stopp programmiert ist.

Unterebenen

Die Ebene WECKALARME enthält weitere Unterebenen, denen jeweils ein spezifisches Programm (OB) zugeordnet ist. Innerhalb der Ebene WECKALARME besitzen die Unterebenen eine eigene Priorität (siehe nachfolgende Tabelle).

Ebene WECKALARME	
Unterebene	Priorität
Verzögerungsalarm	steigende Priorität (Voreinstellung) ↑
zyklischer Weckalarm, kürzeste Periode	
...	
...	
zyklischer Weckalarm, längste Periode	
uhrzeitgesteuerter Weckalarm	

Beispiele

Beispiel zu

"Bearbeitung durch das Systemprogramm":

In der Bearbeitungsebene ZYKLUS aktualisiert das Systemprogramm das Prozeßabbild der Ein- und Ausgänge, triggert die Zyklusüberwachungszeit und ruft die Verwaltung der PG-Schnittstelle auf (Systemkontrollpunkt).

Beispiel zu
"Unterbrechungsstack":

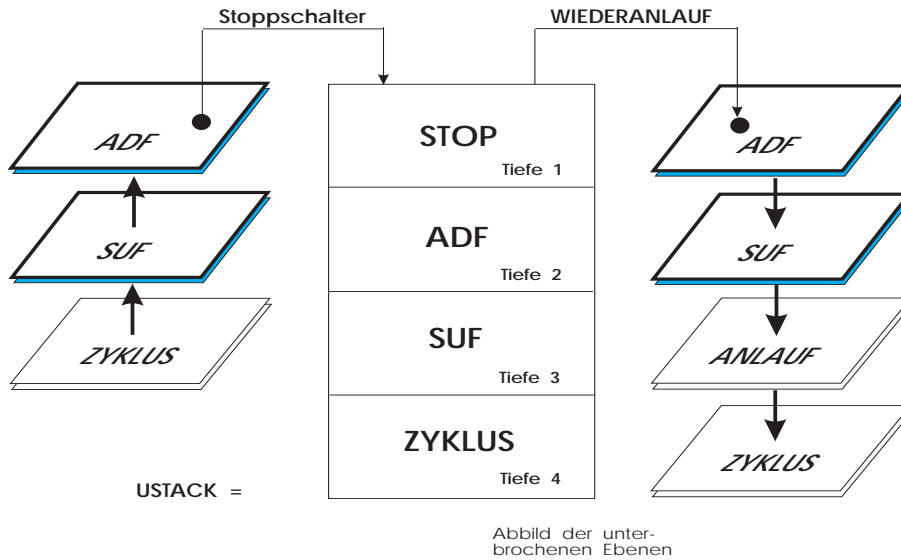


Bild 4-2 Prinzip des Ebenenwechsels und des USTACKs

Beispiel zu

"Unterbrechung einer Grundebene bei Unterbrechbarkeit an Bausteingrenzen":

Während der Bearbeitung eines Prozessalarms tritt ein Weckalarm auf. Da der Weckalarm eine höhere Priorität besitzt, wird die Bearbeitung der PROZESSALARM-Ebene an der nächsten Bausteingrenze unterbrochen und die WECKALARM-Ebene eingeschachtelt. Tritt bei der Bearbeitung des Weckalarms nun z. B. ein Adressierfehler auf, so wird die Weckalarmbearbeitung sofort an der nächsten Befehlsgränze unterbrochen, um die ADF-Ebene einzuschachteln.

4.2 Betriebszustand STOP

Die CPU 948 kennt zwei unterschiedliche STOP-Zustände, den "harten" STOP und den "weichen" (= kommunikationsfähigen) STOP.

4.2.1

WEICHER STOP

Der Betriebszustand WEICHER STOP ist durch folgende Merkmale charakterisiert:

Die CPU ist kommunikationsfähig: Das Systemprogramm ruft nach NETZ EIN einmalig den Organisationsbaustein OB 38 (Ebene KOMMUNIKATIONSANLAUF im WEICHEN STOP) und dann zyklisch den OB 39 (Ebene KOMMUNIKATION im WEICHEN STOP) auf.

Initialisierung der Kommunikation (OB 38)

Zur Initialisierung der Kommunikation ruft das Systemprogramm als Anwenderschnittstelle den **OB 38** auf.

Der OB 38 wird nur nach **NETZ EIN** aufgerufen. Der Aufruf erfolgt unabhängig davon, welche Anlaufart (AUTOMATISCHER NEUSTART/AUTOMATISCHER WIEDERANLAUF) Sie im Datenbaustein DX 0 eingestellt haben.

Ablaufüberwachung des OB 38

Der Ablauf des OB 38 wird vom Systemprogramm nicht auf Zeitüberschreitung überwacht. Sie können ihn jedoch durch Betätigen des Betriebsartenschalters nach STOP abbrechen

Tritt im OB 38 ein Fehler auf, so wird die Bearbeitung des OB 38 abgebrochen und ein evtl. vorhandener OB 39 aufgerufen..

Zyklische Kommunikation (OB 39)

Bei einer Unterbrechung des zyklischen Programms, die in den Betriebszustand WEICHER STOP führt, wird als Anwenderschnittstelle der **OB 39** aufgerufen.

Wird bei der Unterbrechung des zyklischen Programms ein Hantierungsbaustein (Kommunikation) in der Bearbeitung nicht abgeschlossen so ist im OB 39 ein erneuter Aufruf desselben Bausteins bzw. desselben Bausteintyps (z. B. SEND) möglich und zulässig.

<i>Zeitüberwachung des OB 39</i>	Der OB 39 wird vom Systemprogramm auf Überschreitung einer Höchstzeit überwacht: Dauert der Ablauf länger als 2,55 s (fest vorgegeben), so erkennt das Systemprogramm einen Zykluszeitfehler; es ruft den Fehler-OB 26 auf und bearbeitet dann den OB 39 von vorne. Tritt dabei der Zykluszeitfehler erneut auf, so kommt es bei einer USTACK-Tiefe > 5 zu einem USTACK-Überlauf (Reaktion: siehe nächster Absatz).
<i>Reaktion auf Ablauffehler im OB 39</i>	Tritt im OB 39 bzw. in einem dort aufgerufenen Baustein ein Fehler auf (z. B. QVZ), so ruft das Systemprogramm den zugehörigen Fehler-Organisationsbaustein auf. Nach dessen Bearbeitung setzt es die Bearbeitung im OB 39 fort. Ist der Fehler-OB nicht vorhanden, so wird der OB 39 von vorne bearbeitet. (Ausnahme: Bei QVZ, KB und Fehler im Selbsttest erfolgt keine Reaktion.) Treten weitere Fehler auf, so kommt es bei einer USTACK-Tiefe > 5 zu einem USTACK-Überlauf: Das Systemprogramm bricht die Programmbearbeitung ab (der OB 39 wird nicht mehr aufgerufen!), die CPU bleibt jedoch weiterhin im Zustand WEICHER STOP.
<i>Daten werden nicht zurückgesetzt</i>	Hat eine zyklische Bearbeitung im Zustand RUN stattgefunden, so bleiben die Werte von Zählern, Zeiten, Merkern und des Prozeßabbildes beim Übergang in den Stoppzustand erhalten.
<i>Echtzeituhr</i>	Die Echtzeituhr läuft weiter. Die Aktualisierung im BS-Bereich erfolgt im 10-ms-Raster.
<i>BASP-Signal</i>	Das Signal BASP (Befehlsausgabe sperren) ist aktiv. Damit sind alle digitalen Ausgaben gesperrt (Ausnahme: Im Testbetrieb bei Mehrprozessorbetrieb und bei der PG-Funktion "Ausgänge steuern" wird BASP nicht aktiv – siehe dazu Abschnitt 10.1.8).
<i>USTACK</i>	Wurde vor dem Stoppzustand ein Anwenderprogramm bearbeitet, so ist im Stoppzustand ein Informationseintrag im Unterbrechungsstack (USTACK) vorhanden, der über die Unterbrechungsursache Auskunft gibt.
<i>Zeitenverarbeitung im OB 38/39</i>	Während des Ablaufs des OB 38/39 wird die Verarbeitung von Zeiten und Zählern gestoppt. Im Bedarfsfall müssen Zeitinformationen aus dem Systemdatenbereich BS 96 bis BS 99 oder mit dem OB 121 bzw. mit dem OB 150 verarbeitet werden.

Aufruf OB 38/OB 39

Die Bilder 4-3 und 4-4 zeigen Ihnen das Prinzip der Aufrufe von OB 38 und OB 39.

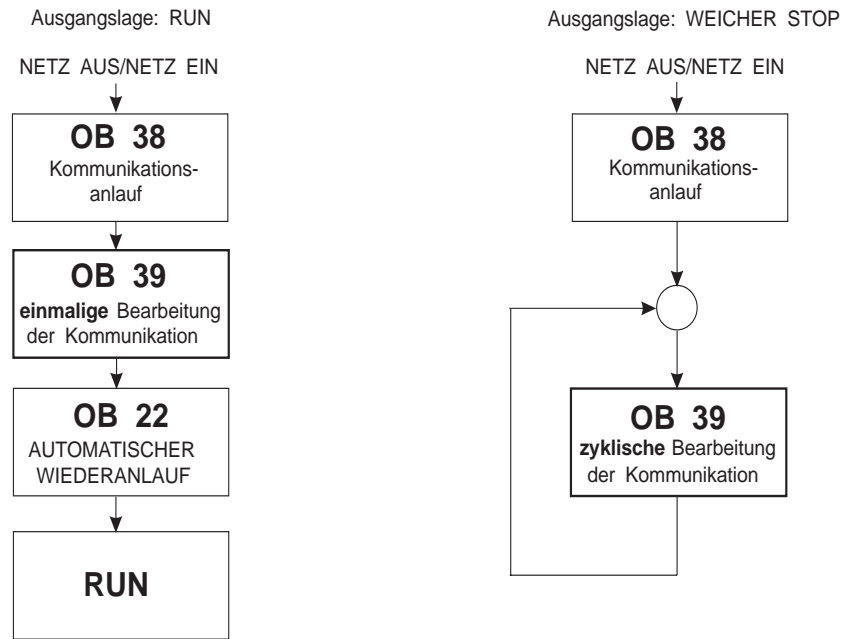


Bild 4-3 Programmbearbeitung nach NETZ EIN

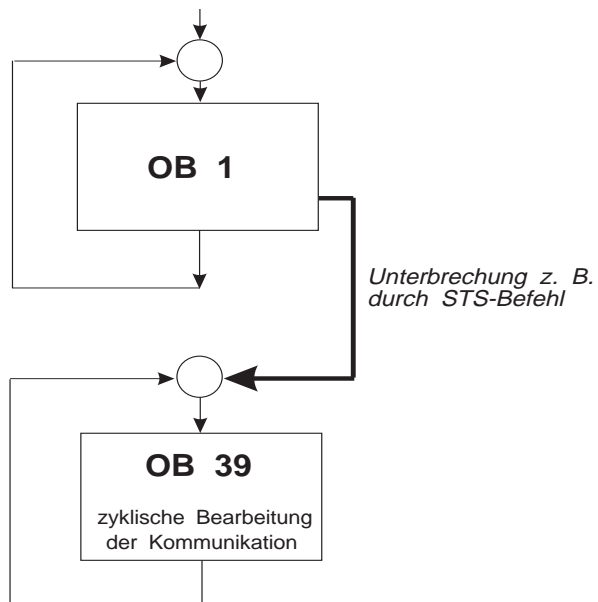


Bild 4-4 Programmbearbeitung nach einer Zyklusunterbrechung

LED-Anzeigen

Den Zustand WEICHER STOP können Sie an den Anzeigen folgender LEDs auf der Frontplatte des Zentralgerätes erkennen:

LED	Status
RUN	aus
STOP	an (Dauerlicht oder blinkend)
SYSFAULT	aus
BASP	an (außer im Testbetrieb bei Mehrprozessorbetrieb oder PG-Funktion "Ausgänge steuern")

Die **STOP-LED** signalisiert Ihnen bei dieser Anzeige, welche Ursachen für den momentanen Stoppzustand in Frage kommen:

*STOP-LED zeigt
Dauerlicht*

Der Betriebszustand WEICHER STOP wurde ausgelöst:

im Einzelprozessorbetrieb:

- durch Betätigen des Betriebsartenschalters von RUN auf STOP,
- durch PG-Funktion AG-STOP,
- durch Gerätefehler (PEU),
- nach URLÖSCHEN.

im Mehrprozessorbetrieb durch:

- Betätigen des Betriebsartenschalters am KOR auf STOP,
- STOP einer anderen CPU aufgrund einer Störung (jede **nicht** fehlerverursachende CPU zeigt Dauerlicht) oder durch STOP-Schalter,
- PG-Funktion AG-STOP,
- PG-Funktion "Bearbeitungskontrolle Ende" an einer anderen CPU.

*STOP-LED blinkt langsam
(ca 0,5 mal pro Sekunde)*

Der Betriebszustand WEICHER STOP wurde ausgelöst durch:

- STP- oder STS-Anweisung im Anwenderprogramm,
- Fehlbedienung (DB-1/DX-0-Fehler, Wahl einer unzulässigen Anlaufart usw.),
- BSTACK- Überlauf (STUEB) oder Klammerzählerüberlauf (KZU),
- Programmier- und Gerätefehler ; als zusätzlicher Hinweis auf die Fehlerursache leuchten:
 - LED "ADF"
 - LED "QVZ"
 - LED "ZYG"
- durch PG-Funktion "Bearbeitungskontrolle Ende" an dieser CPU.

Verlassen des Zustandes WEICHER STOP

Der Zustand WEICHER STOP kann verlassen werden

- a) durch Wahl einer Anlaufart (siehe Abschnitt 4.4),
- b) durch URLÖSCHEN mit anschließendem NEUSTART.

4.2.2 HARTER STOP

Wenn ein einwandfreier Ablauf des Systemprogramms nicht mehr gewährleistet ist, geht die CPU in den HARTEN STOP, um für diesen Fall einen sicheren Betriebszustand zu garantieren.

Folgende Ursachen bewirken einen HARTEN STOP:

- Stoppbefehl (STW) für das Systemprogramm,
- USTACK-Überlauf (STUEU),
- Quittungsverzug (QVZ) oder Parityfehler (PARE) im System-RAM/EPROM,
- Fehler im Systemprogramm.

Hinweis

Die CPU "steht": Ein Verlassen des Zustandes HARTER STOP ist ausschließlich über Bedienung NETZ AUS/NETZ EIN möglich!

LED-Anzeigen

Den Zustand HARTER STOP können Sie an den Anzeigen folgender LEDs auf der Frontplatte des Zentralgerätes erkennen:

LED	Zustand
RUN	aus
STOP	aus
SYSFAULT	an
BASP	an

**4.2.3
URLÖSCHEN**

Durch das URLÖSCHEN werden der gesamte Anwenderspeicher und alle Operandenbereiche (Merker, Prozeßabbilder usw.) gelöscht.

URLÖSCHEN anfordern

Bevor das URLÖSCHEN durchgeführt werden kann, muß es angefordert werden. URLÖSCHEN ist angefordert, wenn die STOP-LED schnell blinkt.

Anforderung durch das Systemprogramm

Nach jedem Einschalten der Netzspannung durchläuft die CPU eine Initialisierungsroutine. Wenn nun bei der Initialisierung Fehler festgestellt werden (z. B. Ausfall der Pufferspannung während NETZ AUS), so wird vom Systemprogramm URLÖSCHEN angefordert.

Die Störungsursache muß beseitigt werden (z. B. durch Batterieaustausch), und danach die CPU urgelöscht werden.

URLÖSCHEN wird auch angefordert bei Auftreten eines CPU- oder Systemfehlers. Sie erkennen diese Fehler daran, daß die Aufforderung nach durchgeführtem URLÖSCHEN erneut auftritt. In diesem Fall wenden Sie sich bitte an Ihre nächste Siemens-Vertretung.

Anforderung durch Bedienung

Durch folgende Bedienschritte können Sie URLÖSCHEN anfordern (die Bedienelemente finden Sie auf der CPU-Frontplatte – Bild 4-1):

Schritt	Aktion	Ergebnis
1	Betätigen Sie den Betriebsartenschalter von RUN nach STOP.	Die CPU befindet sich im Stoppzustand. Die STOP-LED zeigt Dauerlicht.
2	Halten Sie den Betriebsartentaster in Stellung OVERALL RESET fest; betätigen Sie gleichzeitig den Betriebsartenschalter von STOP nach RUN und wieder nach STOP.	URLÖSCHEN ist angefordert. Die STOP-LED blinkt schnell.

Hinweis

Soll das angeforderte URLÖSCHEN doch nicht ausgeführt werden, so führen Sie jetzt einen MANUELLEN NEUSTART oder MANUELLEN WIEDERANLAUF durch.

URLÖSCHEN durchführen

Unabhängig davon, ob die Urlöschanforderung vom Systemprogramm oder vom Anwender kommt, führen Sie das URLÖSCHEN folgendermaßen durch (Ausgangszustand: STOP-LED blinkt schnell):

URLÖSCHEN über Bedienelemente der CPU durchführen

Aktion	Ergebnis
Halten Sie den Betriebsartentaster in Stellung OVERALL RESET fest; betätigen Sie gleichzeitig den Betriebsartenschalter von STOP nach RUN und wieder nach STOP (siehe Bild 4-1).	URLÖSCHEN wird durchgeführt. Die STOP-LED zeigt Dauerlicht.

URLÖSCHEN vom PG aus durchführen

Aktion	Ergebnis
Führen Sie die PG-Funktion "Löschen aller Bausteine" durch.	URLÖSCHEN wird durchgeführt. Die STOP-LED zeigt Dauerlicht.

Hinweis

Im Unterschied zur CPU 946/947 können Sie bei der CPU 948 am PG das URLÖSCHEN auch aus dem Betriebszustand RUN heraus initiieren. Die CPU wird in diesem Fall automatisch in den Stoppzustand versetzt und das URLÖSCHEN dann durchgeführt.

Während des Vorgangs URLÖSCHEN sind außer INIT- und BASP-LED alle LEDs dunkel.

Haben Sie URLÖSCHEN durchgeführt, so ist als Anlaufart anschließend nur ein NEUSTART zulässig!

Laden der Memory Card und Aufruf OB39

Nach Durchführen des URLÖSCHENS wird – sofern sie gesteckt ist – die Memory Card geladen und danach der OB 39 aufgerufen.

4.3 Betriebszustand ANLAUF

Der Betriebszustand ANLAUF ist durch folgende Merkmale charakterisiert:

Zustandswechsel

Der ANLAUF ist der Übergang vom Betriebszustand STOP in den Betriebszustand RUN.

Anlaufarten

Die CPU 948 kennt folgende Anlaufarten:

- NEUSTART (manuell oder automatisch),
 - WIEDERANLAUF (manuell oder automatisch).
- (Sie können die Anlaufart über Bedienung und DX-0-Parametrierung wählen)

NEUSTART

Das zyklische Anwenderprogramm wird von Anfang an bearbeitet.

Als Anwenderschnittstelle ruft das Systemprogramm den **OB 20** auf.

WIEDERANLAUF

Die Bearbeitung des zyklischen Anwenderprogramms wird an der Unterbrechungsstelle fortgesetzt.

Als Anwenderschnittstelle ruft das Systemprogramm auf:

- beim MANUELLEN WIEDERANLAUF den **OB 21**,
- beim AUTOMATISCHEN WIEDERANLAUF den **OB 22**.

keine Zeitüberwachung der Anlauf-OBs

Der Ablauf der Anlauf-Organisationsbausteine wird **zeitlich nicht überwacht**. Sie können in ihnen weitere Bausteine aufrufen. Vermeiden Sie jedoch unbedingt Endlosschleifen!

Zähler, Zeiten, Zähler, Prozeßabbilder

Die Werte von Zählern, Zeiten, Merkern und Prozeßabbildern werden in den einzelnen Anlaufarten unterschiedlich behandelt (siehe Abschnitt 4.4.3).

BASP

Das Signal BASP (Befehlsausgabe sperren) ist aktiv. Damit sind alle digitalen Ausgaben gesperrt (Ausnahme: Im Testbetrieb bei Mehrprozessorbetrieb wird BASP nicht aktiv – siehe dazu Abschnitt 10.1.8).

Alarme

Die Alarme (Interrupts, Prozeßalarne, Weckalarne) sind gesperrt

Anlauf im Mehrprozessorbetrieb

Hinweise zum "Anlaufverhalten im Mehrprozessorbetrieb" finden Sie in Abschnitt 10.1.7.

4.3.1 MANUELLER und AUTOMATISCHER NEUSTART

Wann ist NEUSTART
zulässig ?

Ein NEUSTART ist **immer zulässig**, sofern vom System nicht URLÖSCHEN angefordert ist!

Wann ist NEUSTART
erforderlich?

Ein NEUSTART ist **erforderlich** nach:

- URLÖSCHEN,
- Laden des Anwenderspeichers mit dem Anwenderprogramm während des Stopp-Zustandes der CPU,
- USTACK-/BSTACK-Überlauf,
- NEUSTART-Abbruch (durch NETZ AUS oder Betriebsartenschalter nach "STOP"),
- Stopp nach PG-Funktion "Bearbeitungskontrolle Ende".

MANUELLER NEUSTART

So **lösen Sie** den MANUELLEN NEUSTART aus:

- über Bedienelemente der CPU:

Halten Sie den Betriebsartentaster in Stellung RESET; betätigen Sie gleichzeitig den Betriebsartenschalter von STOP nach RUN. (siehe Bild 4-1).

- über PG:

Wählen Sie die PG-Funktion AG-START/NEUSTART.

AUTOMATISCHER
NEUSTART

Ein AUTOMATISCHER NEUSTART **wird ausgelöst**:

Bei NETZ EIN, wenn

- die Voreinstellung "AUTOMATISCHER WIEDERANLAUF nach NETZ EIN" im Datenbaustein DX 0 geändert wurde in "AUTOMATISCHER NEUSTART nach NETZ EIN",
- die Betriebsartenschalter an allen CPUs und am Koordinator unverändert auf RUN stehen
und
die CPU bei NETZ AUS nicht im STOP war.

Hinweis

Wenn die CPU bei NETZ AUS im Betriebszustand STOP war (z. B. nach einem Adressierfehler), so ist ein **AUTOMATISCHER NEUSTART nicht zulässig**. Der Betriebszustand STOP kann in diesem Fall nur durch einen **MANUELLEN NEUSTART** verlassen werden.

Abbruch des NEUSTARTS

Sie können einen laufenden NEUSTART nur Abbrechen durch Betätigen des Betriebsartenschalters nach "STOP" oder durch NETZ AUS. Einen abgebrochenen NEUSTART müssen Sie wiederholen!

**4.3.2
MANUELLER und
AUTOMATISCHER
WIEDERANLAUF**

*Wann ist ein
WIEDERANLAUF
möglich und zulässig?*

Ein MANUELLER WIEDERANLAUF ist nur **möglich** nach folgenden Stoppursachen:

- Der Betriebsartenschalter wird von Stellung "RUN" in die Stellung "STOP" gebracht,
- Stopp im Mehrprozessorbetrieb durch HALT-Signal vom Koordinator,
- NETZ AUS, mit entsprechender Einstellung im Datenbaustein DX 0,
- PG-Funktion AG-STOP

Hinweis

War ein anderes Ereignis als oben aufgeführt Ursache für den Übergang in den Stoppzustand, so ist grundsätzlich **kein Wiederanlauf** möglich. Das Systemprogramm läßt dann nur einen NEUSTART zu.

Ein MANUELLER bzw. AUTOMATISCHER WIEDERANLAUF ist nur **zulässig**, wenn

- das Anwenderprogramm im Stoppzustand nicht verändert wurde
und
- nicht aus anderen Gründen (siehe Abschnitt 4.4.1) NEUSTART erforderlich ist.

**MANUELLER
WIEDERANLAUF**

So **lösen Sie** einen MANUELLEN WIEDERANLAUF aus:

- über Bedienelemente der CPU:

Ausgangszustand: Der Betriebsartentaster befindet sich in Mittelstellung,

Betätigen Sie den Betriebsartenschalter von STOP nach RUN (siehe Bild 4-1).

- über PG:

Wählen Sie die PG-Funktion AG-START/WIEDERANLAUF.

**AUTOMATISCHER
WIEDERANLAUF**

Ein AUTOMATISCHER WIEDERANLAUF **wird ausgelöst:**

Bei NETZ EIN, wenn

- die Voreinstellung "AUTOMATISCHER WIEDERANLAUF nach NETZ EIN" im Datenbaustein DX 0 eingestellt ist bzw. kein DX 0 vorhanden ist,
- die Betriebsartenschalter an allen CPUs und am Koordinator unverändert auf RUN stehen
und
die CPU bei NETZ AUS nicht im STOP war,
- keine weiteren Fehler bei der Initialisierung und vor NETZ AUS aufgetreten sind,
- kein NEUSTART aus den o. g. Gründen erforderlich ist.

Nach Netzspannungsausfall/NETZ AUS im RUN und anschließender Netzspannungswiederkehr/NETZ EIN durchläuft die CPU eine Initialisierungsroutine und führt dann automatisch einen WIEDERANLAUF durch.

Bei Netzspannungsausfall in einem Erweiterungsgerät (PEU-Signal) geht die CPU in STOP. Sie verharrt dort solange, bis das PEU-Signal inaktiv geschaltet wird und führt dann einen AUTOMATISCHEN WIEDERANLAUF bzw. AUTOMATISCHEN NEUSTART durch.

Hinweis

Beachten Sie beim WIEDERANLAUF folgenden Sonderfall:

Die CPU bearbeitet gerade einen Fehler-OB (z. B. wegen ADF) und geht dabei aufgrund von NETZ AUS, HALT, Stoppschalter oder PG-STP in den STOP. Anschließend wird ein MANUELLER oder AUTOMATISCHER WIEDERANLAUF durchgeführt.

Reaktion der CPU:

- **Vor** Aufruf des OB 21/22 wird die unterbrochene Bearbeitung des Fehler-OBs zu Ende geführt.
- Läuft der Fehler-OB darauf nicht auf eine Stopp-Operation, so erfolgt nach der Restbearbeitung des Fehler-OBs ein **WIEDERANLAUF**.
- Führt der Fehler-OB darauf die CPU in den Stoppzustand, so ist anschließend nur ein **NEUSTART** möglich.

*Abbruch des
WIEDERANLAUFS*

Sie können einen begonnenen WIEDERANLAUF nur abbrechen durch Betätigen des Betriebsartenschalters nach "STOP" oder durch NETZ AUS. Nach einem derart abgebrochenen Wiederanlauf ist sowohl ein NEUSTART als auch ein WIEDERANLAUF möglich.

4.3.3

**Gegenüberstellung
NEUSTART und
WIEDERANLAUF**

Folgende Tabelle enthält eine Gegenüberstellung der Anlaufarten NEUSTART und WIEDERANLAUF.

Tabelle 4-2 Ablaufmerkmale von NEUSTART und WIEDERANLAUF

	NEUSTART	WIEDERANLAUF
Auslösung manuell:	Betriebsartenschalter von Stellung "STOP" in Stellung "RUN" und Betriebsartentaster in Stellung "RESET " oder PG-Funktion AG-START (NEUSTART)	Betriebsartenschalter von Stellung "STOP" in Stellung "RUN" oder PG-Funktion AG-START (WIEDERANLAUF)
Auslösung automatisch:	Einschalten der Versorgungsspannung, wenn im DX 0 "AUTOMATISCHER NEUSTART nach NETZ EIN" eingetragen ist	Einschalten der Versorgungsspannung, wenn im DX 0 die Voreinstellung eingetragen oder kein DX 0 vorhanden ist
Systemprogrammleistungen:	Bausteinadreßliste im DB 0 aufbauen	Bausteinadreßliste im DB 0 erhalten
	Prozeßabbild der Eingänge löschen	Prozeßabbild der Eingänge erhalten
	Prozeßabbild der Ausgänge löschen	Prozeßabbild der Ausgänge erhalten
	Merker, Zeiten und Zähler löschen	Merker, Zeiten und Zähler erhalten
	digitale/analoge Peripherie löschen (je 2*128 byte)	
	Koppelmerker löschen (256 byte)	Koppelmerker erhalten
	Verzögerungsalarme und Zeitaufträge löschen	Verzögerungsalarme löschen, Zeitaufträge erhalten
USTACK/BSTACK löschen	USTACK/BSTACK erhalten	
Semaphore löschen	Semaphore erhalten	
wenn DB 1 vorhanden: dort eingetragene digitale E/A und Koppelmerker in die PA-Listen übernehmen wenn DB 1 nicht vorhanden: tatsächlich existierende Baugruppen (nur digitale E/A) in die PA-Listen übernehmen (Koppelmerker werden ignoriert)	keine Übernahme aus DB 1	

	NEUSTART	WIEDERANLAUF
	Fortsetzung der Tabelle 4-3:	
System- programm- leistungen (Fortsetz.):	Systemparameter entsprechend Vorgabe im DX 0 einstellen	keine Auswertung des DX 0
	Anwenderschnittstelle OB 20 aufrufen (falls vorhanden) ¹⁾	Anwenderschnittstelle OB 21/22 aufrufen (falls vorhanden) ¹⁾
	Anlauf im Mehrprozessorbetrieb synchronisieren	Anlauf im Mehrprozessorbetrieb synchronisieren
	Übergang in den Zyklus: - BASP inaktiv schalten, - OB 1 aufrufen.	Übergang in den Zyklus: - BASP bleibt aktiv, - Prozeßabild der Ausgänge löschen, - Restzyklus bearbeiten, - BASP inaktiv schalten, - OB1 aufrufen.

¹⁾ Nach NETZ EIN werden die Anwenderschnittstellen im ANLAUF in folgender Reihenfolge aufgerufen: OB 38, OB 39, OB 20/OB 22.

4.3.4 NEUSTART MIT GEDÄCHTNIS

Ist im geladenen Datenbaustein DX 0 der Parameter "Neustart mit Gedächtnis" abgelegt, so führt das Systemprogramm an Stelle des WIEDERANLAUFS den NEUSTART MIT GEDÄCHTNIS durch. Wie sich dieser von einem "normalen" NEUSTART unterscheidet, entnehmen Sie bitte dem folgenden Abschnitt.

4.3.5

Gegenüberstellung Folgende Tabelle zeigt Ihnen die Unterschiede zwischen NEU-
NEUSTART und NEUSTART START und NEUSTART MIT GEDÄCHTNIS auf.
MIT GEDÄCHTNIS

Tabelle 4-3 Unterschiede zwischen NEUSTART und NEUSTART MIT GEDÄCHTNIS

	NEUSTART	NEUSTART MIT GEDÄCHTNIS
Auslösung manuell:	Betriebsartenschalter von Stellung "STOP" in Stellung "RUN" und Wahlschalter in Stellung "RESET " oder PG-Funktion AG-START (NEUSTART)	Betriebsartenschalter von Stellung "STOP" in Stellung "RUN" oder PG-Funktion AG-START (WIEDERANLAUF)
Auslösung automatisch:	Einschalten der Versorgungsspannung, wenn im DX 0 "AUTOMATISCHER NEUSTART nach NETZ EIN" eingetragen ist	Einschalten der Versorgungsspannung, wenn im DX 0 "AUTOMATISCHER WIEDERANLAUF nach NETZ EIN " und "NEUSTART MIT GEDÄCHTNIS" eingetragen ist
Systemprogrammleistungen:	Bausteinadreßliste im DB 0 aufbauen	Bausteinadreßliste im DB 0 erhalten
	Prozeßabbild der Eingänge löschen	Prozeßabbild der Eingänge erhalten
	Prozeßabbild der Ausgänge löschen	Prozeßabbild der Ausgänge erhalten
	Verzögerungsalarme und Zeitaufträge löschen	Verzögerungsalarme und Zeitaufträge löschen
	Merker, Zeiten und Zähler löschen	Merker, Zeiten und Zähler erhalten
	digitale/analoge Peripherie löschen (je 2*128 byte)	digitale Peripherie löschen (128 byte), analoge Peripherie erhalten (128 byte)
	Koppelmerker löschen (256 byte)	Koppelmerker erhalten
	USTACK/BSTACK löschen	USTACK/BSTACK löschen
	Semaphore löschen	Semaphore erhalten
	wenn DB 1 vorhanden: dort eingetragene digitale E/A und Koppelmerker in die PA-Listen übernehmen wenn DB 1 nicht vorhanden: tatsächlich existierende Baugruppen (nur digitale E/A) in die PA-Listen übernehmen (Koppelmerker werden ignoriert)	keine Übernahme aus DB 1

	NEUSTART	NEUSTART MIT GEDÄCHTNIS
System- programm- leistungen (Fortsetz.):	Fortsetzung der Tabelle 4-4:	
	Systemparameter entsprechend Vorgabe im DX 0 einstellen	keine Auswertung des DX 0
	Anwenderschnittstelle OB 20 aufrufen (falls vorhanden) ¹⁾	Anwenderschnittstelle OB 21/22 aufrufen (falls vorhanden) ¹⁾
	Anlauf im Mehrprozessorbetrieb synchronisieren	Anlauf im Mehrprozessorbetrieb synchronisieren
	Übergang in den Zyklus: - BASP inaktiv schalten, - OB 1 aufrufen.	Übergang in den Zyklus: - BASP inaktiv schalten, - OB1 aufrufen.

1) Nach NETZ EIN werden die Anwenderschnittstellen im ANLAUF in folgender Reihenfolge aufgerufen:
OB 38, OB 39, OB 20/OB 22.

4.3.6

Anwenderschnittstellen für den Anlauf

Als Anwenderschnittstellen für die verschiedenen Anlaufarten dienen die Organisationsbausteine OB 20, OB 21 und OB 22. In diesen Bausteinen können Sie Ihr STEP-5-Programm für die jeweilige Anlaufart hinterlegen.

OB 20

Wenn die CPU einen MANUELLEN oder AUTOMATISCHEN NEUSTART durchführt, wird vom Systemprogramm **einmal** der OB 20 aufgerufen. Sie können dort ein STEP-5-Programm hinterlegen, das vor Beginn der zyklischen Programmbearbeitung vorbereitende Schritte für einen **Neubeginn** der zyklischen Bearbeitung durchführt.

Sie können z. B.

- Merker setzen,
- Zeiten starten (der Start wird vom Systemprogramm erst beim Eintritt in den RUN durchgeführt),
- Daten, die an Peripheriebaugruppen ausgegeben werden, vorbesetzen,
- die Synchronisierung von CPs durchführen .

Nach der Bearbeitung des OB 20 beginnt die zyklische Programmbearbeitung durch Aufruf des OB 1.

Ist der OB 20 nicht geladen, beginnt die CPU am Ende eines NEUSTARTS (nach den Systemleistungen) sofort mit der zyklischen Programmbearbeitung.

OB 21

Wenn die CPU einen MANUELLEN WIEDERANLAUF oder MANUELLEN NEUSTART MIT GEDÄCHTNIS durchführt, so ruft das Systemprogramm einmalig den OB 21 auf. Sie können dort ein STEP-5-Programm hinterlegen, das vorbereitende Schritte zur **Wiederaufnahme** der zyklischen Programmbearbeitung durchführt.

**MANUELLER
WIEDERANLAUF**

Nach der Bearbeitung des OB 21 wird bei MANUELLEM WIEDERANLAUF die zyklische Programmbearbeitung an der Abbruchstelle mit der nächsten Anweisung fortgesetzt. Es gilt:

- Das BASP-Signal (Befehlsausgabe sperren) bleibt während der Bearbeitung des Restzyklus aktiv und wird erst mit Beginn des nächsten (vollständigen) Zyklus inaktiv.
- Das Prozeßabbild der Ausgänge wird am Ende des Restzyklus zurückgesetzt.
- Ist der OB 21 nicht geladen, beginnt die CPU am Ende eines MANUELLEN WIEDERANLAUFs nach den Systemleistungen sofort an der Unterbrechungsstelle mit der Programmbearbeitung.

**MANUELLER NEUSTART
MIT GEDÄCHTNIS**

Ist im Datenbaustein DX 0 der Parameter "NEUSTART MIT GEDÄCHTNIS" eingetragen, so führt das Systemprogramm nach Bearbeitung des OB 21 einen **NEUSTART** durch (die CPU setzt die Programmbearbeitung mit der **ersten STEP-5-Anweisung im OB 1** fort). Die Signalzustände der Merker, Koppelmerker, Semaphore und die Baustein-Adreßliste (DB 0) **bleiben erhalten**.

OB 22

Wenn die CPU einen AUTOMATISCHEN WIEDERANLAUF oder AUTOMATISCHEN NEUSTART MIT GEDÄCHTNIS durchführt, so ruft das Systemprogramm einmalig den OB 22 auf. Sie können dort ein STEP-5-Programm hinterlegen, das vorbereitende Schritte (im allgemeinen nach einem Netzausfall) zur **Wiederaufnahme** der zyklischen Programmbearbeitung durchführt.

**AUTOMATISCHER
WIEDERANLAUF**

Nach NETZ EIN führt die CPU die in Abschnitt 4.4.4 erwähnten Systemleistungen bei Wiederanlauf aus und versucht das unterbrochene Programm an der Unterbrechungsstelle fortzusetzen.

Dabei wird zunächst der OB 22 aufgerufen.

Nach der Bearbeitung des OB 22 wird die unterbrochene Programmbearbeitung an der Abbruchstelle mit der nächsten Anweisung fortgesetzt.

Nach einem Netzausfall mit anschließender Netzwiederkehr gilt:

- Das BASP-Signal (Befehlsausgabe sperren) ist während der Bearbeitung des Restzyklus aktiv und wird erst mit Beginn des ersten vollständigen Zyklus inaktiv gesetzt.
- Das Prozeßabbild der Ausgänge wird am Ende des Restzyklus zurückgesetzt.
- Ist der OB 22 nicht geladen, so beginnt die CPU am Ende eines AUTOMATISCHEN WIEDERANLAUFS die Programmbearbeitung sofort an der Abbruchstelle.

**AUTOMATISCHER
NEUSTART MIT
GEDÄCHTNIS**

Ist im Datenbaustein DX 0 der Parameter "Neustart mit Gedächtnis" eingetragen, so führt das Systemprogramm nach Bearbeitung des OB 22 einen **NEUSTART** durch (die CPU setzt die Programmbearbeitung mit der **ersten STEP-5-Anweisung im OB 1** fort). Die Signalzustände der Merker, Koppelmerker, Semaphore und die Baustein-Adreßliste (DB 0) **bleiben erhalten**.

4.3.7

Erweiterter automatischer Wiederanlauf mit der CPU 948

Die in der Norm IEC 1131, Teil 1 angegebene Betriebsart "hot restart" (= erweiterter automatischer Wiederanlauf) ist in der CPU 948 ebenfalls möglich. Der erweiterte Wiederanlauf ist ein Wiederanlauf unter Kontrolle einer – laut IEC 1131 – gepufferten Uhr. Mit der Uhr wird die Zeit zwischen Ausschalten und Wiedereinschalten der Netzspannung für die CPU überwacht. Abhängig von dieser Zeitdauer wird ein anschließender Wiederanlauf zugelassen oder nicht.

Der erweiterte automatische Wiederanlauf der CPU 948 wird nicht direkt vom Systemprogramm unterstützt und muß daher von Ihnen programmiert werden.

Die zur Verfügung stehenden Basisfunktionen sind der AUTOMATISCHE WIEDERANLAUF (OB 22) und die zwar interne, jedoch über externe Batteriepufferung laufende Echtzeituhr.

Programmieren Sie den erweiterten automatischen Wiederanlauf nach folgendem Ablaufschema:

Betriebszustand	Maßnahme/Baustein	
RUN	Zeit (Uhrzeit und Datum von externer Uhr regelmäßig in definierte Speicherzellen retten (z. B. in Datenwörtern): <ul style="list-style-type: none"> - am Zyklusende im OB 1 oder - zeitgesteuert über Weckalarm (z. B. OB 10), falls höhere Genauigkeit notwendig. 	
AUTOMATISCHER WIEDERANLAUF	OB 22	
	Ausfallzeit berechnen: $T_{\text{ausfall}} = 1. \text{ Zeitwert nach Netzwiederkehr} - \text{letzter vor Netzausfall geretteter Zeitwert}$	
	WENN ... Ausfallzeit > vorgegebener Maximalwert,	DANN... - Wiederanlauf abbrechen (STP-Operation) oder - modifizierten Wiederanlauf durchführen.
	Ausfallzeit ≤ vorgegebener Maximalwert,	Wiederanlauf fortsetzen.

**4.3.8
Unterbrechungen im
ANLAUF**

Ein Anlaufprogramm kann unterbrochen werden durch

- Netzspannungsausfall im Zentralgerät (NAU) oder im Erweiterungsgerät (PEU),
- Stoppschalter, Stopp-Befehl, HALT oder PG-STP
- Programm- und Gerätefehler (siehe Abschnitt 5.5).

Grundregeln für unterbrochenen ANLAUF

Es gelten folgende Grundregeln für das Anlaufverhalten der CPU 948:

Wird der ANLAUF unterbrochen, so wird der anschließende ANLAUF auf jeden Fall von vorne durchlaufen!
Es gilt immer die zuletzt ausgewählte Anlaufart. Beispiel: 1. NETZ AUS (NAU) im Zyklus 2. Schalter in Stellung STOP 3. NETZ EIN 4. Schalter in Stellung RUN Reaktion: Die CPU führt einen MANUELLEN WIEDERANLAUF durch.
Ein unterbrochener NEUSTART kann nicht durch einen WIEDERANLAUF fortgesetzt werden, sondern muß wiederholt werden.
Nach einem unterbrochenen WIEDERANLAUF ist sowohl NEUSTART als auch ein erneuter WIEDERANLAUF möglich

*Verhalten der CPU bei
Netzwiederkehr nach
Netzausfall bzw. PEU-Signal*

Wird eine Anlaufbearbeitung durch Netzausfall oder PEU-Signal abgebrochen, so wird das Verhalten der CPU bei Netzwiederkehr von der **eingestellten** und von der **abgebrochenen** Betriebsart bestimmt. Die nachfolgende Tabelle gibt Ihnen dazu einen Überblick.

im DX 0 eingestellte Betriebsart	abgebrochene Betriebsart	abgebrochener Anlauf-OB	Reaktion der CPU
AUTOMATISCHER NEUSTART	MANUELLER NEUSTART	OB 20	unabhängig von der Vorgeschichte wird ein AUTOMATISCHER NEUSTART ausgeführt (der OB 20 wird aufgerufen und von vorne bearbeitet); es wird kein "Rest-Anlauf" bearbeitet (keine Fortsetzung von OB 20 bzw. OB 21!).
	MANUELLER WIEDERANLAUF	OB 21	
	AUTOMATISCHER NEUSTART	OB 20	
AUTOMATISCHER WIEDERANLAUF	MANUELLER NEUSTART	OB 20	STOP
	MANUELLER WIEDERANLAUF	OB 21	es wird ein AUTOMATISCHER WIEDERANLAUF ausgeführt (der OB 22 wird aufgerufen und von vorne bearbeitet); es wird kein "Rest-Anlauf" bearbeitet (keine Fortsetzung von OB 21 bzw. OB 22!)
	AUTOMATISCHER WIEDERANLAUF	OB 22	

4.4 Betriebszustand RUN

Wenn die CPU einen ANLAUF fertig bearbeitet hat (und nur dann), geht sie in den Betriebszustand **RUN**. Dieser Betriebszustand ist durch folgende Merkmale charakterisiert:

Bearbeitung des Anwenderprogramms

Das Anwenderprogramm im OB 1 wird zyklisch bearbeitet, wobei zusätzlich alarmgesteuert weitere Programmteile eingeschachtelt werden können.

Zeiten, Zähler, Prozeßabbild

Alle im Programm gestarteten Zeiten und Zähler "laufen", das **Prozeßabbild wird zyklisch aktualisiert**.

BASP

Das Signal BASP (Befehlsausgabe sperren) ist inaktiv gesetzt. Damit sind alle digitalen Ausgaben freigegeben.

Koppelmerker

Die Koppelmerker – falls im DB 1 programmiert – werden zyklisch aktualisiert.

Im Betriebszustand RUN gibt es folgende Ebenen der Programmbearbeitung:

ZYKLUS

Das Anwenderprogramm im OB 1 wird zyklisch bearbeitet.

*PROZESSALARME/
INTERRUPTS*

Das Anwenderprogramm wird alarmgesteuert bearbeitet (4 Interruptebenen oder 1 Prozeßalarmebene mit 8 Unterebenen).

WECKALARME:

Das Anwenderprogramm wird zeitgesteuert bearbeitet (9 zyklische Weckalarme, 1 Verzögerungsalarm, 1 uhrzeitgesteuerter Weckalarm) .

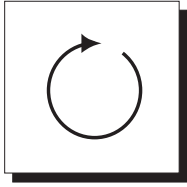
Die Bearbeitungsebenen unterscheiden sich in folgenden Punkten:

- Sie werden durch unterschiedliche Ereignisse ausgelöst.
- Für jede Ebene der Programmbearbeitung existieren als Anwenderschnittstelle ein oder mehrere Organisationsbausteine.

In einer CPU 948 können alle Bearbeitungsebenen gleichzeitig programmiert sein. Der Aufruf der entsprechenden Ebenen erfolgt entsprechend den auftretenden Ereignissen und der vorgegebenen Priorität (siehe Abschnitt 4.2) durch das Systemprogramm.

4.4.1 Zyklische Programm- bearbeitung

Bei den speicherprogrammierbaren Steuerungen herrscht im allgemeinen **die zyklische Programmbearbeitung** (Programmbearbeitungsebene **ZYKLUS**) vor.



Auslösung

Hat die CPU das Anlaufprogramm fehlerfrei beendet, so beginnt sie mit der zyklischen Programmbearbeitung.

Prinzip

Prinzip der zyklischen Programmbearbeitung (Systemleistungen):

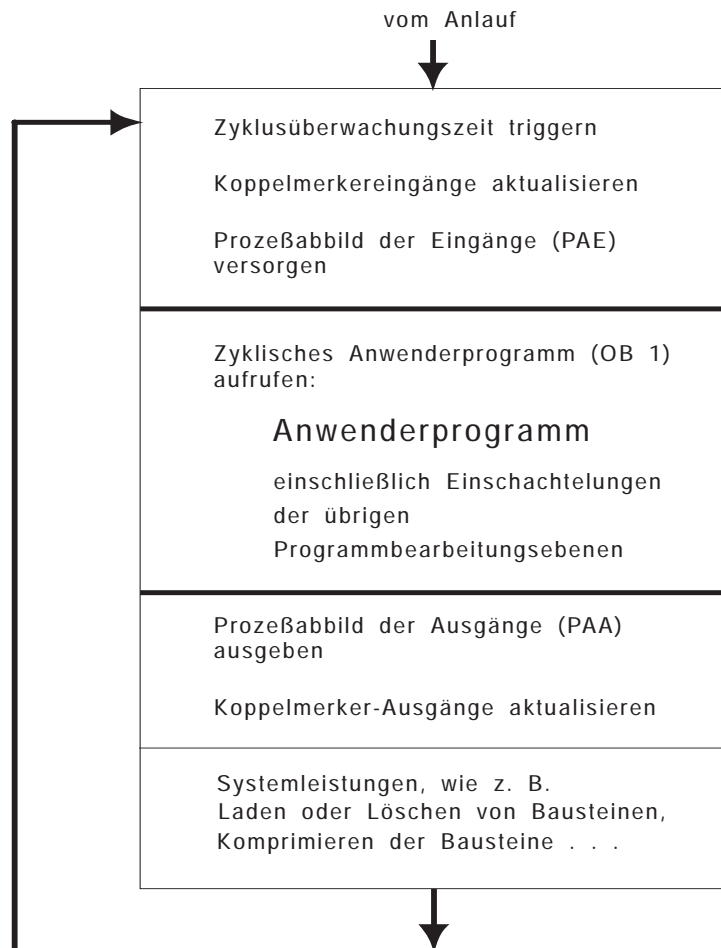


Bild 4-5 Zyklische Programmbearbeitung

Anwenderschnittstelle OB 1

Bei der zyklischen Programmbearbeitung wird als Anwenderschnittstelle regelmäßig der Organisationsbaustein OB 1 aufgerufen. Das STEP-5-Anwenderprogramm im OB 1 wird von Anfang an über die darin von Ihnen programmierten Bausteinaufrufe hinweg durchgehend bearbeitet. Nach den Systemleistungen beginnt die CPU wieder von vorne mit der ersten STEP-5-Anweisung im OB 1.

Im OB 1 programmieren Sie die Aufrufe der Programm-, Funktions- und Schrittbausteine, die im zyklischen Programm bearbeitet werden sollen.

Unterbrechungsstellen

Die zyklische Programmbearbeitung kann unterbrochen werden durch:

- eine zeitgesteuerte Programmbearbeitung an Baustein- oder Befehls Grenzen,
- eine prozeßalarmgesteuerte Programmbearbeitung über Eingangsbyte EB 0 an Bausteingrenzen,
- eine interruptgesteuerte Programmbearbeitung (Interrupts INT A/B/C/D, E, F, G) an Baustein- oder Befehls Grenzen.

Die Unterbrechungsart (an Baustein- oder Befehls Grenzen) legen Sie im Datenbaustein DX 0 fest (siehe Kapitel 7).

Die zyklische Programmbearbeitung kann unabhängig von der DX-0-Parametrierung unterbrochen bzw. **ganz abgebrochen** werden:

- beim Auftreten eines Geräte- oder Programmfehlers (an Befehls Grenzen),
- durch Bedienung:
 - Stoppschalter, HALT (an Befehls Grenzen),
 - PG-Funktion (an Kontrollpunkten – siehe Kapitel 11),
- durch Stopp-Befehl STS (an Befehls Grenzen),
- durch Netzspannungsausfall im Zentralgerät oder im Erweiterungsgerät (an Befehls Grenzen).

4.4.2

Festlegung der zeit- und alarmgesteuerten Programmbearbeitung

Bei der zeit- und alarmgesteuerten Programmbearbeitung stehen Ihnen verschiedenen Typen zur Verfügung, die Sie z. T. nur alternativ nutzen können. Welche der einzelnen Bearbeitungsarten Sie benutzen wollen, legen Sie durch Parametrierung des Datenbausteins DX 0 (siehe Kapitel 7) fest.

Betriebsart "Prozeßalarme über EB 0"

Die Betriebsart "Prozeßalarme über EB 0" ist durch folgende Merkmale charakterisiert:

- es ist **nur Einzelprozessorbetrieb** möglich,
- die Einschachtelung höherpriorer Programmebenen erfolgt **nur an Bausteingrenzen**,
- der Verzögerungsalarm (Bearbeitung durch OB 6) kann **nicht** benutzt werden.
- der uhrzeitgesteuerte Weckalarm (Bearbeitung durch OB 9) kann **nicht** benutzt werden.

Betriebsart "Systeminterrupts"

Die Betriebsart "Systeminterrupts" ist durch folgende Merkmale charakterisiert:

- es ist **Einzel- oder Mehrprozessorbetrieb** möglich,
- die Einschachtelung höherpriorer Programmebenen erfolgt an **Baustein- oder Befehls Grenzen**,
- Verzögerungsalarme werden durch OB 6 **bearbeitet**,
- uhrzeitgesteuerte Weckalarme werden durch OB 9 **bearbeitet**.

Unterbrechungen der zeit- und alarmgesteuerten Programmbearbeitung

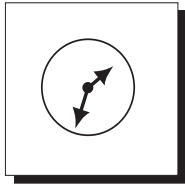
Die zeit- und alarmgesteuerte Programmbearbeitung kann unabhängig von der DX-0-Parametrierung unterbrochen bzw. **ganz abgebrochen** werden:

- beim Auftreten eines Geräte- oder Programmfehlers (an Befehls-
grenzen),
- durch Bedienung:
 - Stoppschalter, MP-STP (an Befehls-
grenzen),
 - PG-Funktion (an Kontrollpunkten – siehe Kapitel 11),
- durch Stopp-Befehl STS (an Befehls-
grenzen),
- durch Netzspannungsausfall im Zentralgerät oder im Erweiterungs-
gerät (an Befehls-
grenzen).

4.4.3

Zeitgesteuerte Programmbearbeitung

Zu dieser Bearbeitungsart gehören der Verzögerungsalarm, der uhrzeitgesteuerte Weckalarm und die zyklischen Weckalarme.



Allen Alarmen ist gemeinsam, daß sie **zeitgesteuert** bearbeitet werden.

Die zeitgesteuerte Programmbearbeitung läuft auf der Ebene **WECKALARME** ab.

Verzögerungsalarm

Auslösung einmalig nach einer freigewählten Verzögerungszeit im Millisekunden-Bereich. Über diesen Alarm wird der Organisationsbaustein OB 6 aufgerufen.

uhrzeitgesteuerter Weckalarm

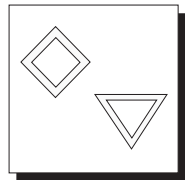
Auslösung in einem freigewähltem Zeitraster oder auch einmalig zu einem absoluten Zeitpunkt. Über diesen Alarm wird der Organisationsbaustein OB 9 aufgerufen.

zyklische Weckalarme

Auslösung in 9 verschiedenen Zeitrastern. Jedem Weckalarm ist ein Organisationsbaustein zugeordnet (OB 10 bis OB 18). Dabei handelt es sich um feste Zyklen, d. h. die Zeitdauer zwischen zwei Programmstarts ist fest.

Prioritäten

Innerhalb der zeitgesteuerten Programmbearbeitung gibt es folgende Einzelprioritäten:



Verzögerungsalarm	OB 6	↑ steigende Priorität
zykl. Weckalarm, Periode 1	OB 10, kürzestes Zeitraster	
zykl. Weckalarm, Periode 2	OB 11	
zykl. Weckalarm, Periode 3	OB 12	
zykl. Weckalarm, Periode 4	OB 13	
zykl. Weckalarm, Periode 5	OB 14	
zykl. Weckalarm, Periode 6	OB 15	
zykl. Weckalarm, Periode 7	OB 16	
zykl. Weckalarm, Periode 8	OB 17	
zykl. Weckalarm, Periode 9	OB 18, längstes Zeitraster)	
uhrzeitgesteuerter Weckal.	OB 9	

Hinweis

Die zeitgesteuerte Alarmbearbeitung im OB 6 und im OB 9 erfolgt nur dann, wenn im DX 0 der Parameter "Prozeßalarme über EB 0= aus" eingestellt ist. In der Voreinstellung des DX-0-Parameters ("Prozeßalarme über EB 0 = ein") werden über OB 6 und OB 9 die entsprechenden Prozeßalarme des EB 0 bearbeitet (siehe Abschnitt 4.5.4).

Verzögerungsalarm

Mit dem Verzögerungsalarm der CPU 948 lassen sich auch kleine Zeitintervalle mit einer Auflösung von 1 ms vorgeben. Wenn die eingestellte Zeit abgelaufen ist, so ruft das Systemprogramm **einmal** den OB 6 auf.

Auslösung

Ein Verzögerungsalarm wird durch Aufruf des Sonderfunktions-Organisationsbausteins OB 153 erzeugt (siehe Abschnitt 6.14). Sobald die mit dem OB 153 parametrisierte Verzögerungszeit abgelaufen ist, unterbricht das Systemprogramm die laufende Programmbearbeitung und ruft den OB 6 (Programmbearbeitungsebene WECKALARME) auf. Danach wird die Programmbearbeitung an der Unterbrechungsstelle wieder aufgenommen.

Voraussetzung für den Betrieb des Verzögerungsalarms ist jedoch, daß im Datenbaustein DX 0 der Parameter "Prozeßalarme über EB 0 = aus" eingestellt ist

Anwenderschnittstelle OB 6

Als Anwenderschnittstelle wird bei einem Verzögerungsalarm der OB 6 aufgerufen. Im OB 6 hinterlegen Sie ein STEP-5-Programm, das in diesem Fall bearbeitet werden soll. Ist der OB 6 nicht geladen, so wird die Programmbearbeitung nicht unterbrochen.

Unterbrechungen

Die Ebene WECKALARME hat aufgrund der Voreinstellung die höchste Priorität der Grundebenen (durch DX-0-Parametrierung änderbar).

Innerhalb der zeitgesteuerten Programmbearbeitung hat die Bearbeitung des Verzögerungsalarms die höchste Priorität.

Wegen dieser Prioritätsverteilung kann die Bearbeitung des Verzögerungsalarms durch kein anderes Anwenderprogramm unterbrochen werden.

Besonderheiten

- Ein Verzögerungsalarm wird nur im Betriebszustand RUN bearbeitet. Verzögerungsalarms, die im STOP, bei **Netzausfall** oder im ANLAUF auftreten, werden **verworfen**.
- Ein einmal erzeugter Verzögerungsalarm (= OB 153-Aufruf wurde bearbeitet) bleibt beim Übergang in den STOP und bei NETZ AUS **nicht** erhalten.

- Wenn Sie einen neuen Verzögerungsalarm erzeugen, d. h. den OB 153 mit neuen Parametern aufrufen, wird ein zuvor eingestellter Verzögerungsalarm storniert. Ein sich bereits in Bearbeitung befindender Verzögerungsalarm wird fortgesetzt. Es ist also immer nur **ein** Verzögerungsalarm gültig.
- Tritt ein Verzögerungsalarm auf, ohne daß der vorangegangene vollständig bearbeitet ist, so wird der neue Alarm verworfen. **Verzögerungsalarme werden nicht auf Weckfehler überprüft!**
- Beachten Sie die Sonderfunktionen OB 122 und OB 142, mit denen Sie die Bearbeitung von Verzögerungsalarmen sperren bzw. verzögern können.

Uhrzeitgesteuerter Weckalarm

Auf der CPU 948 steht eine gepufferte Uhr (zentrale Pufferung über die Stromversorgung des Zentralgerätes) zur Verfügung, die Sie per STEP-5-Programm setzen und auslesen können. Über diese Uhr läßt sich ein Programmteil zeitgesteuert bearbeiten.

Während der Verzögerungsalarm für schnelle Vorgänge eingesetzt wird, ist der uhrzeitgesteuerte Weckalarm besonders geeignet für die Bearbeitung von **einmaligen** Vorgängen oder Vorgängen, die in **großen Zeitabständen zyklisch** auftreten wie z. B. stündlich, täglich oder an jedem Montag. Ist der eingestellte Zeitpunkt erreicht, so ruft das Systemprogramm den OB 9 auf.

Auslösung

Ein uhrzeitgesteuerter Weckalarm (Zeitauftrag) wird durch Aufruf des Sonderfunktions-Organisationsbausteins OB 151 erzeugt (siehe Kapitel 6.13). Ist der mit dem OB 151 eingestellte Zeitpunkt (eine Uhrzeit, ein Datum) erreicht, so wird der Zeitauftrag bearbeitet. Das Systemprogramm unterbricht die laufende Programmbearbeitung und ruft den OB 9 (Programmbearbeitungsebene WECKALARME) auf. Danach wird die Programmbearbeitung an der Unterbrechungsstelle wieder aufgenommen.

Voraussetzung für den Betrieb des Verzögerungsalarms ist jedoch, daß im Datenbaustein DX 0 der Parameter "Prozeßalarms über EB 0 = aus" eingestellt ist.

Anwenderschnittstelle OB 9

Als Anwenderschnittstelle wird bei einem uhrzeitgesteuerten Weckalarm der OB 9 aufgerufen. Im OB 9 hinterlegen Sie ein STEP-5-Programm, das in diesem Fall bearbeitet werden soll. Ist der OB 9 nicht geladen, so wird die Programmbearbeitung nicht unterbrochen.

Unterbrechungen

Die Ebene WECKALARME hat aufgrund der Voreinstellung die höchste Priorität der Grundebenen (durch DX-0-Parametrierung änderbar).

Innerhalb der zeitgesteuerten Programmbearbeitung hat die Bearbeitung des uhrzeitgesteuerten Weckalarms die niedrigste Priorität. Sie kann daher unterbrochen werden durch die Bearbeitung eines Verzögerungsalarms oder eines zyklischen Weckalarms.

Besonderheiten

- Ein uhrzeitgesteuerter Weckalarm wird nur im Betriebszustand RUN bearbeitet. Uhrzeitgesteuerte Weckalarme, die im STOP, bei **Netzausfall** oder im ANLAUF auftreten, werden **verworfen**.
- Ein einmal erzeugter uhrzeitgesteuerter Weckalarm (= OB 151-Aufruf wurde bearbeitet) bleibt bei einem WIEDERANLAUF und über NETZ AUS/ NETZ EIN erhalten; bei einem NEU-START wird er jedoch gelöscht.
- Wenn Sie einen neuen uhrzeitgesteuerten Weckalarm erzeugen, d. h. den OB 151 mit neuen Zeitwerten aufrufen, wird ein zuvor eingestellter uhrzeitgesteuerter Weckalarm storniert. Ein sich bereits in Bearbeitung befindender uhrzeitgesteuerter Weckalarm wird fortgesetzt. Es ist also immer nur **ein** uhrzeitgesteuerter Weckalarm gültig.
- Tritt ein uhrzeitgesteuerter Weckalarm auf, ohne daß der vorangegangene vollständig bearbeitet ist, so wird der neue Weckalarm verworfen. **Uhrzeitgesteuerte Weckalarme werden nicht auf Weckfehler überprüft!**
- Beachten Sie die Sonderfunktionen OB 122 und OB 142, mit denen Sie die Bearbeitung von uhrzeitgesteuerten Weckalarmen sperren bzw. verzögern können.

Zyklische Weckalarme

In der CPU 948 können Sie 9 verschiedene Programme zeitgesteuert bearbeiten lassen, von denen jedes in einem anderen Zeitraster zyklisch aufgerufen wird.

Auslösung

Der Grundtakt für die Weckalarmbeabeitung ist auf 100 ms voreingestellt. Sie können ihn durch einen speziellen Parameter im Datenbaustein DX 0 in Stufen von 10 ms verstellen (Grundtakt = $yy * 10$ ms mit: $01H \leq yy \leq FFH$).

Bei der Einstellung des Grundtaktes können Sie sich an der kleinsten Zeit orientieren, die Ihre Anwendung für die zyklische Bearbeitung benötigt.

Zeitraster

Die Bearbeitung der zyklischen Weckalarme erfolgt in festen Zeitrastern mit 9 verschiedenen Zeitstufen (Perioden). Jeder Zeitstufe ist ein bestimmter Organisationsbaustein zugeordnet. Sie können zwischen zwei Zeitrastern wählen. Die Auswahl erfolgt durch einen speziellen Parameter im Datenbaustein DX 0.

Folgende Tabelle zeigt Ihnen die beiden Zeitraster mit den Zeitstufen und deren Zuordnung zu den Organisationsbausteinen.

Tabelle 4-5 Zeitraster und Zeitstufen der WECKALARME

Zeitraster 1 (Voreinstellung)	Zeitraster 2	aufgerufener OB
1 mal Grundtakt	1 mal Grundtakt	OB 10
2 mal Grundtakt	2 mal Grundtakt	OB 11
5 mal Grundtakt	4 mal Grundtakt	OB 12
10 mal Grundtakt	8 mal Grundtakt	OB 13
20 mal Grundtakt	16 mal Grundtakt	OB 14
50 mal Grundtakt	32 mal Grundtakt	OB 15
100 mal Grundtakt	64 mal Grundtakt	OB 16
200 mal Grundtakt	128 mal Grundtakt	OB 17
500 mal Grundtakt	256 mal Grundtakt	OB 18

Hinweis

Der **erste** Aufruf eines WECKALARM-OBs nach einem Anlauf erfolgt nicht exakt nach Ablauf der dem OB zugeordneten Zeit.

Ist beispielsweise für den OB 18 die Weckzeit "500 s" eingestellt (Grundtakteeinstellung im DX 0 = 1 s), so erfolgt nach einem NEUSTART oder WIEDERANLAUF der erste Aufruf des OB 18 nach ca. 20 s. Alle weiteren Aufrufe erfolgen dann jedoch nach jeweils genau 500 s.

*Anwenderschnittstellen
OB 10 bis OB 18*

Als Anwenderschnittstelle wird bei Auftreten eines bestimmten Weckalarms an der nächsten Bausteingrenze (bzw. Befehls Grenze) der zugehörige Organisationsbaustein eingeschachtelt.

Programmieren Sie beispielsweise im OB 10 denjenigen Programmteil, der alle 100 ms (Voreinstellung) in die zyklische Programmbearbeitung eingeschoben werden soll.

Es wird nur der Weckalarmen bearbeitet, zu dem **der zugeordnete Organisationsbaustein geladen ist**. Ist keiner der Organisationsbausteine OB 10 bis OB 18 geladen, so erfolgt keine zeitgesteuerte Programmbearbeitung und damit auch keine Unterbrechung der zyklischen Programmbearbeitung.

Sie können die Bearbeitung der Weckalarme durch einen Parameter im Datenbaustein DX 0 auch abschalten z. B. für Ihren Programmtest.

Unterbrechungen

Die Ebene WECKALARME hat aufgrund der Voreinstellung die höchste Priorität der Grundebenen (durch DX-0-Parametrierung änderbar).

Aufgrund der Prioritätsverteilung innerhalb der zeitgesteuerten Programmbearbeitung kann es zu folgenden Unterbrechungen bei der Bearbeitung eines zyklischen Weckalarms kommen:

- Die Bearbeitung eines zyklischen Weckalarms kann durch die Bearbeitung eines Verzögerungsalarms unterbrochen werden.
- Organisationsbausteine mit kürzeren Zeitrastern sind höherprior und können Organisationsbausteine mit längeren Zeitrastern unterbrechen (z. B. unterbricht der OB 12 den OB 17).

Hinweis

Bei den **drei kürzesten Zeitrastern (OB 10 bis 12)** ist eine mehrfache Bearbeitung ohne Unterbrechung möglich. Kommt beispielsweise noch während der Bearbeitung eines OB 10 ein weiterer Weckalarm für den OB 10, so wird zunächst die laufende Bearbeitung des OB 10 zu Ende geführt. Danach wird der OB 10 sofort erneut aufgerufen. Stehen allerdings für eines der drei Zeitraster **mehr als drei Weckalarme** an, so kommt es zu einem **Weckfehler**.

Weckfehler

In der CPU 948 gibt es zwei verschiedene Arten von Weckfehlern:

Fehlerart/Ursache	USTACK-Kennung	Reaktion der CPU
<p>Warteschlangenüberlauf bei der Weckalarmbearbeitung:</p> <ul style="list-style-type: none"> - es stehen für eine der drei kürzesten Zeitraster (OB 10 bis 12) mehr als drei Weckalarme zur Bearbeitung an, - einer der übrigen OB (OB 13 bis 18) wird erneut aufgerufen, ehe seine Bearbeitung vollständig abgeschlossen ist 	Am Programmiergerät ist bei "Ausgabe USTACK" in den Steuerbits die Fehlerkennung WEFES angekreuzt	Das Systemprogramm ruft als Anwenderschnittstelle den OB 33 auf. Ist dieser nicht geladen, so geht die CPU in den Stoppzustand .
Bei der Unterbrechbarkeit an Bausteingrenzen wird die Zeitbearbeitung durch die Laufzeit eines Bausteins im zyklischen Anwenderprogramm blockiert; die Laufzeit des Bausteins ist größer als der im DX 0 eingestellte Grundtakt.	Am Programmiergerät ist bei "Ausgabe USTACK" in den Steuerbits die Fehlerkennung WEFEH angekreuzt.	Das Systemprogramm ruft als Anwenderschnittstelle den OB 33 auf. Ist dieser nicht geladen, so setzt das Systemprogramm die Programm-bearbeitung fort .

*Fehlerreaktion
OB 33*

Im OB 33 können Sie die gewünschte Reaktion auf die genannten Weckfehler programmieren. Bei seinem Aufruf übergibt das Systemprogramm im AKKU-1-L (Bit-Nr. 0 bis 9) eine Weckfehlerkennung. Die Bedeutung der Kennungsbits (Bit = '1') können Sie der nachfolgenden Tabelle entnehmen.

Tabelle 4-6 Weckfehlerkennung: Bedeutung der Kennungsbits in AKKU-1-L

Bit-Nummer	Bedeutung
0	Warteschlangen-Überlauf bei Weckalarm-Periode 1 (bei OB 10 stehen mehr als 3 Weckalarme zur Bearbeitung an.)
1	Warteschlangen-Überlauf bei Weckalarm-Periode 2 (bei OB 11 stehen mehr als 3 Weckalarme zur Bearbeitung an.)
2	Warteschlangen-Überlauf bei Weckalarm-Periode 3 (bei OB 12 stehen mehr als 3 Weckalarme zur Bearbeitung an.)
3	Warteschlangen-Überlauf bei Weckalarm-Periode 4 (OB 13 ist erneut aufgerufen worden, bevor der vorhergehende Aufruf abgearbeitet worden ist.)
4	Warteschlangen-Überlauf bei Weckalarm-Periode 5 (OB 14 ist erneut aufgerufen worden, bevor der vorhergehende Aufruf abgearbeitet worden ist.)

Bit-Nummer	Bedeutung
5	Warteschlangen-Überlauf bei Weckalarm-Periode 6 (OB 15 ist erneut aufgerufen worden, bevor der vorhergehende Aufruf abgearbeitet worden ist.)
6	Warteschlangen-Überlauf bei Weckalarm-Periode 7 (OB 16 ist erneut aufgerufen worden, bevor der vorhergehende Aufruf abgearbeitet worden ist.)
7	Warteschlangen-Überlauf bei Weckalarm-Periode 8 (OB 17 ist erneut aufgerufen worden, bevor der vorhergehende Aufruf abgearbeitet worden ist.)
8	Warteschlangen-Überlauf bei Weckalarm-Periode 9 (OB 18 ist erneut aufgerufen worden, bevor der vorhergehende Aufruf abgearbeitet worden ist.)
9	Weckalarmtakt zu lange maskiert

Nach der Bearbeitung des OB 33 wird das Programm im unterbrochenen Weckalarm-OB fortgesetzt.

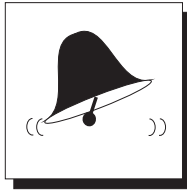
Hinweis

Bei der Einstellung "Unterbrechbarkeit an Bausteingrenzen" zeigt nach einem Weckfehler der SAZ nicht auf den Baustein, an dessen Bausteingrenze (BE-Operation) der Weckfehler aufgetreten ist, sondern auf den Baustein, der den fehlerverursachenden Baustein aufgerufen hat (Rücksprungadresse).

Für die zeitgesteuerte Programmbearbeitung haben Sie im Datenbaustein DX 0 (siehe Kapitel 7) folgende Parametriermöglichkeiten:

- Einstellen des Grundtaktes,
- Einstellen des Zeittaktverteilers,
- Einstellung der Priorität gegenüber der alarmgesteuerten Programmbearbeitung,
- Ein-/Ausschalten der Weckalarmbearbeitung.

4.4.4

**Alarmgesteuerte Programm-
bearbeitung**

Mit der CPU 948 sind – abhängig von der eingestellten Betriebsart – zwei unterschiedliche Arten der alarmgesteuerten Programmbearbeitung möglich:

- **PROZESSALARME**
über Eingangsbyte EB 0 (max. 8 Alarme),
- **INTERRUPTS**
über Signalleitungen des S5-Busses (max. 4 Interrupts).

**PROZESSALARME über
Eingangsbyte EB 0**

Voraussetzung für die Bearbeitung der Prozeßalarme ist, daß Sie die Voreinstellung "Prozeßalarme über EB 0 = ein" im Datenbaustein DX 0 nicht verändert haben.

Eine durch Prozeßalarm gesteuerte Programmbearbeitung liegt vor, wenn ein Signalzustandswechsel im Eingangsbyte EB 0 die CPU veranlaßt, die aktuelle Programmbearbeitung zu unterbrechen und einen spezifischen Programmteil zu bearbeiten.

Hinweis

Wenn Sie "Bearbeitung der Prozeßalarme über EB 0" eingeschaltet haben, können Sie den Verzögerungsalarm, den uhrzeitgesteuerten Weckalarm und die Systeminterrupts **nicht** benutzen!

Auslösung

Der Signalzustandswechsel eines Bits im Eingangsbyte EB 0 löst den Prozeßalarm aus.

*Anwenderschnittstellen
OB 2 bis OB 9*

Als Anwenderschnittstelle wird beim Auftreten eines Prozeßalarms jeweils einer der in der nachfolgenden Tabelle aufgeführten OB aufgerufen.

Tabelle 4-7 Anwenderschnittstellen für Prozeßalarme

Signalzustandswechsel in EB 0 durch Bit	aufgerufener OB
E 0.0	OB 2
E 0.1	OB 3
E 0.2	OB 4
E 0.3	OB 5
E 0.4	OB 6
E 0.5	OB 7
E 0.6	OB 8
E 0.7	OB 9

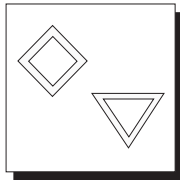
In den Organisationsbausteinen OB 2 bis OB 9 programmieren Sie die Teile Ihres STEP-5-Programms, die beim Auftreten eines entsprechenden Prozeßalarms über das Eingangsbyte EB 0 bearbeitet werden sollen.

Ist der betreffende OB **nicht geladen**, so wird die Programmbearbeitung nicht unterbrochen. Es findet keine alarmgesteuerte Programmbearbeitung statt.

Hinweis

Quittiert die Peripheriebaugruppe beim Zugriff der CPU auf EB 0 im laufenden Betrieb nicht mehr, so erkennt das Systemprogramm einen Quittungsverzug und ruft die Anwenderschnittstelle OB 28 auf. Ist der OB 28 nicht geladen, so geht die CPU in den Stoppzustand.

Prioritäten der Prozeßalarme



Die Ebene PROZESSALARME hat aufgrund der Voreinstellung eine niedrigere Priorität als die Ebene WECKALARME.

Durch einen Parameter im Datenbaustein DX 0 können Sie die Voreinstellung so ändern, daß die Priorität der Ebene PROZESSALARME höher ist als die der WECKALARME.

Innerhalb der Prozeßalarmbearbeitung sind folgende Prioritäten der einzelnen Prozeßalarme festgelegt:

E 0.0	OB 2	↑ steigende Priorität
E 0.1	OB 3	
E 0.2	OB 4	
E 0.3	OB 5	
E 0.4	OB 6	
E 0.5	OB 7	
E 0.6	OB 8	
E 0.7	OB 9	

Bei Prozeßalarmen ist **keine** geschachtelte Bearbeitung möglich!

Wenn ein Prozeßalarm-OB vollständig bearbeitet worden ist und zu diesem Zeitpunkt noch weitere Prozeßalarme anstehen, so wird vom Systemprogramm der OB mit der nächstniedrigeren Priorität aufgerufen und bearbeitet.

Die Programmbearbeitungsebene PROZESSALARME wird erst dann verlassen, wenn jeder Zustandswechsel im Eingangsbyte EB 0 berücksichtigt und der dazugehörige OB vollständig bearbeitet wurde.

Hinweis

Eine prozeßalarmgesteuerte Programmbearbeitung kann **nicht** durch eine erneute prozeßalarmgesteuerte Programmbearbeitung unterbrochen werden.

**INTERRUPTS über
Signalleitungen des
S5-Busses**

Eine durch Interrupt gesteuerte Programmbearbeitung liegt vor, wenn ein S5-Bus-Signal einer interruptfähigen Peripheriebaugruppe (z. B. Digitaleingabe, IP, CP) die CPU veranlaßt, die Programmbearbeitung zu unterbrechen und einen spezifischen Programmteil zu bearbeiten.

Hinweis

Möchten Sie in Ihrer CPU die interruptgesteuerte Programmbearbeitung über S5-Bus-Signalleitungen betreiben, so müssen Sie im DX 0 "Prozeßalarme über EB 0 = aus" parametrieren und die einzelnen Interrupts über DX-0-Parameter aktivieren.

Darüberhinaus müssen die Interrupts über Steckbrücken auf der Baugruppe freigegeben worden sein.

Im Gegensatz zur CPU 946/947 können Sie bei der CPU 948 als Betriebsart im DX 0 einstellen: "Unterbrechung an Bausteingrenzen" oder "Unterbrechung an Befehlsgrenzen".

**Brückeneinstellung für
Systeminterrupts**

Für die interruptgesteuerte Programmbearbeitung mit der CPU 948 stehen Ihnen 4 Systeminterrupts zur Verfügung:

- INT A/B/C/D (abhängig vom Steckplatz der CPU, siehe Systemhandbuch /2/),
- INT E,
- INT F
und
- INT G.

Die Interrupts, die Sie nutzen wollen, müssen Sie mit Hilfe der beiliegenden Steckbrücken freigeben. Der Brückenstecker befindet sich auf der Grundplatine oberhalb des Schachtes für die Memory Card. Die genaue Lage entnehmen Sie dem Anhang 1:

Auslösung

Der aktive Zustand einer Interruptleitung auf dem S5-Bus löst den Interrupt aus. Das Interruptsignal ist pegelgetriggert (Low-Pegel). Zum Quittieren der Interrupts beachten Sie bitte die Betriebsanleitung der interruptauslösenden Baugruppe.

*Anwenderschnittstellen
OB 2 bis OB 5*

Als Anwenderschnittstelle wird beim Auftreten eines Interrupts einer der in der nachfolgenden Tabelle aufgeführten OB aufgerufen.

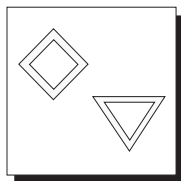
Tabelle 4-8 Anwenderschnittstellen für Interrupts

Interrupt ausgelöst durch	aufgerufener OB
Interruptsignal X (A, B, C oder D, steckplatzabhängig)	OB 2
Interruptsignal E	OB 3
Interruptsignal F	OB 4
Interruptsignal G	OB 5

Tritt z. B. das Interruptsignal 'F' auf, so ruft das Systemprogramm den OB 4 auf.

Ist der betreffende OB **nicht geladen**, so wird die Programmbearbeitung nicht unterbrochen. Es findet keine interruptgesteuerte Programmbearbeitung statt.

Prioritäten der Interrupts



Die Ebene INTERRUPTS hat aufgrund der Voreinstellung eine niedrigere Priorität als die Ebene WECKALARME.

Durch einen Parameter im Datenbaustein DX 0 können Sie die Voreinstellung so ändern, daß die Priorität der Ebene INTERRUPTS höher ist als die der WECKALARME.

Inerhalb der Interruptbearbeitung wird die Priorität der einzelnen Interrupts nach folgendem Schema festgelegt:

- Stehen mehrere Interrupts zur Bearbeitung an, so werden die zugehörigen Organisationsbausteine nach der Prioritätsreihenfolge aufgerufen, die Sie im DX 0 parametrieren haben (Einzelpriorität).

Sie können für die vier Interrupts die Prioritätsstufen 1 bis 5 angeben.

Wenn ein Interrupt-OB vollständig bearbeitet worden ist und zu diesem Zeitpunkt noch weitere Interrupts anstehen, so wird vom Systemprogramm der OB mit der nächstniedrigeren Priorität aufgerufen und bearbeitet.

Die Programmbearbeitungsebene INTERRUPTS wird erst dann verlassen, wenn jeder aktive Signalzustand (Low-Pegel) einer Interruptleitung auf dem S5-Bus berücksichtigt und der dazugehörige OB vollständig bearbeitet wurde.

Hinweis

Eine interruptgesteuerte Programmbearbeitung kann **nicht** durch erneutes Auftreten desselben Interrupts unterbrochen werden.

Sperren der alarmgesteuerten Bearbeitung

Ein alarmgesteuertes Programm wird an einer Bausteingrenze oder einer STEP-5-Befehlsgrenze in das zyklische Programm eingeschoben. Diese Unterbrechung kann sich ungünstig auswirken, wenn ein zyklischer Programmteil in einer bestimmten Zeit bearbeitet werden muß (um z. B. eine bestimmte Reaktionszeit zu erreichen) oder eine Befehlsfolge nicht unterbrochen werden darf (z. B. beim Lesen oder Schreiben von zusammengehörenden Werten).

Wenn ein Programmteil durch eine alarmgesteuerte Bearbeitung **nicht** unterbrochen werden darf, kommen folgende Programmiermöglichkeiten in Frage:

bei Unterbrechbarkeit an Bausteingrenzen

- Programmieren Sie diesen Programmteil so, daß er keinen Bausteinwechsel enthält. Programmteile, die keinen Bausteinwechsel enthalten, können dann auch nicht unterbrochen werden.
- Verwenden Sie den OB 122, mit dem Sie die Bearbeitung von auftretenden Prozeßalarmen für einen bestimmten Programmteil sperren können. Allerdings werden damit auch die Weckalarme gesperrt (siehe Abschnitt 6.3).
- Programmieren Sie die STEP-5-Operation 'AS' (Prozeßalarme sperren). Mit der Operation 'AF' (Prozeßalarme freigeben) geben Sie die Alarmbearbeitung wieder frei.

Zwischen diesen beiden Operationen wird keine prozeßalarmgesteuerte Programmbearbeitung durchgeführt. Der dazwischen stehende Programmteil kann durch auftretende Prozeßalarme **nicht** unterbrochen werden.

'AS' und 'AF' sind nur in Funktionsbausteinen möglich (Ergänzender Operationsvorrat – siehe Abschnitt 3.5.4) und gelten nur für Prozeßalarme über EB 0!

Hinweis

Wird ein Prozeßalarm über OB 122 gesperrt oder über OB 142 verzögert, so ist die AF-Operation unwirksam.

bei Unterbrechbarkeit an Befehls Grenzen

- Programmieren Sie diesen nicht unterbrechbaren Programmteil selbst in einem Interrupt-OB und versehen Sie diesen mit der höchsten Priorität.
- Verwenden Sie die Sonderfunktion OB 122. Mit ihr können Sie Interrupts und Weckalarme sperren (siehe Abschnitt 6.3).
- Mit den Operationen LIM und SIM (Systemoperationen – siehe Abschnitt 3.5.4) lesen bzw. setzen Sie die 32-bit-Interruptmaske.

Die Alarmbearbeitung kann im Datenbaustein DX 0 auch vollständig bzw. für die einzelnen Interrupts getrennt abgeschaltet werden. Dies ist jedoch nur nach einem NEUSTART möglich (siehe Kapitel 7), da der DX 0 nur bei NEUSTART ausgewertet wird.

Reaktionszeit

Die Reaktionszeit auf eine Prozeßalarm-/Interruptanforderung entspricht der Bearbeitungszeit eines Bausteins (bei Unterbrechbarkeit an Bausteingrenzen) bzw. eines STEP-5-Befehls (bei Unterbrechbarkeit an Befehls Grenzen). Wenn jedoch zum Zeitpunkt der Unterbrechung der zyklischen Programmbearbeitung noch höherpriorie Weckalarme anstehen, so wird das alarmgesteuerte Programm erst dann bearbeitet, wenn alle anstehenden Weckalarme vollständig abgearbeitet worden sind.

Die maximale Reaktionszeit zwischen dem Auftreten und der Bearbeitung eines Prozeßalarms/Interrupts wächst in diesem Fall um die Bearbeitungszeit der höherpriorien Weckalarme.

Programmbearbeitungsebenen und Merker

Wenn Sie Ihr Anwenderprogramm nicht nur zyklisch, sondern auch zeit- und/oder alarmgesteuert bearbeiten lassen, besteht die Gefahr, daß Sie Ihre Merker überschreiben. Dies ist der Fall, wenn in unterschiedlichen Programmbearbeitungsebenen auf gleiche Merkerbereiche zugegriffen wird.

Aus diesem Grund sollten Sie entweder die Merker einzelnen Programmbearbeitungsebenen fest zuteilen oder zu Beginn einer zeit- oder alarmgesteuerten Programmbearbeitung die Signalzustände der (doppelt belegten) Merker in einen Datenbaustein "retten" und am Ende der unterbrechenden Bearbeitung wieder zurückschreiben. Dasselbe gilt für den Wiederanlauf.

Um eine Doppelbelegung von Merkern zu vermeiden, können Sie für die meisten Anwendungsfälle auch S-Merker verwenden. Spezielle "Rettungsmaßnahmen" für Merker sind dann nicht erforderlich, sofern Sie S-Merker den einzelnen Programmbearbeitungsebenen fest zuordnen (S-Merker stehen in genügender Anzahl zur Verfügung).

Unterbrechungs- und Fehlerbehandlung

5

Inhalt von Kapitel 5

5.1	Häufige Fehler im Anwenderprogramm	5 - 4
5.2	Fehlerinformationen	5 - 5
5.3	Vorgehen bei der Fehleranalyse	5 - 8
5.4	Steuerbits und Unterbrechungsstack	5 - 9
5.4.1	Steuerbits	5 - 10
5.4.2	USTACK-Inhalt	5 - 14
5.4.3	Beispiel zur Fehlerdiagnose über USTACK	5 - 19
5.5	Fehlerbehandlung über Organisationsbausteine	5 - 20
5.6	Fehlerursachen und Fehlerreaktionen der CPU	5 - 23
5.6.1	OB 19: Aufrufen eines nicht geladenen Codebausteins (KB)	5 - 24
5.6.2	OB 19: Aufschlagen eines nicht geladenen Datenbausteins (KDB)	5 - 24
5.6.3	OB 23/24, OB 28/29: Quittungsverzug (QVZ)	5 - 25
5.6.4	OB 25: Adressierfehler (ADF)	5 - 26
5.6.5	OB 26: Zykluszeitfehler (ZYK)	5 - 27
5.6.6	OB 27: (Substitutionsfehler SUF)	5 - 28
5.6.7	OB 30: Parityfehler und QVZ beim Anwenderspeicher (PARE)	5 - 28
5.6.8	OB 32: Lade- und Transferfehler (TLAF)	5 - 29
5.6.9	OB 33: Weckfehler (WEFES/WEFEH)	5 - 30
5.6.10	OB 34: Fehler bei E DB/EX DX (FEDBX)	5 - 32
5.6.11	OB 35: Kommunikationsfehler	5 - 32
5.6.12	OB 36: Fehler bei Selbsttest	5 - 33
5.7	Selbsttest	5 - 34
5.7.1	Übersicht	5 - 34
5.7.2	Beschreibung der Testfunktionen	5 - 35
5.7.3	Einstellungen	5 - 37
5.7.4	Fehlerbehandlung	5 - 38

5

Unterbrechungs- und Fehlerbehandlung

In diesem Kapitel erfahren Sie, wie Sie Fehler beim Planen und Programmieren Ihrer STEP-5-Programme vermeiden können. Sie werden informiert, welche Hilfen Ihnen das Systemprogramm zur Fehlerdiagnose und evtl. Fehlerreaktion zur Verfügung stellt und in welchen Bausteinen Sie Reaktionen auf bestimmte Fehler programmieren können. Schließlich erfahren Sie, wie Sie integrierte Systemfunktionen für einen Selbsttest der CPU 948 aktivieren können.

5.1 Häufige Fehler im Anwenderprogramm

Das Systemprogramm kann fehlerhaftes Arbeiten der CPU, Fehler in der Systemprogrammbearbeitung oder Auswirkungen einer fehlerhaften Programmierung durch den Anwender feststellen.

Die folgende Liste enthält eine Aufzählung von Fehlern, die bei der Inbetriebnahme des Anwenderprogramms am häufigsten auftreten, die Sie jedoch schon beim Erstellen Ihres Programms leicht vermeiden können.

Beachten Sie aus diesem Grund beim Entwurf, bei der Implementierung und bei der Inbetriebnahme Ihres STEP-5-Programms bitte folgende Punkte:

- Bei der Angabe von Byte-Adressen für Ein- und Ausgänge müssen entsprechend adressierte Baugruppen im Zentralgerät oder Erweiterungsgerät stecken.
- Achten Sie darauf, daß alle Operanden mit den korrekten Parametern versorgt werden.
- Vorsicht beim nachträglichen Ändern von Funktionsbausteinen. Kontrollieren Sie, ob die FB/FX mit den richtigen Operanden parametrisiert sind und ob alle Aktualoperanden angegeben sind.
- Ausgänge, Merker, Zeiten und Zähler sollten nicht an mehreren Stellen im Programm mit entgegengesetzt wirkenden Operationen bearbeitet werden.
- Zeiten sollten nur einmal im Zyklus abgefragt werden (z. B. U T1).
- Sorgen Sie dafür, daß alle im Programm aufgerufenen Datenbausteine vorhanden und ausreichend lang sind.
- Überprüfen Sie, ob alle aufgerufenen Bausteine auch tatsächlich im Speicher vorhanden sind.
- Schmiermerker sollten von alarm- und zeitgesteuerten Programmen gerettet und am Schluß der Bearbeitung wieder geladen werden, wenn diese von anderen Bausteinen (z. B. Standard-FB) benötigt werden.

5.2 Fehlerinformationen

Wenn im Anlauf oder bei der zyklischen Bearbeitung des Anwenderprogramms ein Fehler auftritt, haben Sie verschiedene "Informationsquellen" zur Verfügung, um diesem Fehler auf die Spur zu kommen. Dies sind:

- LEDs auf der Frontplatte der CPU
- Unterbrechungsstack USTACK und Steuerbits
- Bausteinstack BSTACK

In den nachfolgenden Abschnitten können Sie nachlesen, welche Hilfsmittel es zur Auswertung dieser Informationsquellen gibt und wie Sie die Fehlerinformation für die Analyse einer Störung verwenden können.

LEDs auf der Frontplatte der CPU

Orientieren Sie sich im Falle eines unerwünschten Stoppzustandes an den Leuchtdioden. Sie können Ihnen Hinweise auf die Fehlerursache geben:

LED-Anzeige	Bedeutung
STOP-LED zeigt Dauerlicht	Die unterschiedlichen Anzeigen der STOP-LED deuten auf bestimmte Unterbrechungs- und Fehlerursachen hin.
STOP-LED blinkt langsam	
STOP-LED blinkt schnell	
SYS-FAULT-LED zeigt Dauerlicht	Beachten Sie hierzu die Ausführungen im Abschnitt 4.1.
ADF-LED zeigt Dauerlicht	Adressierfehler
QVZ-LED zeigt Dauerlicht	Quittungsverzug
ZYK-LED zeigt Dauerlicht	Zykluszeitfehler

*PG-Online-Funktion
AUSGABE USTACK*

Die PG-Online-Funktion AUSGABE USTACK gibt Ihnen Auskunft über die Zustände CPU-interner Steuerbits und den Inhalt des Unterbrechungsstacks (USTACK).

In den **USTACK** trägt das Systemprogramm beim Übergang in den Stoppzustand alle Informationen ein, die es für einen Wiederanlauf benötigt. Diese Informationen sind:

- Registerinhalte,
- Akkuinhalte,
- STEP-Adreßzähler SAZ

und

- Ergebnisanzeigen.

Die angezeigten Informationen können sich – je nachdem, wo die Unterbrechung erfolgte, die zum STOP geführt hat – auf Anwenderbausteine oder Bausteine des Systemprogramms (OB 0) beziehen.

Diese Einträge sind bei der Fehlerdiagnose eine wertvolle Hilfe.

Vor der Ausgabe des USTACK-**Inhalts** werden zunächst die Zustände der **Steuerbits** angezeigt. Diese markieren den aktuellen Betriebszustand, bestimmte Eigenschaften der CPU und des Anwenderprogramms und geben zusätzliche Hinweise auf die Fehlerursache.

Sie können die Funktion "Ausgabe USTACK" in den Betriebszuständen STOP, ANLAUF und RUN benutzen; allerdings erhalten Sie im ANLAUF und im RUN nur Auskunft über die Steuerbits und nicht über den Inhalt des USTACKS.

Die Bedeutung der Steuerbits und der Aufbau des Unterbrechungsstacks sind in Abschnitt 5.4 ausführlich beschrieben.

*Online-Funktion AUSGABE
BSTACK*

Die PG-Online-Funktion AUSGABE BSTACK gibt Ihnen Auskunft im STOP über den Inhalt des Bausteinstacks (BSTACK — siehe Abschnitt 3.2 "Bausteinschachtelung").

Im BSTACK sind alle Bausteine (Bausteine des Anwenderprogramms und Organisationsbaustein OB 0 des Systemprogramms) aufgeführt, die **nacheinander** bis zum Übergang in den Stoppzustand aufgerufen und noch nicht zu Ende bearbeitet worden sind. Da der BSTACK von unten gefüllt wird, steht in der obersten Zeile der BSTACK-Ausgabe derjenige Baustein, der den fehlerhaften Baustein **aufgerufen** hat.

Bei der Auswertung der **obersten** Zeile erhalten Sie folgende Informationen:

Information	Bedeutung
BAUST.-NR.	Bausteinart und -nummer des Bausteins, der den fehlerhaften Baustein aufgerufen hat
BAUST.-ADR.	absolute Anfangsadresse dieses Bausteins im Anwenderspeicher
RÜCKSPR.-ADR.	Absolutadresse der ersten STEP-5-Operation dieses Bausteins im Anwenderspeicher
REL.-ADR	Relativadresse (= Differenz "RÜCKSPR.-ADR. - BAUST.-ADR.") des nächsten zu bearbeitenden Befehls in diesem Baustein (Relativadressen können vom PG in Betriebsart "Eingabesperre"/Schlüsselschalter und mit S5-DOS ab Stufe IV über Funktionstaste "Adressen" angezeigt werden.)
DB-NR.	Nummer des in diesem Baustein aufgeschlagenen Datenbausteins
DB-ADR	absolute Anfangsadresse dieses Datenbausteins (Adresse des Datenwortes DW 0) im Programmspeicher

Beispiel

AUSGABE BSTACK auswerten:

BAUST.-NR.	BAUST.-ADR.	RUECKSPR.-ADR.	REL.-ADR	DB-NR.	DB-ADR.
PB 3	00090	98	00008		0
PB 2	00050	51	00001		0
PB 1	00040	41	00001		0
OB 1	00010	11	00001		0
OB 66 ¹⁾	E2B10	E2C40	00130		0
OB 63	E0FC0	E12FA	0033A		0
OB 62	E0490	E0CBE	0082E		0
OB 61	E0010	E0273	00263		0

1) Die vor dem OB 1 ausgeführten Bausteine sind interne Bausteine des Systemprogramms (der BSTACK ist chronologisch aufgebaut).

Im Beispiel hat der PB 3 den fehlerhaften Baustein an der relativen Adresse "00008 - 1 = 00007" aufgerufen.

Beim Sprung in den fehlerhaften Baustein war kein Datenbaustein aufgeschlagen.

5.3 Vorgehen bei der Fehleranalyse

Befindet sich die CPU in einem unerwünschten Stoppzustand, so machen Sie sich auf der Suche nach der Fehleranalyse alle Informationen zunutze, die Ihnen zur Verfügung stehen:

Schritt	Aktion
1	Stellen Sie fest, welchen Status die LEDs STOP und SYSFAULT sowie die Fehler-LEDs auf der Frontplatte haben. Daraus können Sie bestimmte Fehler- und Unterbrechungsursachen ableiten.
2	Analysieren Sie mit Hilfe der PG-Online-Funktion AUSGABE USTACK den Zustand der Steuerbits und den Inhalt des USTACKs. Dadurch erhalten Sie weitere Informationen über den Fehlerort und die Fehlerursache.
3	Wählen Sie die PG-Online-Funktion AUSGABE BSTACK an: In der obersten Zeile der BSTACK-Ausgabe finden Sie Angaben zu dem Baustein, der den fehlerverursachenden Baustein aufgerufen hat .
4	Dem Systemdatum BS 75 (siehe Abschnitt 8.3.4) können Sie weitere detaillierte Fehlerinformationen entnehmen.

5.4 Steuerbits und Unterbrechungsstack

Über die Online-Funktionen AG-INFO und AUSGABE USTACK können Sie mit dem Programmiergerät sowohl Betriebszustand, Eigenschaften der CPU und des Anwenderprogramms als auch eventuelle Fehler- und Unterbrechungsursachen analysieren.

Hinweis

Die Ausgabe der **Steuerbits** erhalten Sie in **jedem** Betriebszustand, die Ausgabe des **USTACK-Inhalts** nur im **STOP**.

- **Steuerbits** geben den aktuellen bzw. vorausgegangenen Betriebszustand und die Störungsursache an.
Sind mehrere Fehler aufgetreten, werden in den Steuerbits **alle aufgetretenen** Fehler dargestellt.
- Im **USTACK** wird die jeweilige Unterbrechungsstelle (Adressen) mit den dort aktuellen Anzeigen und Akkuinhalten sowie die Störungsursache angegeben.
Sind mehrere Unterbrechungen aufgetreten, so wird ein **mehrstufiger** Unterbrechungsstack aufgebaut (maximal 5 Stufen):

Tiefe 01 = letzte Unterbrechungsursache,

Tiefe 02 = vorletzte Unterbrechungsursache usw.

Bei USTACK-Überlauf erfolgt ein sofortiger Übergang in den Stoppzustand (HARTER STOP!). Anschließend ist NETZ-AUS/NETZ EIN und ein NEUSTART erforderlich

Die Bedeutung der einzelnen Abkürzungen in den Steuerbits und im Unterbrechungsstack wird nachfolgend erläutert.

Hinweis

Der Text am Bildschirm Ihres Programmiergerätes ist abhängig von der benutzten PG-Software. Daher kann er von der hier abgedruckten Darstellung abweichen. Trotzdem ist die Beschreibung der Bildschirminformation in dieser Programmieranleitung gültig!

5.4.1 Steuerbits

Bei der PG-Ausgabe des USTACKs werden auf der 1. Bildschirmseite die Zustände der Steuerbits angezeigt (siehe Bild 5-1).

Hinweis

Die in Bild 5-1 dargestellte USTACK-Maske bezieht sich auf die PG-Software STEP 5/ST, Vers. 6.3 bzw. STEP 5/MT, Vers. 6.0, jeweils mit "Delta-Diskette CPU 948". Bei älteren Versionen der PG-Software sind einige *Bezeichnungen* der Steuerbits unterschiedlich. Die *Bedeutung* der Steuerbits entspricht jedoch der in den nachfolgenden Tabellen aufgeführten Beschreibung.

S T E U E R B I T S						
SYSTEMBESCHREIBUNG :	E0VH X TEST	GEP X BSTG	BATT BEFG X	EINP MCG	MEHRP	SYNCR
STOPPURSACHE :	PGSTP	HALT	STP	STS	STOPS	BEARBE
	UPROG	USYS	UANL	AFEL	SYSFHL	
ANLAUFKENNUNGEN :	NEUDF X AWEG X	WIEDF ANEG	URLDF MSEG	NEUZU	WIEZU	URLER
FEHLERKENNUNGEN :	QVZIN	PARIN	BSTKF	BSTEF	UMCG	MODUN
	FE2S	SRAMF	UAFEHL	KDB1	KDX0	FDB1
	FDX0	FMODE	FEDBX	QVZNIO	WEFES	DBOUN

Bild 5-1 Beispiel für die 1. Bildschirmseite "AUSGABE USTACK": Steuerbits

Die Steuerbits lassen sich in allen Betriebszuständen ausgeben. Sie markieren den aktuellen oder vorausgegangenen Betriebszustand der CPU und geben Auskunft über bestimmte Eigenschaften der CPU und des STEP-5-Anwenderprogramms.

Die unter FEHLERKENNUNGEN aufgeführten Steuerbits markieren Fehler, die in den Betriebszuständen ANLAUF (z. B. beim ersten Neustart) und RUN (z. B. bei der zeitgesteuerten Programmbearbeitung) auftreten können. Sind mehrere Fehler aufgetreten, so werden in den Steuerbits **alle** aufgetretenen Fehler angezeigt.

In den nachfolgenden Tabellen finden Sie, welche Aussage die einzelnen Bits haben.

Tabelle 5-1 Bedeutung der Steuerbits SYSTEMBESCHREIBUNG

SYSTEMBESCHREIBUNG	
Steuerbit	Bedeutung
E0VH	Das Eingangsbyte EB 0 (Prozeßalarme) ist vorhanden, d. h. die Digitaleingabebaugruppe mit der Adresse '0' war beim letzten NEUSTART gesteckt und hat quittiert.
GEP	Das AG ist mit einer Zentralbatterie gepuffert.
BATT	Batterie im Zentralgerät unklar (BAU)
EINP	Einzelprozessorbetrieb
MEHRP	Mehrprozessorbetrieb
SYNCR	Anlauf der CPUs im Mehrprozessorbetrieb erfolgt synchronisiert
TEST	Testbetrieb
BSTG	DX-0-Einstellung "Unterbrechung an Bausteingrenzen"
BEFG	DX-0-Einstellung "Unterbrechung an Befehls Grenzen"
MCG	Memory Card gesteckt

Tabelle 5-2 Bedeutung der Steuerbits STOPPURSACHE

STOPPURSACHE (siehe BS 7)	
Steuerbit	Bedeutung
PGSTP	Stoppzustand durch PG-Bedienung
HALT	Mehrprozessor-STOP: a) Wahlschalter am KOR in Stellung STOP oder b) Stoppzustand durch STOP-Befehl vom Systemprogramm, wenn beim Auftreten eines Fehlers der zugehörige Fehler-OB nicht geladen ist.
STS	Stoppzustand durch STEP-5-Befehl 'STS' (nach Befehlsausführung)
STOPS	Stoppzustand durch Betriebsartenschalter in Stellung STOP
BEARBE	Stoppzustand nach PG-Funktion BEARBEITUNGS-KONTROLLE-ENDE
UPROG	Stoppzustand durch Anwenderprogramm

STOPPURSACHE (siehe BS 7)	
Steuerbit	Bedeutung
Fortsetzung der Tabelle 5-2:	
USYS	Stoppzustand durch das Systemprogramm (WIEDERANLAUF möglich)
UANL	Stoppzustand durch unzulässige Anlaufart
AFEL	Stoppzustand durch Fehler im Anlaufbaustein
SYSFHL	Stoppzustand durch Systemfehler (möglicherweise hervorgerufen durch einen Anwenderfehler z. B. Überschreiben des System-RAM mit einem Blocktransfer o. ä. (wenn ein Systemfehler angekreuzt ist, erscheint am unteren Bildschirmrand als zusätzliche Information eine vierstellige Hexadezimalzahl/Fehlercode – siehe BS 75 in Kapitel 8)

Tabelle 5-3 Bedeutung der Steuerbits ANLAUFKENNUNGEN

ANLAUFKENNUNGEN (siehe BS 8)	
Steuerbit	Bedeutung
NEUDF	NEUSTART wurde als letzter Anlauf durchgeführt.
WIEDF	WIEDERANLAUF wurde als letzter Anlauf durchgeführt.
URLDF	URLÖSCHEN wurde durchgeführt oder ist aktiv.
NEUZU	NEUSTART ist als nächster Anlauf zulässig.
WIEZU	WIEDERANLAUF ist als nächster Anlauf zulässig.
URLER	URLÖSCHEN ist erforderlich.
AWEG	AUTOMATISCHER WIEDERANLAUF ist eingestellt.
ANEG	AUTOMATISCHER NEUSTART ist eingestellt.
MSEG	Manueller Start ist eingestellt.

Tabelle 5-4 Bedeutung der Steuerbits FEHLERKENNUNGEN

FEHLERKENNUNGEN	
Steuerbit	Bedeutung
QVZIN	Quittungsverzug in Initialisierung
PARIN	Parityfehler in Initialisierung
BSTKF	falsche Bausteinkennung
BSTEF	falsche Bausteinendekennung
UMCG	unzulässige Memory Card gesteckt
MODUN	Inhalt der Memory Card ist größer als der verfügbare interne Anwenderspeicher
FE2S	Fehler auf 2. Schnittstelle
SRAMF	System-RAM-Fehler
UAFEHL	Fehler im Unterbrechungsanzeigenwort (UAW)
KDB1	DB 1 fehlt im Mehrprozessorbetrieb.
KDX0	DX 0 fehlt im Mehrprozessorbetrieb
FDB1	Fehler im DB 1
FDX0	Fehler im DX 0
FMODE	keine Prozeßalarme EB 0 im Mehrprozessorbetrieb zulässig
FEDBX	Fehler bei STEP-5-Operationen E DB, EX DX
QVZNIO	QVZ-Test nicht in Ordnung
WEFES	Weckfehler-Software: Warteschlangenüberlauf
DB0UN	DB 0 seit dem letzten NEUSTART verändert, deshalb kein WIEDERANLAUF möglich

5.4.2 USTACK-Inhalt

Befindet sich die CPU im Stoppzustand, so erscheint nach Ausgabe der Steuerbits und Betätigen der Übernahmetaste auf dem Bildschirm des PGs der Inhalt des USTACKS. Das Systemprogramm trägt beim Übergang in den Stoppzustand alle Informationen in den USTACK ein, die es für einen Wiederanlauf benötigt.

Aus den USTACK-Informationen können Sie bei der Fehlerdiagnose entnehmen, was für ein Fehler aufgetreten ist und an welcher Stelle im Programm er sich ausgewirkt hat.

Hat **ein einziger** Fehler den Stoppzustand verursacht, so wird nur **eine** Ebene mit USTACK-Informationen ausgegeben. Sind **mehrere** Fehler aufgetreten, so werden **entsprechend viele** Ebenen mit USTACK-Informationen angezeigt (TIEFE 01, TIEFE 02 usw.). In allen Ebenen ist unter STOERUNGSURSACHE ein einziger Fehler angekreuzt.

Bei mehreren Fehlern ist in der Ebene TIEFE 01 der Fehler markiert, der unmittelbar vor dem Übergang in den Stoppzustand festgestellt wurde.

Ein Beispiel für die PG-Ausgabe des USTACK-Inhalts sehen Sie in Bild 5-2.

USTACK							
Tiefe 01							
BEF-REG:	1205	SAZ (neu):	000B3	DB-ADR:	00000	BA-ADR:	00108
BST-STP:	EDEFF	PB-NR.:	9	DB-NR.:		OB-NR.:	1
KACHEL-		REL-SAZ:	00013	DBL-REG:	0000	BS-REG:	00000
NUMMER:	00FD	SAZ (alt):	000B2	UAMK:	09DF3FBF	UALW:	FFFFFFFF
KLAMMERN:	KE1 000	KE2 000	KE3 000	KE4 000	KE5 000	KE6 000	
AKKU1:	0000 31BA	AKKU2:	0000 0005	AKKU3:	0004 0005	AKKU4:	0004 0005
ERGEBNISANZEIGE:	ANZ1	ANZ0	OVFL	OVFLS	ODER	$\overline{\text{ERAB}}$	
	STATUS	VKE					
	X	X					
STOERUNGSURSACHE:	KB	KDB	TLAF	SUF	STUEB	STUEU	
	NAU	QVZ	ADF	PARE	ZYK	STOP	
					X		
	STS	WEFEH	PEU	HALT			

Bild 5-2 Beispiel für eine Bildschirmseite "AUSGABE USTACK": Inhalt

Erläuterung der USTACK-Anzeigen

TIEFE

Stufe der Informationsebene des USTACK-Inhalts bei Fehler-schachtelung:

TIEFE 01 = zuletzt aufgetretene Störungsursache,
 TIEFE 02 = vorletzte aufgetretene Störungsursache

 TIEFE05 = (maximale Tiefe)

Angaben über die Fehlerstelle

Die folgenden Tabelle enthält Angaben (USTACK-Kennungen) über die Fehlerstelle, mit denen im Anwenderprogramm die Anweisung gefunden werden kann, bei deren Bearbeitung die CPU in STOP gegangen ist.

Tabelle 5-5 Bedeutung der USTACK-Kennungen zur Fehlerstelle

Angaben zur Fehlerstelle	
USTACK-Kennung	Bedeutung
BEF-REG	Befehlsregister: Es enthält den Machinencode (1. Wort) des zuletzt bearbeiteten Befehls einer unterbrochenen Programmbearbeitungsebene.
SAZ (neu)	STEP-Adreßzähler (neu): Er enthält die Absolut adresse des nächsten zu bearbeitenden Befehls im Programmspeicher . Mit diesem Befehl setzt die CPU im Falle eines WIEDERANLAUFS das Programm fort.
DB-ADR	absolute Anfangsadresse (DW 0) des zuletzt aufgeschlagenen Datenbausteins im Programmspeicher (= 0000, wenn kein Datenbaustein aufgeschlagen wurde)
BA-ADR	Absolutadresse im Programmspeicher für den nächsten zu bearbeitenden Befehl im zuletzt aufgerufenen Baustein
BST-STP	Bausteinstack-Pointer: Er enthält die 20-bit-Offsetadresse des letzten BSTACK-Eintrags (immer Exxxx).
PB-NR (je nach Typ PB, OB ...)	Baustein-Typ und -Nummer des zuletzt bearbeiteten Bausteins

Angaben zur Fehlerstelle	
USTACK-Kennung	Bedeutung
Fortsetzung 1 der Tabelle 5-5:	
DB-NR	Nummer des zuletzt aufgechlagenen Datenbausteins
OB-NR (je nach Typ OB, PB ...)	Baustein-Typ und -Nummer des zuletzt aufzufenden Bausteins
REL-SAZ	relativer STEP-Adreßzähler: Er enthält die Relativ adresse (bezogen auf die Baustein-Anfangsadresse) der nächsten zu bearbeitenden Operation im zuletzt bearbeiteten Baustein.
DBL-REG	Länge des zuletzt aufgeschlagenen Datenbausteins
BS-REG	Inhalt des Basis-Adreßregisters vor Übergang in den Stoppzustand
KACHEL-NUMMER	Nummer der zuletzt angewählten Kachel (Kachelzugriffe beziehen sich auf diese Kachel — Informationen über Kachelzugriffe finden Sie in Kapitel 9)
SAZ (alt)	STEP-Adreßzähler (alt): Er enthält die Absolut adresse der zuletzt im Programmspeicher bearbeiteten Operation einer unterbrochenen Programmebene; bei einem Fehler zeigt SAZ (alt) genau auf die fehlerverursachende Operation.
UAMK	Unterbrechungsanzeigen-Maskenwort: Im UAMK sind alle bisher aufgetretenen und noch nicht zu Ende bearbeiteten Unterbrechungsursachen angezeigt.
UALW	Unterbrechungsanzeigen-Löschwort
KLAMMERN	Anzahl der Klammerebenen: "KEx abc" mit x = 1 bis 7 Ebenen, a = 'OR' (Oder, siehe Bitanzeigen), b = 'VKE' (Verknüpfungsergebnis, siehe Bitanzeigen), c = 1: 'U', c = 0: 'O'.
AKKU1 bis AKKU4	Inhalt der Rechenregister zum Unterbrechungszeitpunkt

ERGEBNISANZEIGE

siehe Abschnitt 3.5

STOERUNGSURSACHE

Die folgenden Abkürzungen (USTACK-Kennungen) stellen die wichtigsten Störungsursachen dar.

Tabelle 5-6 USTACK-Kennungen STOERUNGSURSACHE

STOERUNGSURSACHE	
USTACK-Kennung	Bedeutung (aufgerufener Fehler-OB)
KB	Ein aufgerufener Baustein ist nicht geladen (OB 19).
KDB	Ein aufgeschlagener Datenbaustein ist nicht geladen (OB 19).
TLAF	Lade- oder Transferfehler (OB 32)
SUF	Substitutionsfehler (OB 27): ein bearbeiteter STEP-5-Befehl ist nicht substituierbar
STUEB	Der Bausteinstack ist übergelaufen: Die Schachtelungstiefe ist zu groß; erforderliche Maßnahme: NEUSTART
STUEU	Der Unterbrechungsstack ist übergelaufen: Die Schachtelungstiefe ist zu groß; erforderliche Maßnahmen: NETZ-AUS, NETZ-EIN, dann NEUSTART
NAU	Netzspannungsausfall im Zentralgerät
QVZ	Quittungsverzug (OB 23/OB 24/OB 28/OB 29)
ADF	Adressierfehler bei digitalen Eingängen und Ausgängen mit Prozeßabbild (OB 25)
PARE	Parityfehler (OB 30)
ZYK	Zyklusüberwachungszeit überschritten (OB 26)
STOP	Stoppzustand durch Betriebsartenschalter in Stellung STOP
STS	Stoppzustand durch STEP-5-Operation 'STS' (nach Befehlsausführung)

STOERUNGSURSACHE	
USTACK-Kennung	Bedeutung (aufgerufener Fehler-OB)
Fortsetzung der Tabelle 5-6:	
WEFEH	Weckfehler/Hardware (OB 33): Weckalarmtakt zu lange maskiert
PEU	Peripherie unklar = Spannungsausfall im Erweiterungsgerät: Nach Wegnahme eines statisch anstehenden PEU-Signals wird immer der Organisationsbaustein OB 22 (AUTOMATISCHER WIEDERANLAUF) aufgerufen.
HALT	Mehrprozessor-Stopp: a) Wahlschalter am KOR in Stellung STOP, b) Stopp einer anderen CPU im Mehrprozessorbetrieb.

5.4.3

Beispiel zur Fehlerdiagnose
über USTACK

Bild 5-3 zeigt Ihnen den Aufbau des USTACK in Zusammenhang mit den aufgetretenen Unterbrechungen.

- Die Programmbearbeitungsebene ZYKLUS (**OB 1**) wird unterbrochen durch das Auftreten eines Interrupts.
- Daraufhin wird die Programmbearbeitungsebene INTERRUPT aktiviert und der **OB 3** bearbeitet.
- Durch das Auftreten eines Weckalarms wird die Ebene INTERRUPT verlassen, die Ebene WECKALARM aktiviert und der **OB 13** bearbeitet.
- Ein falscher Adressierbefehl führt dazu, daß die Ebene ADF aktiviert und dort der **OB 25** bearbeitet wird. In seinem Fehlerbehandlungsprogramm hat der Anwender einen Stoppbefehl (STS) programmiert: Die CPU bricht die Programmbearbeitung ab.

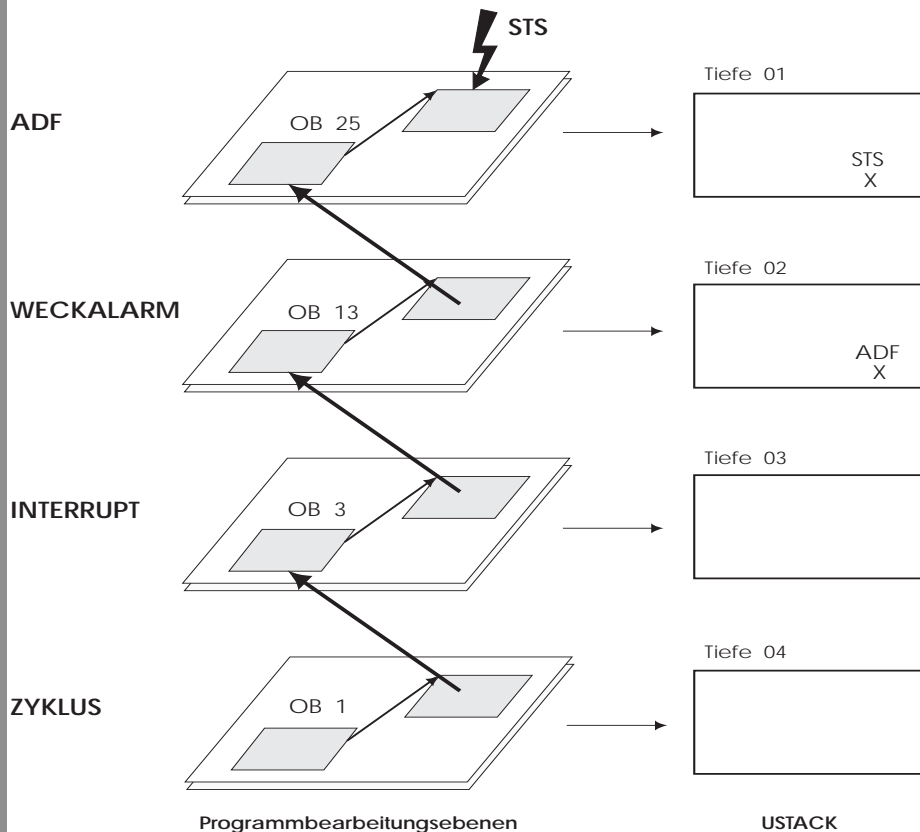


Bild 5-3 Beispiel zur Auswertung des USTACKS

Vor dem endgültigen Übergang in den Stoppzustand sind insgesamt 4 verschiedene Programmbearbeitungsebenen unterbrochen worden. Wenn Sie sich nun am PG den USTACK ausgeben lassen, bekommen Sie entsprechend einen **vierstufigen** USTACK, zuoberst der USTACK mit der Tiefe 01, in dem die Kennung der **zuletzt** unterbrochenen Programmbearbeitungsebene (= ADF) vermerkt ist. Sie können den USTACK nun "hinunterschalten" bis zum USTACK mit der Tiefe 04, der die Programmbearbeitungsebene ZYKLUS repräsentiert, die **als erste** unterbrochen wurde.

5.5 Fehlerbehandlung über Organisationsbausteine

Wenn das Systemprogramm einen bestimmten Fehler erkannt hat, ruft es den für diesen Fall vorgesehenen Organisationsbaustein auf. Durch entsprechende Programmierung dieses Organisationsbausteins können Sie nun das weitere Verhalten der CPU festlegen.

Abhängig davon, wie Sie den Organisationsbaustein programmieren, können Sie

- die normale Programmbearbeitung fortsetzen lassen,
- die CPU in STOP bringen

und/oder

- ein spezielles "Fehlerprogramm" bearbeiten lassen.

Für die folgenden Fehlerursachen sind Organisationsbausteine vorhanden:

Tabelle 5-7 Bei Fehlern aufgerufene Organisationsbausteine

Fehlerursache	Aufruf von	Reaktion bei nicht programm. OB
Aufruf eines nicht geladenen Bausteins (KB)	OB 19	keine
Aufschlagen eines nicht geladenen Datenbausteins DB/DX (KDB)	OB 19	STOP
Quittungsverzug im Anwenderprogramm bei Zugriff auf Peripheriebaugruppen (QVZ)	OB 23	keine
Quittungsverzug beim Aktualisieren des Prozeßabbildes und bei der Koppelmerkerübertragung (QVZ)	OB 24	keine
Adressierfehler (ADF)	OB 25	STOP ¹⁾
Zykluszeitüberschreitung (ZYK)	OB 26	STOP
Substitutionsfehler (SUF)	OB 27	STOP
Quittungsverzug beim Eingangsbyte EB 0 (Prozeßalarme – QVZ)	OB 28	STOP
Quittungsverzug bei dezentraler Peripherie (erweiterter Adreßbereich — QVZ)	OB 29	keine
Parityfehler und Quittungsverzug beim Anwenderspeicher (PARE)	OB 30	STOP

Fehlerursache	Aufruf von	Reaktion bei nicht programm. OB
Fortsetzung der Tabelle 5-7:		
Lade- und Transferfehler (TLAF)	OB 32	STOP
Weckfehler: a) Warteschlangenüberlauf (Steuerbit WEFES) b) Weckalarmtakt zu lange maskiert (WEFEH)	OB 33	STOP keine
Fehler beim STEP-5-Befehl "E DB/EX DX" (Steuerbit FEDBX)	OB 34	STOP
Fehler im Selbsttest (siehe Abschnitt 5.7)	OB 36	keine

1) Dies gilt nur dann, wenn der Adressierfehler nicht durch die STEP-5-Operation "AFS" gesperrt ist.

Beispiele für Reaktion bei nicht geladenem Organisationsbaustein

a) Keine Reaktion; die zyklische Programmbearbeitung wird nicht unterbrochen.
<p>Tritt z. B. ein Quittungsverzug auf und OB 23 oder OB 24 sind nicht geladen, so wird die zyklische Programmbearbeitung gemäß obiger Tabelle nicht unterbrochen. Es erfolgt keine Reaktion der CPU. Soll dagegen die CPU bei QVZ in den Stoppzustand übergehen, so muß der entsprechende Organisationsbaustein (z. B. OB 23 bei QVZ) eine Stopp-Anweisung (STP für STOP am Zyklusende) enthalten und mit 'BE' abgeschlossen werden.</p> <p>Beispiel für OB 23:</p> <pre> : QVZ ist aufgetreten : :STP zyklische Bearbeitung wird abgebrochen :BE CPU geht in den Stoppzustand </pre>
b) Reaktion : Die CPU geht in den Stoppzustand .
<p>Bei Auftreten eines entsprechenden Fehlers – z. B. Zyklus- oder Lade-/Transferfehler – geht die CPU sofort in den Stoppzustand, wenn die zugehörigen Organisationsbausteine nicht geladen sind.</p> <p>Soll in Ausnahmefällen (z. B. während der Inbetriebsetzung) der eine oder andere Fehler die zyklische Programmbearbeitung nicht unterbrechen, so genügt eine Bausteinende-Anweisung im jeweiligen Organisationsbaustein.</p> <p>Beispiel für OB 25:</p> <pre> : ADF ist aufgetreten : :BE zyklische Bearbeitung wird fortgesetzt, kein STOP der CPU </pre>

*Unterbrechungen bei der
Bearbeitung der Fehler-
Organisationsbausteine*

Nachdem das Systemprogramm den betreffenden Organisationsbaustein aufgerufen hat, wird das darin enthaltene Anwenderprogramm bearbeitet.

Tritt während der Bearbeitung eines Organisationsbausteins erneut ein Fehler auf, so wird das Programm wie in der zyklischen Programmbe-
arbeitung an der nächsten Befehls-grenze unterbrochen und der entspre-
chende Organisationsbaustein wird aufgerufen.

Die Organisationsbausteine werden in der Reihenfolge bearbeitet, in
der sie aufgerufen werden.

Hinweis

Es können maximal 5 Fehler-OBs ineinandergeschachtelt werden.
Bei mehr als 5 Fehlern geht die CPU wegen USTACK-Überlauf
in den HARTEN STOP.

5.6 Fehlerursachen und Fehlerreaktionen der CPU

Im Betriebszustand RUN kann eine zyklische, zeit- oder alarmgesteuerte Programmbearbeitung an Befehls Grenzen unterbrochen werden durch das Auftreten bestimmter Unterbrechungsursachen.

Bei der Initialisierung und im Betriebszustand ANLAUF auftretende Unterbrechungsursachen führen ebenfalls dazu, daß das Anlaufprogramm abgebrochen wird. Die CPU geht danach in den Stoppzustand oder ruft den für diesen Fehlerfall vorgesehenen Organisationsbaustein auf. Im Anlaufprogramm auftretende Unterbrechungsursachen werden wie im Zyklus behandelt.

Je nach Unterbrechungsursache ist die Reaktion unterschiedlich:

- sofortiger Übergang der CPU in den Betriebszustand STOP, ohne vorher den Fehler-OB aufzurufen (z. B. NAU → harter Stopp, STUEU → harter Stopp, PEU → weicher Stopp),
- vor Übergang in den Betriebszustand STOP ruft das Systemprogramm einen bestimmten Fehler-OB auf, den Sie programmieren und in dem Sie – je nach Fehlerursache – den Übergang in den Stoppzustand verhindern können (z. B. QVZ/EB 0 → OB 28, ADF → OB 25).

Beachten Sie im Fehlerfall die Einträge der Steuerbits unter FEHLERKENNUNGEN und die Einträge im USTACK unter STOERUNGSURSACHE.

In den folgenden Abschnitten werden mögliche Fehlerursachen näher erläutert.

5.6.1

OB 19: Aufrufen eines nicht geladenen Codebausteins (KB)

Wenn im Anwenderprogramm ein Baustein aufgerufen wird, der nicht vorhanden ist, erkennt das Systemprogramm einen Fehler. Dies gilt für alle Codebausteine und sowohl für unbedingte als auch für bedingte Aufrufe.

Wenn ein nicht vorhandener Codebaustein im Anwenderprogramm aufgerufen wird, so ruft das Systemprogramm stattdessen den Organisationsbaustein **OB 19** auf, wenn dieser geladen ist. In ihm können Sie das weitere Verhalten der CPU festlegen.

Ist der OB 19 nicht vorhanden, so setzt das Systemprogramm die Bearbeitung des unterbrochenen STEP-5-Programms mit der nächsten Operation fort.

5.6.2

OB 19: Aufschlagen eines nicht geladenen Datenbausteins (KDB)

Wird im Anwenderprogramm ein DB- oder DX-Datenbaustein aufgeschlagen, der im Speicher nicht vorhanden ist oder als ungültig geführt wird, so erkennt die CPU einen Fehler und das Systemprogramm ruft den **OB 19** auf, wenn dieser geladen ist. Ist der OB 19 nicht geladen, so geht die CPU in den Stoppzustand. In die Register DBA und DBL wird der Wert '0' eingetragen.

Hinweis

Der OB 19 wird sowohl bei nicht geladenem Code- als auch bei nicht geladenem Datenbaustein aufgerufen.

Durch Auslesen des Systemdatums BS 75 können Sie per STEP-5-Programm feststellen, welche der beiden Fehlerarten aufgetreten ist:

- Inhalt von BS 75 bei KB: 0101H,
- Inhalt von BS 75 bei KDB: 0904H.

5.6.3

**OB 23/24, OB 28/29:
Quittungsverzug (QVZ)**

Ein Quittungsverzug tritt auf, wenn sich bei Schreib- oder Lesezugriffen der adressierbare Speicherbereich nicht innerhalb einer bestimmten und von der Hardware überwachten Zeit mit dem Quittungssignal (Ready) zurückmeldet. Die Ursache für den Quittungsverzug kann z. B. ein Defekt auf der adressierten Baugruppe sein oder das Entfernen der Baugruppe während des Betriebs.

Folgende Quittungsverzugsfehler unterbrechen die Bearbeitung des Anwenderprogramms, verzweigen in die Fehlerbehandlung des Systemprogramms und rufen einen entsprechenden Organisationsbaustein auf, wenn er geladen ist:

OB 23

QVZ bei direkten Peripheriezugriffen:

Fehlerursache	Fehlerreaktion
Quittungsverzug bei Direktzugriff (im Anwenderprogramm) über den S5-Bus auf IP, KOR oder Peripheriebaugruppe (z. B. mit Lade- und Transferbefehlen "L/T P..." bzw. "L/T Q...").	Ist der OB 23 nicht geladen, so setzt das Systemprogramm die Bearbeitung des Anwenderprogramms fort.

OB 24

Fehlerursache	Fehlerreaktion
Quittungsverzug beim Aktualisieren des Prozeßabbildes für Ein- und Ausgänge oder beim Transfer der Koppelmerker.	Ist der OB 24 nicht geladen, so setzt das Systemprogramm die Bearbeitung des Anwenderprogramms fort.

Verlängerung der Ausführungszeit

Sowohl beim Aufruf des OB 23 als auch des OB 24 verlängert ein Quittungsverzug bei Fortsetzung der Programmbearbeitung die Ausführungszeit der ihn verursachenden STEP-5-Operation:
Verlängerung = "Quittungsüberwachungszeit + Zeit der Fehlerbehandlung durch das Systemprogramm + evtl. Bearbeitungszeit des aufgerufenen Fehler-OBs".

OB 28

Fehlerursache	Fehlerreaktion
Quittungsverzug beim Eingangsbyte EB 0 (Prozeßalarme)	Ist der OB 28 nicht geladen, so geht die CPU in den Stoppzustand.

OB 29

Fehlerursache	Fehlerreaktion
Quittungsverzug der dezentralen Peripherie in den Adreßbereichen: - F 0000H bis F EFFFH - F F200H bis F FFFFH	Ist der OB 29 nicht geladen, so setzt das Systemprogramm die Bearbeitung des Anwenderprogramms fort.

Fehleradresse

Bei einem aufgetretenen Quittungsverzug kann die Fehleradresse im Systemdatenbereich ausgelesen werden (siehe Kapitel 8):

BS	Inhalt	Adresse
68	Fehleradresse high	E F044H
69	Fehleradresse low	E F045H

5.6.4

OB 25: Adressierfehler (ADF)

Ein Adressierfehler tritt auf, wenn mit einer STEP-5-Operation ein Ein- oder Ausgang im Prozeßabbild angesprochen wird, dem zum Zeitpunkt des letzten NEUSTARTS keine Peripheriebaugruppe zugeordnet war (Baugruppe war nicht gesteckt, defekt oder nicht im Datenbaustein DB 1 der CPU angegeben).

Die STEP-5-Operation, bei der der Adressierfehler aufgetreten ist, wird vollständig bearbeitet: Bei Bitoperationen wird das Bit im Prozeßabbild abgefragt und verknüpft oder gesetzt bzw. rückgesetzt; ebenso werden Lade- und Transferoperationen durchgeführt. Ein Weiterarbeiten kann allerdings zu falschen oder unerwünschten Reaktionen führen!

Das Systemprogramm unterbricht beim Auftreten eines ADF die weitere Bearbeitung des Anwenderprogramms und ruft den Organisationsbaustein **OB 25** auf. Nach der Bearbeitung des im OB 25 enthaltenen Programms wird mit der nächsten Operation des unterbrochenen Programms fortgefahren.

Wenn der OB 25 nicht geladen ist, geht die CPU beim Auftreten eines Adressierfehlers in den Stoppzustand.

Die Adressierfehler-Überwachung kann für einzelne Programmteile oder das gesamte Programm durch die STEP-5-Operation "AFS" gesperrt und mit "AFF" wieder freigegeben werden (siehe Abschnitt 3.5.4 bzw. Operationsliste).

5.6.5

OB 26: Zykluszeitfehler (ZYK)

Die Zykluszeit ist die Zeitspanne von einem OB-1-Eintritt bis zum nächsten.

Sie umfaßt die gesamte Bearbeitung des zyklischen Programms inklusive Unterbrechungen, Alarmbearbeitungen und Systemprogramm-Leistungen. Eine Überschreitung der in der CPU eingestellten Zyklusüberwachungszeit kann ausgelöst werden z. B. durch fehlerhafte Programmierung (Programmschleife).

Hinweis

Hardwarefehler als Ursache für Zykluszeitfehler sind **äußerst** selten. In der Regel liegen Fehler im Anwenderprogramm vor oder Programme und Zyklusüberwachungszeit sind nicht aufeinander abgestimmt.

Wenn eine Zykluszeitüberschreitung auftritt, unterbricht das Systemprogramm die Bearbeitung des Anwenderprogramms und ruft den Organisationsbaustein **OB 26** auf, wenn dieser geladen ist. Die Überwachungszeit wird dabei neu gestartet (getriggert). Falls die Überwachungszeit erneut abläuft, bevor der OB 26 zu Ende bearbeitet ist, geht die CPU in den Stoppzustand.

Wenn der OB 26 nicht geladen ist, geht die CPU in den Stoppzustand.

Die Zyklusüberwachungszeit ist variabel (10 bis 2550 msec) und nachtriggerbar (siehe oben).

Sie können die Zyklusüberwachungszeit individuell vorgeben durch einen Eintrag im DX 0 (siehe Kapitel 7) oder durch Programmieren des OB 31. Die Voreinstellung der Überwachungszeit beträgt 200 ms.

Im zyklischen Programm kann die Zyklusüberwachungszeit durch einen Aufruf des Sonderfunktions OB 222 "nachgetriggert" werden.

5.6.6

OB 27: (Substitutionsfehler SUF)

Wenn in einem Funktionsbaustein eine Operation mit einem Formaloperanden ausgeführt werden soll, so ersetzt (substituiert) die CPU bei der Bearbeitung des Anwenderprogramms diesen Formaloperanden durch den im Aufruf des Funktionsbausteins stehenden Aktualoperanden.

Erkennt die CPU eine unzulässige Substitution, so unterbricht das Systemprogramm daraufhin die Bearbeitung des Anwenderprogramms und ruft den Organisationsbaustein **OB 27** auf, wenn dieser geladen ist. Ist der OB 27 nicht geladen, so geht die CPU in den Stoppzustand.

Neben der unzulässigen Substitution wird in folgenden weiteren Fällen SUF gemeldet:

- Unzulässiger Operationscode,
- Besonderheit:
Die Datenbausteine DB 0 und DB 1 dürfen vom Anwender nicht aufgeschlagen werden. Eine Operation "A DB 0" oder "A DB 1" wird von der CPU wie ein Substitutionsfehler behandelt. In die Register DBA und DBL wird der Wert '0' eingetragen.

5.6.7

OB 30: Parityfehler und QVZ beim Anwenderspeicher (PARE)

Der Anwenderspeicher ist durch ein Parity-Bit gesichert. Das Systemprogramm prüft bei jedem Zugriff auf den Anwenderspeicher, ob das Parity-Bit korrekt ist. Ist das Parity-Bit falsch gesetzt, so kommt es zu einem Parityfehler.

Das Systemprogramm ruft den Organisationsbaustein **OB 30** auf. Ist dieser nicht geladen, so geht die CPU in den Stoppzustand.

Bei einem Quittungsverzug im Anwenderspeicher erfolgt dieselbe Reaktion.

PARE bei Zugriff auf das Betriebssystem-RAM

Tritt beim Zugriff auf das Betriebssystem-RAM ein Parityfehler auf, so ruft das Systemprogramm **nicht** den OB 30 auf, sondern **geht in den HARTEN STOP**.

Fehleradresse

Bei einem aufgetretenen Parityfehler oder Quittungsverzug kann die Fehleradresse im Systemdatenbereich ausgelesen werden (siehe Kapitel 8):

BS	Inhalt	Adresse
70	Fehleradresse high	E F046H
71	Fehleradresse low	E F047H

5.6.8**OB 32: Lade- und Transferfehler (TLAF)**

Ein Lade- und Transferfehler wird in folgenden Fällen gemeldet:

- Beim Zugriff auf Daten in DB- oder DX-Datenbausteine vergleicht die CPU die Länge des aufgeschlagenen DBs mit dem im Lade-/Transferbefehl stehenden Parameter. Wird durch den angegebenen Parameter die tatsächliche Datenbausteinlänge überschritten, so wird die Lade-/Transferanweisung nicht ausgeführt. Bei Transferfehlern wird auf diese Weise ein irrtümliches Überschreiben von Daten im Speicher verhindert. Bei Ladefehlern bleiben die Akku-Inhalte erhalten.
- Ein Lade-/Transferfehler wird auch festgestellt, wenn ein einzelnes Bit innerhalb eines nicht vorhandenen Datenwortes abgefragt oder verändert werden soll.
- Wurde vor einem Zugriff auf ein Datenwort noch kein Datenbaustein aufgeschlagen (mit "A DBn" bzw. "AX DXn"), so führt dies ebenfalls zu einem Lade-/Transferfehler.
- Ferner kann ein Lade-/Transferfehler durch absolute Speicherzugriffe über das BR-Register hervorgerufen werden oder durch falsche Bereichsgrenzen bei den STEP-5-Operationen "TNW", "TXW" und "TXB".

Beim Erkennen eines Lade-/Transferfehlers ruft das Systemprogramm den Organisationsbaustein **OB 32** auf, wenn dieser geladen ist. Die Operation, die den Lade-/Transferfehler verursacht hat, wird nicht mehr bearbeitet.

Ist der OB 32 nicht geladen, so geht die CPU in den Stoppzustand.

5.6.9

OB 33: Weckfehler (WEFES/WEFEH)

Eine zeitgesteuerte Programmbearbeitung (Weckalarme) erfolgt durch die Organisationsbausteine OB 6, OB 9 und OB 10 bis OB 18.

In der CPU 948 gibt es folgende Weckfehler:

Warteschlangenüberlauf

Ursache:

Warteschlangen-Überlauf bei der Weckalarmbearbeitung:

- Es stehen für eine der drei kürzesten Perioden (OB 10 bis 12) mehr als drei Weckalarme zur Bearbeitung an

oder

- einer der übrigen OBs (OB 13 bis 18) wird erneut aufgerufen, ehe der erste Aufruf vollständig bearbeitet worden ist.

Reaktion:

Das Systemprogramm ruft als Anwenderschnittstelle den **OB 33** auf, wenn dieser geladen ist.

Sie können darin die Reaktion auf diesen Zustand programmieren.

Ist der OB 33 nicht geladen, so geht die CPU in den Stoppzustand.

PG-Anzeige bei "Ausgabe USTACK":

In den Steuerbits ist die Anzeige WEFES angekreuzt.

Maskierung des Weckalarmtaktes

Ursache:

Der Weckalarmtakt ist zu lange maskiert (bei Unterbrechbarkeit an Bausteingrenzen/Prozeßalarme).

Dies bezieht sich auf den Grundtakt der Weckalarme und die Laufzeit eines Bausteins im zyklischen Anwenderprogramm, d. h. wenn ein zyklischer Baustein länger läuft als ein Grundtakt, kommt es zu einem Weckfehler.

Reaktion:

Das Systemprogramm ruft als Anwenderschnittstelle den **OB 33** auf, wenn dieser geladen ist.
Sie können darin die Reaktion auf diesen Zustand programmieren.

Ist der OB 33 nicht geladen, so setzt die CPU die Programmbearbeitung fort.

PG-Anzeige bei "Ausgabe USTACK":

In den Steuerbits ist die Anzeige WEFEH angekreuzt.

Bei Aufruf des OB 33 hinterlegt das Systemprogramm im AKKU-1-L eine Weckfehlerkennung (siehe Abschnitt 4.5.3).

Hinweis

In der Betriebsart "Prozeßalarme über EB 0" **und** "Unterbrechbarkeit an Bausteingrenzen" zeigt der SAZ nicht auf den Baustein, an **dessen Bausteingrenze (BE-Befehl) der Weckfehler** auftrat, sondern auf den Baustein, der den **fehlerverursachenden Baustein aufgerufen** hatte (Rücksprungadresse).

Solange eine Fehlerursache nicht beseitigt ist, d. h. solange der Fehler noch ansteht oder bei der Bearbeitung der entsprechenden STEP-5-Operation in jedem Zyklus immer wieder auftritt, wird auch der dazugehörige Fehler-Organisationsbaustein immer wieder aufgerufen.

Je nach Dauer der Fehlerbehandlung durch das Systemprogramm und der Bearbeitungszeit des Fehler-Organisationsbausteins kann sich die Zykluszeit dadurch beträchtlich verlängern.

5.6.10

OB 34: Fehler bei E DB/EX DX (FEDBX)

Ursachen:

- Bei der Operation E DB/EX DX wurde eine unzulässige Baustein-Nummer angegeben (Nummer eines reservierten Bausteins, Nummer > 255),
- Bei der Operation E DB/EX DX wurde eine unzulässige Bausteinlänge angegeben (> 4091),
- Beim Aufruf der Operation E DB/EX DX ist nicht mehr genügend Platz im Anwenderspeicher.

Reaktion:

Das Systemprogramm ruft den **OB 34** auf, wenn dieser geladen ist. Ist der OB 34 nicht geladen, so geht die CPU in den Stoppzustand.

5.6.11

OB 35: Kommunikationsfehler

Treten auf der zweiten seriellen Schnittstelle bei Rechnerkopplung RK 512, Datenübertragung mit Prozedur 3964/3964R, Datenübertragung mit "offenem Treiber" oder bei Datenübertragung mit SINEC L1 Störungen auf, so ruft das Systemprogramm den Organisationsbaustein **OB 35** auf und übergibt in AKKU 1 zusätzliche Informationen, die die aufgetretenen Störungen näher erläutern.

Reaktion bei nicht geladem OB 35

Haben Sie keinen OB 35 programmiert, so erfolgt **keine** Reaktion des Systemprogramms und die CPU geht **nicht** in den Stoppzustand.

Fehlerinformation in AKKU 1

Alle 100 ms prüft das Systemprogramm, ob Kommunikationsfehler an der zweiten seriellen Schnittstelle aufgetreten sind. Ist dies der Fall, so hinterlegt das Systemprogramm in AKKU 1 Fehlerinformationen.

Es können Fehlernummern zu maximal drei Störungsursachen beim Aufruf des OB 35 übergeben werden. Liegen mehr als drei Störungsursachen vor, so wird dies durch eine spezielle Überlaufkennung angezeigt.

*Aufbau der Fehlerinformation
in AKKU 1*

	31						24 23			18 15		8 7		0
AKKU 1	0	0	0	0	F	U	B	0	Fehlernummer 1	Fehlernummer 2	Fehlernummer 3			

F = '0', wenn kein Fehlereintrag im Fehlerbereich
= '1', wenn Fehler in Fehlerbereich eingetragen

U = '0', wenn kein Fehlerüberlauf (maximal drei Einträge)
= '1', wenn Fehlerüberlauf (mehr als drei Einträge)

B = '0', wenn kein BREAK auf der Schnittstelle
= '1', wenn BREAK auf der Schnittstelle

BREAK

Bei BREAK auf der Schnittstelle wird der OB 35 nur zu Beginn des BREAK-Zustandes aufgerufen.

**Fehlernummer 1 bis
Fehlernummer 3**

Hier werden maximal 3 Fehlernummern zu den auf der Schnittstelle erkannten Störungen eingetragen und zwar in der Reihenfolge, wie sie vom System erkannt werden.

**Bedeutung der
Fehlernummern**

Die Bedeutung der Fehlernummern sowie weitere Informationen zur Behandlung von Schnittstellenfehlern entnehmen Sie bitte dem Handbuch "Kommunikation" /14/.

5.6.12

OB 36: Fehler bei Selbsttest

Der OB 36 wird aufgerufen, wenn eine der Selbsttestroutinen bei ihrem Ablauf einen Fehler feststellt. Lesen Sie dazu bitte die Einzelheiten in Abschnitt 5.7.

5.7 Selbsttest

5.7.1

Übersicht

Die CPU 948 enthält im Systemprogramm integrierte Selbsttestroutinen

Zu- und Abschaltung

Die Funktionen des Selbsttestes können über Bits im Systemdatum BS 137 durch das Anwenderprogramm einzeln zu- oder abgeschaltet werden

Zeitscheibe

Um im RUN die Zyklusbelastung durch den Selbsttest zu reduzieren, wird im Zyklus immer nur ein Teil der Selbsttests ausgeführt (Zeitscheibe). Die dafür zur Verfügung stehende Zeit ist im BS 136 einstellbar (siehe auch Abschnitt 5.7.3).

Was kann getestet werden?

Die Selbsttestroutinen können folgende Tests durchführen:

WAS WIRD GETESTET?	WANN?
der Anwenderspeicher	beim URLÖSCHEN
das BASP-Signal (Befehlsausgabe sperren)	im STOP
die Hardwareuhr	bei NEUSTART
die Zykluszeitüberwachung	im ANLAUF
die Adreßleitungen	zyklisch im RUN
der Code des Systemprogramms (Checksumme)	zyklisch im RUN
der Code der STEP-5-Codebausteine im Anwenderspeicher (Checksumme)	zyklisch im RUN

5.7.2 Beschreibung der Testfunktionen

Test des Anwenderspeichers (beim URLÖSCHEN, ohne Zeitscheibe)

Der Test des Anwenderspeichers wird beim URLÖSCHEN durchgeführt. Er überprüft den Anwenderspeicher sowie die Bytebereiche der Merker und Prozeßabbilder.

Beim Test wird zunächst der gesamte Bereich (einschließlich Bytebereiche) mit einem Testmuster vorbesetzt und anschließend auf Übereinstimmung verglichen. Am Ende wird der Bereich mit Null überschrieben.

Hinweis

Der Test des Anwenderspeichers ist zeitintensiv; er dauert bei

- CPU 948-1 (640 kbyte) ca 5 Sekunden
- CPU 948-2 (1 664 kbyte) ca 22 Sekunden

Test des BASP-Signals (im STOP, ohne Zeitscheibe)

Beim Test wird geprüft, ob ein BASP-Signal von der CPU ausgegeben wird. Diese Testfunktion läuft in der Stoppschleife. Dabei wird zyklisch das BASP-Signal gelesen.

Im Fehlerfall erfolgt ein Eintrag in den Fehlerpuffer. Beim nachfolgenden ANLAUF wird ein evtl. vorhandener OB 36 (Fehler bei Selbsttest) aufgerufen. Ist der OB 36 geladen und darin eine STP-Operation programmiert wurde, wird der ANLAUF abgebrochen. Andernfalls geht die CPU in den zyklischen Betrieb.

Test der Hardwareuhr (im ANLAUF, jedoch nur bei **NEUSTART** der CPU; ohne Zeitscheibe)

Der Test wird vor Aufruf des OB 20 durchgeführt und dauert 1 Sekunde.

Die eingestellte Uhrzeit bleibt erhalten; vorhandene Zeitaufträge (uhrzeitgesteuerte Weckalarne – OB 9) werden dagegen gelöscht.

*Test der
Zykluszeitüberwachung*

(im ANLAUF, ohne Zeitscheibe)

Mit dieser Testfunktion wird in der Anlaufphase die Zykluszeitüberwachung überprüft. Dazu wird die Zyklusüberwachungszeit auf den kleinsten Wert (20 ms) eingestellt und in einer Programmschleife gewartet, bis der Zyklusfehler auftritt.

Test der Adreßleitungen

(zyklisch im RUN, mit Zeitscheibe)

Beim Test werden Unterbrechungen sowie Kurzschlüsse der Adreßleitungen festgestellt, indem zyklisch Testmuster über die Leitungen geschrieben, zurückgelesen und verglichen werden.

*Test des
Systemprogrammcodes*

(zyklisch im RUN, mit Zeitscheibe)

Beim Betriebssystem-Code-Test wird der Inhalt des CPU-Betriebssystems im internen RAM geprüft (Prüfbereich D 0000H bis E 7FFDH).

Die Prüfung erfolgt durch Aufaddieren des Prüfbereichinhalts und anschließendem Vergleich mit der Checksumme im EPROM.

*Test des Baustein-codes der
STEP-5-Code-Bausteine*

(zyklisch im RUN, mit Zeitscheibe)

Es wird für jeden gültigen STEP-5-Code-Baustein die Checksumme überprüft.

Wenn eine Memory Card gesteckt ist, wird die Checksumme für die Code-Bausteine von der CPU 948 nach dem URLÖSCHEN und nach dem Kopieren des Memory-Card-Inhalts in den internen Anwenderspeicher gebildet. Auch nachgeladene Codebausteine werden überprüft.

Hinweis

Beim Code-Test der STEP-5-Bausteine wird grundsätzlich ein Fehler erkannt, wenn ein oder mehrere Code-Bausteine dynamisch verändert wurden!

Das Verändern mit dem PG ist möglich. Dabei wird die Checksumme vom Systemprogramm der CPU 948 gebildet.

5.7.3 Einstellungen

Berechnen und Einstellen der Anzahl Zeitscheiben

Die Bearbeitungszeit der Selbsttestfunktionen ist auf Zeitscheiben aufgeteilt, die einmal pro Zyklus aufgerufen werden. Die Anzahl der Zeitscheiben ist einstellbar. Dadurch können Sie die pro Zyklus benötigte Bearbeitungszeit für die Selbsttestfunktionen erhöhen.

Berechnen der Anzahl

Dazu müssen Sie zunächst abschätzen, welche Zeit Sie im Zyklus für den Selbsttest erübrigen können: Die Länge einer Zeitscheibe beträgt etwa 500 µs, d. h. der Selbsttest benötigt in einem Zyklus 500 µs.

Wenn Sie die verfügbare Zeit abgeschätzt haben, können Sie daraus die Anzahl der Zeitscheiben in je 500 µs berechnen.

Einstellen der Anzahl

Sie können die Anzahl der Zeitscheiben im Systemdatenwort **BS 136** (16-bit-breit) einstellen. Die Vorbesetzung darin ist **1** Zeitscheibe (Minimalwert). Sie können maximal 10 Zeitscheiben einstellen (das entspricht einer Zykluszeitbelastung von 5 ms).

Die Anzahl der Zeitscheiben wird folgendermaßen aus dem Wert des Systemdatenwortes BS 136 abgeleitet:

BS 136 = 0 oder 1: 1 Zeitscheibe
 BS 136 = 2: 2 Zeitscheiben
 BS 136 = 3: 3 Zeitscheiben
 usw.

Ein-/ Ausschalten der Tests

Sie können die einzelnen Tests z. B. in einem Anlaufbaustein aktivieren, indem Sie die entsprechenden Bits in BS 137 auf '1' setzen bzw. zum Deaktivieren auf '0'.

Hinweis

Nach dem URLÖSCHEN einer neu gesteckten CPU sind **alle** Testfunktionen **ausgeschaltet**.

Beim nächsten URLÖSCHEN bleiben nur die Testfunktionen, die **beim URLÖSCHEN** ablaufen sollen, **eingeschaltet**. Alle anderen Testfunktionen werden ausgeschaltet.

Dies bedeutet, daß Sie den Anwenderspeicher einer neu gesteckten CPU nur prüfen können, indem Sie nach URLÖSCHEN die Testfunktion im BS 137 einschalten und anschließend das URLÖSCHEN wiederholen.

Belegung des
Systemdatenwortes BS 137

Testfunktion	Bit-Nr.
Code des Systemprogramms prüfen	2
Code der STEP-5-Codebausteine im Anwenderspeicher prüfen	5
Adreßleitungen prüfen	7
Uhr prüfen	10
BASP-Signal prüfen	11
Zykluszeitüberwachung prüfen	13
Anwenderspeichertest	15

Die in der Tabelle nicht genannten Bit-Nummern sind nicht belegt.

5.7.4 Fehlerbehandlung

Stellt eine Testfunktion einen Fehler fest, so ruft Sie – sofern Sie nicht im URLÖSCHEN abläuft – den Fehler-OB 36 auf (siehe Abschnitt 5.6.11) und übergibt in AKKU 1 den Inhalt von BS 137 mit den Bits der eingeschalteten Testroutinen.

Zusätzlich hinterlegen alle Testroutinen in den Systemdatenwörtern BS 75 bis BS 78 Informationen über Art des Tests und den erkannten Fehler.

Für Testkomponenten, die nur beim URLÖSCHEN ablaufen, wird die Fehlerursache in BS 75 angezeigt. **Die STOP-LED blinkt dann schnell, nachdem die Tests im URLÖSCHEN mit Fehler beendet sind.**

Fehler, die von der Selbsttestkomponente "BASP-Signal" im STOP gefunden wurden, werden ebenfalls ab BS 75 angezeigt. Der nachfolgende ANLAUF führt nur dann in den zyklischen Betrieb, wenn im OB 36 **keine STP-Operation** programmiert ist.

Fehlerinformation*Test des Anwenderspeichers*

Systemdatenwort	Fehlerinformation.
BS 75	Fehler-Nr. 640CH bei Prüfung des Wortspeichers Fehler-Nr. 650CH bei Prüfung des Bytespeichers
BS 76	Prüfmuster, bei dem Fehler aufgetreten ist
BS 77	fehlerhafte Adresse, high
BS 78	fehlerhafte Adresse, low

Test des BASP-Signals

Systemdatenwort	Fehlerinformation.
BS 75	Fehler-Nr. 6700H
BS 76	FFFFH
BS 77	FFFFH
BS 78	FFFFH

Test der Hardwareuhr

Systemdatenwort	Fehlerinformation.
BS 75	Fehler-Nr. 6800H
BS 76	FFFFH
BS 77	FFFFH
BS 78	FFFFH

Test der Zykluszeitüberwachung

Systemdatenwort	Fehlerinformation.
BS 75	Fehler-Nr. 6600H
BS 76	FFFFH
BS 77	FFFFH
BS 78	FFFFH

Test der Adreßleitungen

Systemdatenwort	Fehlerinformation.
BS 75	Fehler-Nr. 630BH
BS 76	FFFFH
BS 77	fehlerhafte Adresse, high
BS 78	fehlerhafte Adresse, low

*Test des
Systemprogrammcodes*

Systemdatenwort	Fehlerinformation.
BS 75	Fehler-Nr. 610BH
BS 76	FFFFH
BS 77	Ist-Checksumme, high
BS 78	Ist-Checksumme, low

*Test des Bausteincode der
STEP-5-Code-Bausteine*

Systemdatenwort	Fehlerinformation.
BS 75	Fehler-Nr. 620AH
BS 76	Bausteintyp/Bausteinnummer (Kennungen aus Bausteinkopf)
BS 77	Soll-Checksumme
BS 78	Ist-Checksumme

6

Integrierte Sonderfunktionen

Inhalt von Kapitel 6

6.1	Einführung	6 - 4
6.2	OB 121: Systemzeit stellen/lesen.....	6 - 8
6.3	OB 122: "Alarmer sperren" ein-/ausschalten	6 - 12
6.4	OB 124: STEP-5-Bausteine löschen	6 - 14
6.5	OB 125: STEP-5-Bausteine erzeugen	6 - 17
6.6	OB 126: Prozeßabbilder definieren, übertragen	6 - 20
6.7	OB 129: Batteriezustand ermitteln.....	6 - 25
6.8	OB 131: AKKU 1, 2, 3 und 4 löschen.....	6 - 26
6.9	OB 132/133: AKKU-Roll-Up/AKKU-Roll-Down	6 - 27
6.10	OB 141: "Zyklische Weckalarmer einzeln sperren" ein-/ausschalten	6 - 29
6.11	OB 142 "Alarmer gemeinsam verzögern" ein-/ausschalten	6 - 32
6.12	OB 143 "Zyklische Weckalarmer einzeln verzögern" ein-/ausschalten.....	6 - 35
6.13	OB 150: Systemzeit stellen/lesen.....	6 - 38
6.14	OB 151: Zeit für uhrzeitgesteuerten Weckalarm stellen/lesen	6 - 43
6.15	OB 153: Zeit für Verzögerungsalarm stellen/lesen	6 - 50
6.16	OB 180: Variabler Datenbaustein-Zugriff	6 - 53
6.17	OB 181: Datenbausteine (DB/DX) testen	6 - 57

6.18	OB 182: Datenbereich kopieren.....	6 - 59
6.19	OB 202 bis 205: Mehrprozessorkommunikation.....	6 - 62
6.20	OB 222: Zyklusüberwachungszeit neu starten	6 - 63
6.21	OB 223: Anlaufarten vergleichen.....	6 - 64
6.22	OB 254/255: Datenbausteine kopieren/duplizieren.....	6 - 65

Integrierte Sonderfunktionen

6

Das nachfolgende Kapitel schildert Ihnen, welche integrierten Sonderfunktionen das Systemprogramm enthält, wo Sie diese anwenden können und wie Sie die Sonderfunktions-OBs aufrufen und parametrieren müssen.

Ferner erfahren Sie, wie Sie Fehler bei der Bearbeitung einer Sonderfunktion erkennen und u. U. per Programm bearbeiten können.

6.1 Einführung

Das Betriebssystem der CPU 948 bietet Ihnen Sonderfunktionen an, die Sie bei Bedarf mit einem bedingten (SPB OB x) oder einem unbedingten (SPA OB x) Bausteinaufruf aufrufen können. Für diese Sonderfunktionen sind die Organisationsbausteine OB 100 bis 255 reserviert.

Diese Funktionen werden als **Integrierte** Sonderfunktionen bezeichnet, da sie ein fester Bestandteil des Systemprogramms sind. Als Anwender können Sie diese Sonderfunktionen zwar aufrufen, jedoch das entsprechende Programm nicht lesen oder ändern.

Die nachfolgende Tabelle gibt Ihnen eine Übersicht der vorhandenen Sonderfunktionen.

Tabelle 6-1 Übersicht der bei der CPU 948 vorhandenen Sonderfunktionen

Baustein	Funktion	siehe Abschnitt/Seite
OB 121	Systemzeit stellen/lesen (kompatibel zu CPU 946/947)	6.2/6 - 8
OB 122	"Alarmer sperren" ein-/auschalten	6.3/6 - 12
OB 124	STEP-5-Bausteine löschen	6.4/6 - 14
OB 125	STEP-5-Bausteine erzeugen	6.5/6 - 17
OB 126	Prozeßabbilder definieren/übertragen	6.6/6 - 20
OB 129	Batteriezustand ermitteln	6.7/6 - 25
OB 131	AKKU 1, 2, 3 und 4 löschen	6.8/6 - 26
OB 132	AKKU-Roll-Up	6.8/6 - 27
OB 133	AKKU-Roll-Down	6.9/6 - 27
OB 141	"Zyklische Weckalarmer einzeln sperren" ein-/auschalten	6.10/6 - 29
OB 142	"Alarmer gemeinsam verzögern" ein-/auschalten	6.11/6 - 32
OB 143	"Zyklische Weckalarmer einzeln verzögern" ein-/auschalten	6.12/6 - 35
OB 150	Systemzeit stellen/lesen	6.13/6 - 38
OB 151	Zeit für uhrzeitgesteuerten Weckalarm stellen/lesen	6.14/6 - 44
OB 153	Zeit für Verzögerungsalarm stellen/lesen	6.15/6 - 50
OB 181	Datenbaustein testen	6.16/6 - 53
OB 182	Datenbereich kopieren	6.17/6 - 55
OB 200, 202 203, 204, 205	Funktionen zur Mehrprozessor-Kommunikation	6.18/6 - 58
OB 222	Zyklusüberwachungszeit neu starten	6.19/6 - 59
OB 223	Anlaufarten im Mehrprozessorbetrieb vergleichen	6.20/6 - 60
OB 254, 255	DB- und DX-Datenbausteine kopieren/duplizieren	6.21/6 - 61

Schnittstellen

Als Schnittstellen zu den Sonderfunktionen stehen Ihnen bei der Programmierung zur Verfügung:

Bausteinanruf

- Aufruf mit bedingtem/unbedingtem Bausteinanruf SPB .. / SPA ..

Parameter

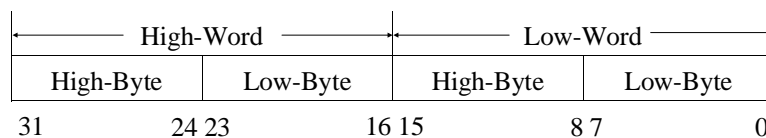
- Parameter zur Voreinstellung über AKKU 1 und evtl. AKKU 2 und/oder Speicherzellen.

Unter dem Begriff **Parameter** sind in der folgenden Beschreibung der einzelnen Sonderfunktionen alle Daten aufgelistet, die die CPU benötigt, um die Sonderfunktion korrekt ausführen zu können. Vor Aufruf der Sonderfunktion im STEP-5-Programm müssen Sie diese Daten in die Akkumulatoren oder in die jeweils angegebenen Speicherzellen laden.

Akku-Schreibweise

Bei den Angaben zur Parametrierung der einzelnen Sonderfunktions-Organisationsbausteine berücksichtigen Sie bitte folgende Schreibweise:

AKKU 1:	AKKU 1,	32 bit
AKKU-1-L:	AKKU 1, Low-Wort,	16 bit
AKKU-1-LL:	AKKU 1, Low-Wort, Low-Byte,	8 bit
AKKU-1-LH:	AKKU 1, Low-Wort, High-Byte,	8 bit



Fehlerbearbeitung

Tritt bei der Bearbeitung der aufgerufenen Sonderfunktion ein Fehler auf, so reagiert das Systemprogramm darauf mit einer speziellen Fehlerreaktion.

Hinsichtlich dieser Fehlerreaktion des Systemprogramms können zwei Gruppen von Sonderfunktionen unterschieden werden.

- **Gruppe 1:**

Akku-Anzeigen

Zur Gruppe 1 zählen alle Sonderfunktionen, bei denen im Fehlerfall im AKKU 1 Kennungen übergeben werden, die den aufgetretenen Fehler näher erläutern.

- **Gruppe 2:**

VKE, ANZ 0/ANZ 1

Bei einigen Sonderfunktionen werden für das Anzeigen von sonderfunktionsspezifischen Fehlern das VKE oder die Anzeigen ANZ 0/ANZ 1 beeinflusst.

Wenn bei der Bearbeitung dieser Sonderfunktionen ein Fehler auftritt, wird in den meisten Fällen das VKE gesetzt (VKE = 1). Sie können in Ihrem STEP-5-Programm bei diesen Sonderfunktionen mit einer SPB-Operation (Springe bedingt) das VKE auswerten und damit auf einen Fehler reagieren.

Bei manchen Sonderfunktionen werden die Ergebnisanzeigen ANZ 0 und ANZ 1 durch die Bearbeitung der Sonderfunktion beeinflusst. Diese Anzeigen können Sie in Ihrem STEP-5-Programm mit Vergleichsoperationen abfragen und somit ebenfalls auf einen Fehler reagieren.

Welche der geschilderten Fehlerreaktionen bei den einzelnen Sonderfunktions-OB ausgeführt werden, wird in den folgenden Unterkapiteln zu den Sonderfunktions-OB beschrieben.

Hinweis

Der Aufruf eines Sonderfunktions-OB mit der Operation "SPA OB 131/132/133" bzw. "SPB OB 131/132/133" wirkt nicht wie ein "echter" Bausteinwechsel, sondern wie eine STEP-5-Operation ohne Baustein-Operand. Es werden **keine Alarme** eingeschachtelt (bei Voreinstellung "Unterbrechung an Bausteingrenzen")!

6.2 OB 121: Systemzeit stellen/lesen

Funktion

Über den OB 121 können Sie kompatibel zur CPU 946/947 die Systemzeit (Datum und Uhrzeit) stellen bzw. lesen.

Parameter

1. Datenfeld

4 Wörter im wortorientierten Speicherbereich.

Bei der Funktion "**Systemzeit stellen**" muß der Bereich **vor** dem Aufruf des OB 121 mit den einzustellenden Zeitwerten geladen werden. Das Systemprogramm überprüft diese Werte auf ihre logische Richtigkeit.

Bei der Funktion "**Systemzeit lesen**" trägt der OB 121 die aktuellen Zeitwerte in diesen Bereich ein.

Aufbau des Datenfeldes

Bit-Nr.	15	12 11	9 8	4 3	0
1. Wort	10er Sek.	1er Sek.	1/10 Sek.	1/100 Sek.	
2. Wort	10er Std.	1er Std.	10er Min.	1er Min.	
3. Wort	10er Tag	1er Tag	Wochentag	0	
4. Wort	10er Jahr	1er Jahr	10er Monat	1er Monat	

Hinweis

Der Aufbau des Datenfeldes entspricht dem Aufbau der Systemdaten BS 96 bis BS 99 (aktuelle Uhrzeit).

Mögliche Zeitwerte:

1/100 Sekunde: 0 bis 9

1/10 Sekunde: 0 bis 9

1er Sekunde: 0 bis 9

10er Sekunde: 0 bis 5

1er Minute: 0 bis 9

10er Minute: 0 bis 5

1er Stunde: 0 bis 9

10er Stunde:

Bit-Nr.

12 /13: 0 bis 1 bei 12-Stunden-Format

0 bis 2 bei 24-Stunden-Format

Bit-Nr. 14: 0 = AM bei 12-Stunden-Format

1 = PM " " "

0 bei 24-Stunden-Format

Bit-Nr. 15: 0 = 12-Stunden-Format

1 = 24-Stunden-Format

Wochentag:	0 bis 6 für Mo bis So
1er Tag:	0 bis 9
10er Tag:	0 bis 3
1er Monat:	0 bis 9
10er Monat:	0 bis 1
1er Jahr:	0 bis 9
10er Jahr:	0 bis 9

2. BR-Register (Basisadrefregister)

Anfangsadresse eines Datenfeldes in einem Bereich des wortweisen organisierten Speichers, aus dem die einzustellenden Zeitwerte gelesen bzw. in den die aktuellen Zeitwerte gespeichert werden sollen. Das BR-Register muß **vor** dem Aufruf des OB 121 mit der Adresse geladen werden.

3. AKKU-1-L

Funktions-Nr.,
zulässige Werte: 1 = Systemzeit stellen
 2 = Systemzeit lesen

Ergebnis

Nach korrekter und fehlerfreier Bearbeitung hinterlegt das Systemprogramm im AKKU-1-L den Wert '0'. Nach der Funktion "Stellen" ist die Systemzeit auf die im Datenfeld hinterlegten Werte eingestellt; nach der Funktion "Lesen" enthält das Datenfeld die aktuellen Zeitwerte.

Fehlerfälle

Es können die in der nachfolgenden Tabelle aufgeführten Fehlerfälle auftreten. Kommt es zu einem Fehler, so hinterlegt das Systemprogramm im AKKU-1-L die in der Tabelle aufgeführten Fehlerkennungen.

Tabelle 6-2 Fehlerkennungen des OB 121 im AKKU-1-L

Kennung	Bedeutung
F001H	unzulässige Funktions-Nr.
F00FH	Mehrfachaufruf des Bausteins
F101H	Jahresangabe unzulässig
F102H	Monatsangabe unzulässig
F103H	Tagesangabe unzulässig
F104H	Wochentagsangabe unzulässig
F105H	Stundenangabe unzulässig
F106H	Minutenangabe unzulässig
F107H	Sekundenangabe unzulässig
F108H	1/100 ...1/10 Sekunde im Datenfeld ungleich 0 Beim Stellen: Einträge bei 1/100...1/10 Sekunde müssen 0 sein
F109H	Stundenformat ungleich OB-151-Einstellung

Beispiele

Programmierbeispiel für "Systemzeit stellen"

Für die Funktion "Systemzeit stellen" wird der FB 13 programmiert. Die neuen Werte werden im Datenbaustein DB 10 (Datenwort DW 0 bis DW 3) übergeben.

STEP-5-Programm:

```

FB13
NAME :UHRWR
      :A DB 10 DB 10 aufschlagen
      :L KH 1500 15 Sekunden (1/10 .. 1/100 Sek. = 0!)
      :T DW 0
      :L KH 9555 24-Stunden-Mode, 15 Uhr 55
      :T DW 1
      :L KH 1010 den 10., Dienstag
      :T DW 2
      :L KH 9308 1993, August
      :T DW 3
      :MBR EEC00 Anfangsadresse der DB-Liste in das
      : BR laden
      :LRW +10 Anfangsadresse des DB 10 im Speicher
      : (Paragrafenadresse) in AKKU 1 laden
      :SLD 4 Absolutadresse des DB 10 (DW 0)
      :MAB Inhalt AKKU 1 in das BR-Register laden
      :L KB 1 Funktions-Nr. '1' in AKKU-1-L laden
      :SPA OB 121 Systemzeit stellen
      :L KB 0
      :><F Fehleranzeigen abfragen
      :SPB =FEHL Sprung zur Fehlerbehandlung
      :BEA
      :
FEHL : Fehlerbehandlung
      :
      :BE
    
```

Der Datenbaustein DB 10 enthält beim Aufrufen des OB 121 folgende Information:

```

0: KH = 1500;
1: KH = 9555;
2: KH = 1010;
3: KH = 9308;
4:
    
```

Der OB 121 überträgt die gewünschten Zeitparameter aus dem DB 10 in den Systemdatenbereich BS 96 bis BS 99.

Programmierbeispiel für "Systemzeit lesen"

Für die Funktion "Systemzeit lesen" wird der FB 14 programmiert. Die aktuellen Werte sollen im Datenbaustein DB 11 (Datenwort DW 0 bis DW 3) abgelegt werden.

STEP-5-Programm:

```

FB14
NAME      :UHRRD
          :MBR EEC00      Anfangsadresse der DB-Liste in das
          :              BR laden
          :LRW +11       Anfangsadresse des DB 11 im Speicher
          :              (Paragrafenadresse) in AKKU 1 laden
          :L   KB  0
          :!=F           überprüfen, ob DB 11 geladen
          :SPB =NIVO     zur Fehlerbehandlung springen,
          :              falls DB-Anfangsadresse = 0
          :TAK
          :SLD 4         Absolutadresse des DB 11 (DW 0)
          :MAB           Inhalt AKKU 1 in das BR-Register laden
          :L   KB  2     Funktions-Nr. '2' in AKKU-1-L laden
          :SPA OB 121    Systemzeit lesen
          :BEA
          :
NIVO      :              Fehlerbehandlung
          :
          :BE

```

Der Datenbaustein DB 11 enthält nach dem Aufrufen des OB 121 z. B. folgende Information:

```

0: KH = 2994;          29 Sek., 940 Millisek.
1: KH = 9555;         24-Std.-Format, 15 Std., 55 Min.
2: KH = 1010;        10 Tage, Wochentag '1'(Dienstag), 0
3: KH = 9308;        93 Jahre, 8 Monate
4:

```

Es ist Dienstag, der 10. August 1993, 15:57 Uhr, 29 Sekunden und 940 Millisekunden (9 Zehntel- und 4 Hundertstel-Sekunden).

6.3 OB 122: "Alarme sperren" ein-/ausschalten

Ein STEP-5-Programm kann an Bausteingrenzen oder Befehls­grenzen von Programmen in einer Bearbeitungsebene mit höherer Priorität unterbrochen werden. Zu diesen Programmbearbeitungsebenen mit höherer Priorität gehören

- WECKALARME
- und
- PROZESSALARME/INTERRUPTS.

Die Laufzeit des unterbrochenen Programms verlängert sich dabei jeweils um die Laufzeit der eingeschachtelten Programme.

Funktion

Mit Hilfe des OB 122 können Sie das Einschachteln von Alarmbearbeitungsbausteinen an einer oder mehreren aufeinanderfolgenden Baustein- oder Befehls­grenzen verhindern.

Der OB 122 beeinflusst das Entgegennehmen von Alarmen:

- "Alarme sperren" einschalten heißt:
es werden ab sofort keine Alarme mehr registriert.
- "Alarme sperren" ausschalten heißt:
es werden ab sofort alle auftretenden Alarme wieder registriert und die zugehörigen Bearbeitungsbausteine an der nächsten Befehls- oder Bausteingrenze (abhängig von der im DX 0 eingestellten Betriebsart) eingeschachtelt und bearbeitet.

Alarme, die bereits registriert worden waren, werden dann nicht mehr bearbeitet, wenn im DX 0 die Betriebsart "Unterbrechbarkeit an Bausteingrenzen" eingestellt ist.

Parameter

AKKU-1-L

Funktions-Nr.,
zulässige Werte: 1 = alle Alarme sperren
 2 = alle Alarme freigeben

Ergebnis

Nach korrekter und fehlerfreier Bearbeitung hinterlegt das Systemprogramm im AKKU-1-L den Wert '0'.

Hinweis

Durch den Aufruf des OB 122 wird das VKE (undefiniert) beeinflusst. Das BR-Register wird nicht verändert.
 Zum Sperren und Freigeben der Prozeßalarmer können Sie statt des OB 122 auch die STEP-5-Operationen AS und AF verwenden.
 Die Sperrung der Alarmer wird am nächsten Systemkontrollpunkt (siehe Kapitel 11) aufgehoben.

Fehlerfälle

Es kann der in der nachfolgenden Tabelle aufgeführte Fehlerfall auftreten. Kommt es zu diesem Fehler, so hinterlegt das Systemprogramm im AKKU-1-L die in der Tabelle aufgeführte Fehlerkennung.

Tabelle 6-3 Fehlerkennungen des OB 122 im AKKU-1-L

Kennung	Bedeutung
F001H	unzulässige Funktions-Nr.

Beispiel

Ausschnitt aus einem STEP-5-Programm, in dem vor einem kritischen Programmteil alle Alarmer über den OB 122 gesperrt und danach wieder freigegeben werden:

```

:L   KB 1      Funktionskennung in AKKU-1-L lad.
:SPA OB 122   alle Alarmer sperren
:           )
:           )
:           )
:           } kritischer Programmteil
:           )
:           )
:L   KB 2      Funktionskennung in AKKU-1-L lad.
:SPA OB 122   alle Alarmer freigegeben
:
:
    
```

6.4 OB 124: STEP-5-Bausteine löschen

Funktion

Mit dem OB 124 können Sie beliebige STEP-5-Bausteine (Code- und Datenbausteine) im Anwenderspeicher löschen. Der gelöschte Baustein wird aus der Adreßliste im DB 0 ausgetragen. Die beim Löschen entstehende Speicherlücke wird beim Laden neuer Bausteine wiederverwendet.

Parameter

1. AKKU-1-LH

Bausteintyp des zu löschenden Bausteins

2. AKKU-1-LL

Bausteinnummer des zu löschenden Bausteins

Zulässige Bausteintypen und -nummern:

AKKU-1-LH (Bausteintyp)	AKKU-1-LL (Bausteinnummer)
1 = PB	0 bis 255
2 = SB	0 bis 255
3 = FB	0 bis 255
4 = FX	0 bis 255
5 = DB	3 bis 255
6 = DX	3 bis 255
7 = OB	1 bis 39

Ergebnis

Nach korrekter und fehlerfreier Bearbeitung setzt das Systemprogramm das VKE auf '0' und löscht die Anzeigen ANZ 1 und ANZ 0.

Hinweis

Während des eigentlichen Löschvorgangs sind die Anwenderalarme gesperrt: Es kommen keine Alarme bzw. Interrupts durch.

Durch den Aufruf des OB 124 werden die Inhalte von AKKU 1 bis AKKU 4 verändert. Das BR-Register bleibt erhalten.

Fehlerfälle und Warnungen

Im Fehlerfall oder bei einer Warnung bricht das Systemprogramm die Bearbeitung des OB 124 ab und setzt die Programmbearbeitung mit der nächsten STEP-5-Operation fort. Ferner setzt es das VKE auf '1' und hinterlegt im AKKU-1-LL eine Kennung (siehe Tabelle 6-5).

Bei Abbruch der Funktion mit einer Warnung kann u. U. durch einen erneuten Aufruf der Sonderfunktion (evtl. mehrmals wiederholen) eine korrekte Bearbeitung des OB 124 erreicht werden.

Im folgenden Fall wird der OB 124 mit einer **Warnung** abgebrochen:

Während der letzten 10 ms ist bereits ein OB 124, OB 125, OB 254 oder OB 255 aufgerufen worden. (Innerhalb 10 ms ist jeweils nur ein einziger Aufruf dieser Sonderfunktionen zugelassen. Dadurch wird verhindert, daß durch mehrfachen Aufruf der o. g. OBs die Schnittstelle zum PG nicht mehr bearbeitet wird.)

Anzeigen

Nach Aufruf des OB 124 können Sie an Hand des Verknüpfungsergebnisses und der Anzeigen ANZ 1 und ANZ 0 feststellen, ob die Sonderfunktion ordnungsgemäß abgelaufen ist oder mit "Fehler" oder "Warnung" abgebrochen wurde. Das Ergebnis kann mit bedingten Sprungbefehlen ausgewertet werden.

Ergebnisanzeigen

Tabelle 6-4 Ergebnisanzeigen des OB 124

VKE	ANZ 1	ANZ 0	Bedeutung	Abfrage
0	0	0	Sonderfunktion wurde korrekt bearbeitet	SPB SPZ
1	1	0	Bearbeitung der Sonderfunktion wurde mit "Warnung" abgebrochen	SPB SPP SPN
1	0	1	Bearbeitung der Sonderfunktion wurde mit "Fehler" abgebrochen	SPB SPM SPN

Kennungen im AKKU-1-LL

Im AKKU-1-LL werden vom Systemprogramm Kennungen über das Bearbeitungsergebnis abgelegt, mit denen Ursachen, die zu einer Warnung oder einem Fehler führten, näher spezifiziert sind.

Bit-Nr.	7	6	5	0
	W	F	Ursache für Fehler/Warnung	

Dabei sind folgende Sammelanzeigen festgelegt:

Bit-Nr. 7 (W) = 1: Warnung
 Bit-Nr. 6 (F) = 1: Fehler

Tabelle 6-5 Ergebniskennungen des OB 124 im AKKU-1-LL

Kennung	Bedeutung
01H	Funktion wurde korrekt bearbeitet
45H	Fehler: Bausteinart nicht zulässig Baustein nicht vorhanden Online-Funkt. SPEICHER KOMPRIMIEREN tätig
47H	
4DH	
8DH	Warnung Konflikt mit einer Online-Funktion (außer "Speicher komprimieren") 10-ms-Wartezeit noch nicht abgelaufen
8EH	

Beispiel

```
:L   KY 6,100   diese Befehlsfolge loescht den
:SPA OB 124    Datenbaustein DX 100 im Arbeits-
:              speicher
```

6.5 OB 125: STEP-5-Bausteine erzeugen

Funktion

Mit dem OB 125 können Sie beliebige STEP-5-Bausteine (Code- und Datenbausteine) im Anwenderspeicher erzeugen. Das Erzeugen von Code-Bausteinen sollte jedoch Spezialisten vorbehalten bleiben.

Der angegebene Baustein wird im internen RAM mit Bausteinkopf und Bausteinrumpf angelegt und in den DB 0 eingetragen. Der Bausteinrumpf enthält zufällige Daten. Aus diesem Grunde muß ein neu erzeugter Baustein zuerst beschrieben werden, bevor aus ihm sinnvolle Daten gelesen werden können.

Parameter

1. AKKU-1-LH

Bausteintyp des zu erzeugenden Bausteins

2. AKKU-1-LL

Bausteinnummer des zu erzeugenden Bausteins

Zulässige Bausteintypen und -nummern:

AKKU-1-LH (Bausteintyp)	AKKU-1-LL (Bausteinnummer)
1 = PB	0 bis 255
2 = SB	0 bis 255
3 = FB	0 bis 255
4 = FX	0 bis 255
5 = DB	3 bis 255
6 = DX	3 bis 255
7 = OB	1 bis 39

3. AKKU-2-L

Anzahl Wörter (gewünschte Bausteinlänge ohne Bausteinkopf)
Die maximal parametrierbare Bausteinlänge beträgt 32762
Datenwörter. Momentan sind mit einem PG ca 2 K Wörter editierbar.

Ergebnis

Nach korrekter und fehlerfreier Bearbeitung setzt das Systemprogramm das VKE auf '0' und löscht die Anzeigen ANZ 1 und ANZ 0.

Hinweis

Während des Erzeugungsvorgangs sind die Anwenderalarme gesperrt: Es kommen keine Alarme bzw. Interrupts durch.

Durch den Aufruf des OB 125 werden die Inhalte von AKKU 1 bis AKKU 4 verändert. Das BR-Register bleibt erhalten.

Fehlerfälle und Warnungen

Im **Fehlerfall** bricht das Systemprogramm die Bearbeitung des OB 125 ab und setzt die Programmbearbeitung mit der nächsten STEP-5-Operation fort. Ferner setzt es das VKE auf '1' und hinterlegt im AKKU-1-LL eine Kennung (siehe Tabelle 6-7).

Bei Abbruch der Funktion mit einer Warnung kann u. U. durch einen erneuten Aufruf der Sonderfunktion (evtl. mehrmals wiederholen) eine korrekte Bearbeitung des OB 125 erreicht werden.

Im folgenden Fall wird der OB 125 mit einer **Warnung** abgebrochen:

Während der letzten 10 ms ist bereits ein OB 124, OB 125, OB 254 oder OB 255 aufgerufen worden. (Innerhalb 10 ms ist jeweils nur ein einziger Aufruf dieser Sonderfunktionen zugelassen. Dadurch wird verhindert, daß durch mehrfachen Aufruf der o. g. OBs die Schnittstelle zum PG nicht mehr bearbeitet wird.)

Anzeigen

Nach Aufruf des OB 125 können Sie an Hand des Verknüpfungsergebnisses und der Anzeigen ANZ 1 und ANZ 0 feststellen, ob die Sonderfunktion ordnungsgemäß abgelaufen ist oder mit "Fehler" oder "Warnung" abgebrochen wurde. Das Ergebnis kann mit bedingten Sprungbefehlen ausgewertet werden.

Ergebnisanzeigen

Tabelle 6-6 Ergebnisanzeigen des OB 125

VKE	ANZ 1	ANZ 0	Bedeutung	Abfrage
0	0	0	Sonderfunktion wurde korrekt bearbeitet	SPB SPZ
1	1	0	Bearbeitung der Sonderfunktion wurde mit "Warnung" abgebrochen	SPB SPP SPN
1	0	1	Bearbeitung der Sonderfunktion wurde mit "Fehler" abgebrochen	SPB SPM SPN

Kennungen im AKKU-1LL

Im AKKU-1-LL werden vom Systemprogramm Kennungen über das Bearbeitungsergebnis abgelegt, mit denen Ursachen, die zu einer Warnung oder einen Fehler führten, näher spezifiziert sind.

Bit-Nr.	7	6	5	0
	W	F	Ursache für Fehler/Warnung	

Dabei sind folgende Sammelanzeigen festgelegt:

Bit-Nr. 7 (W) = 1: Warnung
 Bit-Nr. 6 (F) = 1: Fehler

Tabelle 6-7 Ergebniskennungen des OB 125 im AKKU-1-LL

Kennung	Bedeutung
01H	Funktion wurde korrekt bearbeitet
42H	Fehler: Baustein bereits vorhanden
43H	Speicherplatz reicht nicht aus
44H	Bausteinlänge ist nicht zulässig
45H	Bausteinart nicht zulässig
4DH	Online-Funkt. SPEICHER KOMPRIMIEREN tätig
8DH	Warnungen: Konflikt mit einer Online-Funktion (außer "Speicher komprimieren")
8EH	10-ms-Wartezeit noch nicht abgelaufen

Beispiel

```
:L   KF +2000   diese Befehlsfolge erzeugt den
:L   KY 5,24   DB 24 mit einer Laenge von 2000
:SPA OB 125   Datenwoertern
:
:              (Gesamtlaenge inkl. Kopf:
:              2005 Woerter)
```

6.6 OB 126: Prozeßabbilder definieren, übertragen

In jedem Zyklusdurchlauf wird vom Systemprogramm das Prozeßabbild der digitalen Ein- und Ausgänge bzw. der Koppelmerker aktualisiert. Welche Ein-, Ausgänge und Koppelmerker im Prozeßabbild geführt werden, ist im System-Datenbaustein DB 1 hinterlegt (siehe Kapitel 10).

Der OB 126 ermöglicht es Ihnen, zusätzliche Prozeßabbilder zu benutzen.

Funktion

Mit dem OB 126 können Sie im **NEUSTART** zusätzlich zu dem im DB 1 definierten Prozeßabbild bis zu vier weitere Prozeßabbilder vereinbaren.

Diese zusätzlichen Prozeßabbilder lassen sich mit dem STEP-5-Programm in **jeder beliebigen Programmbearbeitungsebene** einlesen und ausgeben.

Parameter

1. Datenfeld

6 Merkerbytes mit folgendem Aufbau:

Bit-Nr.	7	0
MB n	Funktions-Nr.	
MB n+1	Adreßlisten-Nr.	
MB n+2	Bausteintyp	
MB n+3	Bausteinnummer	
MB n+4	Datenwort-Nr. des ersten Kennwortes	
MB n+5	in der Adreßliste	

Parameter des Datenfeldes:

Funktions-Nr.

Mit der Funktions-Nr. legen Sie fest, welchen Auftrag der OB 126 ausführen soll (siehe Tabelle).

zulässige Werte: 1 bis 5

Funktions-Nr.	Funktion
1	Prozeßabbild der digitalen Eingänge einlesen
2	Prozeßabbild der digitalen Ausgänge ausgeben
3	Prozeßabbild der Eingangskoppelmerker einlesen
4	Prozeßabbild der Ausgangskoppelmerker ausgeben
5	systeminterne Adreßliste (analog zu DB 1) aufbauen (nur im NEUSTART – OB 20 zulässig!)

<i>Adreßlisten-Nr.</i>	Nummer der Adreßliste für das zusätzlich definierte Prozeßabbild zulässige Werte: 1 bis 4
<i>Bausteintyp</i>	Typ des Datenbausteins, der die Adreßliste enthält zulässige Werte: 1 = DB 2 = DX
<i>Bausteinnummer</i>	Nummer des Datenbausteins, der die Adreßliste enthält zulässige Werte: 3 bis 255
<i>DW-Nr. 1. Kennwort</i>	Tragen Sie hier die Nummer des Datenwortes ein, in dem das erste Kennwort der Adreßliste steht (siehe Aufbau des DB 1, Abschnitt 10.1.6). Die möglichen Kennwörter sind: KH = DE00 (digitale Eingänge) KH = DA00 (digitale Ausgänge) KH = CE00 (Eingangskoppelmerker) KH = CA00 (Ausgangskoppelmerker) Der Parameter belegt 2 Merkerbytes!

Hinweis

Das vollständige Datenfeld muß nur dann angelegt werden, wenn der OB 126 die Adreßliste im NEUSTART generieren soll (= Funktion 5). Bei Ausführung der Funktionen 1 bis 4 genügt es, im Datenfeld neben der Funktions-Nr. die Adreßlisten-Nr. einzutragen. Die übrigen Einträge entfallen.

Den Datenbaustein, mit dem Sie die Adreßliste für ein zusätzliches Prozeßabbild einrichten wollen (= Funktion 5), müssen Sie analog zum DB 1 aufbauen.

In diesem Datenbaustein können Sie für jedes der zusätzlichen Prozeßabbilder die Adreßlisten-Information hinterlegen, wenn Sie jeden Informationsblock mit der Ende-Kennung wie beim DB 1 versehen.

Zum Einrichten der Adreßliste müssen Sie jedoch den OB126 für jedes zusätzliche Prozeßabbild mit der Funktion '5' einzeln aufrufen (nur im NEUSTART!).

2. AKKU-1-L

Nr. des Merkerbytes **MB n**, bei dem das Datenfeld beginnt

zulässige Werte: 0 bis 250

Ergebnis

Nach korrekter und fehlerfreier Bearbeitung setzt das Systemprogramm das VKE auf '0' und trägt in den AKKU-1-LL eine '1' ein.

Hinweis

Während der Bearbeitung des OB 126 sind die Anwenderalarme gesperrt: Es kommen keine Alarmer bzw. Interrupts durch.

Durch den Aufruf des OB 126 werden die Inhalte von AKKU 1 bis AKKU 4 verändert. Das BR-Register bleibt erhalten.

Bei WIEDERANLAUF wird der Restzyklus unter BASP bearbeitet. Alle digitalen Ausgänge sind gesperrt. Am Zyklusende werden **alle** Ausgänge (auch die in den Adreßlisten 1 bis 4 angegebenen) zurückgesetzt.

Fehlerfälle

Wenn die Sonderfunktion nicht ausgeführt werden kann, bricht das Systemprogramm die Bearbeitung des OB 126 ab und setzt die Programmbearbeitung mit der nächsten STEP-5-Operation fort. Ferner setzt es das VKE auf '1' und hinterlegt im AKKU-1-LL eine Kennung (siehe nachfolgende Tabelle).

Sonderfall bei der Fehlerbearbeitung:

Soll der OB 126 die Funktion '5' (systeminterne Adreßliste aufbauen) ausführen, so überprüft das Systemprogramm den korrekten Aufbau der Adreßliste. Es überprüft ferner, ob die darin enthaltenen Ein- und Ausgänge bzw. Koppelmerker auf entsprechenden Baugruppen quittieren.

Ist eine fehlerhafte Adreßliste übergeben worden, so verhält sich die CPU wie bei einem DB-1-Fehler: Sie geht in den weichen Stoppzustand mit langsamen Blinken der STOP-LED. Als Fehlerursache wird ein DB-1-Fehler gemeldet.

Kennungen im AKKU-1-LL

Tabelle 6-8 Ergebniskennungen des OB 125 im AKKU-1-LL

Kennung	Bedeutung
01H	Funktion wurde korrekt bearbeitet
02H	Funktions-Nr. ist unzulässig
03H	Zeiger in AKKU-1-L (Merker-Nr.) ist unzulässig
04H	Bausteintyp/-nummer ist unzulässig bzw. Baustein DB/DX ist nicht vorhanden
05H	im angegebenen Datenwort des Datenbausteins steht nicht das 1. Kennwort (falsche DW-Nr.) oder die Adreßliste enthält ein falsches Kennwort
06H	Adreßlisten-Nr. ist unzulässig
07H	Aufruf der Funktion ist in der aktuellen Programmbearbeitungsebene unzulässig

Beispiele**Adreßliste im DB 5 erstellen:**

Über die Softkeys <Eingabe>, <Maske>, "Baustein: DB 5" programmieren Sie am PG einen Datenbaustein DB 5 mit folgenden Parametern:

```
Digitale Eingänge:      1, 2,
Digitale Ausgänge:     3,
Koppelmerker-Eingänge: 5, 6, 7,
Koppelmerker-Ausgänge: 20, 22,
```

Wenn Sie den DB 5 manuell erstellen, muß er gleich aufgebaut sein wie ein DB 1 (mit Anfangskennung, Kennwörtern der Operandenbereiche, Endekennung; siehe Abschnitt 10.1.6).

Adreßliste im NEUSTART/OB 20 übernehmen lassen:

Zuerst müssen Sie das Datenfeld im Merkerbereich anlegen. Es belegt die Merkerbytes MB 20 bis MB 25:

```
:L   KB 5   Funktions-Nr. '5'
:T   MB 20  ins MB 20 transferieren
:L   KB 1   Adreßlisten-Nr. '1 '
:T   MB 21  ins MB 21 transferieren
:L   KH 0105 Bausteintyp DB ('1') und -nummer '5'
:T   MW 22  in MB 22 und 23 transferieren
:L   KB 3   Datenwort-Nr. '3' (DW 3 im DB 5 ent-
:T   MW 24  hält 1. Kennwort) nach MB 24 und 25
```

Nachdem das Datenfeld korrekt angelegt ist, muß im AKKU-1-L die Nummer des 1. Merkerbytes im Datenfeld übergeben werden. Anschließend wird der OB 126 aufgerufen, der die Adreßliste aufbaut:

```
:L   KB 20  Datenfeld beginnt mit MB 20
:SPA OB 126 Aufruf für Adreßlistengenerierung
:
:          evtl. Anzeigen auswerten
:          ...
```

Anmerkung:

Die Adreßlisten mit der Nummer 1 bis 4 werden nur über einen Aufruf des OB 126 im OB 20 (NEUSTART) von der CPU übernommen: Dazu muß im OB 20 der OB 126 mit der Funktions-Nr. '5' aufgerufen werden.

Prozeßabbild der Ausgänge ausgeben

Die folgende STEP-5-Programmsequenz kann in einer beliebigen Programmbearbeitungsebenen stehen (im OB 1, in einem Weckalarm-OB oder in einem Prozeßalarm-OB usw.) und hat zur Folge, daß das Prozeßabbild aller Ausgänge in der Adreßliste 1 ausgegeben wird.

```
:L   KB 2   Funktions-Nr. '2'  
:T   MB 50  ins MB 50 transferieren  
:L   KB 1   Adreßlisten-Nr. '1'  
:T   MB 51  ins MB 51 transferieren  
:L   KB 50  Datenfeld beginnt mit MB 50  
:SPA OB 126 Aufruf zur Ausgabe des PAA  
:           evtl. Anzeigen auswerten  
:
```

6.7 OB 129: Batteriezustand ermitteln

Funktion

Mit dem OB 129 können Sie per STEP-5-Programm prüfen, ob die Pufferbatterie noch in gutem Zustand oder ausgefallen ist (der OB 129 fragt dabei das BAU-Signal ab). Abhängig vom Prüfungsergebnis können Sie dann beispielsweise eine Störungsanzeige (Lampe) setzen.

Bildung des BAU-Signals

Die Stromversorgung enthält 2 Pufferbatterien, eine Lithiumzelle (MB für "Main-Batterie") sowie einen Akku (RB für "Reserve-Batterie") als Reservebatterie. Das BAU-Signal wird gebildet durch UND-Verknüpfung der beiden Batterieüberwachungssignale.

Brückeneinstellung in der Stromversorgung

In der Stromversorgung des AGs befinden sich 2 Steckbrücken, über die die Überwachung der Batterien beeinflusst werden kann. Die Brücke MB-NB bestimmt den Zeitpunkt der Bildung des BAU-Signals: Bei nicht gesteckter Brücke MB-NB wird BAU einmalig nach NETZ EIN gebildet. Andernfalls erfolgt die Überwachung zyklisch im Betrieb. Das Überwachungssignal des Akkus ist über die Brücke MA-NA abschaltbar. Aus den möglichen Brückeneinstellungen ergeben sich die in der folgenden Tabelle aufgeführten Möglichkeiten der Batterieüberwachung durch den OB 129:

Brücke MB-NB	Brücke MA-NA	Zeitpunkt der BAU-Signal-Bildung	BAU-Signal wird gebildet aus dem Überwachungssignal der ...
offen	offen	nach NETZ EIN	Lithiumzelle
offen	geschlossen	nach NETZ EIN	Lithiumzelle und des Akkus
geschlossen	offen	zyklisch	Lithiumzelle
geschlossen	geschlossen	zyklisch	Lithiumzelle und des Akkus

Parameter

keine

Ergebnis

VKE = '0': Der Batteriezustand ist in Ordnung

VKE = '1': Die Batterie ist ausgefallen

Beispiel

Mit folgender Operationsfolge können Sie testen, ob die Pufferbatterie noch in Ordnung ist und im Fehlerfall eine Warnlampe einschalten:

```
      :  
      :SPA OB 129  
      :SPB =BATL      VKE = 1 -> Batterie ist  
      :                ausgefallen  
      :  
      :  
      :BEA  
BATL :SU  A  22.5    Warnlampe an Ausgabebyte 22,  
      :                Bit 5 einschalten  
      :BE
```

6.8 OB 131: AKKU 1, 2, 3 und 4 löschen

Funktion

Durch den einmaligen Aufruf des Sonderfunktions-Organisationsbausteins OB 131 können Sie die Inhalte der AKKUs 1 bis 4 auf einfache Weise löschen: Der OB 131 überschreibt alle vier Register mit '0'.

Parameter

keine

Ergebnis

Die AKKUs 1 bis 4 (je 32 bit) sind gelöscht ('0').

Fehlerfälle

keine

6.9 OB 132/133: AKKU-Roll-Up/AKKU-Roll-Down

Funktion

OB 132 und OB 133 bewirken ein "Rollen" der AKKU-Inhalte in aufsteigender bzw. absteigender Richtung:

- Der OB 132 (Roll Up) verschiebt den Inhalt von AKKU 4 in den AKKU 1, den Inhalt von AKKU 1 in den AKKU 2, den Inhalt von AKKU 2 in den AKKU 3 usw...
- Der OB 133 (Roll Down) verschiebt die AKKU-Inhalte in entgegengesetzte Richtung: Inhalt von AKKU 1 in den AKKU 4, AKKU 4 in AKKU 3 usw.

Parameter

keine

Ergebnis

Die Bilder 6-1 und 6-2 zeigen die AKKU-Inhalte **vor** und **nach** dem Aufruf von OB 132 und 133.

Hinweis

Mit den STEP-5-Befehlen ENT (Ergänzender Operationsvorrat) und TAK (Systemoperation) lassen sich AKKU-Inhalte ebenfalls verschieben (siehe Abschnitt 3.4.3).

Fehlerfälle

keine

6.10 OB 141: "Zyklische Weckalarme einzeln sperren" ein-/ausschalten

Mit Hilfe des OB 141 können Sie das Einschachteln von bestimmten zyklischen Weckalarm-OB (Weckalarme mit **festem Zeitraster**) an einer oder an mehreren aufeinanderfolgenden Baustein- oder Befehlsgrenzen verhindern. Beispielsweise können Sie für einen bestimmten Programmteil festlegen, daß er nicht unterbrochen werden kann durch einen OB 10 (Periode 1) und einen OB 11 (Periode 2). Dagegen werden alle übrigen programmierten Weckalarme wie üblich bearbeitet

Funktion

Der OB 141 hat Auswirkungen auf die Reaktion auf zyklische Weckalarme:

"Zyklische Weckalarme einzeln sperren" **einschalten** heißt, es werden ab sofort keine der angegebenen zyklischen Weckalarme mehr registriert und diejenigen Alarme, die bereits registriert worden sind (die z.B. auf eine Bausteingrenze "warten"), werden gelöscht. Nur falls ein Weckalarm-OB (zur Bearbeitung eines Weckalarms mit festem Zeitraster) bereits begonnen wurde, wird dieser vollständig bearbeitet.

"Zyklische Weckalarme einzeln sperren" **ausschalten** heißt, es werden ab sofort alle auftretenden zyklischen Weckalarme wieder registriert und an der nächsten Baustein- oder Befehlsgrenze (je nach Einstellung im DX 0) bearbeitet.

Parameter

1. Steuerwort

Der OB 141 vermerkt die zu sperrenden Weckalarme in einem systeminternen Steuerwort:



Die Bits des Steuerwortes haben folgende Bedeutung:

Bit-Nr.	Alarm
0 bis 2	reserviert, diese Bits müssen '0' sein!
3 = '1'	zyklische Weckalarme mit festem Zeitraster: Periode 1 (OB 10) Periode 2 (OB 11) Periode 3 (OB 12) Periode 4 (OB 13) Periode 5 (OB 14) Periode 6 (OB 15) Periode 7 (OB 16)
4 = '1'	
5 = '1'	
6 = '1'	
7 = '1'	
8 = '1'	
9 = '1'	

Bit-Nr.	Alarm
(Fortsetzg.) 10 = '1' 11 = '1'	Periode 8 (OB 17) Periode 9 (OB 18)
12 bis 15	reserviert; diese Bits müssen '0' sein!

Solange ein Bit auf '1' gesetzt ist, ist der betreffende Alarm gesperrt.

2. Akkus

2a) AKKU-2-L

Funktions-Nr.,
zulässige Werte: 1, 2 oder 3 mit:

- 1: Der Inhalt von AKKU 1 wird in das Steuerwort geladen.
- 2: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '1' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.
- 3: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '0' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.

2b) AKKU 1

neues Steuerwort oder Maske, abhängig von der gewünschten Funktion

Ergebnis

Nach korrekter und fehlerfreier Bearbeitung setzt das Systemprogramm das VKE auf '0'.
Der Aufruf des OB 141 bringt folgende Ergebnisse:

Funkt.-Nr. in AKKU-2-L	Inhalt von AKKU 1	
	vorher	nachher
1	Steuerwort	Steuerwort
2	Maske	neues Steuerwort
3	Maske	neues Steuerwort

Fehlerfälle

Im Fehlerfall setzt das Systemprogramm das VKE auf '1'.
 Es können die in der nachfolgenden Tabelle aufgeführten Fehlerfälle auftreten. Kommt es zu einem Fehler, so hinterlegt das Systemprogramm im AKKU-1-L die in der Tabelle aufgeführten Fehlerkennungen.

Tabelle 6-9 Fehlerkennungen des OB 141 im AKKU-1-L

Kennung	Bedeutung
8D01H 8D02H	unzulässige Funktions-Nr.in AKKU-2-L ¹⁾ eines der reservierten Bits in AKKU 1 ist '1' ¹⁾

¹⁾ im AKKU-2-L befindet sich der fehlerhafte Wert

Steuerwort abfragen

- Der Zustand des Steuerwortes läßt sich durch folgende Programmsequenz abfragen:
 1. Funktions-Nr. '2' oder '3' in den AKKU-2-L laden,
 2. Wert '0' in den AKKU 1 laden,
 3. OB 141 aufrufen,
 4. AKKU 1 auslesen.

6.11 OB 142 "Alarmer gemeinsam verzögern" ein-/ausschalten

Ein STEP-5-Programm kann an Baustein- oder Befehls Grenzen von Programmen höherer Priorität unterbrochen werden. Zu diesen höherprioritären Programmbearbeitungsebenen gehören die Prozeß- und alle Weckalarmer. Die Laufzeit des unterbrochenen Programms verlängert sich dabei jeweils um die Laufzeit der eingeschachtelten Programme. Mit Hilfe des OB 142 können Sie das Einschachteln von höherprioritären Programmbearbeitungsebenen an einer oder an mehreren aufeinanderfolgenden Baustein- oder Befehls Grenzen (je nach Einstellung im DX 0) verhindern.

Funktion

Der OB 142 hat Auswirkungen auf das Bearbeiten von Alarmen:

"Alarmer verzögern" **einschalten** heißt, es werden weiterhin alle auftretenden Alarme registriert und bereits anstehende Alarme bleiben registriert. Eine Bearbeitung der registrierten Alarme findet jedoch zunächst nicht statt. Vorübergehend werden alle Befehls- bzw. Bausteingrenzen für die Bearbeitung von Alarmen unwirksam gemacht. Nur falls ein OB für die Prozeßalarm-Bearbeitung oder ein OB für die Weckalarmbearbeitung bereits begonnen wurde, wird dieser vollständig bearbeitet.

"Alarmer verzögern" **ausschalten** heißt, es werden alle registrierten Alarme an der nächsten Baustein- oder Befehls Grenze bearbeitet.

Hinweis

Die Zeitphase, in der die Alarme verzögert werden, muß kürzer sein als **der dreifache Wert der kürzesten Weckalarmperiode**. Ist dies nicht der Fall, so kommt es zu einem Weckfehler.

Parameter

1. Steuerwort

Der OB 142 vermerkt die zu verzögernden Alarme in einem systeminternen Steuerwort:



Die Bits des Steuerwortes haben folgende Bedeutung:

Bit-Nr.	Alarmart
0 = '1'	zyklische Weckalarmer, festes Zeitraster
1 = '1'	uhrzeitgesteuerter Weckalarm
2 = '1'	Prozeßalarmer
3 = '1'	Verzögerungsalarm
4 bis 15	reserviert; diese Bits müssen '0' sein!

Solange ein Bit auf '1' gesetzt ist, ist der betreffende Alarm gesperrt.

2. Akkus

2a) AKKU-2-L

Funktions-Nr.,
zulässige Werte:

1, 2 oder 3 mit:

- 1: Der Inhalt von AKKU 1 wird in das Steuerwort geladen.
- 2: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '1' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.
- 3: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '0' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.

2b) AKKU 1

neues Steuerwort oder Maske, abhängig von der gewünschten Funktion

Ergebnis

Nach korrekter und fehlerfreier Bearbeitung setzt das Systemprogramm das VKE auf '0'.
Der Aufruf des OB 142 bringt folgende Ergebnisse:

Funkt.-Nr. in AKKU-2-L	Inhalt von AKKU 1	
	vorher	nachher
1	Steuerwort	Steuerwort
2	Maske	neues Steuerwort
3	Maske	neues Steuerwort

Fehlerfälle

Im Fehlerfall setzt das Systemprogramm das VKE auf '1'.
Es können die in der nachfolgenden Tabelle aufgeführten Fehlerfälle auftreten. Kommt es zu einem Fehler, so hinterlegt das Systemprogramm im AKKU-1-L die in der Tabelle aufgeführten Fehlerkennungen.

Tabelle 6-10 Fehlerkennungen des OB 142 im AKKU-1-L

Kennung	Bedeutung
8E01H 8E02H	unzulässige Funktions-Nr.in AKKU-2-L ¹⁾ eines der reservierten Bits (Nr. 4 bis 15) in AKKU 1 ist '1' ¹⁾
8EFFH	falsche Betriebsart (z. B. wenn der Verzögerungs- alarm gesperrt werden soll und der DX 0 den Parameter "Prozeßalarmer über EB 0 = ein" enthält)

¹⁾ im AKKU-2-L befindet sich der fehlerhafte Wert

Steuerwort abfragen

Der Zustand des Steuerwortes läßt sich durch folgende Programmsequenz abfragen:

1. Funktions-Nr. '2' oder '3' in den AKKU-2-L laden,
2. Wert '0' in den AKKU 1 laden,
3. OB 142 aufrufen,
4. AKKU 1 auslesen.

6.12 OB 143 "Zyklische Weckalarme einzeln verzögern" ein-/ausschalten

Mit Hilfe des OB 143 können Sie das Einschachteln von bestimmten zyklischen Weckalarm-OB (Weckalarme mit **festem Zeitraster**) an einer oder an mehreren aufeinanderfolgenden Baustein- oder Befehls-grenzen verhindern. Beispielsweise können Sie für einen bestimmten Programmteil festlegen, daß er nicht unterbrochen werden kann durch einen OB 10 (Periode 1) und einen OB 11 (Periode 2). Dagegen werden alle übrigen programmierten Weckalarme wie üblich bearbeitet

Funktion

Der OB 143 hat Auswirkungen auf das Bearbeiten von zyklischen Weckalarmen:

"Zyklische Weckalarme einzeln verzögern" **einschalten** heißt, es werden weiterhin alle auftretenden Alarmer registriert und bereits anstehende Weckalarmer bleiben registriert. Eine Bearbeitung der im Steuerwort angegebenen Weckalarmer findet jedoch zunächst nicht statt. Vorübergehend werden alle Befehls- bzw. Bausteingrenzen für die Bearbeitung dieser Weckalarmer unwirksam gemacht. Nur falls ein Weckalarm-OB (zur Bearbeitung eines Weckalarms mit festem Zeitraster) bereits begonnen wurde, wird er vollständig bearbeitet.

"Zyklische Weckalarme einzeln verzögern" **ausschalten** heißt, es werden alle registrierten Alarmer an der nächsten Baustein- oder Befehls-grenze (je nach Einstellung im DX 0) bearbeitet.

Parameter

1. Steuerwort

Der OB 143 vermerkt die zu verzögernden Weckalarmer in einem systeminternen Steuerwort:



Die Bits des Steuerwortes haben folgende Bedeutung:

Bit-Nr.	Alarm
0 bis 2	reserviert, diese Bits müssen '0' sein!
3 = '1'	zyklische Weckalarmer mit festem Zeitraster: Periode 1 (OB 10) Periode 2 (OB 11) Periode 3 (OB 12) Periode 4 (OB 13) Periode 5 (OB 14) Periode 6 (OB 15) Periode 7 (OB 16)
4 = '1'	
5 = '1'	
6 = '1'	
7 = '1'	
8 = '1'	
9 = '1'	

Bit-Nr.	Alarm
(Fortsetzg.) 10 = '1' 11 = '1'	Periode 8 (OB 17) Periode 9 (OB 18)
12 bis 15	reserviert; diese Bits müssen '0' sein!

Solange ein Bit auf '1' gesetzt ist, ist der betreffende Alarm gesperrt.

2. Akkus

2a) AKKU-2-L

Funktions-Nr.,
zulässige Werte: 1, 2 oder 3 mit:

- 1: Der Inhalt von AKKU 1 wird in das Steuerwort geladen.
- 2: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '1' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.
- 3: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '0' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.

2b) AKKU 1

neues Steuerwort oder Maske, abhängig von der gewünschten Funktion

Ergebnis

Nach korrekter und fehlerfreier Bearbeitung setzt das Systemprogramm das VKE auf '0'.
Der Aufruf des OB 143 bringt folgende Ergebnisse:

Funkt.-Nr. in AKKU-2-L	Inhalt von AKKU 1	
	vorher	nachher
1	Steuerwort	Steuerwort
2	Maske	neues Steuerwort
3	Maske	neues Steuerwort

Fehlerfälle

Im Fehlerfall setzt das Systemprogramm das VKE auf '1'.
 Es können die in der nachfolgenden Tabelle aufgeführten Fehlerfälle auftreten. Kommt es zu einem Fehler, so hinterlegt das Systemprogramm im AKKU-1-L die in der Tabelle aufgeführten Fehlerkennungen.

Tabelle 6-11 Fehlerkennungen des OB 143 im AKKU-1-L

Kennung	Bedeutung
8F01H 8F02H	unzulässige Funktions-Nr.in AKKU-2-L ¹⁾ eines der reservierten Bits in AKKU 1 ist '1' ¹⁾

¹⁾im AKKU-2-L befindet sich der fehlerhafte Wert

Steuerwort abfragen

- Der Zustand des Steuerwortes läßt sich durch folgende Programmsequenz abfragen:
 1. Funktions-Nr. '2' oder '3' in den AKKU-2-L laden,
 2. Wert '0' in den AKKU 1 laden,
 3. OB 143 aufrufen,
 4. AKKU 1 auslesen.

6.13 OB 150: Systemzeit stellen/lesen

Eigenschaften der Systemzeit

- Die Systemzeit wird über die Batterie im AG-Rahmen gepuffert. Wurde sie gestellt, so hat sie auch nach einem Spannungsausfall den aktuellen Wert.
- Die Auflösung beträgt 10 ms beim Lesen und 1 s beim Setzen.
- Schaltjahre werden berücksichtigt.
- Stundendarstellung wahlweise 24 Stunden oder 12 Stunden "am" (Vormittagszeit) und "pm" (Nachmittagszeit).
- Angabe des Wochentags
- Ein- und Ausgaben erfolgen BCD-codiert.

Funktion

Mit Hilfe des OB 150 können Sie in Ihrem Anwenderprogramm Datum und Uhrzeit der CPU 948 stellen oder auslesen. Datum und Uhrzeit werden im folgenden als "Systemzeit" bezeichnet.

Hinweis

Die Systemzeit wird – ausgehend von einem vorbesetzten Wert – nach dem Stecken der CPU gestartet.

Parameter

1. Datenfeld für die Zeitparameter

Beim **Stellen** der Systemzeit übernimmt der OB 150 die einzustellenden Werte aus einem Datenfeld, beim **Lesen** der Zeit überträgt der OB 150 die aktuellen Werte in das Datenfeld. Sie können dieses Datenfeld in einem **Datenbaustein** oder in einem der beiden **Merkerbereiche** (M- oder S-Merker) anlegen.

Das Datenfeld besteht aus vier Wörtern.

1a) Format des Datenfeldes beim Stellen der Systemzeit

Bit-Nr.	15			12	11			8	7			4	3			0	
1. Wort	Sekunden								0								
2. Wort	Format	Stunden								Minuten							
3. Wort	Monatstag								Wochentag				0				
4. Wort	Jahr								Monat								

1b) Format des Datenfeldes beim Lesen der Systemzeit

Bit-Nr.	15			12	11			8	7			4	3			0	
1. Wort	Sekunden								1/100 Sekunde								
2. Wort	Format	Stunden								Minuten							
3. Wort	Monatstag								Wochentag				0				
4. Wort	Jahr								Monat								

Die Zeitparameter haben folgende Bedeutung, zulässige Wertebereiche und Darstellung:

Parameter	zulässiger Wertebereich	Darstellung
Sekunden	0 bis 59	BCD-Format
1/100 Sekunden	0 bis 99 (bei "Systemzeit stellen" = 0)	
Minuten	0 bis 59	
Stunden	0 bis 23 oder 1 bis 12, je nach Angabe zu "Format"	
Wochentag	0 bis 6 für Mo bis So	
Monatstag ¹⁾	1 bis 31	
Monat	1 bis 12	
Jahr	0 bis 99	
Format	Formatangabe für das Stundenfeld mit folgender Bedeutung: Bit 15 = 0: 12-Stunden-Format (Wahl "am" oder "pm" in Bit 14) Bit 15 = 1: 24-Stunden-Format (Bit 14 = 0) Bit 14 = 0: "am" Bit 14 = 1: "pm"	–

1) Der angegebene Wert wird nach Aufruf des OB 150 logisch auf richtiges Datum unter Berücksichtigung der Schaltjahre überprüft.

Datenfeld im Merkerbereich

Wenn Sie das Datenfeld in einem Merkerbereich anlegen, müssen Sie folgende Zuordnung der Datenfeldwörter zu den Merkerbytes berücksichtigen. Dabei ist 'x' der Parameter "Nr. des 1. Datenfeldwortes", den Sie beim Aufruf des OB 150 im AKKU-1-L hinterlegen müssen:

Bit-Nr.	15	8	7	0	
1. Datenfeldwort	Merkerbyte x		Merkerbyte x+1		
2. Datenfeldwort	Merkerbyte x+2		Merkerbyte x+3		
3. Datenfeldwort	Merkerbyte x+4		Merkerbyte x+5		
4. Datenfeldwort	Merkerbyte x+6		Merkerbyte x+7		

2. Akkus

2a) AKKU-2-L

Der AKKU-2-L enthält Angaben zur gewünschten Funktion und zum verwendeten Datenfeld. Er muß folgenden Aufbau haben:

Bit-Nr.	15			12	11			8	7			4	3			0
	Funktions-Nr.				Adreßbereichs-Typ				Datenbaustein-Nr.							

Parameter im AKKU-2-L

Funktions-Nr.,
 zulässige Werte: 1 = Systemzeit stellen
 2 = Systemzeit lesen

Adreßbereichs-Typ,
 zulässige Werte: 1 = DB-Datenbaustein
 2 = DX-Datenbaustein
 3 = M-Merkerbereich
 4 = S-Merkerbereich

Datenbaustein-Nr.,
 zulässige Werte: 3 bis 255 (nur bei Adreßbereichs-Typ '1' oder '2';
 bei Adreßbereichs-Typ '3' oder '4' irrelevant)

2b) AKKU-1-L

Nummer des 1. Datenfeldwortes,
 zulässige Werte (in Abhängigkeit vom Adreßbereichs-Typ):

DB, DX:	0 bis 2039
M-Merker:	0 bis 248 (= Nr. Merkerbyte 'x')
S-Merker	0 bis 4088 (= Nr. Merkerbyte 'x')

Ergebnis

Nach korrekter Bearbeitung des OB 150 sind das VKE, die Anzeigenbits OR, ERAB und OS = 0.

Fehlerfälle

Es können die in der nachfolgenden Tabelle aufgeführten Fehlerfälle auftreten. Kommt es zu einem Fehler, so setzt das Systemprogramm das VKE auf '1' und hinterlegt im AKKU 1 die in der Tabelle aufgeführten Fehlerkennungen.

Tabelle 6-12 Fehlerkennungen des OB 150 im AKKU-1-L

Kennung	Bedeutung
9601H	Datenbaustein nicht geladen
960FH	Mehrfachaufruf des Bausteins
9611H	unzulässige Funktions-Nr.
9612H	Adreßbereichs-Typ unzulässig
9613H	Datenbaustein-Nr. unzulässig
9614H	"Nummer des ersten Datenfeldwortes" unzulässig
9615H	Datenbausteinlänge < 4 Wörter
9621H	Jahresangabe im Datenfeld unzulässig
9622H	Monatsangabe im Datenfeld unzulässig
9623H	Monatstagangabe im Datenfeld unzulässig
9624H	Wochentagangabe im Datenfeld unzulässig
9625H	Stundenangabe im Datenfeld unzulässig
9626H	Minutenangabe im Datenfeld unzulässig
9627H	Sekundenangabe im Datenfeld unzulässig
9628H	1/100 Sekunde im Datenfeld ungleich 0
9629H	Stunden-Format ungleich Einstellung bei OB 151

Hinweis

Wird bei der Funktion "Systemzeit stellen" eine fehlerhafte Parametrierung vorgenommen und es wurde vorher schon mindestens einmal die Uhrzeit korrekt gestellt, so führt dies zwar zu der geschilderten Übergabe der Fehlerkennungen; die vorher eingestellte Systemzeit läuft jedoch weiter.

Beispiele**"Systemzeit stellen":**

Die Systemzeit soll auf folgende Werte eingestellt werden:

"Do, 24.10.1993, 11:30 Uhr 0 Sekunden, 24-Stunden-Darstellung"

Die Zeitparameter sind im Datenbaustein DB 10 ab Datenwort DW 0 hinterlegt.

DB 10

0: KH = 0 0 0 0 linkes Byte = Sekunden (BCD), rechtes Byte = 0

1: KH = 9 1 3 0 91 = Format (=80H) + Stunde (=11 BCD)
30 = Minuten (BCD)

2: KH = 2 4 3 0 24 = Monatstag (BCD)
30 = Wochentag (3 = Donnerstag) + Bit 0 bis 3 = 0

3: KH = 9 3 1 0 93 = Jahr (BCD)
10 = Monat (BCD)

Fortsetzung auf der nächsten Seite

"Systemzeit stellen" (Fortsetzung)

Im OB 1 hinterlegte STEP-5-Operationen zum Aufrufen des OB 150:

```

:
:
:L   KH  1 1 0 A  Werte für AKKU-2-L:
:           |   |   |
:           |   |   |----- DB-Nr. = 10
:           |   |   |----- Adreßbereichs-Typ = 1 fuer "Datenfeld im DB"
:           |   |   |----- Funktions-Nr. = 1 fuer "Stellen"
:L   KF +0          AKKU-1-L:
:                   Nr. 1. Datenfeldwort = 0
:SPA OB 150        OB 150 aufrufen
:

```

"Systemzeit auslesen":

Die aktuelle Systemzeit soll in den Datenbaustein DB 10 ab Datenwort DW 4 geschrieben werden. Dazu müssen Sie den OB 150 mit folgenden Parametern aufrufen:

```

:
:L   KH  2 1 0 A  Werte fuer AKKU-2-L:
:           |   |   |
:           |   |   |----- DB-Nr. = 10
:           |   |   |----- Adreßbereichs-Typ = 1 fuer "Datenfeld im DB"
:           |   |   |----- Funktions-Nr. = 2 fuer "Lesen"
:L   KF +4          AKKU-1-L:
:                   Nr. 1. Datenfeldwort = 4
:SPA OB 150        OB 150 aufrufen
:A   DB 10          DB 10 aufschlagen
:                   DB 10 auswerten
:

```

Nach dem Aufruf des OB 150 ist die aktuelle Systemzeit in folgender Form im Datenbaustein DB 10 hinterlegt ("Do, 24.10.93, 11:30 Uhr 20 Sekunden, 13 Hundertstel-Sekunden, 24-Stunden-Darstellung"):

```

DW 4:  KH = 2 0 1 3   Sekunden = 20 (BCD)
                        1/100 Sekunden = 13 (BCD)
DW 5:  KH = 9 1 3 0   Format = 24 Std. (Bit 15 = 1, Bit 14 = 0),
                        Stunden = 11(BCD), Minuten = 30 (BCD)
DW 6:  KH = 2 4 3 0   Monatstag = 24 (BCD)
                        Wochentag = 3 = Donnerstag
DW 7:  KH = 9 3 1 0   Jahr = 93 (BCD)
                        Monat = 10 (BCD)

```


6.14 OB 151: Zeit für uhrzeitgesteuerten Weckalarm stellen/lesen

Funktion

Durch Aufruf des OB 151 können Sie

- die CPU 948 dazu veranlassen, zu einer vorgegebenen Zeit den uhrzeitgesteuerten Weckalarm ("Zeitauftrag" – OB 9, siehe Abschnitt 4.5.3) zu aktivieren:
 - minütlich
 - stündlich
 - täglich
 - wöchentlich
 - monatlich
 - jährlich
 - einmalig,
- den momentanen Status eines Zeitauftrags auslesen,
- einen bereits erzeugten Zeitauftrag stornieren.

Der OB 151 kann in den Betriebszuständen ANLAUF und RUN aufgerufen werden. Ein erzeugter uhrzeitgesteuerter Weckalarm bleibt bei WIEDERANLAUF (automatisch oder manuell) erhalten. Bei NEUSTART wird ein vorhandener Zeitauftrag gelöscht.

Wenn Sie einen neuen Zeitauftrag erzeugen, wird ein laufender automatisch storniert. Es ist also immer nur **ein** uhrzeitgesteuerter Weckalarm aktiv.

Parameter

1. Datenfeld für Auftragsparameter

Beim **Erzeugen** bzw. **Stornieren** eines Zeitauftrags entnimmt der OB 151 die benötigten Auftragsparameter einem Datenfeld. Beim **Auslesen** des aktuellen Zustands der Auftragsverwaltung überträgt der OB 151 die aktuellen Auftragsparameter in ein Datenfeld.

Sie können dieses Datenfeld in einem **Datenbaustein** oder in einem der beiden **Merkerbereiche** (M- oder S-Merker) anlegen.

Das Datenfeld besteht aus vier Wörtern und hat beim Erzeugen und Auslesen eines Zeitauftrags folgendes Format:

Bit-Nr.	15			12	11			8	7			4	3			0
1. Wort	Sekunden							0								
2. Wort	Format	Stunden							Minuten							
3. Wort	Monatstag							Wochentag				Auftragsart				
4. Wort	Jahr							Monat								

Die Parameter haben folgende Bedeutung, zulässige Wertebereiche und Darstellung:

Parameter	zulässiger Wertebereich	Darstellung
Auftragsart	0 bis 7 mit: 0: Auftrag stornieren bzw. kein Auftrag aktiv 1: minütlich 2: stündlich 3: täglich 4: wöchentlich 5: monatlich 6: jährlich 7: einmalig	BCD-Format
Sekunden 1/100 Sekunden Minuten Stunden Wochentag Monatstag ¹⁾ Monat Jahr	0 bis 59 0 0 bis 59 0 bis 23 oder 1 bis 12, je nach Angabe zu "Format" 0 bis 6 für Mo bis So 1 bis 31 1 bis 12 0 bis 99	BCD-Format
Format ²⁾	Formatangabe für das Stundenfeld mit folgender Bedeutung: Bit 15 = 0: 12-Stunden-Format Bit 15 = 1: 24-Stunden-Format (Bit 14 = 0) Bit 14 = 0: "am" Bit 14 = 1: "pm"	–

- 1) Der angegebene Wert wird nach Aufruf des OB 151 logisch auf richtiges Datum unter Berücksichtigung der Schaltjahre überprüft.
- 2) Bedeutung "am und "pm": siehe OB 150 im vorigen Abschnitt: "Format" muß mit der Formatangabe beim Stellen der Systemzeit durch OB 150 übereinstimmen!

Datenfeld im Merkerbereich

Wenn Sie das Datenfeld in einem Merkerbereich anlegen, müssen Sie folgende Zuordnung der Datenfeldwörter zu den Merkerbytes berücksichtigen. Dabei ist 'x' der Parameter "Nr. des 1. Datenfeldwortes", den Sie beim Aufruf des OB 151 im AKKU-1-L hinterlegen müssen:

Bit-Nr.	15	8 7	0
1. Datenfeldwort	Merkerbyte x	Merkerbyte x+1	
2. Datenfeldwort	Merkerbyte x+2	Merkerbyte x+3	
3. Datenfeldwort	Merkerbyte x+4	Merkerbyte x+5	
4. Datenfeldwort	Merkerbyte x+6	Merkerbyte x+7	

2. Akkus

2a) AKKU-2-L

Der AKKU-2-L enthält Angaben zur gewünschten Funktion und zum verwendeten Datenfeld. Er muß folgenden Aufbau haben:

Bit-Nr.	15			12	11			8	7			4	3			0
	Funktions-Nr.				Adreßbereichs-Typ				Datenbaustein-Nr.							

Parameter im AKKU-2-L

Funktions-Nr.,
 zulässige Werte: 1 = Auftrag erzeugen
 2 = Auftragsstatus lesen

Adreßbereichs-Typ,
 zulässige Werte: 1 = DB-Datenbaustein
 2 = DX-Datenbaustein
 3 = M-Merkerbereich
 4 = S-Merkerbereich

Datenbaustein-Nr.,
 zulässige Werte: 3 bis 255 (nur bei Adreßbereichs-Typ '1' oder '2';
 bei Adreßbereichs-Typ '3' oder '4' irrelevant)

2b) AKKU-1-L

Nummer des 1. Datenfeldwortes,
 zulässige Werte (in Abhängigkeit vom Adreßbereichs-Typ:

DB, DX:	0 bis 2039
M-Merker:	0 bis 248 (= Nr. Merkerbyte 'x')
S-Merker:	0 bis 4088 (= Nr. Merkerbyte 'x')

Hinweis

Es ist nicht sinnvoll, einen Zeitauftrag zyklisch zu erzeugen (z. B. durch einen absoluten Aufruf des OB 151 mit der Funktions-Nr. '1' im OB 1).

Ergebnis

Nach korrekter Bearbeitung des OB 151 sind das VKE, die Anzeigenbits OR, ERAB und OS = 0.

Hinweis

Wenn beim Auslesen des Zeitauftrags im Datenfeld die **Auftragsart '0'** und alle restlichen Parameter **'F'** oder **'FF'** (hexadezimal) sind, ist kein Zeitauftrag aktiv. Dieser Zustand kann entstehen

- a) wenn NEUSTART durchgeführt wurde, ohne einen Zeitauftrag zu erzeugen,
- b) wenn ein einmaliger Zeitauftrag fällig war
oder
- c) wenn Sie einen Auftrag storniert haben.

Fehlerfälle

Es können die in der nachfolgenden Tabelle aufgeführten Fehlerfälle auftreten. Kommt es zu einem Fehler, so setzt das Systemprogramm das VKE auf '1' und hinterlegt im AKKU 1 die in der Tabelle aufgeführten Fehlerkennungen.

Tabelle 6-13 Fehlerkennungen des OB 151 im AKKU-1-L

Kennung	Bedeutung
9701H	Datenbaustein nicht geladen
970FH	Mehrfachaufruf des Bausteins
9710H	falsche Betriebsart ("Prozeßalarme über EB 0 = ein")
9711H	unzulässige Funktions-Nr.
9712H	Adreßbereichs-Typ unzulässig
9713H	Datenbaustein-Nr. unzulässig
9714H	"Nummer des ersten Datenfeldwortes" unzulässig
9715H	Datenbausteinlänge < 4 Wörter
9721H	Jahresangabe im Datenfeld unzulässig
9722H	Monatsangabe im Datenfeld unzulässig
9723H	Monatstagangabe im Datenfeld unzulässig
9724H	Wochentagangabe im Datenfeld unzulässig
9725H	Stundenangabe im Datenfeld unzulässig
9726H	Minutenangabe im Datenfeld unzulässig
9727H	Sekundenangabe im Datenfeld unzulässig
9728H	1/100 Sekunde im Datenfeld ungleich 0
9729H	Stunden-Format ungleich Einstellung bei OB121/OB 150
972AH	Auftragsart unzulässig

Hinweis

Wird eine fehlerhafte Parametrierung vorgenommen **und** es wurde vorher schon ein gültiger Zeitauftrag erzeugt, so führt dies zwar zu der oben geschilderten Übergabe der Fehlerkennungen; **der vorher erzeugte Zeitauftrag bleibt jedoch weiter aktiv.**

Was Sie bei den Zeitparametern beachten müssen

Abhängig davon, wann ein uhrzeitgesteuerter Weckalarm (Zeitauftrag) ausgelöst werden soll, müssen Sie die einzelnen Zeitparameter in bestimmten Kombinationen vorgeben. Dabei müssen Sie – abhängig vom gewählten Zeitpunkt für den uhrzeitgesteuerten Weckalarm – bestimmte Parameter unbedingt angeben, während andere vom Systemprogramm nicht ausgewertet werden.

Nachfolgende Tabelle gibt Ihnen an, welche Zeitparameter bei welchem Zeitauftrag angegeben werden müssen (XXX = notwendige Angabe, --- = irrelevant).

Tabelle 6-14 Zuordnung "Zeitauftrag – Zeitparameter"

Weckzeit	Sekunden	Minuten	Stunden	Wochentag	Montagstag	Monat	Jahr
minütlich	XXX	---	---	---	---	---	---
stündlich	XXX	XXX	---	---	---	---	---
täglich	XXX	XXX	XXX	---	---	---	---
wöchentlich	XXX	XXX	XXX	XXX	---	---	---
monatlich	XXX	XXX	XXX	---	XXX	---	---
jährlich	XXX	XXX	XXX	---	XXX	XXX	---
einmalig	XXX	XXX	XXX	---	XXX	XXX	XXX

Beim **Auslesen** der Zeitparameter werden die irrelevanten Parameter mit FFH besetzt.

Besonderheiten

- Wenn bei Auftragsart "jährlich" (= 6) der "29. Februar" als Montagstag und Monat gewählt wird, so resultiert daraus, daß der OB 9 nur in jedem Schaltjahr aufgerufen wird.
- Wenn bei Auftragsart "monatlich" (= 5) als Montagstag der Wert "29", "30" oder "31" gewählt wird, so wird der OB 9 nur in den Monaten aufgerufen, bei denen dieser Tag vorhanden ist.

Beispiele

Verschiedene Zeitaufträge (im 24-Stunden-Anzeigeformat):

1. "Auftrag jede Minute beim Sekundenwert 29"
(12:44:29, 12:45:29 usw.):

Sie müssen angeben: Auftragsart = 1 (Funktions-Nr. in AKKU-2-L = 1)
 Sekunden = 29

2. "Auftrag stündlich jeweils um xx:14:15 Uhr":

Sie müssen angeben: Auftragsart = 2 (Funktions-Nr. in AKKU-2-L = 1)
 Sekunden = 15
 Minuten = 14

3. "Auftrag täglich um 5:32:47 Uhr":

Sie müssen angeben: Auftragsart = 3 (Funktions-Nr. in AKKU-2-L = 1)
 Sekunden = 47
 Minuten = 32
 Format/Std. = 85

4. "Auftrag wöchentlich, dienstags um 10:50:00 Uhr":

Sie müssen angeben: Auftragsart = 4 (Funktions-Nr. in AKKU-2-L = 1)
 Sekunden = 00
 Minuten = 50
 Format/Std. = 90
 Wochentag = 01

5. "Auftrag monatlich, jeweils am 14. um 7:30:15 Uhr":

Sie müssen angeben: Auftragsart = 5 (Funktions-Nr. in AKKU-2-L = 1)
 Sekunden = 15
 Minuten = 30
 Format/Std. = 87
 Monatstag = 14

6. "Auftrag jährlich, jeweils am 1.5. um 00:01:45 Uhr":

Sie müssen angeben: Auftragsart = 6 (Funktions-Nr. in AKKU-2-L = 1)
 Sekunden = 45
 Minuten = 01
 Format/Std. = 80
 Monatstag = 01
 Monat = 05

Fortsetzung auf der nächsten Seite

**Verschiedene Zeitaufträge (im 24-Stunden-Anzeigeformat),
Fortsetzung :**

7. "Auftrag einmalig am 31.12.1999 um 23:55:00 Uhr":

Sie müssen angeben:

Auftragsart =	7	(Funktions-Nr. in AKKU-2-L = 1)
Sekunden =	00	
Minuten =	55	
Format/Std. =	A3	
Monatstag =	31	
Monat =	12	
Jahr =	99	

8. "Auftrag stornieren":

Sie müssen angeben: Auftragsart = 0 (Funktions-Nr. in AKKU-2-L = 1)

9. "Zeitauftrag auslesen":

Sie müssen angeben: Funktions-Nr. in AKKU-2-L = 2

Falls kein Auftrag aktiv ist, erhalten Sie als Ergebnis im Datenfeld:

Datenfeldwort 0:	FFFF	H
Datenfeldwort 1:	FFFF	H
Datenfeldwort 2:	FFF0	H
Datenfeldwort 3:	FFFF	H

6.15 OB 153: Zeit für Verzögerungsalarm stellen/lesen

Über den OB 153 können Sie sogenannte "Verzögerungsaufträge" an das Systemprogramm übergeben. Diese führen dazu, daß nach Ablauf einer vorgegebenen Verzögerungszeit ein "Verzögerungsalarm" bearbeitet wird (siehe OB 6, Abschnitt 4..5.3).

Funktion

Durch Aufruf des OB 153 können Sie

- eine Verzögerungszeit definieren und starten,
- eine aktivierte Verzögerungszeit stoppen (Verzögerungsauftrag stornieren),
- die aktuelle Restzeit einer aktivierten Verzögerungszeit lesen.

Ein Verzögerungsauftrag kann über den OB 153 in den Betriebszuständen ANLAUF und RUN **abgegeben** werden.

Lebensdauer eines Verzögerungsauftrags

Der durch einen Verzögerungsauftrag ausgelöste Verzögerungsalarm wird jedoch vom Systemprogramm **nur im RUN aktiviert** (Aufruf des OB 6).

Aufträge, die außerhalb des RUN fällig werden, werden vom Systemprogramm **ohne Meldung verworfen**.

Ein laufender (und noch nicht fälliger) Auftrag wird außerdem verworfen beim Übergang in den STOP sowie bei NETZ AUS.

Parameter

Akkus

a) AKKU-2-L

Der AKKU-2-L muß nur bei Aufruf des OB 153 mit der Funktions-Nr. '1' ("Verzögerungszeit definieren") versorgt werden:

Verzögerungszeit in Millisekunden (max. 65535)

zulässige Werte: 0001H bis FFFFH

b) AKKU-1-L

Funktions-Nr.

zulässige Werte: 1 = Verzögerungszeit definieren und starten
2 = Verzögerungszeit stoppen (= Auftrag storn.)
3 = aktuelle Restzeit lesen

Hinweis

Ist beim Definieren einer Verzögerungszeit eine vorher definierte Verzögerungszeit noch nicht abgelaufen, so wird diese "vergessen" und die neue Verzögerungszeit gestartet.

Ergebnis

Nach korrekter Bearbeitung des OB 153 sind das VKE, die Anzeigebits OR, ERAB und OS = 0.

Bei Aufruf des OB 153 mit der Funktions-Nr. '2' oder '3' enthält AKKU-1-L die Restzeit in Millisekunden.

Ist bei Aufruf des OB 153 mit der Funktions-Nr. '2' oder '3' kein Verzögerungsauftrag aktiv, so enthält AKKU-1-L den Wert '0'.

Fehlerfälle

Es können die in der nachfolgenden Tabelle aufgeführten Fehlerfälle auftreten. Kommt es zu einem Fehler, so setzt das Systemprogramm das VKE auf '1' und hinterlegt im AKKU 1 die in der Tabelle aufgeführten Fehlerkennungen.

Tabelle 6-15 Fehlerkennungen des OB 153

Kennung	Bedeutung
990FH	Mehrfachaufruf des Bausteins
9910H	falsche Betriebsart ("Prozeßalarme über EB 0 = ein")
9911H	unzulässige Funktions-Nr.
9921H	Verzögerungszeit unzulässig

Beispiele

Verzögerungszeit definieren und starten:

Bei einem AUTOMATISCHEN WIEDERANLAUF soll nach 5 Sekunden einmalig eine bestimmte STEP-5-Operationsfolge durchlaufen werden. Dazu wird im Anlauforganisationsbaustein OB 22 die Verzögerungszeit definiert und gestartet.

Im OB 22 hinterlegte STEP-5-Operationen zum Aufrufen des OB 153:

```

:
:
:L   KF +5000      Wert fuer AKKU-2-L: 5000 ms
:L   KF +1         Wert fuer AKKU-1-L: Funktions-Nr. = 1 fuer
:                 "Verzoegerungszeit definieren und starten"
:SPA OB 153       OB 153 aufrufen
    
```

Verzögerungszeit stoppen (Auftrag stornieren):

STEP-5-Operationen zum Aufrufen des OB 153:

```
:
:
:L   KF  +2      Wert fuer AKKU-1-L: Funktions-Nr. = 2 fuer
:               "Verzoegerungszeit stoppen"
:SPA  OB 153    OB 153 aufrufen
:
:
```

Restzeit eines Verzögerungsauftrags auslesen):

STEP-5-Operationen zum Aufrufen des OB 153:

```
:
:
:L   KF  +3      Wert fuer AKKU-1-L: Funktions-Nr. = 3 fuer
:               "Restzeit auslesen"
:SPA  OB 153    OB 153 aufrufen
:
:               AKKU-1-L enthaelt die Restzeit des Verzoegerungs-
:               auftrags
:
```

6.16 OB 180: Variabler Datenbaustein-Zugriff

Anwendung des OB 180

Sie können den OB 180 anwenden beim Arbeiten mit Datenbausteinen, die eine Länge größer 261 Wörter (inkl. 5 Wörter Kopf) haben: Mit Hilfe des OB 180 können Sie ein "Zugriffsfenster" von 256 Datenwörtern im 16-Wort-Raster über einem Datenbaustein (auf Paragrafenadressen) verschieben. Durch mehrfaches Aufrufen des OB 180 können Sie das Zugriffsfenster weiter verschieben.

Im Unterschied zur CPU 928B können Sie bei der CPU 948 das Zugriffsfenster nicht kontinuierlich, sondern nur um **Vielfache von 16** positionieren.

Funktion

Mit dem OB 180 wird die Anfangsadresse des aktuellen Datenbausteins um einen vorgegebenen Wert zum Bausteinende hin verschoben. Dabei wird berücksichtigt, daß die noch verfügbare Länge des DBs abnimmt (DBA- und DBL-Register – siehe Abschnitte 8.3 und 9.2.1 – werden entsprechend der Verschiebung geladen).

Hinweis

Vor Aufruf des OB 180 muß ein **ausreichend langer** Datenbaustein (DB oder DX) aufgeschlagen sein.

Parameter

AKKU-1-L

Verschiebezahl V: Anzahl der Datenwörter, um die die Datenbausteinanfangsadresse verschoben werden soll

zulässige Werte: $0 \leq V < \text{DBL}$, $V = n * 16$ (16, 32, 48 ...)

Ergebnis

Nach **erfolgreichem** Aufruf des OB 180

- Ist die relative Adresse des DW 0 entsprechend dem Wert in AKKU-1-L verschoben (DBA- und DBL-Register wurden entsprechend nachgeführt.),
- ist das VKE gelöscht (VKE = 0),
- sind alle übrigen Bit- und Wortanzeigen gelöscht,
- ist der Inhalt von AKKU-1-L = 0.

Fehlerfälle

Es können die in der nachfolgenden Tabelle aufgeführten Fehlerfälle auftreten. Kommt es zu einem Fehler, so setzt das Systemprogramm das VKE auf '1' und hinterlegt im AKKU 1 die in der Tabelle aufgeführten Fehlerkennungen. Die übrigen Bit- und Wortanzeigen werden gelöscht.

Die Werte von DBA und DBL bleiben unverändert.

Tabelle 6-16 Fehlerkennungen des OB 180

Kennung	Bedeutung
B401H	Es kein Datenbaustein aufgeschlagen.
B410H	Die Verschiebezahl V ist kein Vielfaches von 16.
B411H	a) Die Verschiebezahl V ist zu groß; das Bausteinende würde durch die neue Position des Fensters überschritten. b) Die Verschiebezahl V ist negativ.

Zugriffsfenster auf Anfangswert zurückstellen

Ein erneutes Aufschlagen des Datenbausteins mit den Operationen A DB oder AX DX stellt den Grundzustand wieder her.

Verhalten bei verschachtelten Bausteinablauf

Wird in einem Codebaustein das Zugriffsfenster durch Aufruf des OB 180 verschoben und anschließend ein weiterer Codebaustein aufgerufen, so bleibt im aufgerufenen Codebaustein die Position des Zugriffsfensters **bis zu einem neuen Aufruf des OB 180 erhalten** (DBA/DBL-Werte ändern sich nicht).

Wird dagegen das Zugriffsfenster in einem aufgerufenen Codebaustein durch Aufruf des OB 180 verschoben, so wird es beim Rücksprung aus dem aufgerufenen Baustein (Bausteinende-Operation) wieder **auf die Position zurückgestellt, die es beim Aufruf des eingeschachtelten Codebausteins eingenommen hat.**

Beispiel

Die Datenbausteinanfangsadresse (DBA = 4152H im DB 17 (Länge = 256 DW) soll relativ um 32 Datenwörter zum Bausteinende hin verschoben werden.

```
:A  DB   17   DB 17 aufgeschlagen
:L  KB   32   Verschiebung / Versatz als Konstante
:SPA OB  180  OB 180 aufrufen: Zugriffsfenster verschieben
```

Nach Aufruf des OB 180 läßt sich z.B. das unter der Adresse 4 1543H gespeicherte Datenwort nicht mehr mit DW 35, sondern mit DW 3 ansprechen usw.. (siehe Bild 6-3).

Aufgrund des gleichzeitig veränderten DBL-Registers bleibt die **Fehler-Überwachung gewährleistet**: Die Operation T DW 223 ist erlaubt, während T DW 224/L DW 224 fehlerhaft sind.

Durch erneute Aufrufe des OB 180 kann das DBA weiter erhöht werden (wobei das DBL weiter verringert wird): Die Operation A DB 17 stellt den Grundzustand (DBA = 4152H, Länge = 256 DW) wieder her. Hätte der DB 17 eine Länge von z.B. 258 Datenwörtern, könnten Sie mit STEP-5-Operationen nicht mehr auf DW 256 und DW 257 zugreifen. Durch Verschieben des DBA-Registers um 16 lassen sich die Datenwörter 256 und 257 mit "DW 240" und "DW 241" ansprechen.

Fortsetzung auf der nächsten Seite

Fortsetzung des Beispiels:

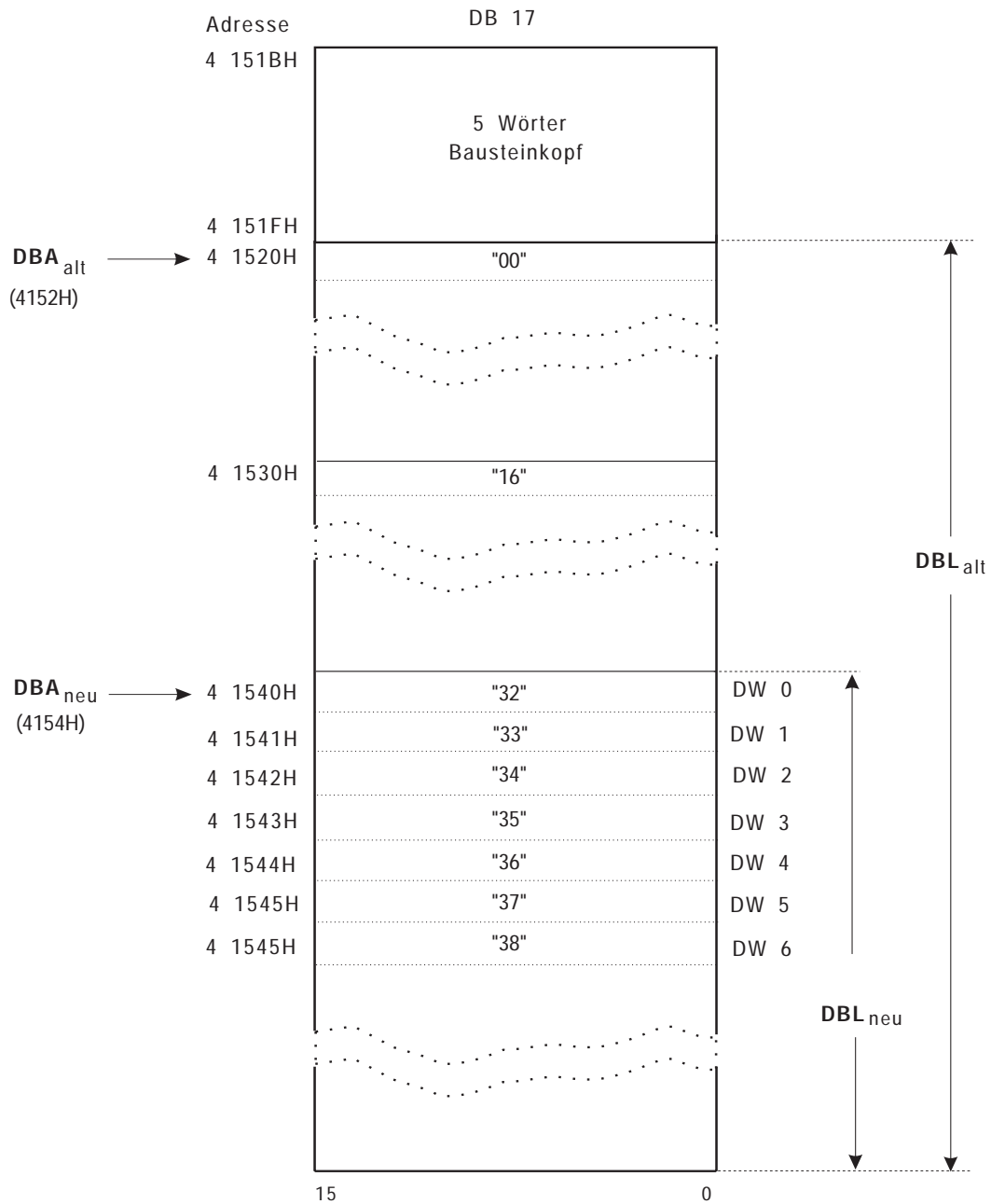


Bild 6-3 Verschieben der DB-Anfangsadresse

6.17 OB 181: Datenbausteine (DB/DX) testen

Mit dem Sonderfunktions-Organisationsbaustein OB 181 können Sie prüfen,

- ob ein bestimmter DB- oder DX-Datenbaustein vorhanden ist,
- unter welcher Adresse das erste Datenwort des Datenbausteins abgelegt ist,
- wieviele Datenwörter dieser Datenbaustein enthält,

Anwendung des OB 181

Eine Anwendung der Funktion "DB/DX testen" ist sinnvoll vor den Befehlen TNB/TNW, E DB/EX DX und vor Aufruf der Sonderfunktions-Organisationsbausteine OB 182, OB 254 und OB 255.

So können Sie beispielsweise vor einem Blocktransfer von Datenwörtern den OB 181 aufrufen, um sicherzustellen, daß der Zieldatenbaustein gültig und lang genug ist, um alle zu kopierenden Datenwörter aufzunehmen.

Funktion

Der OB 181 prüft, ob ein vorgegebener Datenbaustein vorhanden ist, und gibt als Ergebnis die charakteristischen Parameter des Datenbausteins zurück.

Parameter

AKKU-1-L

a) AKKU-1-LL

Bausteinnummer,
zulässige Werte: 1 bis 255

b) AKKU-1-LH

Bausteinkennung,
zulässige Werte: 1 = DB
2 = DX

Ergebnis

Wir die Funktion fehlerfrei durchgeführt und ist der geprüfte Baustein in der CPU **vorhanden**, so übergibt das Systemprogramm folgende Ergebniswerte:

- **AKKU-1-L:** Adresse des 1. Datenwortes (DW 0),
20-bit-Adresse,
- **AKKU-2-L:** Länge des Datenbausteins in Wörtern (ohne Bausteinkopf)
Beispiel: In AKKU-2-L steht der Wert '7':
Der Datenbaustein besteht aus den Datenwörtern DW 0 bis DW 6,
- **VKE:** = 0.

Fehlerfälle

Es können die in der nachfolgenden Tabelle aufgeführten Fehlerfälle auftreten. Kommt es zu einem Fehler, so setzt das Systemprogramm das VKE auf '1' und weitere in der nachfolgenden Tabelle aufgeführte Anzeigen. Außerdem hinterlegt es im AKKU-1-L eine Fehlerkennung.

Tabelle 6-17 Fehleranzeigen des OB 181 und deren Abfrage

VKE	ANZ 1	ANZ 0	AKKU-1-L	Bedeutung	Abfrage
1	0	1	B501H	Baustein nicht vorhanden	SPB SPM SPN
1	1	0	B502H	falsche Bausteinnummer	SPB SPP SPN
1	1	0	B503H	falsche Bausteinkennung	SPB SPP SPN

6.18 OB 182: Datenbereich kopieren

Funktion

Der OB 182 kopiert einen Datenblock variabler Länge von einem Datenbaustein in einen anderen. Als Quell- und Zielbausteine können DB- und DX-Datenbausteine verwendet werden. Der Blockanfang im Quell- und Zieldatenbaustein ist frei wählbar. Der OB 182 kann maximal 4091 Datenwörter kopieren.

Hinweis

Quell- und Zielbaustein können identisch sein. Die Datenbereiche von Quelle und Ziel dürfen sich überlappen. Die **Originaldaten** des Quellbereichs werden auch bei Überlappung unverändert in den **Zielbereich** kopiert. Der **Überlappungsbereich in der Quelle** ist nach dem Kopiervorgang überschrieben. Diese Funktionseigenschaft können Sie einsetzen, um einen Datenbereich innerhalb eines Bausteins zu verschieben.

Parameter

1. Datenfeld mit Kopierparametern

Vor dem Aufruf des OB 182 versorgen Sie ein Datenfeld mit den erforderlichen Parametern für den gewünschten Kopiervorgang. Dieses Datenfeld kann in einem DB- oder DX-Datenbaustein, im M- oder S-Merkbereich angelegt werden.

Das Datenfeld bezeichnet Quell- und Zieldatenbaustein, die Blockanfangsadresse in beiden Bausteinen sowie die Anzahl der zu übertragenden Datenwörter. Es besteht aus fünf Wörtern:

Bit-Nr.	15	8	7	0
1. Wort	Quell-DB-Typ		Quell-DB-Nr.	
2. Wort	Nr. des 1. zu übertragenden Datenwortes im Quell-DB			
3. Wort	Ziel-DB-Typ		Ziel-DB-Nr.	
4. Wort	Nr. des 1. übertragenen Datenwortes im Ziel-DB			
5. Wort	Anzahl der Datenwörter			

Die Parameter haben folgende Bedeutung und zulässige Wertebereiche:

Parameter	zulässiger Wertebereich
Datenbaustein-Typ (Quelle und Ziel)	1 = DB 2 = DX
Datenbaustein-Nr. (Quelle und Ziel)	3 bis 255
Nr. des 1. Datenwortes (Quelle und Ziel)	0 bis 4090
Anzahl der Datenwörter	1 bis 4091

Datenfeld im Merkerbereich

Wenn Sie das Datenfeld in einem Merkerbereich anlegen, müssen Sie folgende Zuordnung der Datenfeldwörter zu den Merkerbytes berücksichtigen. Dabei ist 'x' der Parameter "Nr. des 1. Datenfeldwortes", den Sie beim Aufruf des OB 182 im AKKU-1-L hinterlegen müssen:

Bit-Nr.	15	8	7	0
1. Datenfeldwort	Merkerbyte x		Merkerbyte x+1	
2. Datenfeldwort	Merkerbyte x+2		Merkerbyte x+3	
3. Datenfeldwort	Merkerbyte x+4		Merkerbyte x+5	
4. Datenfeldwort	Merkerbyte x+6		Merkerbyte x+7	
5. Datenfeldwort	Merkerbyte x+8		Merkerbyte x+9	

2. Akkus

2a) AKKU-2-L

Der AKKU-2-L enthält Angaben zum verwendeten Datenfeld. Er muß folgenden Aufbau haben:

Bit-Nr.	15	8	7	0
	Adreßbereichs-Typ		Datenbaustein-Nr.	

Parameter im AKKU-2-L

Adreßbereichs-Typ,

- zulässige Werte:
- 1 = DB-Datenbaustein
 - 2 = DX-Datenbaustein
 - 3 = M-Merkerbereich
 - 4 = S-Merkerbereich

Datenbaustein-Nr.,

- zulässige Werte:
- 3 bis 255 (nur bei Adreßbereichs-Typ '1' oder '2'; bei Adreßbereichs-Typ '3' oder '4' irrelevant)

2b) AKKU-1-L

Nummer des 1. Datenfeldwortes,
mögliche Werte (in Abhängigkeit
vom Adreßbereichs-Typ:

DB, DX:	0 bis 2038
M-Merker:	0 bis 246 (= Nr. Merkerbyte 'x')
S-Merker	0 bis 4086 (= Nr. Merkerbyte 'x')

Ergebnis

Nach korrekter Bearbeitung des OB 182 sind das VKE, die Anzeigenbits OR, ERAB und OS = 0.

Fehlerfälle

Es wird in AKKU 1 eine Fehlerkennung hinterlegt (siehe nachfolgende Tabelle).

Tabelle 6-18 Fehlerkennungen des OB 182 im AKKU-1-L

AKKU-1-L	Bedeutung
B601H	Datenbaustein nicht geladen
B60FH	Bausteinmehrfachaufruf
B611H	Beschreibung des Datenfeldes fehlerhaft
B612H	Adreßbereich-Typ unzulässig
B613H	Datenbaustein-Nr. unzulässig
B621H	"Nummer des ersten Datenfeldwortes" unzulässig
B622H	"Quell-Datenbaustein-Typ" unzulässig
B623H	"Quell-Datenbaustein-Nr." unzulässig
B624H	"Nr. des 1. zu übertragenden Datenwortes im Quell-DB" unzulässig
B625H	Länge Quelldatenbaustein im Bausteinkopf < 5 Wörter
B626H	"Ziel-Datenbaustein-Typ" unzulässig
B627H	"Ziel-Datenbaustein-Nr." unzulässig
B628H	"Nr. des 1. zu beschreibenden Datenwortes im Ziel-DB" unzulässig
B629H	Länge Zieldatenbaustein im Bausteinkopf < 5 Wörter
B62AH	"Anzahl zu übertragende Datenwörter" unzulässig (= 0 oder > 4091)
B62BH	Quelldatenbaustein zu kurz
B62CH	Zieldatenbaustein zu kurz

6.19 OB 202 bis 205: Mehrprozessorkommunikation

Eine detaillierte Anleitung zu diesen Sonderfunktions-Organisationsbausteinen finden Sie in Kapitel 10.

Die Sonderfunktions-Organisationsbausteine OB 200 und OB 202 bis OB 205 ermöglichen im Mehrprozessorbetrieb Datenübertragungen zwischen den einzelnen CPUs unter Verwendung des Koordinators KOR C.

- **OB 200: Initialisieren**

Dieser Sonderfunktions-Organisationsbaustein richtet im Koordinator KOR C den Speicher ein, in dem die zu übertragenden Datenblöcke zwischengespeichert werden.

- **OB 202: Senden**

Diese Funktion übergibt einen Datenblock in den Zwischenspeicher des KOR C und gibt an, wieviele Datenblöcke noch gesendet werden können.

- **OB 203: Sende-Test**

Der Sonderfunktions-OB 203 ermittelt die Anzahl der freien Speicherblöcke im Zwischenspeicher des Koordinators KOR C.

- **OB 204: Empfangen**

Diese Funktion übernimmt einen Datenblock vom Zwischenspeicher des KOR C und zeigt an, wieviele Datenblöcke noch empfangen werden können.

- **OB 205: Empfangs-Test**

Der Sonderfunktions-Organisationsbaustein OB 205 ermittelt die Anzahl belegter Speicherblöcke im Zwischenspeicher des KOR C.

6.20 OB 222: Zyklusüberwachungszeit neu starten

Die Sonderfunktion OB 222 bewirkt ein Nachtriggern der Zyklusüberwachungszeit, d. h. der Timer für die Überwachung wird neu gestartet. Durch Aufruf dieser Sonderfunktion wird die maximal zulässige Zykluszeit für den aktuellen Zyklus um den eingestellten Wert vom Zeitpunkt des Aufrufs an verlängert.

Parameter keine

Fehlerfälle keine

6.21 OB 223: Anlaufarten vergleichen

Funktion	Durch Aufruf des OB 223 können Sie überprüfen, ob die Anlaufarten aller beteiligten CPUs im Mehrprozessorbetrieb gleich sind, um im Fehlerfall per Programm darauf reagieren zu können.
Parameter	keine
Ergebnis	Nach Aufruf des OB 223 setzt das Systemprogramm das VKE auf '0' und hinterlegt im AKKU-1-LL den Wert 01H, wenn die Anlaufarten gleich sind.
Fehlerfälle	<ul style="list-style-type: none"> • Anlaufarten sind ungleich • sonstige Fehler, siehe Anzeigen
Anzeigen	Im Fehlerfall setzt das Systemprogramm das VKE auf '1' und übergibt in AKUU-1-LL eine Fehlerkennung.

Tabelle 6-19 Ergebniskennungen des OB 223 im AKKU-1-LL

Kennung	Bedeutung
01H	Anlaufarten sind gleich
02H	interner Systemfehler
03H	Anlaufarten sind ungleich
04H	Einzelprozessorbetrieb: ein Vergleich der Anlaufarten ist nicht möglich

6.22 OB 254/255: Datenbausteine kopieren/duplizieren

Mit den Sonderfunktionen OB 254/255 kopieren Sie einzelne Datenbausteine von einer Memory Card in den Anwenderspeicher oder duplizieren Sie einzelne Datenbausteine innerhalb des Anwenderspeichers.

Die Sonderfunktionen OB 254 und OB 255 laufen identisch ab, wobei der OB 254 ausschließlich für DX-Datenbausteine und der OB 255 für DB-Datenbausteine zuständig ist.

Hinweis

Während des Kopier- bzw. Dupliziervorgangs sind die Anwenderalarme gesperrt: Es kommen keine Weckalarme und keine Prozeßalarme bzw. Interrupts durch.

Anwendung

Datenbausteine von der Memory Card kopieren oder Datenbausteine im Anwenderspeicher duplizieren und mit einer neuen Bausteinnummer versehen.

Kopieren

Randbedingungen

Bei der Anwendung "**Kopieren**" der beiden Sonderfunktionen (Aufruf des OB 254 bzw. OB 255) müssen Sie folgende Randbedingungen beachten:

- **Die Memory Card muß vor dem URLÖSCHEN gesteckt werden und darf danach nicht gezogen werden,**
- Der Zieldatenbaustein darf noch nicht vorhanden sein,
- Die Online-Funktion "Speicher komprimieren" darf nicht aktiv sein.

Funktion

Ein Datenbaustein wird unter Beibehaltung seiner ursprünglichen Bausteinnummer von der Memory Card in den Anwenderspeicher übertragen. Die Anfangsadresse wird in die Adreßliste im DB 0 eingetragen.

Parameter

1. AKKU-1-LL

Nummer des zu kopierenden Bausteins.

Zulässig sind folgende Bausteinnummern:

Baustein-Typ	Bausteinnummer
DB (OB 255)	3 bis 255
DX (OB 254)	3 bis 255

2. AKKU-1-LH

AKKU-1-LH muß Null sein.

Duplizieren

Funktion

Ein Datenbaustein wird innerhalb des Anwenderspeichers dupliziert, wobei er eine andere Bausteinnummer erhält. Die Anfangsadresse des neuen Datenbausteins wird in die Adreßliste im DB 0 eingetragen. Die Anfangsadresse des alten Bausteins im DB 0 bleibt erhalten, d.h. der ursprüngliche Datenbaustein ist weiterhin gültig.

Die Eintragung der Anfangsadresse in den DB 0 erfolgt erst dann, wenn die Übertragung vollständig abgeschlossen ist und alle Kennungen im Bausteinkopf richtig eingetragen sind. Der duplizierte Baustein wird also vom Systemprogramm erst nach der kompletten Übertragung als gültig bzw. vorhanden erkannt.

Parameter

1. AKKU-1-LL

Nummer des zu duplizierenden Bausteins (Quelle).

2. AKKU-1-LH

Nummer des neuen Bausteins (Ziel).

Zulässig sind folgende Bausteinnummern:

Baustein-Typ	Bausteinnummer
DB (OB 255)	3 bis 255
DX (OB 254)	3 bis 255

Ergebnis nach dem Kopieren und Duplizieren

Nach korrekter und fehlerfreier Bearbeitung setzt das Systemprogramm das VKE auf '0' und löscht die Anzeigen ANZ 1 und ANZ 0.

Durch Aufruf des OB 254/255 werden die Inhalte von AKKU 1 bis AKKU 4 verändert. Das BR-Register bleibt erhalten.

Fehlerfälle und Warnungen beim Kopieren und Duplizieren

Im Fehlerfall oder bei einer Warnung bricht das Systemprogramm die Bearbeitung des OB 254/255 ab und setzt die Programmbearbeitung mit der nächsten STEP-5-Operation fort. Ferner setzt es das VKE auf '1' und hinterlegt im AKKU-1-LL eine Kennung (siehe Tabelle 6-18).

Bei Abbruch der Funktion mit einer Warnung kann u. U. durch einen erneuten Aufruf der Sonderfunktion (evtl. mehrmals wiederholen) eine korrekte Bearbeitung des OB 254/255 erreicht werden.

Im folgenden Fall wird der OB 254/255 mit einer **Warnung** abgebrochen:

Während der letzten 10 ms ist bereits ein OB 124, OB 125, OB 254 oder OB 255 aufgerufen worden. (Innerhalb 10 ms ist jeweils nur ein einziger Aufruf dieser Sonderfunktionen zugelassen. Dadurch wird verhindert, daß durch mehrfachen Aufruf der o. g. OBs die Schnittstelle zum PG nicht mehr bearbeitet wird.)

Anzeigen nach dem Kopieren und Duplizieren

Nach Aufruf des OB 254/255 können Sie an Hand des Verknüpfungsergebnisses und der Anzeigen ANZ 1 und ANZ 0 feststellen, ob die Sonderfunktion ordnungsgemäß abgelaufen ist oder mit "Fehler" oder "Warnung" abgebrochen wurde. Das Ergebnis kann mit bedingten Sprungbefehlen ausgewertet werden.

Ergebnisanzeigen

Tabelle 6-20 Ergebnisanzeigen des OB 254/255

VKE	ANZ 1	ANZ 0	Bedeutung	Abfrage
0	0	0	Sonderfunktion wurde korrekt bearbeitet	SPB SPZ
1	1	0	Bearbeitung der Sonderfunktion wurde mit "Warnung" abgebrochen	SPB SPP SPN
1	0	1	Bearbeitung der Sonderfunktion wurde mit "Fehler" abgebrochen	SPB SPM SPN

Kennungen im AKKU-1LL

Im AKKU-1-LL werden vom Systemprogramm Kennungen über das Bearbeitungsergebnis abgelegt, mit denen Ursachen, die zu einer Warnung oder einen Fehler führten, näher spezifiziert sind.

Bit-Nr.	7	6	5	0
	W	F	Ursache für Fehler/Warnung	

Dabei sind folgende Sammelanzeigen festgelegt:

Bit-Nr. 7 (W) = 1: Warnung
 Bit-Nr. 6 (F) = 1: Fehler

Tabelle 6-21 Ergebniskennungen des OB 254/255 im AKKU-1-LL

Kennung	Bedeutung
01H	Funktion wurde korrekt bearbeitet
41H	Fehler: Bausteinkopf auf der Memory Card ungültig
43H	Speicherplatz reicht nicht aus
48H	Quelldatenbaustein nicht vorhanden
4AH	Bausteinnummer oder -Typ unzulässig/Quell-DB
4BH	Bausteinnummer oder -Typ unzulässig/Ziel-DB
4CH	Zieldatenbaustein ist im Anwenderspeicher bereits vorhanden
4DH	Online-Funkt. SPEICHER KOMPRIMIEREN tätig
4EH	es ist keine Memory Card gesteckt
8DH	Warnungen: Konflikt mit einer Online-Funktion außer "Speicher komprimieren")
8EH	10 ms Wartezeit noch nicht abgelaufen

Beispiele

1. "Kopieren":

```
:L   KY 0,120  Diese Befehlsfolge kopiert den
:SPA OB 254   Datenbaustein DX 120 von der
:              Memory Card in den Anwenderspeicher.
:
```

2. "Duplizieren":

```
:L   KY 80,85  Diese Befehlsfolge dupliziert den
:SPA OB 255   Datenbaustein DB 85; der neue
:              Datenbaustein hat die Nummer 80.
:              Die Inhalte von DB 80 und DB 85
:              sind identisch.
```

Erweiterter Datenbaustein DX 0

7

Inhalt von Kapitel 7

7.1	Anwendung.....	7 - 4
7.2	Aufbau des DX 0	7 - 5
7.2.1	Beispiel für Eingabe des DX 0.....	7 - 7
7.3	Parameter für DX 0	7 - 8
7.4	Parametrierungsbeispiele	7 - 12
7.4.1	STEP-5-Programmierung	7 - 12
7.4.2	Parametrierung über PG-Maske.....	7 - 14

Erweiterter Datenbaustein DX 0

7

Im nachfolgenden Kapitel erfahren Sie, wofür Sie den Datenbaustein DX 0 einsetzen können und wie er aufgebaut ist. Sie werden informiert, welche Bedeutung die verschiedenen DX-0-Parameter haben, und Sie lernen an Hand von Beispielen, wie Sie einen Datenbaustein DX 0 erstellen bzw. über eine Maske parametrieren können

7.1 Anwendung

Als Anwender können Sie bestimmte Leistungen des Systemprogramms Ihren Erfordernissen anpassen, wenn Sie im DX 0 alternativ zu den Standard-Voreinstellungen (in der Parameter-Tabelle mit "V" gekennzeichnet) durch Parametereingabe andere Einstellungen vornehmen.

Die Voreinstellungen des Systemprogramms (V) werden automatisch bei jedem NEUSTART gesetzt. Danach wird der DX 0 ausgewertet. Haben Sie den Baustein DX 0 nicht parametrisiert und geladen, so gelten weiterhin die Voreinstellungen; ansonsten werden die Einstellungen gültig, mit denen Sie den DX 0 parametrisiert haben.

Sie können die Einstellungen im DX 0 vornehmen, indem Sie die Werte wie bei einem anderen Datenbaustein programmieren (siehe Abschnitte 7.2 bis 7.4.1) oder – mit PG-Systemsoftware S5-DOS ab Version 3.0 – die Werte über eine spezielle Maske an Ihrem PG als Parameter eingeben (siehe Abschnitt 7.4.2).

Den vollen Funktionsumfang der DX-0-Maske können Sie benutzen, wenn Sie die PG-Software STEP 5/ST, Vers. 6.3 bzw. STEP 5/MT, Vers. 6.0 + die jeweilige "Delta-Diskette CPU 948" auf Ihrem PG installiert haben.

Hinweis

Eine Eingabe oder eine Änderung des DX 0 wird nur über einen NEUSTART wirksam.

Wird bei NEUSTART ein **geänderter DX 0** übernommen, so bleiben die **nicht geänderten** Parametrierungen **erhalten!**

Unterschiede zur CPU 946/947

Gegenüber der DX-0-Parametrierung für die CPU 946/947 gibt es bei den DX-0-Parametern für die CPU 948 folgende Unterschiede:

- Betriebsarten:
Es gibt keinen Modus 150U/155U mehr, sondern stattdessen die "**Unterbrechbarkeit an Bausteingrenzen**" und die "**Unterbrechbarkeit an Befehls Grenzen**".
- Bearbeitung der System-Interrupts:
Bei der CPU 948 können Sie jetzt auch "Prozeßalarme über EB 0 = aus" (= Systeminterrupt-Bearbeitung) mit "**Unterbrechbarkeit an Bausteingrenzen**" kombinieren.
Damit ist jetzt auch in der Betriebsart "Unterbrechbarkeit an Bausteingrenzen" Mehrprozessorbetrieb möglich.
- Zusätzliche Weckalarme:
Wenn Sie die Betriebsart "Prozeßalarme über EB 0" ausgeschaltet haben, stehen Ihnen bei den Weckalarmen zusätzlich der **Verzögerungsalarm (OB 6)** und der **uhrzeitgesteuerte Weckalarm (OB 9)** zur Verfügung.

7.2 Aufbau des DX 0

Der DX 0 setzt sich aus drei Teilen zusammen:

- der Anfangskennung für den DX 0 (DW 0, 1 und 2) ,
 - mehreren Blöcken unterschiedlicher Länge (je nach Parameteranzahl)
- und
- der Endekennung.

Anfangskennung

ASCII-Zeichen MASKX0 in DW 0 bis DW 2

Block

Ein **Block** im DX 0 besteht aus 1 bis n Datenwörtern. Diese enthalten:

- die Blockkennung,
 - die Blocklänge
- und
- die Blockparameter.

Blockkennung

Die **Blockkennung** gibt an, welche Bedeutung die darauffolgenden Parameter haben. Jeder Block ist einem bestimmten Systemprogrammteil oder einer bestimmten Systemfunktion zugeordnet (z.B. kennzeichnet die Blockkennung '04' den Parameterblock für zyklische Programmbearbeitung).

Blocklänge

Die **Blocklänge** gibt an, wieviele Datenwörter die nachfolgenden Parameter belegen.

Parameter

Die möglichen **Parameter** finden Sie in Abschnitt 7.3. Die angegebenen Zahlenwerte sind im Hexadezimalformat (KH).

Endekennung

Sie kennzeichnet das Ende des DX 0 durch EEEEH im letzten Datenwort.

Formaler Aufbau

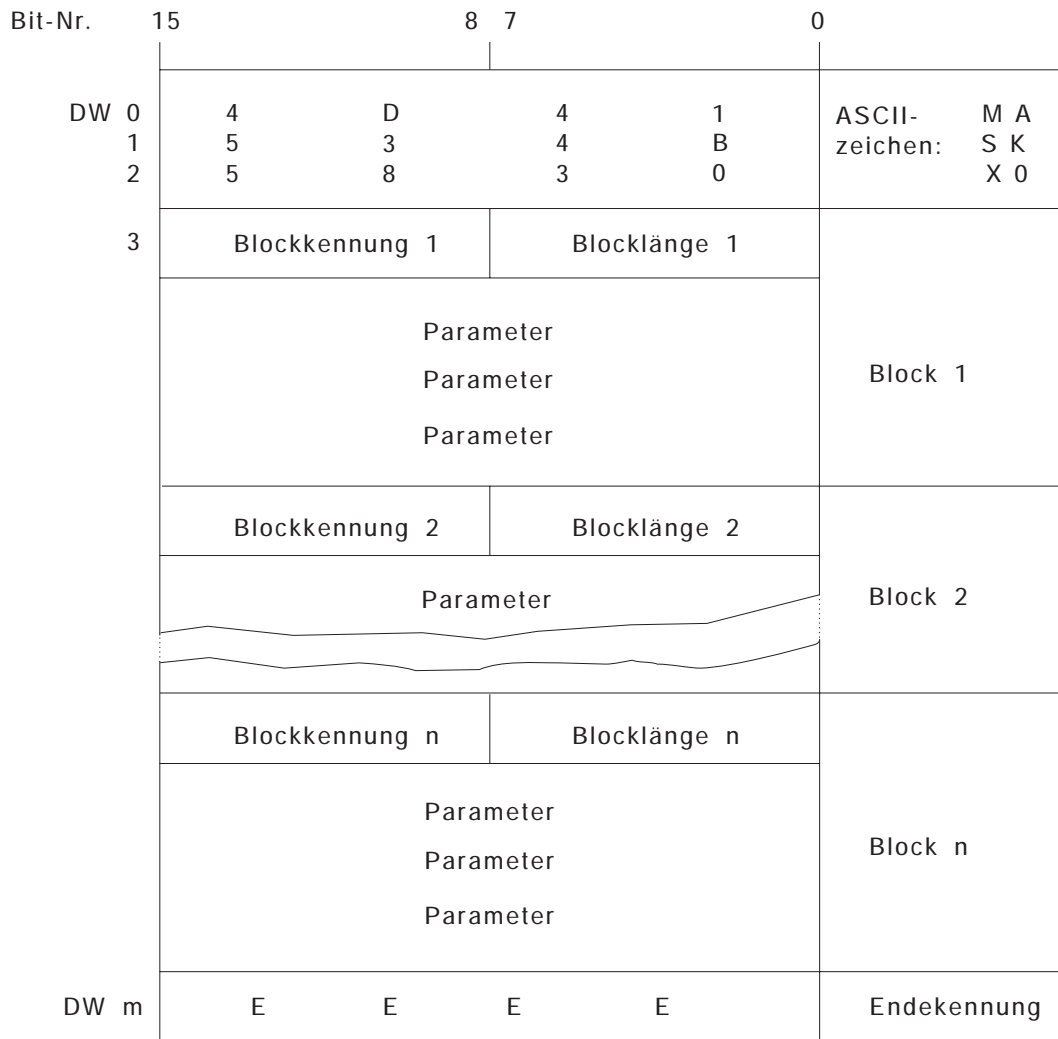


Bild 7-1 Aufbau des DX 0

7.2.1 Beispiel für Eingabe des DX 0

Anfangskennung	DW 0:	KH = 4D41	
	DW 1:	KH = 534B	
	DW 2:	KH = 5830	
Blockkennung/-länge Parameter (belegt 1 DW)	DW 3:	KH = 0101	Block 1
	DW 4:	KH = 1001	
Blockkennung/-länge Parameter (belegt 2 DW)	DW 5:	KH = 0402	Block 2
	DW 6:	KH = 1000	
	DW 7:	KH = 0040	
Endekennung	DW10:	KH = EEEE	

Beachten Sie bei der Parametrierung des DX 0 folgende Punkte:

- Nicht benötigte Blöcke brauchen Sie nicht anzugeben.
- Halten Sie die Reihenfolge der Blöcke ein (geben Sie z. B. den Block mit der Kennung '02' **vor** dem Block mit der Kennung '05' ein).
- Ein bestimmter Block darf nur **einmal** im DX 0 vorkommen.
- Die Anzahl der Parameter muß der am Blockanfang angegebenen Blocklänge entsprechen.
- Halten Sie die Reihenfolge der Parameter ein. Nicht benötigte Parameter eines Blockes, die zum **Blockanfang** hin liegen, müssen Sie wegen der einzuhaltenden Reihenfolge mit der Vorbesetzung notieren.
- Schließen Sie den DX 0 nach Eingabe des letzten Blocks unbedingt mit der Endekennung "KH=EEEE" ab.

7.3 Parameter für DX 0

Tabelle 7-1 DX-0-Parameter und ihre Bedeutung

Block- kennung/-länge	Parameter 1. Wort/2. Wort	Bedeutung ¹⁾	
Betriebsarten			
01xx ²⁾	1000	V Unterbrechung an Bausteingrenzen ³⁾ (CPU 946/947: 150U-Mode)	
	1001	V Unterbrechung an Befehlsgrenzen ⁴⁾ (CPU 946/947: 155U-Mode)	
Anlaufprogrammbearbeitung			
02xx	1000	V AUTOMATISCHER WIEDERANLAUF nach NETZ EIN AUTOMATISCHER NEUSTART nach NETZ EIN MANUELLER NEUSTART/WIEDERANLAUF nach NETZ EIN	
	1001		
	1002		
	2000	V Synchronisation des ANLAUFs im Mehrprozessorbetrieb	
	2001	V keine Synchronisation des ANLAUFs im Mehrprozessorbetrieb	
	BB00	00yy ⁵⁾	Anzahl der zu aktualisierenden Zeitzellen yy: ⁶⁾ zulässige Werte: 00H bis FFH V yy = FF (Zeitzelle T 0 bis T 255)
	4000		V Wiederanlaufart = WIEDERANLAUF
	4001		V Wiederanlaufart = NEUSTART MIT GEDÄCHTNIS
Zyklische Programmbearbeitung			
04xx	1000	00yy	Einstellen der Zyklusüberwachungszeit: ⁷⁾ Zyklusüberwachungszeit = (yy * 10 ms) zulässige Werte: 01H bis FFH V yy = 14H (200 ms)
	4000		V Aktualisierung des Prozeßabbildes und der Koppelmerker ohne Semaphorschutz
	4001		V Aktualisierung des Prozeßabbildes und der Koppelmerker mit Semaphorschutz (im Block, siehe Abschnitt 10.1.3)
Alarmbearbeitung: Weckalarme			
05xx	1000	000c	V Weckalarmbearbeitung "ein"
	1001	0000	V Weckalarmbearbeitung "aus" c = Ebenenpriorität, zulässige Werte: 1 bis 5 V c = 1 (höchste Ebenenpriorität)

Block- kennung/-länge	Parameter 1. Wort/2. Wort	Bedeutung ¹⁾	
Fortsetzung der Tabelle 7-1:			
05xx	Alarmbearbeitung: Weckalarme (Forts.)		
	2000 00yy	Grundtakt für die Weckalarmbearbeitung: Grundtakt = (yy * 10 ms) zulässige Werte: 01H bis FFH V yy = 0A (100 ms)	
	3000 3001	V Zeittaktverteilung nach Zeitraster 1 (1, 2, 5, 10 ...) Zeittaktverteilung nach Zeitraster 2 (2 ⁿ) (siehe Abschnitt 4.5.2)	
	Alarmbearbeitung: Prozeßalarme über S5-Bus/Systeminterrupts		
	4000 000c 4001 0002	Systeminterrupt X "ein" V Systeminterrupt X "aus"	c = Ebenenpriorität, zulässige Werte: 1 bis 5 X = A, B, C oder D V c = 2 (Ebenenpriorität 2)
	5000 000c 5001 0002	Systeminterrupt E "ein" V Systeminterrupt E "aus"	
	6000 000c 6001 0002	Systeminterrupt F "ein" V Systeminterrupt F "aus"	(Die Bearbeitung der System- interrupts kann bei CPU 948 auch mit "Unterbrechbarkeit an Bausteingrenzen" kombiniert werden.)
	7000 000c 7001 0002	Systeminterrupt G "ein" V Systeminterrupt G "aus"	
	8000 000c 8001 0000	V Prozeßalarme über EB 0 "ein" Prozeßalarme über EB 0 "aus" c = Ebenenpriorität, zulässige Werte 1 bis 2 V c = 2 (Ebenenpriorität 2) Wenn "Prozeßalarme über EB 0 = ein": - nur Einzelprozessorbetrieb, - nur "Unterbrechbarkeit an Bausteingrenzen".	
	EEEE	Endekennung	

¹⁾ V = Voreinstellung bei nicht geladenem DX 0 bzw. fehlendem Block

²⁾ xx = Blocklänge (Anzahl der von den Parametern belegten Datenwörtern)

³⁾ Kann bei CPU 948 auch mit Systeminterrupt kombiniert werden.

⁴⁾ Darf nicht mit Prozeßalarmen über EB 0 kombiniert werden.

⁵⁾ Parameter, die zwei Datenwörter belegen, müssen bei Angabe der Blocklänge mit dem Wert '2' berücksichtigt werden.

⁶⁾ Zur Aktualisierung der Zeitzellen lesen Sie bitte die Erklärung auf der nächsten Seite.

⁷⁾ Zyklusüberwachungszeit mit OB 31 oder DX 0 einstellen:
Wenn die Zyklusüberwachungszeit sowohl mit OB 31 als auch mit DX 0 eingestellt wird, so übernimmt das Systemprogramm die Einstellung durch den OB 31, weil es den DX 0 schon früher ausgewertet hat. Aus diesem Grunde sollten Sie nicht beide Möglichkeiten gleichzeitig benutzen. Wir empfehlen Ihnen, die Zyklusüberwachungszeit nur über DX 0 einzustellen.

Aktualisierung der Zeitzellen

- Standardmäßig werden die Zeitzellen T 0 bis T 255 aktualisiert.
- Wenn Sie im DX 0 die Anzahl '0' eintragen, werden **keine** Zeitzellen aktualisiert, auch dann nicht, wenn Sie im Programm enthalten sind. Es gibt dann auch keine Fehlermeldung.
- Es wird immer wie folgt aktualisiert:

Eintrag	'0'	'1'	'2'	'3'	'4'
Aktualisierung	keine	T0 bis T1	T0 bis T2	T0 bis T3	T0 bis T4

Ebenenprioritäten

Für Programmbearbeitungsebenen können unterschiedliche Prioritäten festgelegt werden. Dies geschieht entweder über die Voreinstellung oder Vorgabe über DX-0-Parameter.

Prioritäten bei der Parametrierung "Prozeßalarme über EB 0 = ein" (Ebene PROZESSALARME)

Diese Prioritäten sind im DX 0 voreingestellt mit folgenden Werten:

- Weckalarme: Ebenenpriorität 1 (höhere Priorität)
- Prozeßalarme über Eingangsbyte EB 0: Ebenenpriorität 2 (niedrigere Priorität)

Sie können die beiden Prioritäten im DX 0 vertauschen.

Prioritäten bei der Parametrierung "Prozeßalarme über EB 0 = aus" (= Bearbeitung der Systeminterrupts – Ebene INTERRUPTS)

In dieser Betriebsart sind durch Voreinstellung folgende Prioritäten vergeben:

- Weckalarme: Ebenenpriorität 1 (höhere Priorität)
- Systeminterrupts: Ebenenpriorität 2 (niedrigere Priorität)

Sie können diese Prioritäten für folgende Programmbearbeitungsebenen einzeln im DX 0 ändern, und zwar durch Vorgabe der Prioritätswerte von '1' bis '5' (der Wert '1' bedeutet die höchste Priorität):

- Weckalarne,
- Systeminterrupt INT X (X = A, B, C oder D),
- Systeminterrupt INT E,
- Systeminterrupt INT F,
- Systeminterrupt INT G.

Beispiel

Zuteilung der Prioritäten bei der Alarmbearbeitung
"Systeminterrupts" :

Systeminterrupt INT A/B/C/D	Ebenenpriorität 1	↓ fallende Priorität ↓
Weckalarne	Ebenenpriorität 2	
Systeminterrupt INT E	Ebenenpriorität 3	
Systeminterrupt INT F	Ebenenpriorität 4	
Systeminterrupt INT G	Ebenenpriorität 5	

7.4 Parametrierungsbeispiele

7.4.1

STEP-5-Programmierung

Beispiel A:

Sie wollen im Mehrprozessorbetrieb drei CPUs einsetzen: CPU A, B und C. CPU A und B arbeiten eng miteinander zusammen, tauschen häufig Daten aus und bearbeiten ein umfangreiches Anlaufprogramm.

CPU C bearbeitet weitgehend unabhängig davon ein kurzes, zeitkritisches Programm.

Standardmäßig beginnen im Mehrprozessorbetrieb alle CPUs gemeinsam mit der zyklischen Programmbearbeitung, d. h. die CPUs warten solange aufeinander, bis alle ihren Anlauf beendet haben, und gehen dann gemeinsam in die zyklische Programmbearbeitung.

Da CPU C ihr Programm unabhängig von den anderen CPUs ausführt und ein **sehr kurzes Anlaufprogramm** bearbeitet, ist bei ihr keine Synchronisation des Anlaufs notwendig. Durch Parametrierung des DX 0 erreichen Sie, daß CPU C nach beendetem Anlauf sofort in die zyklische Programmbearbeitung geht, ohne auf CPU A und B zu warten.

Der Parameter für die Synchronisation der CPUs im Mehrprozessorbetrieb ist der 2. Parameter im 1. Block. Um die Reihenfolge der Parameter einzuhalten, muß daher auch der 1. Parameter für das Anlaufverhalten mit der Vorbesetzung (AUTOMATISCHER WIEDER-ANLAUF nach NETZ EIN) notiert werden.

Programmieren Sie den DX 0 für CPU C:

DX 0	Anfangskennung "MASKX0"	DW 0:	KH = 4D41
		DW 1:	KH = 534B
		DW 2:	KH = 5830
	1. Blockkennung/-länge	DW 3:	KH = 0202
	Parameter 1	DW 4:	KH = 1000
	Parameter 2	DW 4:	KH = 2001
	Endekennung	DW 5:	KH = EEEE

Haben Sie diesen DX 0 in den Programmspeicher geladen, wird er mit dem nächsten NEUSTART wirksam. Da CPU C ein sehr kurzes Anlaufprogramm bearbeitet und nicht auf A und B wartet, geht bei ihr sofort nach dem Anlauf die grüne RUN-LED an. Das BASP-Signal (Befehlsausgabesperre) wird jedoch erst inaktiv geschaltet, wenn alle drei CPUs ihren Anlauf beendet haben. Dies bedeutet, daß CPU C nicht auf die Digitalperipherie zugreifen darf.

Beispiel B:

Mit der folgenden Parametrierung des DX 0 wird

- die Betriebsart "Unterbrechung an Befehlsgrenzen" eingestellt,
 - die Zeitzellenaktualisierung abgeschaltet,
 - die Zykluszeit auf 2,5 s eingestellt,
 - die Ebenenpriorität der Weckalarme auf '2' eingestellt
- und
- der Systeminterrupt INT E mit der Ebenenpriorität '1' eingeschaltet.

DX 0	Anfangskennung "MASKX0"	DW 0:	KH = 4D41
		DW 1:	KH = 534B
		DW 2:	KH = 5830
	1. Blockkennung/-länge Parameter	DW 3:	KH = 0101
	2. Blockkennung/-länge Parameter ¹⁾	DW 4:	KH = 1001
		DW 5:	KH = 0202
		DW 6:	KH = BB00
		DW 7:	KH = 0000
	3. Blockkennung/-länge Parameter ¹⁾	DW 8:	KH = 0402
		DW 9:	KH = 1000
		DW10:	KH = 00FA
	4. Blockkennung/-länge Parameter ¹⁾	DW11:	KH = 0504
		DW12:	KH = 1000
	Parameter ¹⁾	DW13:	KH = 0002
		DW14:	KH = 5000
		DW15:	KH = 0001
	Endekennung	DW16:	KH = EEEE

Diese Parametrierung des DX 0 hat folgende Auswirkungen auf die Programmbearbeitung:

Die Unterbrechung der Programmbearbeitung durch höherprioritäre Ebenen erfolgt statt an Bausteingrenzen an Befehlsgrenzen.

Die Laufzeit des Systemprogramms verkürzt sich geringfügig, da keine Zeitzellen aktualisiert werden.

Ein Zyklusfehler wird erst dann erkannt, wenn die Laufzeit des Anwenderprogramms und des Systemprogramms zusammen 2,5 s übersteigt.

Es werden keine Prozeßalarme von Eingangsbyte EB 0 bearbeitet, sondern der Systeminterrupt INT E. Dieser unterbricht wegen seiner höheren Priorität die Weckalarmbearbeitung, die Bearbeitung des Verzögerungsalarms sowie die Bearbeitung eines Zeitauftrags.

¹⁾ Unter "Blocklänge" geben Sie die Anzahl Datenwörter an, die ein Parameter belegt.

**7.4.2
Parametrierung über
PG-Maske**

Für die Parametrierung des DX 0 für die CPU 948 stehen Ihnen bei der PG-Systemsoftware Masken zur Verfügung. Die PG-Software generiert entsprechend den Parametervorbesetzungen (fettgedruckte Werte) und den von Ihnen eingegebenen Parametern automatisch den Datenbaustein DX 0. Für die Parametrierung werden zwei Masken benötigt.

Die grundsätzlichen Bedienungen zur Anwahl und zum Ausfüllen von PG-Masken entnehmen Sie bitte Ihrem STEP-5-Handbuch.

Aufbau der Masken

Die PG-Maske zum Ausfüllen des DX 0 besteht aus zwei Teilmasken:

Die erste Teilmaske (Bild 7-2) enthält die Parametergruppen

- Unterbrechbarkeit,
- Anlaufart nach "NETZ EIN",
- Wiederanlaufverhalten,
- Anzahl der Zeitzellen,
- Zykluszeitüberwachung,
- Mehrprozessoranlauf Synchronisieren,
- Blockübertragung der Koppelmerker.

DX0 - Parametrierung (AG 155U CPU 948)		DX 0
Unterbrechbarkeit	an Bausteingrenzen ¹⁾	
Anlauf nach "NETZ EIN" :	1 (1 = Wiederanlauf 2 = Neustart 3 = Manueller Start)	
Wiederanlaufverhalten :	1 (1 = Wiederanlauf 2 = Neustart mit Gedächtnis)	
Anzahl der Zeitzellen :	256 (0 . . . 256)	
Zykluszeitüberwachung (X 10 MS) :	20 (1 . . . 255)	
Mehrprozessoranlauf Synchronisieren :	JA	
Blockübertragung der Koppelmerker :	NEIN	
F 1	F 2	F 3 Wählen
F 4	F 5	F 6 Weiter
F 7	F 8	

¹⁾ bei älteren PG-Software-Versionen wird angezeigt für:
 Unterbrechbarkeit an Bausteingrenzen: **BETRIEBSART 150U**
 Unterbrechbarkeit an Befehlsgrenzen: **BETRIEBSART 155U**

Bild 7-2 PG-Maske zum Parametrieren des DX 0 / Teil 1

Weitere Parameter finden Sie in der zweiten Teilmaske (Bild 7-3):

- Zeitalarme,
- Hardwareprozessalarne,
- Prozessalarne EingangsByte 0.

DX0 - Parametrierung (AG 155U CPU 948)		DX 0
Zeitalarme : ¹⁾		
Zeitalarbearbeitung :	JA	Prioritaet : 1
Grundtakt (X 10 MS) :	10 (1 . . . 255)	
Zeittaktverarbeitung :	1 (1 = Faktor 1, 2, 5, 10 2 = Faktor 1, 2, 4, 8)	
Hardwareprozessalarne : ²⁾		
Systeminterrupt A/B :	NEIN	Prioritaet : 2
Systeminterrupt E :	NEIN	Prioritaet : 2
Systeminterrupt F :	NEIN	Prioritaet : 2
Systeminterrupt G :	NEIN	Prioritaet : 2
Prozessalarne EingangsByte 0 (nur bei Unterbrechbarkeit an Bausteingrenzen)		
Prozessalarne :	JA	Prioritaet : 2
F 1	F 2	F 3
1	2	3
F 4	F 5	F 6
4	5	6
F 7	F 8	
7	8	
Wahlen		Weiter

1) Verzögerungsalarm und uhrzeitgesteuerter Weckalarm müssen bei Bedarf zusätzlich aktiviert werden durch Abschalten der Prozessalarne (Einschalten der Interruptbearbeitung)

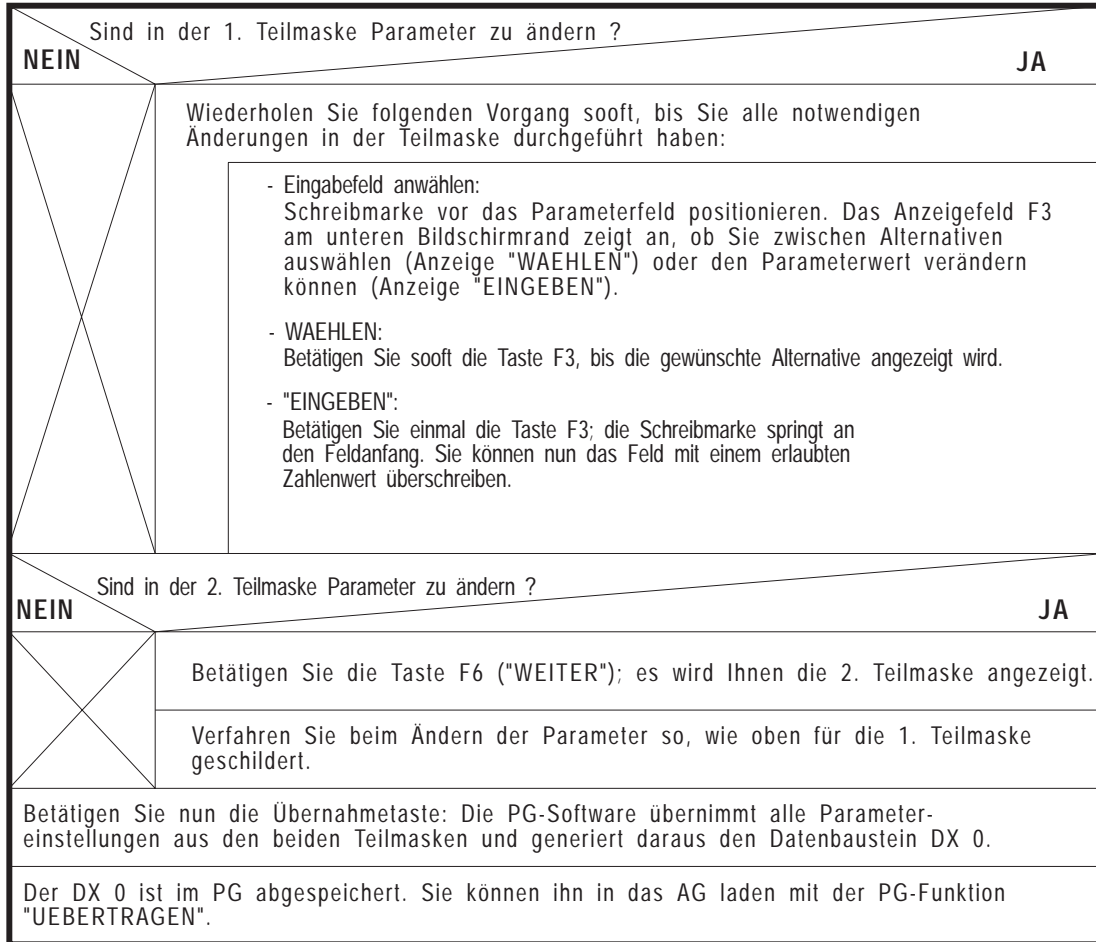
2) CPU 948: Systeminterrupts können nur bearbeitet werden bei "Unterbrechbarkeit an Bausteingrenzen" oder "Unterbrechbarkeit an Befehlsgrenzen"

Bild 7-3 PG-Maske zum Parametrieren des DX 0 / Teil 2

Hantierung

Das nachfolgende Ablaufdiagramm erläutert Ihnen, wie Sie die Teilmasken und Maskenfelder ausfüllen, die Parameter abspeichern und den erzeugten Datenbaustein DX 0 laden können.

Ablaufdiagramm zum Ausfüllen der DX-0-Masken



Beispiel

Sie wollen den DX 0 für folgendes Verhalten des Systemprogramms abweichend von den Vorbesetzungen parametrieren:

- Betriebsart "Unterbrechung an Befehlsgrenzen",
- keine Zeitzellenaktualisierung
- Zyklusüberwachungszeit = 2,5 s,
- Ebenenpriorität für Zeitalarme = 2
- Systeminterrupt INT E mit Priorität = 1 .

Fortsetzung auf der nächsten Seite

Fortsetzung des Beispiels

Füllen Sie für diese Wünsche die Maske folgendermaßen aus:

Erste DX-0-Maske:

- Wählen Sie beim Parameter BETRIEBSART mit der Funktionstaste F3 die Einstellung "Unterbrechbarkeit an Befehls Grenzen" an.
- Betätigen Sie beim Parameter ANZAHL ZEITZELLEN zunächst die Funktionstaste F3 und geben Sie die Ziffer 0 (= **keine** Zeitzelle) ein
- Betätigen Sie beim Parameter ZYKLUSUEBERWACHUNG zunächst die Funktionstaste F3 und geben Sie die Ziffern 250 (= 2,5 s) ein.
- Betätigen Sie die Funktionstaste F6 (WEITER). Es wird danach die zweite DX-0-Maske vorgelegt.

Zweite DX-0-Maske:

- Wählen Sie bei Parameter ZEITALARMBEARBEITUNG/PRIORITAET mit der Funktionstaste F3 den Wert '2' an.
- Wählen Sie bei Parameter SYSTEMINTERRUPT E mit der Funktionstaste F3 die Einstellung 'JA' an.
- Wählen Sie bei Parameter SYSTEMINTERRUPT E/PRIORITAET mit der Funktionstaste F3 den Wert '1' an.
- Wählen Sie beim Parameter PROZESSALARME mit der Funktionstaste F3 die Einstellung 'NEIN' an.
- Bestätigen Sie mit der Übernahmetaste die gesamten Eingaben. Der Datenbaustein DX 0 wird nun von der Systemsoftware generiert.

Speicherbelegung und Speicherorganisation

8

Inhalt von Kapitel 8

8.1	Struktur des Speicherbereiches	8 - 4
8.2	Adreßbraumaufteilung der CPU 948	8 - 5
8.2.1	Adreßbraumaufteilung des System-RAMs	8 - 6
8.2.2	Adreßbraumaufteilung der Peripherie	8 - 8
8.3	Organisation des Anwenderspeichers in der CPU 948	8 - 10
8.3.1	Bausteinköpfe im Anwenderspeicher	8 - 12
8.3.2	Bausteinadreßlisten im Datenbaustein DB 0	8 - 13
8.3.3	BA-/BB-Bereich	8 - 14
8.3.4	BS-/BT-Bereich	8 - 15
8.3.5	Bitbelegung der Systemdatenwörter	8 - 18
8.3.6	Frei adressierbarer Systemdatenbereich	8 - 42

Speicherbelegung und Speicherorganisation

8

In diesem Kapitel können Sie bei Bedarf nachschlagen, wie der Speicher der CPU 948 organisiert ist. Sie finden ferner wichtige und für den Anwender zugängliche Informationen, die in einigen Systemdatenwörtern hinterlegt sind.

8.1 Struktur des Speicherbereiches

Der Speicherbereich in der CPU 948 enthält im wesentlichen folgende Bereiche:

Tabelle 8-1 Struktur des Speicherbereiches

Speicherbereich		Datenbreite	Lage
Anwenderspeicher für:	OB, FB, FX, PB, SB, DB, DX	16 bit	CPU- intern
Bereich Anschaltung:	BA, BB	16 bit	
Bereich System:	BS, BT	16 bit	
Zähler:	Z	16 bit	
Zeiten:	T	16 bit	
Merker:	M	8 bit	
Merker:	S	8 bit ¹⁾	
Prozeßabbild (PA) der Ein- und Ausgänge:	PAE, PAA	8 bit	
Peripheriebereich, unterteilt in:			auf
	P-Peripherie	8 bit	S5-Bus
	Q-Peripherie	8 bit	
	Koppelmerker	8 bit	
	KOR (Semaphor, ...)	8 bit	
	Kacheln (CP, IP, KOR 923C)	8/16 bit	
	Dezentrale Peripherie	8 bit	
	HW-Register	8/16 bit	

¹⁾ S-Merker liegen mit einer Datenbreite von 8 bit im 16-bit-Bereich. Das High-Byte ist undefiniert.

Die genauen Adressen dieser Bereiche entnehmen Sie den Speicherbelegungsplänen im nächsten Abschnitt.

Hinweis

Der STEP-5-Zugriff auf eine Speicherzelle innerhalb eines Operandenbereiches (z.B. Merker) sollte nie direkt über die absolute Adresse dieser Speicherzelle erfolgen, sondern ausschließlich relativ zur Basisadresse des jeweiligen Operandenbereichs.

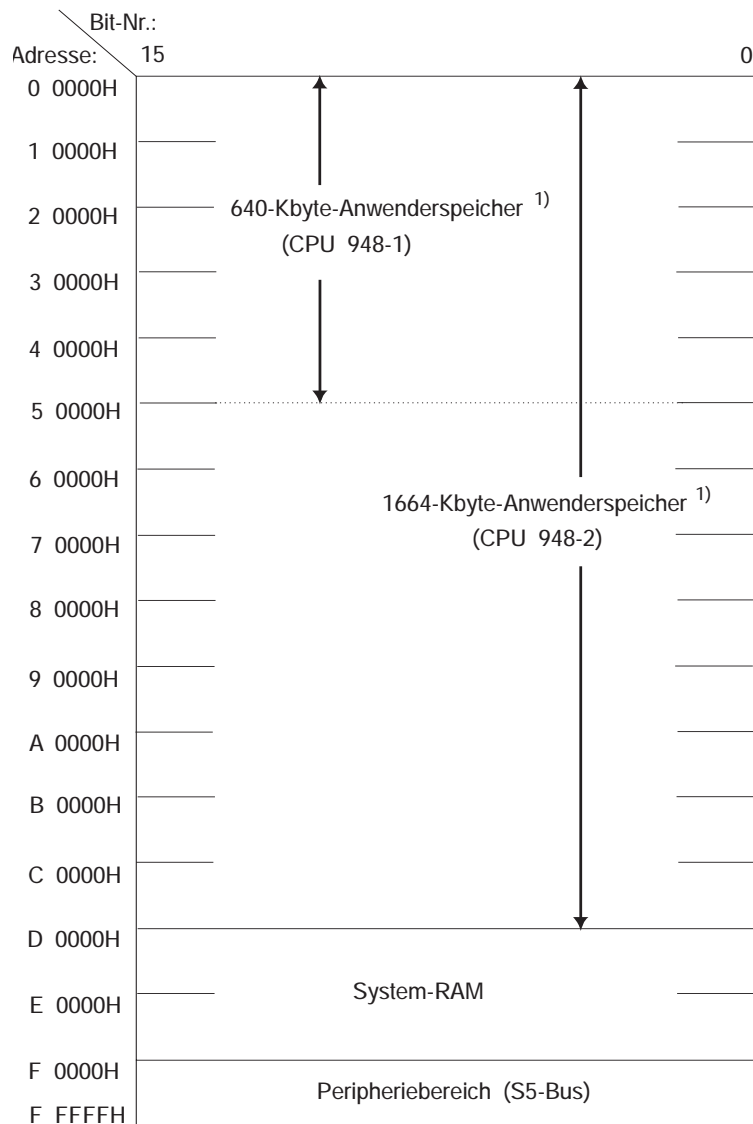
Direkte Zugriffe auf die Bereiche E, A und M liefern im High-Byte 'FFH', im Low-Byte das Datum. Bei Direktzugriffen auf S-Merker ist das **High-Byte undefiniert!**

8.2 Adreßraumaufteilung der CPU 948

Die CPU 948 ist in zwei Versionen des Anwenderspeichers (RAM) lieferbar:

- die CPU 948-1 mit 640 Kbyte Anwenderspeicher,
- die CPU 948-2 mit 1664 Kbyte Anwenderspeicher.

In Bild 8-1 ist die Adreßraumaufteilung der CPU 948 mit der Lage der Anwenderspeicher-Versionen dargestellt.



¹⁾ Die letzten 20 Wörter des Anwenderspeichers sind nicht nutzbar.

Bild 8-1 Adreßraumaufteilung der CPU 948/Übersicht

**8.2.1
Adreßraumteilung des
System-RAMs**

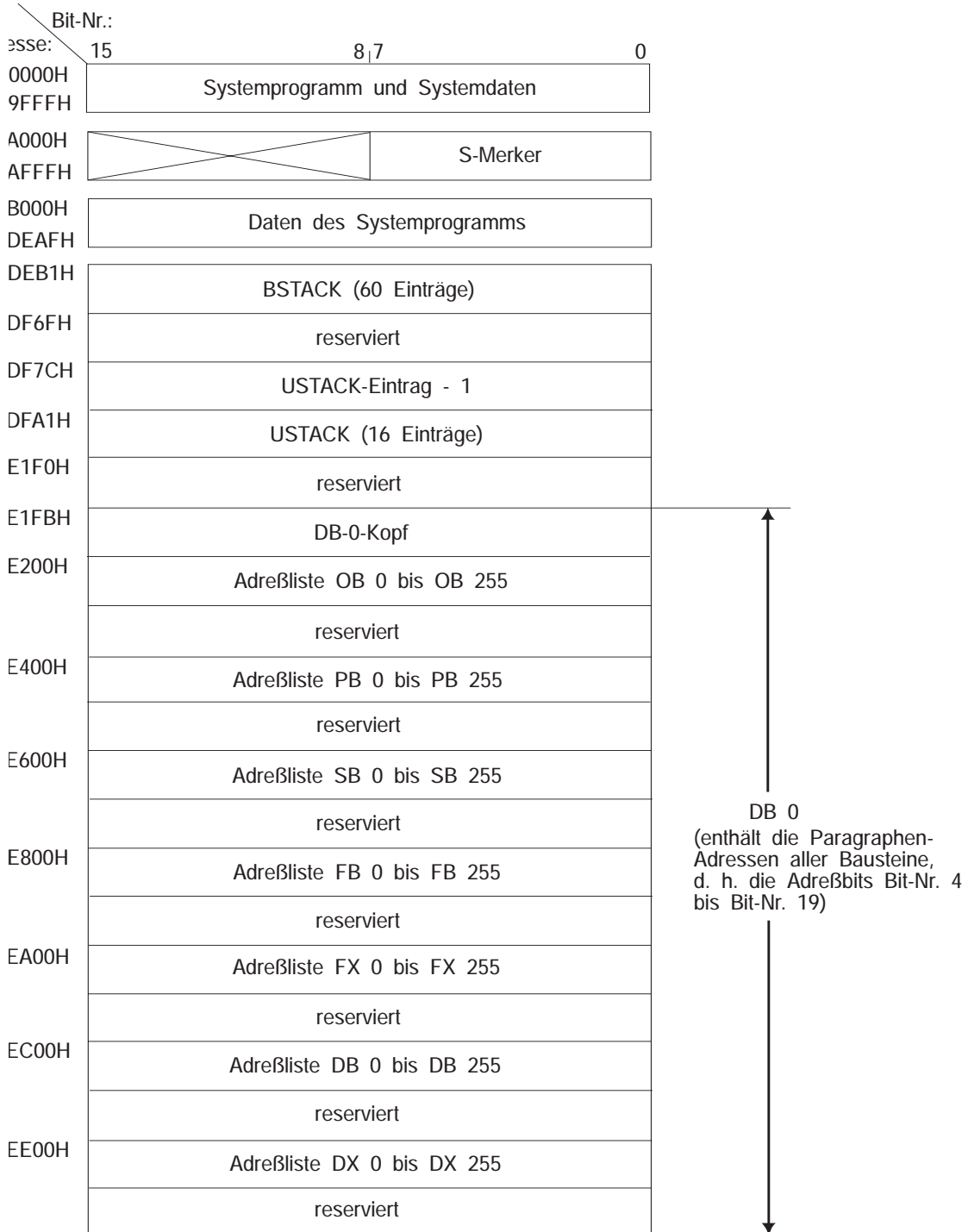


Bild 8-2 Adreßraumteilung System-RAM/Teil 1

Adresse:	Bit-Nr.: 15	8,7	0
E F000H	BS-Bereich (Systemdaten, 256 Wörter)		
	reserviert		
E F200H	BT-Bereich (Systemdaten-Erweiterung, 256 Wörter)		
	reserviert		
E F400H	BA-Bereich (Anschaltung, 256 Wörter)		
	reserviert		
E F600H	BB-Bereich (Anschaltung-Erweiterung, 256 Wörter)		
	reserviert		
E F800H	Zähler (256)		
	reserviert		
E FA00H	Zeiten (256)		
	reserviert		
E FC00H			M-Merker
E FD00H			reserviert
E FE00H			PAE
E FE80H			PAA
E FF00H			reserviert
E FFFFH			

Bild 8-3 Adreßraumteilung System-RAM/Teil 2

8.2.2 Adreßraumaufteilung der Peripherie

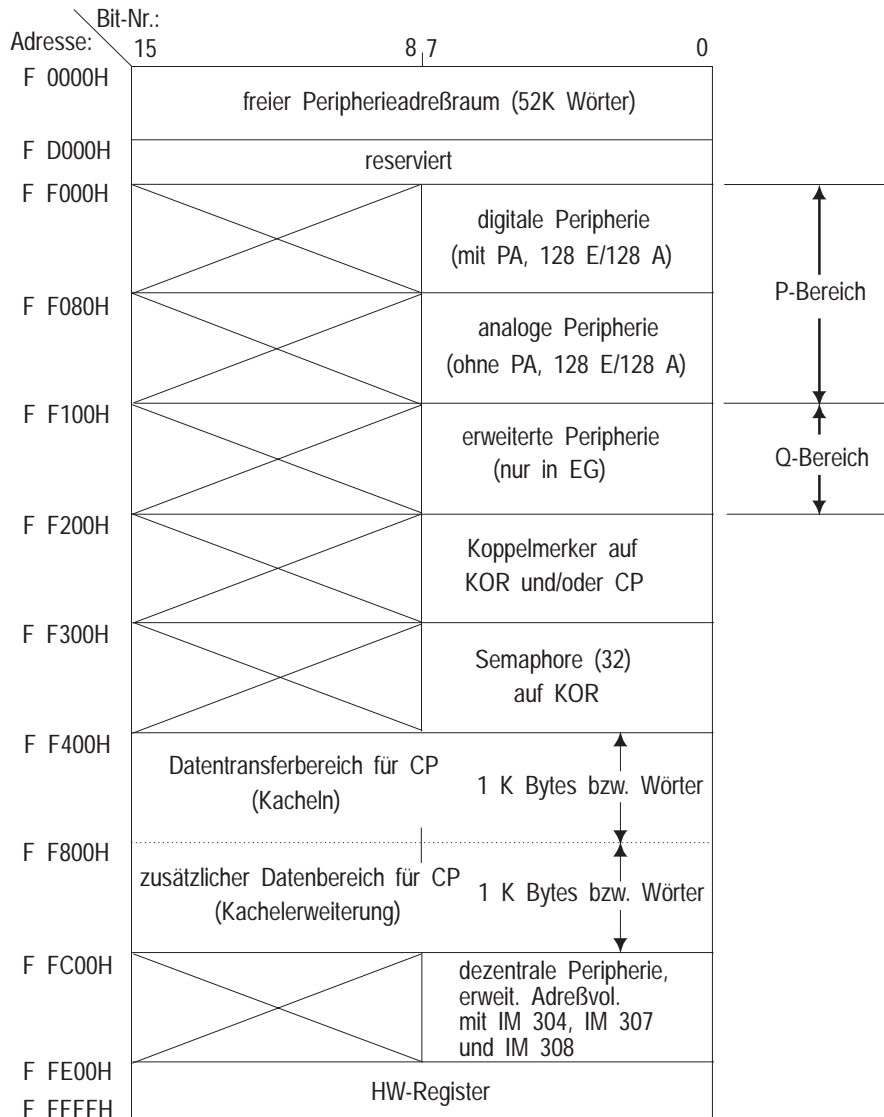


Bild 8-4 Adreßraumaufteilung der Peripherie (8 bit) auf dem S-5-Bus

**Adreßbereiche für
Peripherie und deren
Programmierung**

Bereich (absolute Adresse)		Wird angesprochen mit	Parameter
E FE00 E FE7F	PAAE (Prozeßabbild Eingänge)	LEB / TEB LEW / TEW LED / TED UE / UNE / OE / ONE SE / RE = E	0 bis 127 0 bis 126 0 bis 124 0.0 bis 127.7
E FE80 E FEFF	PAA (Prozeßabbild Ausgänge)	LAB / TAB LAW / TAW LAD / TAD UA / UNA / OA / ONA SA / RA = A	0 bis 127 0 bis 126 0 bis 124 0.0 bis 127.7
P-Peripherie mit Prozeßabbild		Zum Zeitpunkt der Operationsbearbeitung wird nur das Prozeßabbild verändert. Erst am Ende des Zyklus wird der neue Zustand des Prozeßabbilds der Ausgänge an die Peripherie ausgegeben.	
F F000 F F07F	Digitale Peripherie Eingänge/ Ausgänge	LPY / TPY LPW / TPW	0 bis 127 0 bis 126
F F080 F F0FF	Digitale oder ana- logie Peripherie Eingänge/Ausgänge	LPY / TPY LPW / TPW	128 bis 255 128 bis 254
P-Peripherie		Die Ein- und Ausgänge werden byte- oder wortweise direkt angesprochen.	
F F100 F F1FF	Erweiterte Peripherie Eingänge/Ausgänge	LQB / TQB LQW / TQW	0 bis 255 0 bis 254
Q-Peripherie		Die Ein- und Ausgänge werden byte- oder wortweise direkt angesprochen.	

Mit STEP-5-Operationen können Sie entweder direkt oder über das Prozeßabbild auf die Peripherie zugreifen. Beachten Sie dabei, daß ein Prozeßabbild nur für Ein- und Ausgabebytes der P-Peripherie mit Byteadressen von 0 bis 127 existiert!

Hinweis

Über die Anschaltungen IM 304, IM 307 und IM 308 können Sie mit Ihrem Programm auf dezentrale Adreßbereiche zugreifen. Hierdurch gewinnen Sie zwei neue, dem Q-Bereich gleichwertige Adreßbereiche. Ein Zugriff auf diese Bereiche ist aber im Gegensatz zum Q-Bereich nur über absolute Adressierung oder mit dem FB 196 aus dem Softwarepaket "Grundfunktionen" (siehe Katalog ST59) möglich.

8.3 Organisation des Anwenderspeichers in der CPU 948

Der Anwenderspeicher umfaßt – abhängig von der verwendeten Version der CPU 948 – den Speicherbereich von 0 0000H bis C FFFFH. Beim Laden der einzelnen Bausteine des Anwenderprogramms werden diese in beliebiger Reihenfolge im Speicher abgelegt (aufsteigende Adressen).

Verwaltung der Bausteine

Beim Korrigieren von Bausteinen wird der "alte" Baustein im Speicher für ungültig erklärt und ein neuer Baustein in den Speicher und in die Adreßliste eingetragen. Ebenso werden beim Löschen von Bausteinen die Bausteine im Speicher nicht wirklich gelöscht, sondern nur für ungültig erklärt. Lücken, die durch Löschen von Bausteinen entstanden sind, werden als freie Speicherplätze verwaltet und beim Laden neuer Bausteine wieder verwendet.

Speicher komprimieren

Die PG-Online-Funktion SPEICHER KOMPRIMIEREN schiebt alle gültigen Bausteine im Speicher zusammen. Bei SPEICHER KOMPRI-MIEREN im **STOP** werden **alle** nicht dicht gepackt liegenden Bausteine verschoben. Im **RUN** hingegen werden aus Gründen der Datenkonsistenz **lange DB/DX-Bausteine** (> 512 Datenwörter) **nicht** verschoben. Durch das Komprimieren entstehen große zusammenhängende freie Speicherbereiche, die zum Laden von Bausteinen genutzt werden können.

Wird die Online-Funktion SPEICHER KOMPRIMIEREN unterbrochen (z. B. durch NETZ AUS), so wird das Komprimieren beendet und bei NETZ EIN nicht wieder aufgenommen.

Lage der Bausteine im Anwenderspeicher

In der CPU 948 sind Bausteine grundsätzlich so abgelegt, daß das Datenwort DW 0 bzw. die erste STEP-5-Anweisung jedes Bausteins auf einer **Paragrafenadresse** liegt. Paragrafenadressen sind 16-Wort-Grenzen. Im Speicher beginnen somit alle Bausteinköpfe auf der Adresse xxxxBH (Bit-Nr. 0 bis 3 = BH) und alle Bausteinrumpfe auf der Adresse yyyy0H (Bit-Nr. 0 bis 3 = 0H). Die Lücken, die zwischen den Bausteinen im Speicher entstehen, werden durch ungültige Datenbausteine aufgefüllt, so daß alle Bausteine weiterhin aufeinanderfolgend im Speicher stehen.

Füllbausteine

Diese ungültigen Datenbausteine werden "Füllbausteine" genannt. Sie werden von der Speicherverwaltung wie die übrigen Bausteine behandelt und haben folgenden Aufbau:

Anfangskennung:	7070H	;
Baustein-/nummer:	01FBH	;DB 251 ungültig
PG-Kennung:	00FFH	;irrelevant
Bibliotheksnummer:	FFFFH	;irrelevant
Bausteinlänge:	00XXH	;Länge 5 - 20 Wörter
Daten:	FFFFH	;je
	:	;nach
	:	;Länge;
	FFFFH	;können ganz entfallen

Beispiel

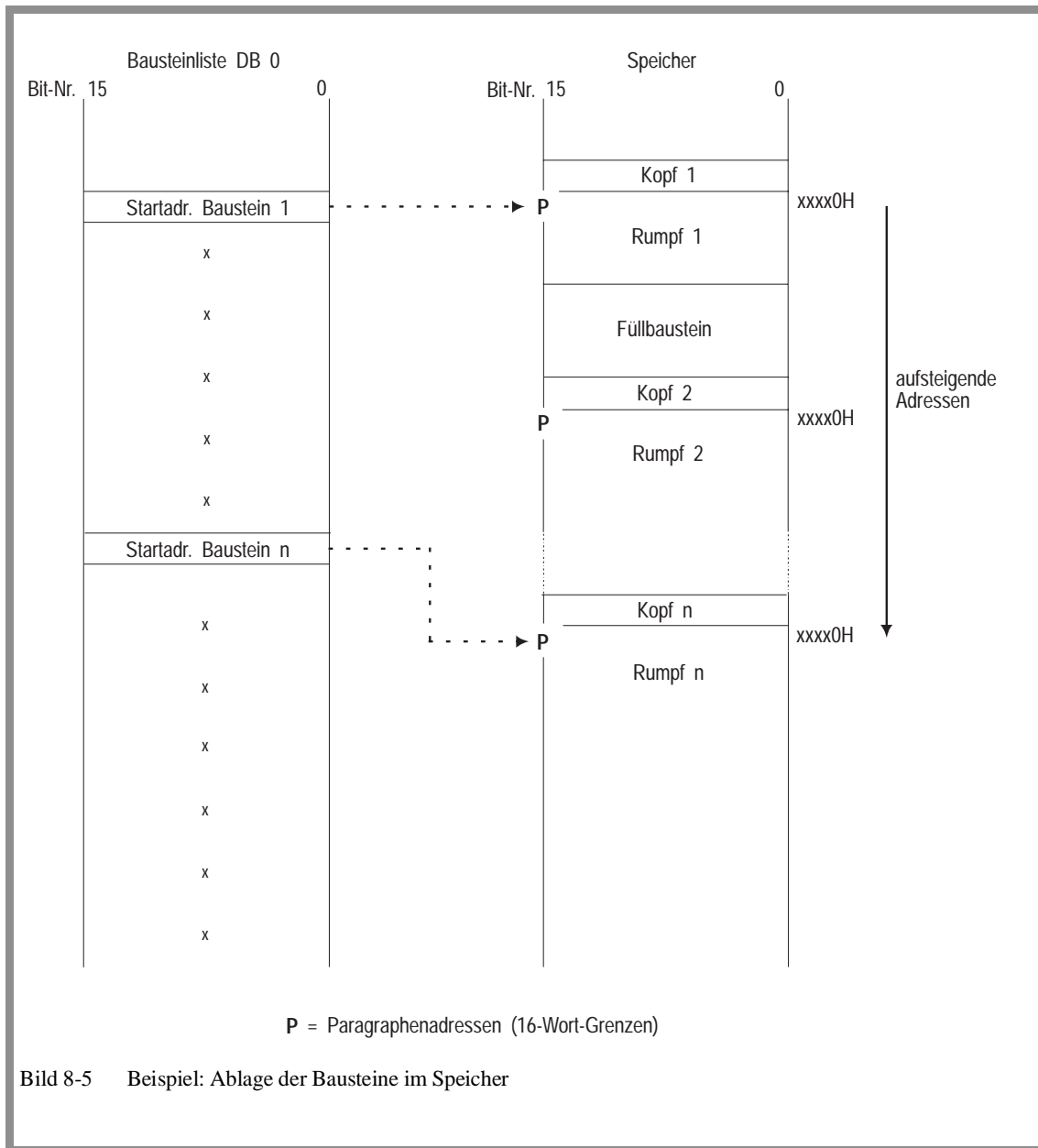


Bild 8-5 Beispiel: Ablage der Bausteine im Speicher

Die Länge der einzelnen Füllbausteine ergibt sich aus der Differenz zwischen Endadresse des zuletzt gespeicherten Bausteins und der nächsten Paragraphenadresse.

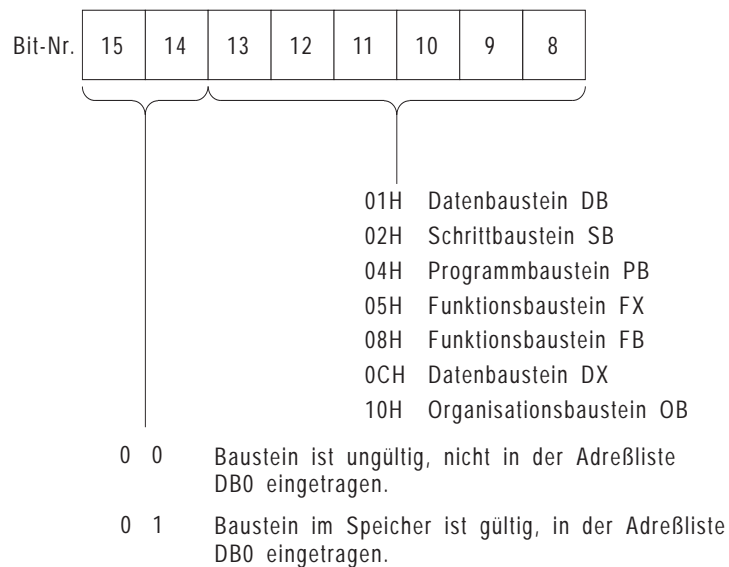
Differenz	Länge des Füllbausteins (inkl. Kopf)
0 bis 5	Differenz + 10
6	kein Füllbaustein
7 bis 10	Differenz + 10
11 bis 15	Differenz - 6

8.3.1 Bausteinköpfe im Anwenderspeicher

Jeder Baustein im Speicher beginnt mit einem 5 Wörter langen Bausteinkopf.

1. Wort: Baustein-Anfangskennung: 7070H

2. Wort: High-Byte = Bausteintyp



Low-Byte = Bauteinnummer

Die Bauteinnummer (0 bis 255) liegt im Low-Byte des 2. Kopfwortes und wird als Dualzahl codiert: 00 bis FFH.

3. Wort: Im High-Byte des 3. Wortes stehen die Kennungen für das Programmiergerät, im Low-Byte ein Teil der Bibliotheksnummer.

4. Wort: Das vierte Wort enthält den Rest der Bibliotheksnummer.

5. Wort: Im 5. Wort (Low- und High-Byte) steht die Länge des Bausteins inklusive Bausteinkopf. Die Angabe erfolgt in Wörtern.

8.3.2

Bausteinadreßlisten im Datenbaustein DB 0

Der Datenbaustein DB 0 befindet sich im System-RAM der CPU (ab Adresse E E200H) und enthält die Liste mit den Anfangsadressen aller Bausteine, die sich im Anwenderspeicher der CPU befinden. Diese Adreßliste wird nach NETZ EIN vom Systemprogramm bei NEU-START aufgebaut, bei WIEDERANLAUF überprüft und bei jeder Eingabe oder Änderung von Bausteinen mit dem PG automatisch aktualisiert.

Anfangsadressen der Adreßlisten

Für jeden Bausteintyp gibt es im DB 0 eine eigens reservierte, 256 Wörter lange Adreßliste. Nicht geladene und gelöschte Bausteine haben die Anfangsadresse '0'. Die Anfangsadressen der einzelnen Bausteinadreßlisten sind festgelegt (siehe Abschnitt 8.2.1).

Baustein-Anfangsadressen

Die Baustein-Anfangsadressen in den Adreßlisten zeigen immer auf das erste Wort **nach** dem Bausteinkopf:

- bei Datenbausteinen jeweils auf das Datenwort DW 0.
- bei Codebausteinen jeweils auf die erste STEP-5-Anweisung (bei FBs auf den 'SPA'-Befehl vor dem Namen und der Parameterliste).

Da jeder Baustein auf einer Paragraphenadresse (16-Wort-Grenze) liegt, beschränkt sich jeder Adreßlisteneintrag im DB 0 auf 1 Wort mit den Bits Nr. 4 bis 19 der Adresse.

Ablage der Bausteinadressen im DB 0

n = E E400H (Anfangsadresse der PB-Adreßliste)

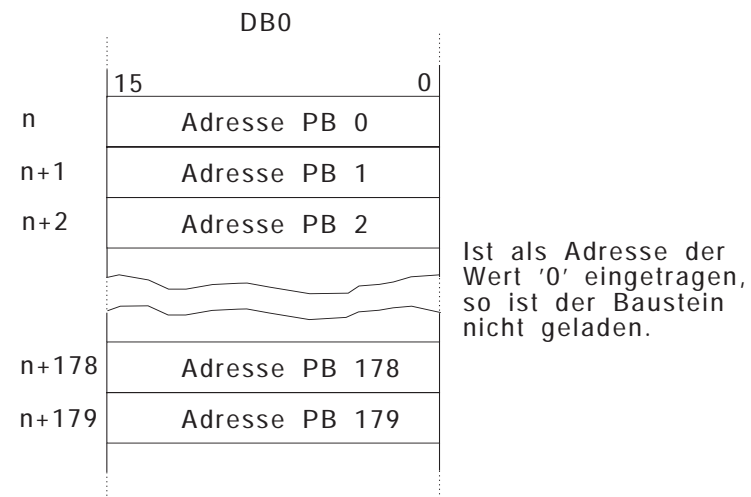


Bild 8-6 Bausteinadressen im DB 0

*Beispiel für die Ermittlung
einer Bausteinadresse*

Die Bausteinanfangsadressen der **Programmbausteine** stehen im DB 0 und beginnen bei der Adresse E E400H. Die Anfangsadresse des PB 22 läßt sich somit durch einen Speicherzugriff auf die Adresse E E416H (= Anfangsadresse der PB + 16H) auslesen.

**8.3.3
BA-/BB-Bereich**

Der BA-Bereich ist ein 256 Wörter langer Bereich im internen System-RAM der CPU. Er belegt die Adressen E F400H bis E F4FFH.

Der BB-Bereich ist ein 256 Wörter langer Bereich im internen System-RAM der CPU. Er belegt die Adressen E F600H bis E F6FFH.

Der gesamte BA-Bereich (BA 0 bis BA 255) und der gesamte BB- Bereich (BB 0 bis BB 255) können vom Anwender für eigene Zwecke genutzt werden.

Der BA-/BB-Bereich wird nur bei URLÖSCHEN mit Nullen vorbe-
setzt!

8.3.4 BS-/BT-Bereich

BS- und BT-Bereich enthalten teils Informationen für den Systemprogrammierer, teils systeminterne Daten.

Der **BS-Bereich** ist ein 256 Wörter langer Bereich im internen System-RAM der CPU. Er belegt die Adressen E F000H bis E F0FFH.

**Warnung**

Es dürfen ausschließlich die Systemdatenwörter BS 60 bis BS 63 beschrieben werden:

Alle übrigen Systemdaten dürfen nur gelesen werden:

Ein Beschreiben dieser Systemdaten kann Rückwirkungen auf die Funktionsfähigkeit des Automatisierungsgerätes sowie angeschlossener Programmiergeräte zur Folge haben: Es können schwere Störungen entstehen, die Mensch und Maschine in Gefahr bringen können.

Der **BT-Bereich** ist ein 256 Wörter langer Bereich im internen System-RAM der CPU. Er belegt die Adressen E F200H bis E F2FFH.

Den gesamten BT-Bereich (BT 0 bis BT 255) können Sie für Ihre eigenen Zwecke nur dann nutzen, wenn Sie,

1. keine Standard-FB verwenden

und

2. keine PG-Funktionen über SINEC-H1 und den parallelen S5-Bus benutzen.

Der BS-/BT-Bereich wird nur bei URLÖSCHEN gelöscht.

Die Informationen einiger Systemdaten (über den internen Aufbau der CPU, den Ausgabestand der Software, die CPU-Kennung usw.) erhalten Sie außerdem über die Online-Funktion SYSTEM-PARAMETER.

Belegung des BS-Bereiches

Tabelle 8-2 Belegung des BS-Bereiches

BS	Bezeichnung	Adresse
0	Abbild des Eingangsbyte EB 0 (Prozeßalarme)	E F000H
1	aktuell anstehende Prozeßalarme (EB 0)	E F001H
2 bis 4	Systemprogramm	
5	aktuell benötigte Zykluszeit	E F005H
6	Systemprogramm	
7	Stopp-Kennungen (USTACK)	E F007H
8	Start/Anlauf-Kennungen (USTACK)	E F008H
9 bis 15	Systemprogramm	
16	Fehlerbereich Ausgangsbytes 0 bis 15	E F010H
17 bis 23	Fehlerbereich Ausgangsbytes 16 bis 127	E F011H bis E F017H
24 bis 31	Fehlerbereich Eingangsbytes 0 bis 127	E F018H bis E F01FH
32 bis 47	Fehlerbereich Koppelmerkerbytes 0 bis 255	E F020H bis E F02FH
48 bis 49	Systemprogramm	
50	PAFE-Byte für "Rückwandbusfunktionen"	E F032H
51 bis 59	Systemprogramm	
60 bis 63	frei für Anwenderzwecke	E F03CH bis E F03FH
64 bis 67	Systemprogramm	
68 bis 71	Fehleradresse bei QVZ- und PARE-Fehler	E F044H bis E F047H
72 bis 74	Systemprogramm	
75	Systemmeldung, Funktionsnummer	E F04BH

BS	Bezeichnung	Adresse
Fortsetzung der Tabelle 8-2:		
76	Systemmeldung, Parameter 1	E F04CH
77	Systemmeldung, Parameter 2	E F04DH
78	Systemmeldung, Parameter 3	E F04EH
79 bis 95	Systemprogramm	
96	aktuelle Uhrzeit (Sekunden)	E F060H
97	aktuelle Uhrzeit (Stunden)	E F061H
98	aktuelle Uhrzeit (Tage)	E F062H
99	aktuelle Uhrzeit (Jahr/Monat)	E F063H
100 bis 119	Systemprogramm	
120	Softwareschutz/Paßwort	E F078H
121 bis 135	Systemprogramm	
136 bis 137	Zellen für Selbsttestfunktion	E F088H bis E F089H
138	Systemprogramm	
139	verbrauchte Zykluszeit beim Nachtriggern	E F08BH
140 bis 252	Systemprogramm	
253	frei für dezentrale Peripherie	E F0FDH
254 bis 255	Systemprogramm	

Ergänzend zur Darstellung in der Tabelle werden nachfolgend die Bit-Belegungen einiger Systemdaten angegeben, die Sie über STEP-5-Operationen oder mit dem PG auswerten können (die Erklärung für die dort aufgeführten Abkürzungen entnehmen Sie bitte dem Abschnitt 5.4).

**8.3.5
Bitbelegung der System-
datenwörter**

Systemdatum BS 0

Abbild Eingangsbyte EB 0 (Prozeßalarme)

Adresse E F000H

Tabelle 8-3 Belegung BS 0 (Abbild EB 0)

High-Byte	
Bit-Nr.	Belegung
15	belegt vom Systemprogramm
14	
13	
12	
11	
10	
9	
8	
Low-Byte	
7	Zustand E 0.7
6	Zustand E 0.6
5	Zustand E 0.5
4	Zustand E 0.4
3	Zustand E 0.3
2	Zustand E 0.2
1	Zustand E 0.1
0	Zustand E 0.0

Systemdatum BS 1

Anzeige der aktuell anstehenden Prozeßalarme

Adresse: E F001H

Tabelle 8-4 Belegung BS 1 (aktuelle Prozessalarme)

High-Byte	
Bit-Nr.	Belegung
15	alle Bits haben den Wert '0'
14	
13	
12	
11	
10	
9	
8	
Low-Byte	
7	Bit = '1': Flanke E 0.7
6	Bit = '1': Flanke E 0.6
5	Bit = '1': Flanke E 0.5
4	Bit = '1': Flanke E 0.4
3	Bit = '1': Flanke E 0.3
2	Bit = '1': Flanke E 0.2
1	Bit = '1': Flanke E 0.1
0	Bit = '1': Flanke E 0.0

Systemdatum BS 5

Aktuelle Zykluszeit

Adresse: E F005H

Tabelle 8-5 Belegung BS 5 (Zykluszeit)

High-Byte und Low-Byte	
Bit-Nr.	Belegung
15	eingetragene Dualzahl * 10 ms = Zykluszeit des zuletzt bearbeiten Zyklus
14	
13	
12	
11	
10	
9	
8	
7	
6	
5	
4	
3	
2	
1	
0	

Beispiel

Bit-Nr.	15	14	13	12	11	10	9	8	7	5	4	3	2	1	0
Wert	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Die Zeit des letzten Zyklus beträgt:

$$(2^4 + 2^3) * 10 \text{ ms} = (16 + 8) * 10 \text{ ms} = 240 \text{ ms}$$

Systemdatum BS 7

AG-Stoppkennungen (USTACK)

Adresse: E F007H

Tabelle 8-6 Belegung BS 7 (AG-Stoppkennungen)

High-Byte	
Bit-Nr.	Belegung
15	reserviert
14	
13	
12	fehlerhafte USTACK-Tiefe
11	unzulässige Anlaufart (UANL)
10	Unterbrechung in Stoppschleife
9	unzulässiger Aufruf Systembaustein (SYSFHL)
8	Fehler im Anlaufbaustein (AFEL)
Low-Byte	
7	Unterbrechung durch System (USYS – Wiederanlauf möglich)
6	Unterbrechung durch Programmierfehler (UPROG – Neustart erforderlich)
5	"Bearbeitungskontrolle Ende" (BEARBE)
4	Stoppschalter (STOPS)
3	Stopp nach Befehlende (STS)
2	Stopp am Zyklusende (STP)
1	Stopp im Mehrprozessorbetrieb (HALT)
0	Stopp durch PG (PGSTP)

Systemdatum BS 8

Start- und Anlaufkennungen (USTACK)

Adresse: E F008H

Tabelle 8-7 Belegung BS 8 (Start- und Anlaufkennungen)

High-Byte	
Bit-Nr.	Belegung
15	Voreinstellung: MANUELLER NEUSTART/ WIEDERANLAUF (MSEG)
14	Voreinstellung: AUTOMATISCHER NEUSTART (ANEG)
13	Voreinstellung: AUTOMATISCHER WIEDERANLAUF (AWEG)
12	URLÖSCHEN erforderlich (URLER)
11	WIEDERANLAUF zulässig (WIEZU)
10	NEUSTART zulässig (NEUZU)
9	URLÖSCHEN durchgeführt (URLDF)
8	WIEDERANLAUF durchgeführt (WIEDF)
Low-Byte	
7	NEUSTART durchgeführt (NEUDF)
6	automatischer Start nach NAU
5	manueller Start
4	NEUSTART MIT GEDÄCHTNIS
3	PG-Urlöschen
2	PG-Systemstart
1	PG-Wiederanlauf
0	PG-Neustart

**Systemdatenwörter
BS 16 bis BS 47**

Fehlerbereiche

BS xx	Adresse(n)	Fehlerbereich
BS 16	E F010	Ausgangsbytes 0 bis 15
BS 17 bis BS 23	E F011 bis E F017	Ausgangsbytes 16 bis 127
BS 24 bis BS 31	E F018 bis E F01F	Eingangsbytes 0 bis 127
BS 32 bis BS 47	E F020 bis E F02F	Koppelmerkerbytes 0 bis 255

BS 16

Adresse: E F010H

Tabelle 8-8 Belegung BS 16 (Fehlerbereich Ausgangsbytes 0 bis 15))

High-Byte	
Bit-Nr.	Belegung
15	Ausgangsbyte 0
14	Ausgangsbyte 1
13	Ausgangsbyte 2
12	Ausgangsbyte 3
11	Ausgangsbyte 4
10	Ausgangsbyte 5
9	Ausgangsbyte 6
8	Ausgangsbyte 7
Low-Byte	
7	Ausgangsbyte 8
6	Ausgangsbyte 9
5	Ausgangsbyte 10
4	Ausgangsbyte 11
3	Ausgangsbyte 12
2	Ausgangsbyte 13
1	Ausgangsbyte 14
0	Ausgangsbyte 15

Wenn bei der Prozeßabbildaktualisierung der Ausgänge (bzw. Eingänge oder Koppelmerker) Fehler auftreten, werden die entsprechenden Bits auf '1' gesetzt. – Die Systemdatenwörter BS 17 bis 47 sind entsprechend BS 16 aufgebaut.

Beispiel zu BS 16

Der Inhalt des Systemdatum BS 16 ist

"8020" hexadezimal bzw. "1000 0000 0010 0000" binär.

Dies bedeutet, daß die Aktualisierung des Prozeßabbildes für die Ausgangsbytes 0 und 10 fehlerhaft verlaufen ist.

Systemdatum BS 50

PAFE-Byte für "Rückwandbusfunktionen"

Im Systemdatum BS 50 wird das Parametrierfehler-Byte für die "Rückwandbusfunktionen" einer CPU eingetragen.

	15	8 7	0
BS 50	PAFE	-	E F032H

Systemdatenwörter BS 68 bis BS 71

Fehleradressen von QVZ- und PARE-Fehlern

Bei Auftreten eines Fehlers QVZ bzw. PARE werden hier die Adresse eingetragen, bei der der Fehler auftrat.

	15		0
BS 68	QVZ-Fehleradr. high		E F044H
BS 69	QVZ-Fehleradr. low		E F045H
BS 70	PARE-Fehleradr. high		E F046H
BS 71	PARE-Fehleradr. low		E F047H

**Systemdatenwörter
BS 75 bis BS 78**

Systemmeldung

Die Einträge in den Systemdatenwörtern BS 75 bis BS 78 beziehen sich auf den zuletzt aufgetretenen Fehler. Die Meldung setzt sich aus den 4 Systemdatenwörtern zusammen und hat folgenden Aufbau:

	15	0	
BS 75	Fehlernummer	Parametertyp	E F04BH
BS 76	Parameter 1		E F04CH
BS 77	Parameter 2		E F04DH
BS 78	Parameter 3		E F04EH

BS 75

Das **High-Byte** enthält die **Fehlernummer**, die den aufgetretenen Fehler klassifiziert. Durch die Fehlernummer läßt sich der Fehler einem der folgenden Bereiche zuordnen:

Fehlergruppen

- 01H bis 2FH: Anwenderfehler,
- 30H bis 3FH: Fehler im DX 0 oder DB 1,
- 40H Systemfehler.

Parametertyp

Das **Low-Byte** enthält den **Parametertyp**, der den Aufbau des nachfolgenden Parameterblocks in BS 76 bis BS 78 beschreibt. Es gibt die Parametertypen von 00H bis 10H. Der dazugehörige Aufbau des Parameterblocks wird später beschrieben.

In den beiden nachfolgenden Tabellen werden die Fehler der Fehlergruppen "allgemeine Fehler" und "Fehler im DX 0 oder DB 0" aufgeführt.

allgemeine Fehler

Tabelle 8-9 BS 75: Allgemeine Fehler

Fehler- nummer	Parame- tertyp	Bedeutung
01H	01H	Aufgerufener Baustein nicht geladen
02H	01H	Adressierfehler
03H	01H	Zykluszeitfehler
04H	01H	Substitutionsfehler
05H	02H	Quittungsverzug dezentrale Peripherie
06H	03H	Quittungsverzug Anwenderspeicher
07H	01H	Lade-/Transferfehler bei DB/DX-Datenbausteinen
08H	01H	Klammerzähler-Überlauf
09H	04H	aufzuschlagener Datenbaustein nicht vorhanden
0AH	05H	Fehler bei Weckalarmen/Interrupts
0BH	03H	Quittungsverzug Kachelbereich
0CH	03H	Quittungsverzug Globaler Kommunikationsbereich
0DH	07H	Quittungsverzug bei der Prozeßabbildaktualisierung
0EH	03H	Quittungsverzug Koppelmerker Synchronisationsbereich
0FH	03H	Quittungsverzug P/Q-Peripherie
10H	03H	Quittungsverzug, weil dezentrale Anschaltung nicht vorhanden (IM 3/IM 4)
11H	03H	Parityfehler im Anwenderspeicher
12H	01H	Quittungsverzug Prozeßabbildtransfer
13H	01H	Quittungsverzug Eingangsbyte EB 0
14H	01H	BSTACK-Überlauf
15H	01H	STS-Befehl
16H	01H	Stoppeschalter
17H	01H	Halt-Signal vom Koordinator
18H	01H	nicht belegt
19H	01H	Lade-/Transferfehler bei der Prozeßabbildaktualisierung
1AH	01H	Lade-/Transferfehler bei Adressierung über das BR-Register
1BH	01H	Peripherie unklar
1CH	08H	Quittungsverzug/Parityfehler bei Initialisierung

Fehler- nummer	Parame- tertyp	Bedeutung
Fortsetzung der Tabelle 8-9:		
1DH	08H	AUTOMATISCHER WIEDERANLAUF nicht möglich; NEUSTART erforderlich
1EH	00H	unzulässige Anlaufart
1FH	01H	Lade-/Transferfehler bei Blocktransfer-Befehl (falsche Bereichsgrenzen bei TNW, TXB, TXW)
20H	09H	unzulässige Länge bei E DB/EX DX
21H	09H	DB/DX existiert schon bei E DB/EX DX
22H	09H	Speicherplatz reicht nicht aus bei E DB/EX DX
23H	05H	maskierter Systeminterrupt kommt durch
24H	00H	Bausteinfunktion (Komprimieren, Übertragen, Eingabe) im STOP, NEUSTART erforderlich
25H	00H	Batterieausfall; kein Anlauf möglich
26H	00H	Wechsel von Ein- zu Mehrprozessorbetrieb verhindert WIEDERANLAUF
27H	01H	STOP durch STP-Operation
28H	01H	Dauer-Ready-Signal (periph. Baugruppe defekt)
29H	08H	Fehler im DB-0-Aufbau nach URLÖSCHEN

DX-0-/DB1-Fehler

Tabelle 8-10 BS 75: Fehler im DX 0 oder DB 1

Fehler- nummer	Parame- tertyp	Bedeutung
Fehler im DX 0		
30H	00H	DX 0 fehlt im Mehrprozessorbetrieb
32H	00H	Prozeßalarm und Systeminterrupt gleichzeitig angewählt
33H	00H	Alarmer und Betriebsart unverträglich
34H	06H	ungültiger DX-0-Kopf
35H	07H	Fehler bei DX-0-Block-Kennung
36H	07H	Fehler bei DX-0-Parameter
Fehler im DB 1		
38H	06H	DB 1 fehlt im Mehrprozessorbetrieb
39H	06H	ungültiger DB-1-Kopf
3AH	07H	DB-1-Kennung mehrmals gesetzt
3BH	07H	DB-1-Byte-Offset ohne Kennung

Fehler- nummer	Parame- tertyp	Bedeutung
Fortsetzung der Tabelle 8-10:		
3CH	07H	in DB 1 eingetragene Peripherie ist nicht gesteckt
3DH	07H	Offset zu groß (Parameterfehler)
3EH	07H	zu viele Offsets

Fehleranzeigen der Selbsttestfunktionen

Tabelle 8-11 BS 75: Fehleranzeigen der Selbsttestfunktionen

Fehler- nummer	Parame- tertyp	Bedeutung
61H	0BH	Checksummenfehler im Sytemprogrammcode
62H	0AH	Checksummenfehler im Code der STEP-5-Codebausteine
63H	0BH	Adressdecoder-Fehler
64H	0CH	Fehler beim Test des wortweise organisierten Anwenderspeichers
65H	0CH	Fehler beim Test des byteweise organisierten Anwenderspeichers
66H	00H	Fehler beim Test der Zykluszeitüberwachung
67H	00H	Fehler beim Test des BASP-Signals
68H	00H	Fehler beim Test der Hardwareuhr

Aufbau des Parameterblocks
(BS 76 bis BS 78)

Tabelle 8-12 BS 76 bis BS 78:Parametertypen

Parametertyp	Aufbau des Parameterblocks	
00H	keine Parameter; Parameter 1, 2, 3 = 0	
01H	Parameter 1:	Bausteintyp/Bausteinnummer (Kennungen aus Bausteinkopf)
	Parameter 2:	Befehl, der Unterbrechung verursachte
02H	Parameter 1:	Anschaltnummer (dezentrale Peripherie)
	Parameter 2:	Nummer der defekten Schnittstelle
	Parameter 3:	fehlerhafter Byteoffset
03H	Parameter 1:	nicht belegt
	Parameter 2:	Fehleradresse, high
	Parameter 3:	Fehleradresse, low
04H	Parameter 1:	Bausteintyp/Bausteinnummer (Kennungen aus Bausteinkopf)
	Parameter 2:	Befehl, der Unterbrechung verursachte
	Parameter 3:	Typ/Nummer des aufzuschlagenden DB/DX
05H	Parameter 1:	folgende Bits werden fehlerabhängig gesetzt:
	Bit-Nr. 0	= 1: Weckalarm/Periode 1
	Bit-Nr. 1	= 1: Weckalarm/Periode 2
	: :	: :
	Bit-Nr. 7	= 1: Weckalarm/Periode 8
	Bit-Nr. 8	= 1: Weckalarm/Periode 9
	Bit-Nr. 9	= 1: Wecktakt zu lange maskiert = 0: Warteschlangen-Überlauf
	Bit-Nr. 10	reserviert
	Bit-Nr. 11	reserviert
	Bit-Nr. 12	= 1: Interrupt G
	Bit-Nr. 13	= 1: Interrupt F
	Bit-Nr. 14	= 1: Interrupt E
Bit-Nr. 15	= 1: Interrupt X	
06H	Parameter 1	Datenwort "Soll" im DX-0- bzw. DB-1-Kopf
	Parameter 2	Datenwort "Ist" im DX-0- bzw. DB-1-Kopf
07H	Parameter 1	Block-Kennung oder Kennwort im DX 0 bzw. DB 1
	Parameter 2	fehlerhafter Parameter 1 im DX 0 bzw. DB 1 (FFFFH: Parameter irrelevant)
	Parameter 3	fehlerhafter Parameter 2 im DX 0 bzw. DB 1 (FFFFH: Parameter irrelevant)

Parametertyp	Aufbau des Parameterblocks	
Fortsetzung der Tabelle 8-12:		
08H	Parameter 1	folgende Bits werden fehlerabhängig gesetzt:
		Bit-Nr. 0 = 1: QVZ in Initialisierung
		Bit-Nr. 1 = 1: PARE bei Initialisierung
		Bit-Nr. 2 = 1: Inhalt der memory Card ist zu groß
		Bit-Nr. 3 = 1: Betriebssystemfehler
		Bit-Nr. 4 = 1: Bausteinkennung falsch
		Bit-Nr. 5 = 1: Bausteinendekennung falsch
		Bit-Nr. 6 reserviert
		Bit-Nr. 7 reserviert
		Bit-Nr.8 = 1: DB 0 seit letztem NEUSTART verändert
		Bit-Nr. 9 bis 15 reserviert
	Parameter 2	Fehleradresse high, (wenn Bit 2, 4 oder 5 von Parameter 1 = '1' ist
Parameter 3	Fehleradresse low, (wenn Bit 2, 4 oder 5 von Parameter 1 = '1' ist	
09H	Parameter 1	Opcode "EX DX" oder "E DB" (daraus ergibt sich der Bausteintyp)
	Parameter 2	Bausteinnummer
	Parameter 3	Datenbaustein-Länge
10H	systeminterne Fehlernummer	
0AH	Parameter 1	Bausteintyp/Bausteinnummer (Kennungen aus Bausteinkopf)
	Parameter 2	Soll-Checksumme
	Parameter 3	Ist-Checksumme
0BH	Parameter 1	FFFFH
	Parameter 2	Fehleradresse high bei Adresscodefehler, Ist-Checksumme high beim Test des Systemprogrammcodes
	Parameter 3	Fehleradresse low bei Adresscodefehler, Ist-Checksumme low beim Test des Systemprogrammcodes
0CH	Parameter 1	Prüfmuster bei Test des Anwenderspeichers
	Parameter 2	Fehleradresse high
	Parameter 3	Fehleradresse low

Beispiel einer Systemmeldung

BS 75	21H	09H	E F04BH
BS 76	7804H		E F04CH
BS 77	0064H		E F04DH
BS 78	0078H		E F04EH

BS 75, Fehlernummer = 21H: Der Fehler ist im STEP-5-Anwender-Programm beim Erzeugen eines DB/DX aufgetreten.

BS 75, Parametertyp = 09H: Der Parameterblock in BS 76 bis BS 78 enthält 3 Parameter.

BS 76, Parameter 1 = 7804H: Opcode = "EX DX", daraus ergibt sich als Bausteintyp "DX".

BS 77, Parameter 2 = 0064H: Bausteinnummer = 100 (dez.)

BS 78, Parameter 3 = 0078H: Datenbausteinlänge = 120 Datenwörter

Informationsgehalt der Meldung:

Im STEP-5-Anwenderprogramm sollte der Datenbaustein DX 100 mit einer Länge von 120 Datenwörtern erzeugt werden. Dieser ist aber schon vorhanden.

**Systemdatenwörter
BS 96 bis BS 99**

Echtzeituhr

Im Systemdatenbereich BS 96 bis BS 99 wird das aktuelle Datum und die aktuelle Uhrzeit mitgeführt und kann von dort bei Bedarf ausgelesen werden.

	15		0	
BS 96	Sekunden	1/10, 1/100 Sekunden		E F060H
BS 97	Stunden	Minuten		E F061H
BS 98	Tag	Wochentag		E F062H
BS 99	Jahr	Monat		E F063H

Die Uhr wird im 10-ms-Takt aktualisiert.

BS 96

Sekunden und 1/100 Sekunden (Adresse: E F060H):

Tabelle 8-13 Belegung BS 96 (Echtzeituhr: Sekunden, 1/100 Sekunden)

High-Byte	
Bit-Nr.	Belegung
15	Sekunden, Zehner, mögliche Werte: 00H bis 05H
14	
13	
12	
11	Sekunden, Einer, mögliche Werte: 00H bis 09H
10	
9	
8	
Low-Byte	
7	1/10 Sekunde, mögliche Werte: 00H bis 09H
6	
5	
4	
3	1/100 Sekunde, mögliche Werte: 00H bis 09H
2	
1	
0	

BS 97

Stunden und Minuten (Adresse: E F061H):

Tabelle 8-14 Belegung BS 97 (Echtzeituhr: Stunden, Minuten)

High-Byte	
Bit-Nr.	Belegung
15	0 = 12-Std.-Format, 1 = 24-Std.-Format
14	
13	0 = AM, 1 = PM
12	
11	Stunden, Zehner, mögliche Werte: 00/01H bei 12-Std.-Format, 00/02H bei 24-Std.-Format
10	
9	
8	
Low-Byte	
7	Minute, Zehner, mögliche Werte: 00H bis 05H
6	
5	
4	Minute, Einer, mögliche Werte: 00H bis 09H
3	
2	
1	
0	

BS 98

Tag und Wochentag (Adresse: E F062H):

Tabelle 8-15 Belegung BS 98 (Echtzeituhr: Tag, Wochentag)

High-Byte	
Bit-Nr.	Belegung
15	Tag, Zehner, mögliche Werte: 00H bis 03H
14	
13	
12	Tag, Einer, mögliche Werte: 00H bis 09H
11	
10	
9	
8	
Low-Byte	
7	Wochentag, mögliche Werte: 00H bis 06H für Mo bis So
6	
5	
4	0
3	
2	
1	
0	

BS 99

Jahr und Monat (Adresse: E F063H):

Tabelle 8-16 Belegung BS 99 (Echtzeituhr: Jahr, Monat)

High-Byte	
Bit-Nr.	Belegung
15	Jahr, Zehner, mögliche Werte: 00H bis 09H
14	
13	
12	
11	Jahr, Einer, mögliche Werte: 00H bis 09H
10	
9	
8	
Low-Byte	
7	Monat, Zehner, mögliche Werte: 00/01H
6	
5	
4	
3	Monat, Einer, mögliche Werte: 00 bis 09H
2	
1	
0	

Systemdatum BS 120

Softwareschutz

Das Systemdatum BS 120 steuert die Systemfunktion "Softwareschutz". Mit dieser Funktion können Sie das Lesen, Überschreiben und Löschen von Bausteinen durch das Programmiergerät (z. B. durch unbefugtes Bedienpersonal) verhindern, indem Sie ein Paßwort vereinbaren.

Paßwort

Die Funktion "Softwareschutz" ist mit einem Paßwort verknüpft, das dem Systemprogramm über BS 120 mitgeteilt wird.

Paßwort vereinbaren/Softwareschutz aktivieren

Mit dem Vereinbaren eines Paßwortes im BS 120 wird automatisch der Softwareschutz aktiviert. Das Paßwort kann nur (neu) vereinbart werden, wenn es zuvor gelöscht worden ist.

Paßwort löschen/Softwareschutz deaktivieren

Mit dem Löschen des Paßwortes wird automatisch der Softwareschutz deaktiviert. Beim Löschen eines Paßwortes muß dieses dem Systemprogramm über BS 120 mitgeteilt werden.

maximal 5 Löschversuche

Fehlerhafte Paßwortangaben zum Löschen des Paßwortes werden vom Systemprogramm abgewiesen und gezählt. Nach maximal 5 fehlerhaften Versuchen stellt das Systemprogramm die Paßwortbearbeitung ein. Danach kann das Paßwort erst nach einem NEUSTART erneut gelöscht werden. Der "Fehlerzähler" wird nach erfolgreichem Löschen des Paßwortes und bei einem NEUSTART wieder gelöscht.

Wie wird das Paßwort vereinbart bzw. gelöscht?

Das Paßwort wird vereinbart/gelöscht (und der Softwareschutz aktiviert/deaktiviert) durch Beschreiben des Systemdatums BS 120 mit einem bestimmten Bitmuster (siehe "Belegung beim Beschreiben") durch

- das STEP-5-Programm oder
- einen PG-Auftrag "Ausgabe Adresse".

Hinweis

Im Lieferzustand der CPU und nach URLÖSCHEN ist das Paßwort gelöscht und der Softwareschutz abgeschaltet.

Wann wird der Softwareschutz aktiviert bzw. deaktiviert?

Die Vereinbarung eines Paßwortes ist zu **jedem Zeitpunkt** möglich. Nach einer Paßwortvereinbarung wird der Softwareschutz jedoch zu **bestimmten Zeitpunkten** aktiviert:

- im Betriebszustand WEICHER STOP:
einmalig nach dem Aufruf des OB 38 ¹⁾,
zyklisch vor dem Aufruf des OB 39 ¹⁾,
- im Betriebszustand ANLAUF:
einmalig nach Aufruf der Anlauf OBs (OB 20, OB 21 OB 22),
- im Betriebszustand RUN:
zyklisch vor Aufruf des OB 1.

Belegung des Systemdatums beim Beschreiben

Um die Softwareschutzfunktion aufzurufen, beschreiben Sie das Systemdatum BS 120 mit einem Bitmuster entsprechend der auszuführenden Funktion. Der nachfolgenden Tabelle können Sie entnehmen, wie Sie das Bitmuster aufbauen müssen.

Adresse: E F078H

Tabelle 8-17 Belegung BS 120 (Softwareschutz) beim Beschreiben

High-Byte	
Bit-Nr.	Belegung
15	Aktionsbit: 1 = Funktion ausführen
14	Funktionsbit: 1 = Paßwort setzen, 0 = PW lösch.
13	Bit-Nr. 8 bis 13 eines 14-bit-Paßwortes
12	
11	
10	
9	
8	
Low-Byte	
7	Bit-Nr. 0 bis 7 eines 14-bit-Paßwortes
6	
5	
4	
3	
2	
1	
0	

¹⁾ Die Bearbeitung einer Anforderung erfolgt unabhängig davon, ob der OB 38 bzw. OB 39 geladen ist. Dadurch ist z. B. ein Aktivieren des Softwareschutzes im STOP möglich.

**Auslesen des Systemdatums
BS 120**

Durch Auslesen des Systemdatums BS 120 können Sie feststellen, ob ein "Auftrag" durch Beschreiben des Systemdatums ausgeführt wurde. Das Systemprogramm hinterlegt dort eine Rückmeldung.

Belegung des Systemdatums beim Auslesen:

Nach dem Aufruf der Softwareschutzfunktion können Sie durch Auswerten der Rückmeldung Informationen über den Erfolg des Auftrags erhalten.

Adresse: E F078H

Tabelle 8-18 Belegung BS 120 (Softwareschutz) beim Auslesen

High-Byte	
Bit-Nr.	Belegung
15	0
14	Fehlerbit: 0 = kein Fehler, 1 = Fehler
13	0
12	0
11	0
10	dualer Zähler der Löschversuche
9	
8	
Low-Byte	
7	0
6	0
5	0
4	1 = kein Paßwort aktiv
3	1 = Löschen nicht möglich, falsches Paßwort
2	1 = Softwareschutz(Paßwort) ist bereits aktiviert
1	1 = unzulässiges Paßwort
0	1 = Versuchszählerüberlauf

Gültige Rückmeldungen

Anzeige	Erläuterung
0000H	kein Fehler
4x01H	Der Zähler für die Löschversuche ist übergelaufen. Um den Zähler rückzusetzen, ist ein NEUSTART erforderlich.
4x02H	unzulässiges Paßwort (0000H oder 3FFFH)
4x04H	Bei aktivem Softwareschutz wurde versucht, ein neues Paßwort zu vereinbaren (x = Anzahl der Löschversuche)
4x08H	Es wurde mit falschem Paßwort versucht, das bestehende Paßwort zu löschen (den Softwareschutz zu deaktivieren). Der Zähler für die Löschversuche wurde inkrementiert. Der Zählerstand 'x' wird mit der Rückmeldung übergeben (Dualzahl in Bit-Nr. 8 bis Bit-Nr. 10).
4010H	Es wurde versucht, ein nicht vorhandenes Paßwort zu löschen.

Bester Zeitpunkt zum Aktivieren des Softwareschutzes

Die größte Schutzwirkung erreichen Sie durch Aktivieren des Softwareschutzes im OB 38/OB 39 (Betriebszustand WEICHER STOP). Damit ist der Schutz auch bei gesteckter Memory Card nach dem URLÖSCHEN sofort aktiv.

Reaktionen bei Softwareschutzverletzung

Wenn Sie den Softwareschutz aktiviert haben, reagiert das Systemprogramm auf Schutzverletzungen durch PG-Aufträge. Der nachfolgenden Tabelle können Sie die Reaktion auf die unterschiedlichen PG-Aufträge entnehmen.

PG-Funktion	Ausgabe am PG
Baustein löschen	Meldungsausgabe "Baustein-Typ und -Nummer unzulässig"
Baustein lesen	Ausgabe eines Dummy-Bausteins: FB/FX: FB nummer NAME :DUMMY :BE DB/DX: DW0 6500 OB/PB/SB: :BE
Baustein überschreiben (Baustein ist noch nicht vorhanden)	der Baustein wird eingetragen
Baustein überschreiben (Baustein ist bereits vorhanden)	Meldungsausgabe "Baustein vorhanden"; nach Bestätigung durch die Übernahmetaste wird die Meldung "Baustein-Typ und -Nummer falsch" ausgegeben.

*Beispiele zum Schreiben und
Lesen des BS 120*

Aktivieren des Softwareschutzes in den Anlaufbausteinen:

(Die Aktivierung des Softwareschutzes per Programm sollte zweckmäßigerweise in einem der Anlauf-OBs (OB 20, OB 21, OB 22, OB 38) vorgenommen werden.)

```

:
:L   KH C0AF      KH = Bitmuster "Passwort vereinbaren"
:                               (Passwort = 00AFH)
:T   BS 120
:

```

Rückmeldung in BS 120 auswerten:

Mit folgender STEP-5-Operationsfolge im **OB 1** oder **OB 39** können Sie durch Auswertung der Rückmeldung auf einen Fehler beim Vereinbaren des Paßwortes reagieren.

Beachten Sie, daß die Rückmeldung erst nach bestimmten Aktionen des Systemprogramms (siehe S. 8 - 35 unten) ausgewertet werden kann!

```

:
:L   BS 120
:L   KB 0
:><F
:SPB FB yyy      Funktionsbaustein fuer Fehlerbearbeitung aufrufen
NAME :   PW-FEHLER
:

```

**Löschen und Ändern des Paßwortes am PG über die Funktion
AUSGABE ADRESSE:**

Ausgangszustand: Die CPU befindet sich im Betriebszustand RUN oder STOP.

Führen Sie folgende Hantierungen am PG durch:

1. Lassen Sie sich den Inhalt der Adresse E F078H ausgeben.
2. Löschen Sie das alte Paßwort durch Überschreiben des Inhaltes mit der Hexadezimalzahl 80AFH ("00AFH" = altes Paßwort).
3. Warten Sie mindestens solange, wie die Zykluszeit von OB 39 bzw. OB 1 ist.
4. Lassen Sie sich erneut den Inhalt der Adresse E F078H ausgeben.
5. Geben Sie das neue Paßwort "1234H" ein, indem Sie den Inhalt mit der Hexadezimalzahl D234H überschreiben.

Systemdatenwörter BS 136 bis BS 137 Für Selbsttestfunktion

Die Systemdatenwörter BS 136 bis BS 137 werden für die Selbsttestfunktion benutzt.

BS 136 Anzahl der Zeitscheiben (Adresse: E F088H)

BS 137 Steuerbits (Adresse: E F089H)

Über die Steuerbits werden die einzelnen Selbsttestfunktionen zu- oder abgeschaltet (siehe Abschnitt 5.7).

Bit = '1': Selbsttestfunktion ist eingeschaltet.

Bit = '0': Selbsttestfunktion ist ausgeschaltet.

Tabelle 8-19 Belegung BS 137 (Steuerbits für Selbsttestfunktionen)

High-Byte	
Bit-Nr.	Belegung
15	Speichertest
14	nicht belegt
13	Test der Zykluszeitüberwachung
12	nicht belegt
11	Test des BASP-Signals
10	Uhrentest
9	nicht belegt
8	nicht belegt
Low-Byte	
7	Test der Adreßleitungen
6	nicht belegt
5	Codetest der STEP-5-Codebausteine im Anwenderspeicher
4	nicht belegt
3	nicht belegt
2	Codetest des Systemprogramms
1	nicht belegt
0	nicht belegt

Systemdatum BS 139

Verbrauchte Zykluszeit beim Nachtriggern

Adresse E F08BH

In diesem Systemdatenwort wird die Zeit hinterlegt, die seit dem letzten Systemkontrollpunkt (Beginn OB 1) bis zum Nachtriggern über OB 222 (bei mehreren Aufrufen des OB 222 innerhalb des Zyklus die Zeit bis zum letzten Nachtriggern) verbraucht wurde.

Der Zeitwert beträgt: Inhalt von BS 139 * 10 ms.

Systemdatum BS 253

Verzeichnis gesteckter Anschaltungen

Adresse: E F0FDH

Tabelle 8-20 Belegung BS 253 (Verzeichnis gesteckter Anschaltungen)

High-Byte	
Bit-Nr.	Belegung
15	reserviert
14	
13	
12	
11	Anschaltung Nr. 11
10	Anschaltung Nr. 10
9	Anschaltung Nr. 9
8	Anschaltung Nr. 8
Low-Byte	
7	reserviert (Anschaltung Nr. 8)
6	reserviert (Anschaltung Nr. 7)
5	reserviert (Anschaltung Nr. 6)
4	reserviert (Anschaltung Nr. 5)
3	reserviert (Anschaltung Nr. 4)
2	reserviert (Anschaltung Nr. 3)
1	reserviert (Anschaltung Nr. 2)
0	reserviert (Anschaltung Nr. 1)

8.3.6

Frei adressierbarer Systemdatenbereich

Der Speicherbereich E 8200H bis E DEF0H wird vom Systemprogramm als frei adressierbarer Systemdatenbereich genutzt.

AG-Identifikationsblock

Am Anfang dieses Bereiches steht ein Informationsblock von 12 Wörtern, in dem eine Identifikation des benutzten AG eingetragen wird.

Dieser Block hat folgende Struktur:

Wort

0	'A'	'G'	E 8200H
1	'1'	'5'	E 8201H
2	'5'	'U'	E 8202H
3	'C'	'P'	E 8203H
4	'U'	'9'	E 8204H
5	'4'	'8'	E 8205H
6	'V'	'x'	E 8206H
7	'.'	'y'	E 8207H
8	0		
9	0		
10	0		
11	0		E 820BH

Für 'x' und 'y' wird die jeweils aktuelle Versionsnummer eingetragen.

Systemparameter

Ab der Adresse E 8210H sind folgende Systemparameter enthalten:

Wort

0	Anfangsadr. Eingangssignalformer		E 8210H
1	Anfangsadr. Ausgangssignalformer		E 8211H
2	Anfangsadr. Prozeßabbild Eingänge		E 8212H
3	Anfangsadr. Prozeßabbild Ausgänge		E 8213H
4	Anfangsadr. Merker		E 8214H
5	Anfangsadr. Zeiten		E 8215H
6	Anfangsadr. Zähler		E 8216H
7	Anfangsadr. Systemdaten		E 8217H
8	Statuskennung	Ausg. AG-Software	E 8218H
9	Anwenderspeicher-Endadresse		E 8219H
10	Systemprogrammspeicher		E 821AH
11	Länge der DB-Liste		E 821BH
12	Länge der SB-Liste		E 821CH
13	Länge der PB-Liste		E 821DH
14	Länge der FB-Liste		E 821EH
15	Länge der OB-Liste		E 821FH
16	Länge der FX-Liste		E 8220H
17	Länge der DX-Liste		E 8221H
18	Länge des Adreßlisten-DBs (DB 0)		E 8222H
19	Steckplatzkennng. (s. u.)	CPU-Kennung 2 (s. u.)	E 8223H
20	Bausteinkopflänge		E 8224H
21	CPU-Kennung 1 (s. u.)	Ausgabe der PG-Anschaltungssoftware	E 8225H

Die Informationen zu einigen Systemdaten (z. B. über den internen Aufbau der CPU, den Ausgabestand der Software, die CPU-Kennung usw.) erhalten Sie außerdem über die PG-Online-Funktion SYSTEMPARAMETER.

Wort 19 und Wort 21

Aufbau der Wörter 19 und 21:

Wort 19

Bit-Nr.	Belegung High-Byte
15	0
14	0
13	0
12	0
11	Steckplatzkennung CPU 4
10	Steckplatzkennung CPU 3
9	Steckplatzkennung CPU 2
8	Steckplatzkennung CPU 1
Belegung Low-Byte	
7	CPU-Typ: 0010 = CPU 948 (nur gültig in Zusammenhang mit CPU-Kennung!)
6	
5	
4	
3	CPU-Kennung 2: 1000 = AG 155U
2	
1	
0	

Wort 21

Bit-Nr.	Belegung High-Byte
15	reserviert
14	
13	
12	
11	
10	
9	
8	
Bit-Nr.	Belegung Low-Byte
7	Ausgabestand der PG-Anschaltungssoftware in der Form "xyH" Beispiel: 13H entspricht dem Ausgabestand "V1.3"
6	
5	
4	
3	
2	
1	
0	

Speicherzugriffe über absolute Adressen

9

Inhalt von Kapitel 9

9.1	Einführung	9 - 4
9.2	Speicherzugriffe über Adresse in AKKU 1	9 - 8
9.2.1	LIR/TIR: 16-bit-Register indirekt laden/transferieren	9 - 9
9.2.2	Beispiele für Zugriffe auf DW > 255	9 - 15
9.2.3	LDI/TDI: 32-bit-Register indirekt laden/transferieren	9 - 17
9.3	Speicherblöcke transferieren	9 - 19
9.4	Operationen mit dem Basisadressregister (BR-Register)	9 - 22
9.4.1	Transferoperationen zwischen Registern	9 - 23
9.4.2	Zugriffe auf den lokalen Speicher	9 - 24
9.4.3	Zugriffe auf den globalen Speicher	9 - 25
9.4.4	Zugriffe auf den Kachelspeicher	9 - 29

9

Speicherzugriffe über absolute Adressen

In diesem Kapitel erfahren Sie, wie Sie mit Hilfe von STEP-5-Operationen und speziellen STEP-5-Registern Daten in bestimmten Speicherbereichen über Absolutadressen ansprechen können.

9.1 Einführung

Die Programmiersprache STEP 5 enthält Operationen, mit denen Zugriffe auf den gesamten Adreßraum möglich sind. Diese Operationen gehören zu den "Systemoperationen".

Die in diesem Kapitel beschriebenen Operationen arbeiten mit 20-bit-Absolutadressen und sind daher abhängig von der Speichergröße, von der Speicherart, von der Peripherie und von den verwendeten CPUs, IPs usw. Ihres Automatisierungsgerätes.



Warnung

Bei einer nicht sachgerechten Anwendung dieser Operationen können STEP-5-Bausteine und Systemdaten überschrieben werden. Dies kann unerwünschte Betriebszustände zur Folge haben. Operationen, die mit absoluten Adressen arbeiten, sollten deshalb nur von Anwendern mit sehr guten Systemkenntnissen benutzt werden.

Lokaler Speicher

Als lokaler Speicher wird der Speicherbereich bezeichnet, der auf jeder CPU vorhanden ist (Anwenderspeicher, BA-, BB-, BS-, BT-Bereich, Zähler, Zeiten, Merker, Prozeßabbild).

Globaler Speicher

Der globale Speicher ist nur einmal für alle CPUs vorhanden und wird über den S5-Bus adressiert.

Speicherorganisation

Speicherbereiche sind **byteweise** oder **wortweise** organisiert.

- Byteweise Organisation: Jede Adresse adressiert ein Byte.
- Wortweise Organisation: Jede Adresse adressiert ein 16-bit-Wort (= 2 Bytes).

Die Organisation des **lokalen** Speichers ist vorgegeben (siehe Kapitel 8)

Die Organisation des **globalen** Speichers hängt vom Typ der gesteckten Baugruppen ab:

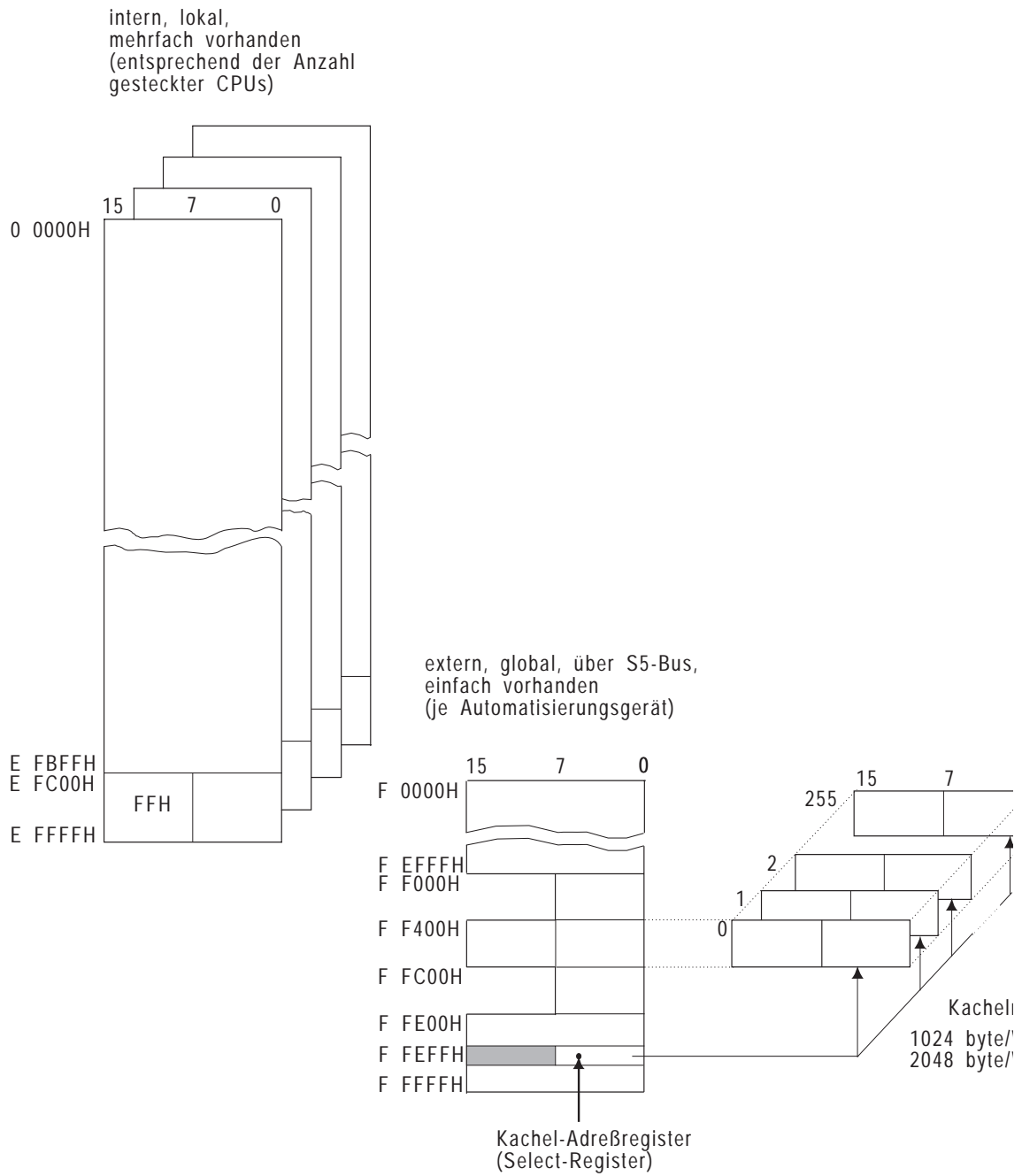


Bild 9-1 Globaler und lokaler Speicher

Speicherzugriffe

Die folgenden Operationen ermöglichen Zugriffe auf lokale bzw. globale Speicherbereiche über absolute Adressen (siehe auch Bild 9-2).

Zugriffe auf den Lokal- und Globalbereich

Sie können sowohl auf den Lokal- als auch auf den Globalbereich zugreifen:

- auf Lokalbereich (Adressen 0 0000H bis E FFFFH) und den Globalbereich (Adressen F 0000H bis F FFFFH) mit:

LIR, TIR, LDI, TDI, TNW, TXB, TXW,

- auf den wortweise organisierten Teil (Adressen 0 0000H bis E FBFFH) bzw. byteweise organisierten Teil (Adressen E FC000 bis E FFFF) des Lokalbereiches mit:

LRW, TRW, LRD, TRD.

Zugriffe nur auf den Globalbereich

Sie können auf folgende Teile des Globalbereichs zugreifen:

- auf den byteweise organisierten Teil des Globalbereiches (Adressen F 0000H bis F FFFFH) mit:

LB GB, LB GW, LB GD, TB GB, TB GW, TB GD, TSG,

- auf den wortweise organisierten Teil des Globalbereichs (Adressen F 0000H bis F FFFFH) mit:

LW GW, LW GD, TW GW, TW GD, TSG .

Zugriffe auf den Kachelbereich

Sie können auf folgende Teile des Kachelbereichs zugreifen:

- auf den byteweise organisierten Teil des Globalbereichs (Adressen F F400H bis F FBFFH, = Kachelbereich):

LB CB, LB CW, LB CD, TB CB, TB CW, TB CD, TSC,

- auf den wortweise organisierten Teil des Globalbereichs (Adressen F F400H bis F FBFFH, = Kachelbereich):

LW CW, LW CD, TW CW, TW CD, TSC

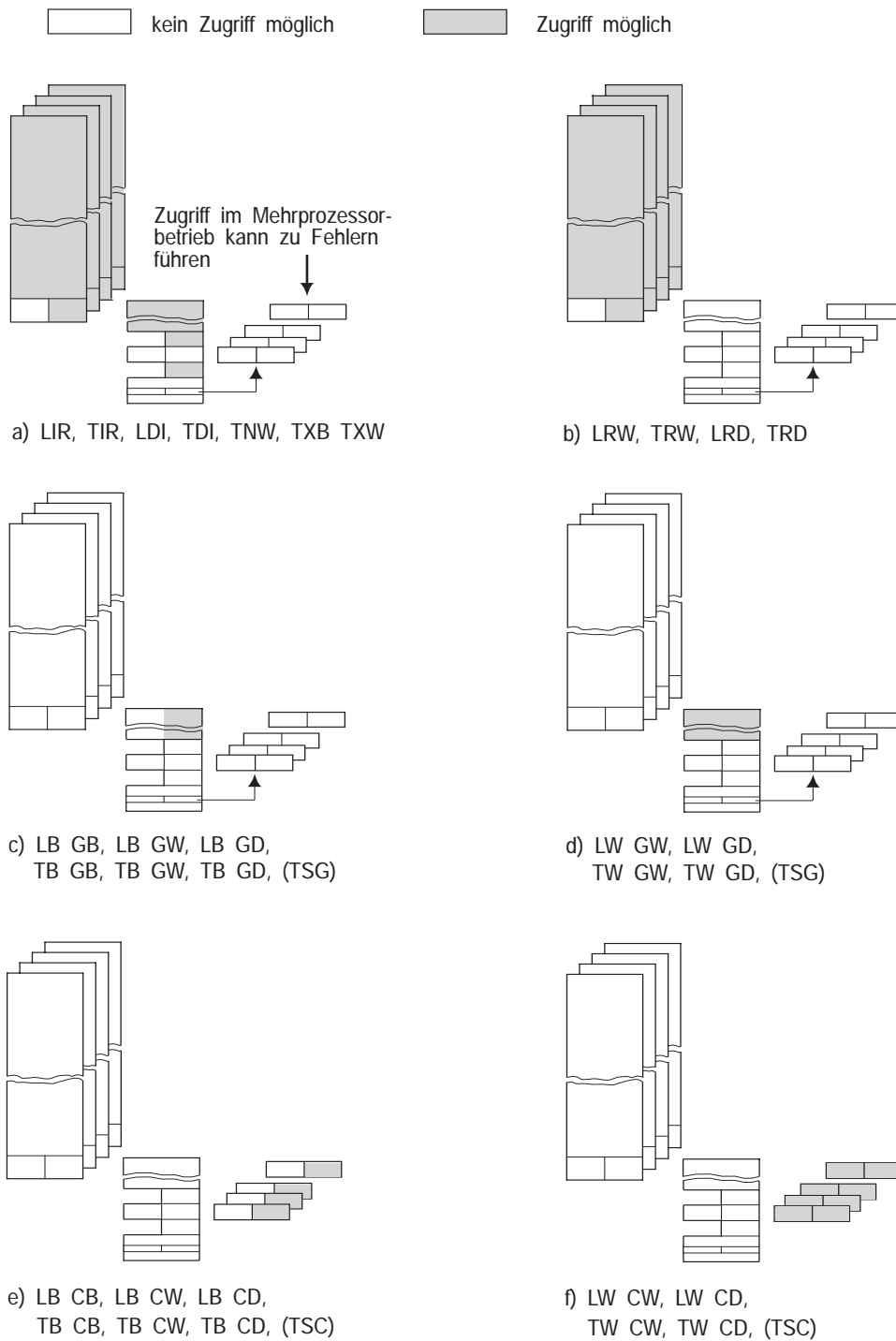


Bild 9-2 Zugriffe auf lokale bzw. globale Speicherbereiche über absolute Adressen

9.2 Speicherzugriffe über Adresse in AKKU 1

Anwendung

Die in diesem Abschnitt aufgeführten Operationen eignen sich vor allem für den Zugriff auf Datenbausteine und sonstige Operandenbereiche. Auf Bausteine, die STEP-5-Programme enthalten (OBs, FBs, PBs und SBs), sollten Sie mit diesen Operationen jedoch **nicht** zugreifen.

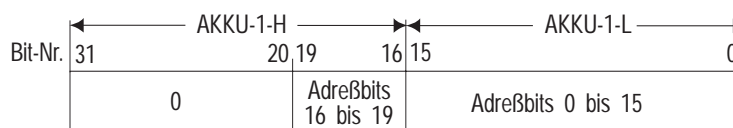
Für die Zugriffe können Sie verschiedene 16- oder 32-bit-breite Register verwenden. Zu diesen Registern gehören die Akkumulatoren AKKU 1 bis AKKU 4 und andere Spezialregister, die von der CPU als Betriebsmittel verwendet werden.

Operationen

Tabelle 9-1 Operationen für indirekte Speicherzugriffe über Register

Operation	Operand	Funktion
LIR	Register-Nr. 0 bis 15	das 16-bit-Register mit dem Inhalt eines durch AKKU 1 adressierten (20-bit-Adresse) Speicherwortes laden
TIR	Register-Nr. 0 bis 15	den Inhalt des 16-bit-Registers in das durch AKKU 1 adressierten (20-bit-Adresse) Speicherwortes laden
LDI	Reg.-Name	das 32-bit-Register mit dem Inhalt der durch AKKU 1 adressierten (20-bit-Adresse) Speicherwörter 'n' und 'n+1' laden
TDI	Reg.-Name	den Inhalt des 32-bit-Registers in die durch AKKU 1 adressierten (20-bit-Adresse) Speicherwörter 'n' und 'n+1' laden

Die Absolutadresse des Speicherwortes bzw. des ersten der beiden



Speicherwörter steht immer im AKKU 1 in folgender Darstellung:

Auf den folgenden Seiten erfahren Sie, **welche Register** Sie bei den Operationen verwenden können.

Einige Beispiele erläutern Ihnen dann, **wie** Sie die Operationen anwenden.

9.2.1

**LIR/TIR: 16-bit-Register
indirekt laden/transferieren**

Die folgende Tabelle zeigt Ihnen, welche Registernummern Sie bei der CPU 948 für die Operationen LIR und TIR verwenden dürfen und wie diese belegt sind.

Tabelle 9-2 16-bit-Register für LIR/TIR

Register-Nr.	Registerbelegung (je 16 bit breit)
0	AKKU-1-H (linkes Wort von AKKU1, Bit 16 - 31) ¹⁾
1	AKKU-1-L (rechtes Wort von AKKU1, Bit 0 - 15) ¹⁾
2	AKKU-2-H
3	AKKU-2-L
5	Baustein-Pointer (Offset)
6	DBA (Datenbaustein-Anfangsadresse)
8	DBL (Datenbaustein-Länge)
9	AKKU-3-H
10	AKKU-3-L
11	AKKU-4-H
12	AKKU-4-L

¹⁾ Soll der Inhalt der adressierten Speicherzelle in das Register '0' oder '1' geladen werden, so wird die in AKKU 1 hinterlegte Adresse überschrieben.

Die Register 4, 7, 13, 14 und 15 sind bei der CPU 948 nicht vorhanden. LIR/TIR-Operationen mit diesen Registernummern dürfen nicht verwendet werden.

*LIR/TIR auf
8-bit-Speicherbereiche*

Wird mit LIR/TIR auf Speicherbereiche zugegriffen, die nur 8 bit breit sind, so beachten Sie, daß

- bei LIR das High-Byte des Registers mit **nicht definierten Werten** überschrieben wird (außer bei Merkern, PAE und PAA; bei diesen Bereichen wird das High-Byte mit FFH beschrieben)
- und
- bei TIR nur das Low-Byte des Registers übertragen wird (das High-Byte des Registers geht verloren).

Die Bilder 9-3 und 9-4 zeigen Ihnen den Unterschied bei Zugriffen mit LIR/TIR auf wort- und byteweise orientierte Speicherbereiche:

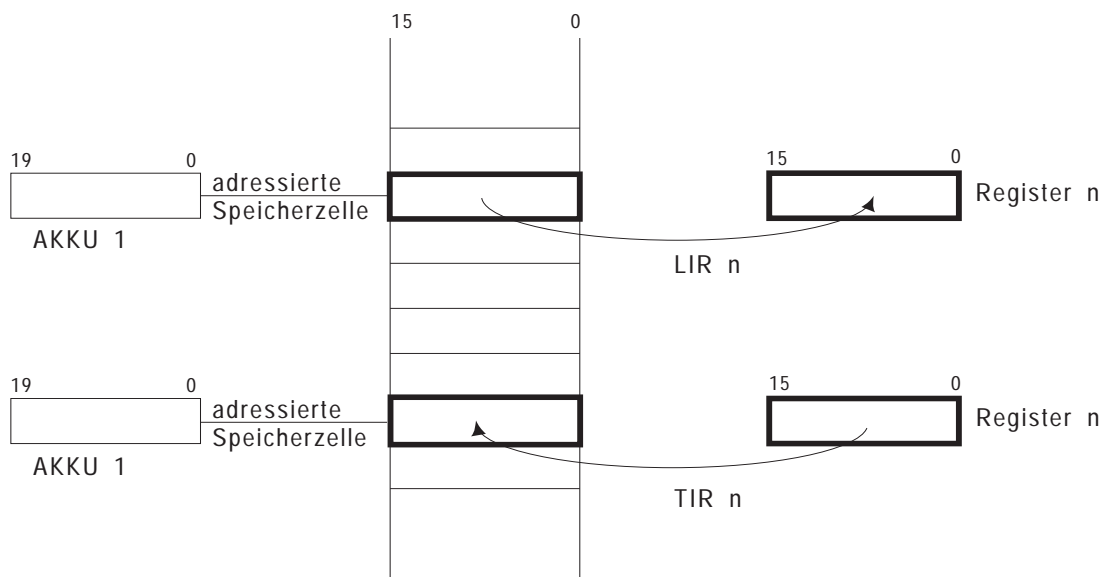


Bild 9-3 LIR/TIR auf 16-bit-Speicherbereiche (wortweise orientiert)

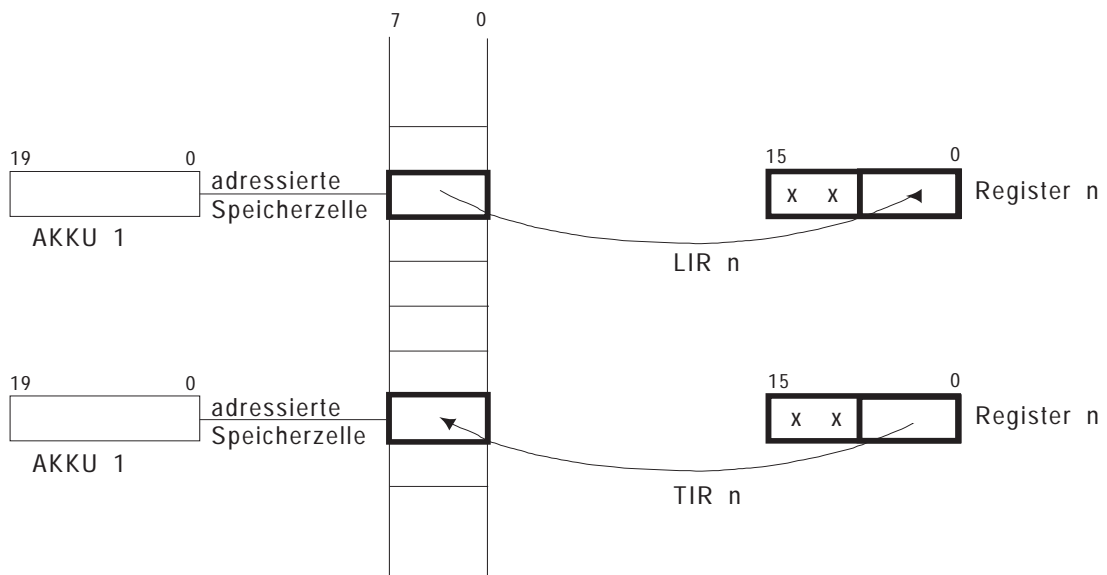


Bild 9-4 LIR/TIR auf 8-bit-Speicherbereiche (byteweise orientiert)

Register 0 bis 3 und 9 bis 12: Die Akkumulatoren werden von der CPU bei der Programmbearbeitung als Zwischenspeicher verwendet. Mit den Befehlen TIR und LIR können Sie die Inhalte der AKKUs in absolut adressierte Speicherzellen transferieren bzw. die Inhalte absolut adressierter Speicherzellen in die AKKUs laden. Die Absolutadresse der Speicherzelle steht jeweils im AKKU 1, Bit-Nr. 0 bis 19.

Beispiele

Der Inhalt der Speicherzelle mit der Adresse E F800 wird ins Merkerwort MW 100 geladen.

```
:L   DH 000E F800  Adresse E F800 der Speicherzelle in AKKU 1 laden
:LIR 1              Inhalt der durch AKKU 1 adressierten Speicherzelle
:              ins Register 1 = AKKU-1-L laden
:T   MW 100        Inhalt der Adresse E F800 im Merkerwort MW 100 ablegen
:BE
```

Den Inhalt des Merkerwortes 200 wird in die Speicherzelle mit der Adresse E F800 transferiert.

```
:L   MW 200        Merkerwort MW 200 in den AKKU 1 laden
:L   DH 000E F800  Adresse E F800, auf die transferiert werden soll, in
:              den AKKU 1 laden (Merkerwort MW 200 nach AKKU 2)
:TIR 3             Inhalt von Register 3 = AKKU-2-L in die durch AKKU 1
:              adressierte Speicherzelle transferieren
:BE
```

Register 6: DBA (Datenbaustein-Anfangs- adresse)

Beim Aufschlagen eines Datenbausteins mit den Befehlen A DB und AX DX wird das Register 6 mit der Adresse des DW 0 im aufgeschlagenen Datenbaustein geladen. Diese Adresse ist in der Bausteina-dreßliste im DB 0 enthalten.

Das DBA-Register wird vor jedem Aufruf des OB 1 gleich '0' gesetzt.

Das DBA-Register bleibt **erhalten**, wenn

- durch eine Sprunganweisung (SPA/SPB) die Programm-bearbeitung in einem anderen Baustein fortgesetzt wird

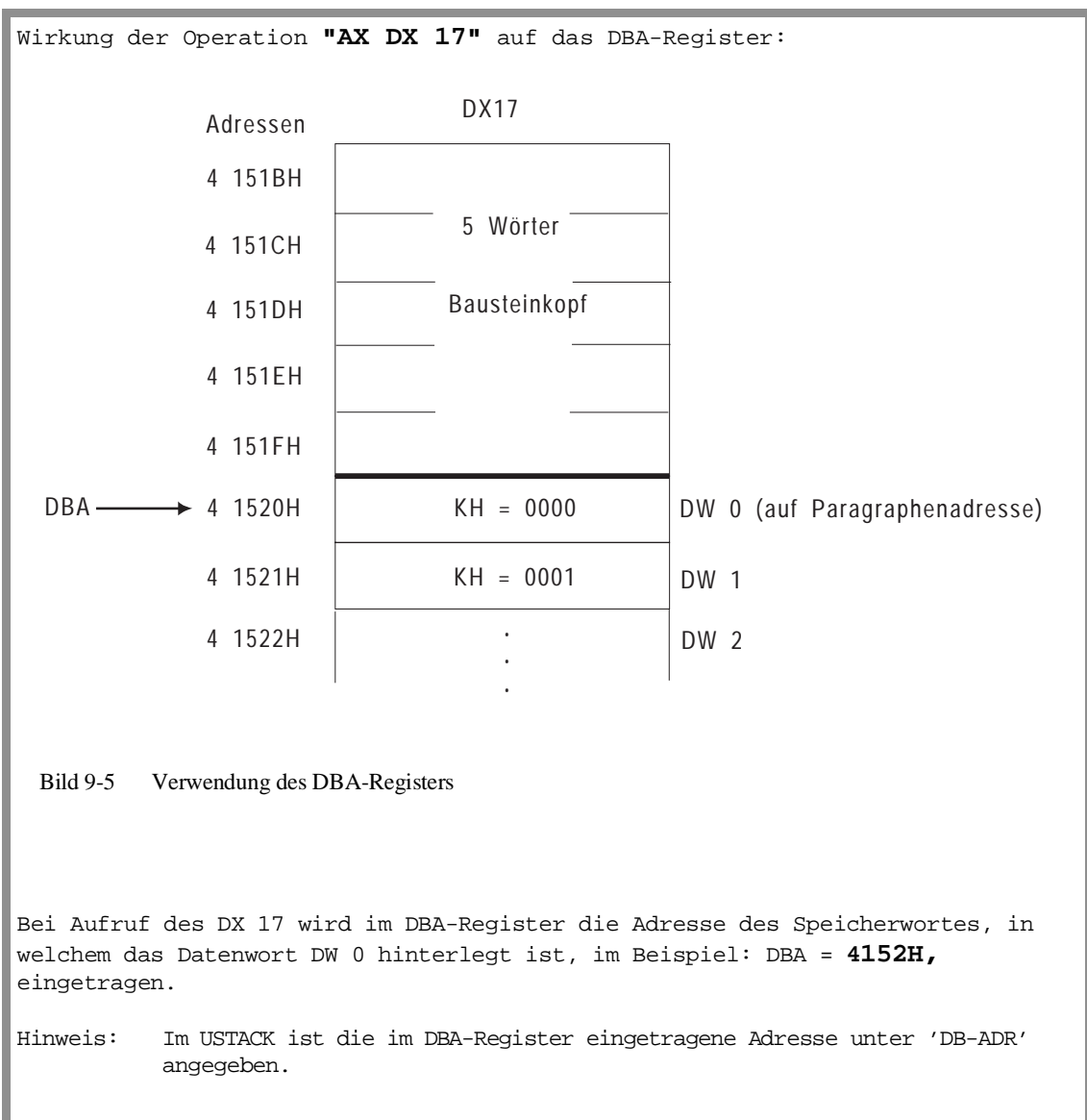
oder

- eine andere Programmbearbeitungsebene eingeschachtelt wird.

Es **ändert** sich, wenn

- ein anderer Datenbaustein aufgeschlagen wird
- oder
- ein Rücksprung in einen übergeordneten Baustein erfolgt nachdem im aufgerufenen Baustein ein neuer Datenbaustein aufgeschlagen wurde (siehe auch Abschnitt 2.4.3).

Beispiel



Register 8:
DBL = Datenbaustein-Länge

Zusätzlich zum DBA-Register wird bei jedem Aufruf eines Datenbausteins das DBL-Register geladen. Es enthält die Länge (in Wörtern) des aufgerufenen Datenbausteins **ohne** Baustein-Kopf. Das DBL-Register wird vor jedem Aufruf des OB 1 gleich '0' gesetzt.

Das DBL-Register bleibt **erhalten**, wenn

- durch eine Sprunganweisung (SPA/SPB) die Programmbearbeitung in einem anderen Baustein fortgesetzt wird
- oder
- eine andere Programmbearbeitungsebene eingeschachtelt wird.

Es **ändert** sich, wenn

- ein anderer Datenbaustein aufgeschlagen wird
- oder
- ein Rücksprung in einen übergeordneten Baustein erfolgt, nachdem im aufgerufenen Baustein ein neuer Datenbaustein aufgeschlagen wurde (siehe auch Abschnitt 2.4.2).

Hinweis

Sie können DBA- und DBL-Register über LIR-Operationen verändern, um auf Datenbaustein-Adressen zuzugreifen, die größer als 255 sind. Das DBA-Register enthält **Paragraphen-adressen**. Achten Sie jedoch darauf, daß mit einer Veränderung des DBA-Registers das DBL-Register nicht automatisch nachgeführt wird und umgekehrt. Dadurch ist eine **Transferfehlerüberwachung nicht mehr gewährleistet!**

Bei der CPU 948 werden Veränderungen am DBA-/DBL-Register **rückgängig gemacht**, sobald der **aktuelle Baustein beendet** wird (BSTACK-Einträge!). Manipulationen am DBA-/DBL-Register wirken daher nur in dem Baustein, in dem sie durchgeführt wurden!

Beispiel

Wirkung der Operation **"AX DX 17"** auf das DBL:

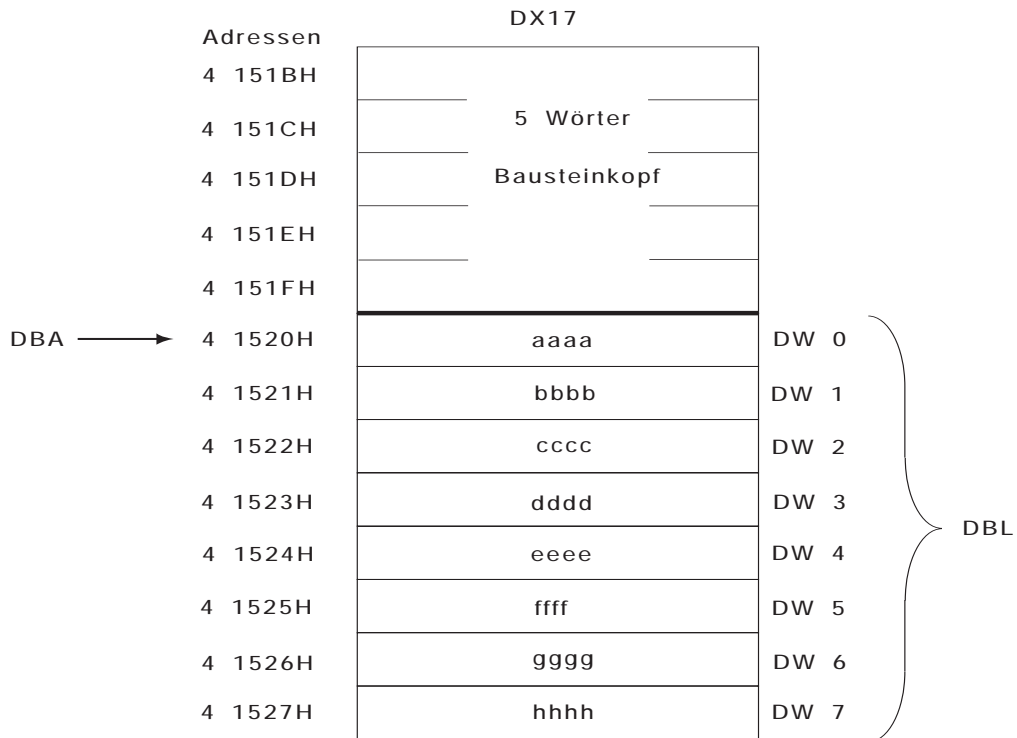


Bild 9-6 Verwendung des DBL-Registers

Bei Aufruf des DX 17 wird im DBL-Register die Anzahl der vorhandenen Datenwörter eingetragen, im Beispiel: DBL = 8 (DW 0 bis DW 7)

Hinweis: Im USTACK ist die im DBL-Register eingetragene Anzahl unter "DBL-REG" angegeben.

9.2.2

Beispiele für Zugriffe auf DW > 255

Beispiel 1:

Der Inhalt des Datenwortes DW 300 im DB 100 soll gelesen und in das Merkerwort MW 100 transferiert werden. (Durch Ändern der fettgedruckten STEP-5-Befehle kann es auch zum Lesen anderer Datenbausteine (DB oder DX) benutzt werden.)

FB 5

NETZWERK 1 0000 Auslesen des DW 300 von DB 100
NAME :LIR DW

0005	:L	DH	000E EC00	Anf.-Adr der DB-Liste
0008	:L	KF	+100	plus der DB-Nummer
000A	:+D			= Adresslisteneintrag des DB 100
000B	:			(Bit 4 bis 19)
000C	:LIR	1		Anf.-Adr. DB 100 nach AKKU 1
000D	:SLD	4		Adr. in physikalische Adr. umwandeln
000E	:L	KF	+300	DW 300 soll ausgelesen werden
0010	:+D			Adr. = Anf.-Adr. des DB + DW-Adr.
0011	:LIR	1		Inhalt des DW 300 nach AKKU 1
0012	:T	MW 100		abspeichern in MW 100
0013	:BE			

Beispiel 2:

Alle Datenwörter eines Datenbausteins sollen mit einer Konstanten beschrieben werden.

Das unten dargestellte Programm beschreibt alle Datenwörter des DB 100 mit der Konstanten KH = A5A5. Nach Ändern der fettgedruckten STEP-5-Befehle kann es auch zum Beschreiben anderer Datenbausteine (DB oder DX) mit beliebigen Werten benutzt werden. Nicht vorhandene Datenbausteine werden erkannt und führen zum Sprung zur Marke NIVO.

Das Programm nutzt drei Akkumulatoren. Innerhalb der Schleife ändert sich die Akkumulatorbelegung nicht.

Der AKKU 1 enthält zunächst die Adresse des ersten Datenwortes und wird mit jedem Schleifendurchlauf um eins erhöht.

Der AKKU 2 enthält die Adresse des letzten Datenwortes + 1. Die Schleife wird abgebrochen, sobald der Inhalt des AKKU 1 gleich ist wie der Inhalt des AKKU 2.

Zum Beschreiben der Datenwörter wird der Befehl TIR 10 verwendet, der den Inhalt des AKKU-3-L (die Konstante) unter der im AKKU 1 stehenden Adresse abspeichert.

AKKU-Belegung innerhalb der Schleife:

AKKU 1: Adresse des **aktuell** zu beschreibenden Datenwortes
AKKU 2: Adresse des **letzten** zu beschreibenden Datenwortes + 1
AKKU 3: Konstante

Fortsetzung von Beispiel 2:

Belegung der Merker:

MW 10: Bit 4 bis 19 der Anfangsadresse des DB/DX (zeigt auf DW 0)
 MW 12: Länge des DB/DX (Anzahl der Datenwörter)
 MD 14: Adresse des letzten vorhandenen Datenwortes im DB/DX + 1 (phys. Adr.)

FB 6

```

NETZWERK 1          0000          Fuellen eines DB mit einer Konstanten
NAME :FILL DB
0005      :
0006      :L   DH   000E EC00  Anf-Adr. der DB-Liste
0009      :L   KF   +100    plus der DB-Nummer
000B      :+D              = Adresslisteneintrag des DB 100
000C      :              (Bit 4 bis 19)
000D      :LIR          1      Anf-Adr. DB 100 nach AKKU 1
000E      :T   MW   10      Anf-Adr. zwischenspeichern
000F      :L   KB   0      (Paragraphenadresse)
0010      :!=F          wenn Anf-Adr. = 0, dann ist
0011      :SPB   =NIVO      der DB nicht vorhanden
0012      :
0013      :L   MW   10      Anf-Adr. des DB (1. DW)
0014      :SLD          4      Adr. umwandeln in physikalische Adresse
0015      :L   KB   1      Ermitteln der DB-Laenge ueber
0016      : -D          5. Wort im Bausteinkopf
0017      :LIR          1      Laenge inkl. Bausteinkopf nach AKKU 1
0018      :ADD   BF   -5      Anzahl DW = Gesamtlaege - 5 Woerter (Bst.-Kopf)
0019      :
001A      :T   MW   12      Laenge zwischenspeichern
001B      :
001C      :L   MW   12      Anzahl der Datenwoerter +
001D      :L   MW   10      Anfangsadresse (DW 0 - in physik. Adresse
001E      :SLD          4      umgewandelt -)
001F      :+D          ergibt
0020      :T   MD   14      Adresse des letzten DW + 1
0021      :
0022      :L   KH   A5A5    Konstante, mit der alle Datenwoerter
0024      :              beschrieben werden sollen
0025      :L   MD   14      Adresse des letzten DW + 1
0026      :ENT          Konstante nach AKKU-3-L schieben
0027      :              (= Register 10)
0028      :L   MW   10      Adresse des 1. Datenwortes (DW 0)
0029      :SLD          4      umwandeln in physikalische Adresse
002A      :
002B SCHL      :          Schleife:
002C      :              AKKU 1: Adresse des zu beschreibenden DW
002D      :              AKKU 2: Adresse letztes DW + 1
002E      :              AKKU 3: Konstante
002F      :TIR          10      Wert von AKKU-3-L in das DW mit der Adresse, die
0030      :              in AKKU 1 steht, abspeichern
0031      :
0032      :ADD   DH   0000 0001  Adresse um 1 erhoehen
0035      :
0036      :><D          Abfrage, ob letztes DW
0037      :SPB   =SCHL      erreicht (wenn nicht, zurueck in die Schleife)
0038      :
0039 WEIT      :          Fortsetzung des Programms ...
003A      :          nachdem alle DW beschrieben wurden ...
003B      :
003C      :BEA
003D      :
003E NIVO      :          falls DB 100 nicht vorhanden ist
    
```

9.2.3

**LDI/TDI: 32-bit-Register
indirekt laden/transferieren**

Die folgende Tabelle zeigt Ihnen, welche Registernamen Sie bei der CPU 948 für die Operationen LDI und TDI verwenden dürfen und wie diese belegt sind.

Tabelle 9-3 32-bit-Register für LDI/TDI

Register-Name	Registerbelegung (je 32 bit breit)
A1	AKKU 1 (AKKU1, Bit 0 - 31) ¹⁾
A2	AKKU 2 (AKKU1, Bit 0 - 31)
SA	STEP-Adreßzähler (Bit 0 - 19)
BA	BA-Register (Bausteinanfangsadresse, Bit 0 - 19)
BR	BR-Register (Basisadreßregister, Bit 0 - 19)

¹⁾ Soll der Inhalt der adressierten Speicherzelle in das A1-Register geladen werden, so wird die in AKKU 1 hinterlegte Adresse überschrieben.

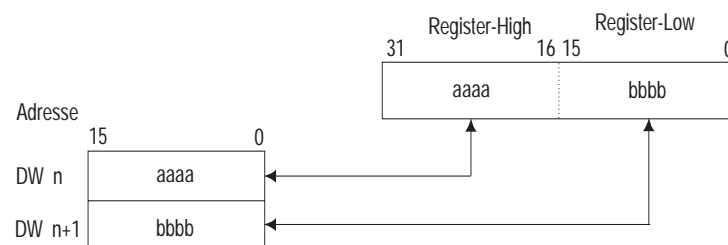
Byteadressen

Werden mit LDI/TDI Byteadressen angesprochen, so beachten Sie, daß

- bei LDI die High-Bytes des angesprochenen Registers mit **nicht definierten Werten** überschrieben werden (außer bei Merkern, PAE und PAA; bei diesen Bereichen wird das High-Byte mit FFH beschrieben)

und

- bei TDI nur die Low-Bytes des angesprochenen Registers übertragen werden (die High-Bytes gehen verloren – siehe Beispiel auf der folgenden Seite).

Datenablage bei LDI/TDI

SA-Register:
SAZ = STEP-Adreßzähler

Im SA-Register ist am Ende einer Operationsbearbeitung die 20-bit-Absolutadresse **der Operation** eingetragen, die als **nächste** zu bearbeiten ist.

BA-Register:
Bausteinanfangsadresse

Im BA-Register ist während der Programmbearbeitung des STEP-5-Anwenderprogramms die 20-bit-Absolutadresse für die nächste zu bearbeitende Operation im übergeordneten Baustein (entspricht der Rücksprungadresse) eingetragen .

BR-Register:
Freies Basisadreßregister

Das Basisadreßregister (20 bit) ermöglicht Adreßrechnungen und indirekte Register-Lade- und Transferoperationen ohne Verwendung der Akkumulatoren für die Adreßangabe. Es kann während der STEP-5-Programmbearbeitung frei genutzt werden.

Beispiel für TDI im Bytebereich

```
:L   DH 1234 5678   Datum laden
:L   DH 000E FC00   Adresse von Merkerbyte
:                                     MB 0 laden
:TDI A2             Inhalt von AKKU 2 abspei-
:                                     chern

E FC00 = 34   (die Werte '12H' und '56H' aus AKKU 2
E FC01 = 78   gehen verloren)
```

9.3 Speicherblöcke transferieren

Anwendung

Mit den in diesem Abschnitt aufgeführten Operationen können Sie Datenbereiche mit einer Länge bis zu 255 Wörtern, die in bestimmten Adreßbereichen liegen, umspeichern.

Operationen

Tabelle 9-4 Operationen für Blocktransfer

Operation	Operand	Funktion
TNW	0 bis 255	Blocktransfer 0 bis 255 Wörter ¹⁾ (im 16-bit-Speicherbereich)
TXB	--	Blocktransfer vom 8-bit- in den 16-bit-Speicher
TXW	--	Blocktransfer vom 16-bit- in den 8-bit-Speicher

¹⁾ Kann auch für Transfers "von Byte- in Bytebereich" benutzt werden.

Parameter

Blocklänge

Bei TNW: Operand = Anzahl Wörter (0 bis 255)

Bei TXB/TXW AKKU 3 = Anzahl Wörter (0 bis 127)

Endadresse des Quellbereichs

AKKU 2 = Endadresse des Quellbereichs (20 bit)

Endadresse des Zielbereichs

AKKU 1 = Endadresse des Zielbereichs (20 bit)

Quellbereich und Zielbereich müssen **vollständig** in einem der in Tabelle 9-5 genannten Speicherbereiche liegen und **dürfen sich nicht überschneiden**.

erlaubte Speicherbereiche

Tabelle 9-5 Für TNW, TXB und TXW erlaubte Speicherbereiche

Adressen	Speicherbereich
0 0000H bis C FFFFH	Anwenderspeicher: 16-bit-Bereich (abhängig vom Speicherausbau)

Adressen	Speicherbereich
Fortsetzung der Tabelle 9-5:	
E 8000H bis E 9FFFH	System-RAM: Systemdaten, 16 bit
E B000H bis E FBFFH	Systemdaten (BA/BS, Zeiten,Zähler usw.), 16 bit
E A000H bis E AFFFH	S-Merker, 8 bit, Low-Byte im 16-bit Wort (High-Byte nicht definiert)
E FC00H bis E FFFFH	Merker, Prozeßabbild, 8 bit (High-Byte = FFH)
F 0000H bis F FFFFH	Peripherie, 8/16 bit

(Siehe auch Kapitel 8)

Ablauf

Der Blocktransfer erfolgt dekrementierend, d. h. er beginnt seine Übertragung mit der höchsten Adresse des Quellbereichs (=Endadresse) und beendet sie mit der niedrigsten.

TNW, TXB und TXW

Die Befehle TNW, TXB und TXW sind langlaufende STEP-5-Operationen, die nur durch NETZ AUS und QVZ unterbrochen werden können.

Besonderheiten

Unterbrechungen durch NETZ AUS

Wird die Ausführung einer der Operationen durch Netzausfall (NAU) unterbrochen und anschließend ein Wiederanlauf durchgeführt, so läuft die Operation danach nicht vom Unterbrechungspunkt weiter, sondern wird **von Anfang an** bearbeitet.

Unterbrechungen durch QVZ

Tritt während der Übertragung ein Quittungsverzug (QVZ) auf, so wird die Operation unterbrochen und der entsprechende Fehler-OB aufgerufen.
Als Fehleradresse wird immer **die** Adresse angegeben, an der ein Fehler aufgetreten ist (siehe Abschnitt 5.6.3).

ADF während der Ausführung

Tritt während der Übertragung einmal oder mehrmals ein Adressierfehler (ADF) auf, so werden zuerst alle Teilblöcke übertragen und dann vor Ausführung der nächsten Operation einmalig der OB 25 aufgerufen.

Beispiel

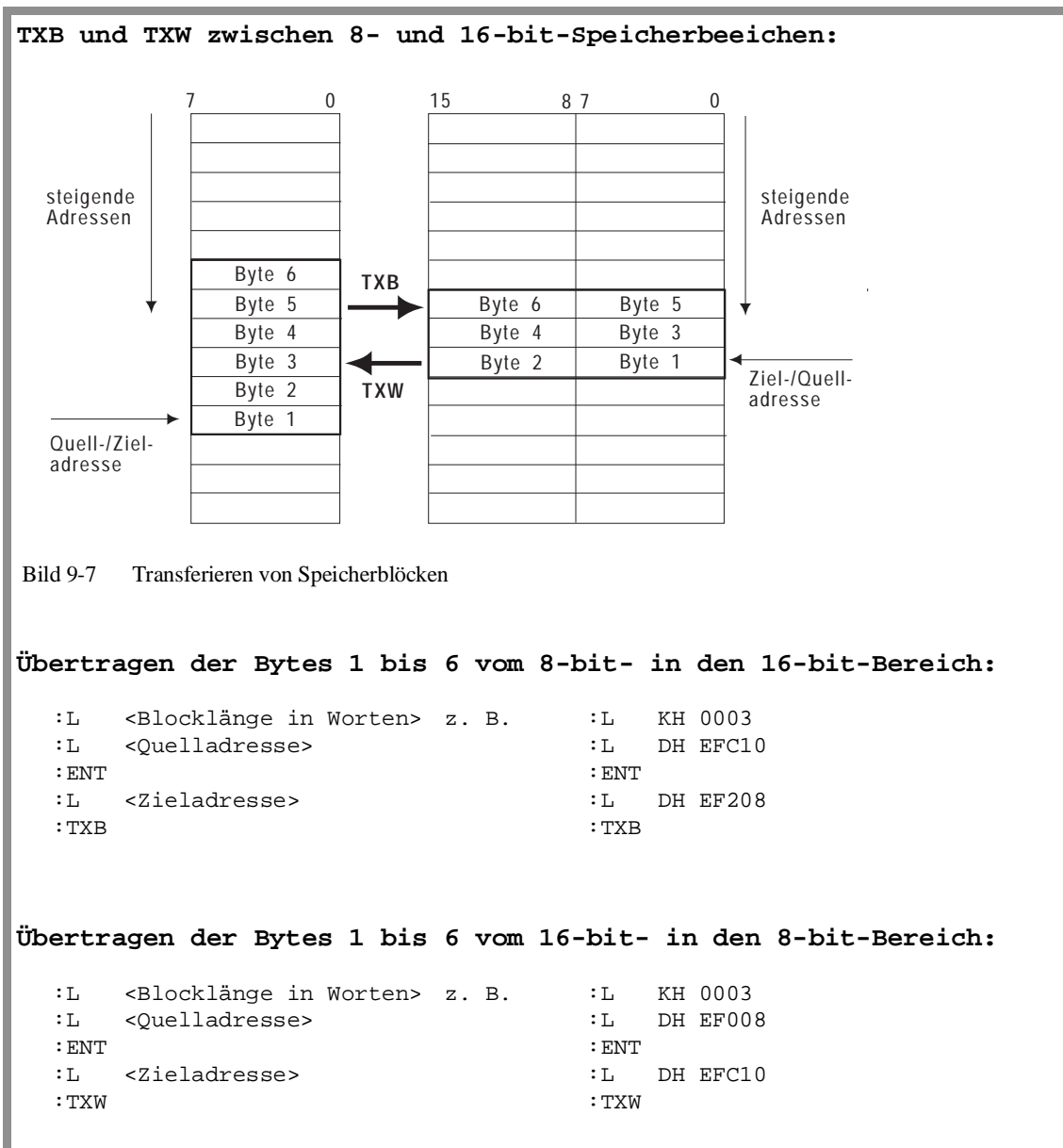


Bild 9-7 Transferieren von Speicherblöcken

9.4 Operationen mit dem Basisadressregister (BR-Register)

Anwendung

Das Basisadressregister (20 bit) ermöglicht Ihnen Adreßrechnungen und indirekte Register-Lade- und -Transferoperationen ohne Verwendung der AKKUs für die Adressierung.

Zugegriffen wird auf die Speicherzelle, deren absolute Adresse sich als Summe von BR-Registerinhalt und einer Konstanten errechnet:

$$\text{Absolute Adresse} = \text{BR-Registerinhalt} + \text{Konstante}$$

Operationen

Tabelle 9-6 Lade- und Rechenoperationen mit dem BR-Register

Operation	Operand	Funktion
MBR	Konstante (0H bis F FFFFH)	das BR-Register mit einer 20-bit-Konstanten laden
ABR	Konstante (-32 768 bis +32 767)	eine 16-bit-Konstante zum Inhalt des BR-Registers addieren

Veränderung des BR-Registers

- Das BR-Register bleibt **erhalten**, wenn durch eine Sprunganweisung ('SPA FB'/'SPB FB') in einem **anderen Baustein derselben Programmverarbeitungsebene** fortgesetzt wird.
- Das BR-Register bleibt nach **Einschachtelung** einer anderen Programmbearbeitungsebene **erhalten**.
- Bei **Aufruf** einer anderen Programmbearbeitungsebene durch das Systemprogramm wird das BR-Register auf '0' gesetzt

9.4.1 Transferoperationen zwischen Registern

Anwendung

Die in diesem Abschnitt aufgeführten Operationen können Sie dazu benutzen, schnell Werte zwischen den Registern AKKU 1 (32 bit), STEP-Adreßzähler (SAZ – 20 bit) und BR-Register (20 bit) auszutauschen.

Operationen

Tabelle 9-7 Register-Register-Operationen

Operation	Operand	Funktion
MAS	--	den Inhalt des AKKU 1 in den STEP-Adreßzähler (20 bit) transferieren
MAB	--	den Inhalt des AKKU 1 in das Basisadreßregister (20 bit) transferieren
MSA	--	den Inhalt des STEP-Adreßzählers (20 bit) in den AKKU 1 transferieren ¹⁾
MSB	--	den Inhalt des STEP-Adreßzählers (20 bit) in das Basisadreßregister (20 bit) transferieren
MBA	--	den Inhalt des Basisadreßregisters (20 bit) in den AKKU 1 transferieren ¹⁾
MBS	---	den Inhalt des Basisadreßregisters (20 bit) in den STEP-Adreßzähler (20 bit) transferieren

¹⁾ Die Bits 2^{20} bis 2^{31} werden auf '0' gesetzt.

Wie die Register bei den Operationen verändert werden, zeigt Ihnen das folgende Bild.

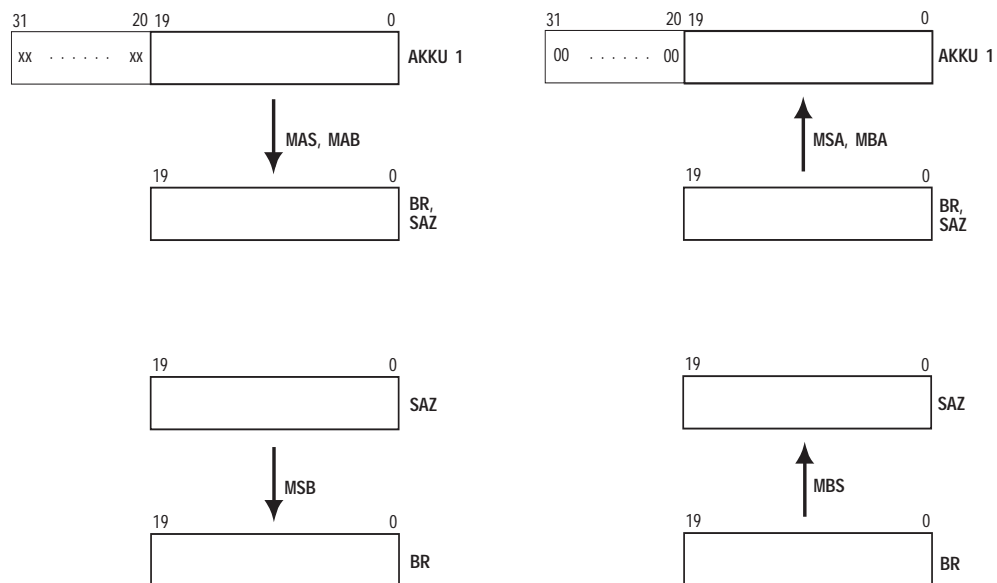


Bild 9-8 Register-Register-Transferoperationen

9.4.2 Zugriffe auf den lokalen Speicher

Anwendung

Die folgenden Operationen ermöglichen den Zugriff auf den lokalen, wortweise organisierten Speicher über eine absolute Speicheradresse. Die Absolutadresse ist die Summe vom BR-Registerinhalt und der im Befehl enthaltenen 16-bit-Konstanten (-32768 bis +32767).

Operationen

Tabelle 9-8 Operationen für Zugriffe auf den lokalen Speicher

Operation	Operand	Beschreibung
LRW	Konstante (-32768 bis +32767)	die angegebene Konstante zum Inhalt ¹⁾ des BR-Registers addieren und das so adressierte Wort in den AKKU-1-L laden
LRD	Konstante (-32768 bis +32767)	die angegebene Konstante zum Inhalt ¹⁾ des BR-Registers addieren und das so adressierte Doppelwort in den AKKU 1 laden
TRW	Konstante (-32768 bis +32767)	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-L in das so adressierte Wort transferieren

Operation	Operand	Beschreibung
Fortsetzung der Tabelle 9-8:		
TRD	Konstante (-32768 bis +32767)	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU 1 in das so adressierte Doppelwort transferieren

¹⁾ AKKU 2_{neu} = AKKU 1_{alt}

Fehlerreaktion

Liegt die errechnete Adresse der Speicherzelle nicht zwischen 0 0000H und E FFFFH, so erkennt die CPU einen Lade-/Transferfehler (TLAF) und ruft den **OB 32** auf. Ist der OB 32 nicht geladen, so geht die CPU mit der Fehleranzeige TLAF (USTACK) in den Stoppzustand.

9.4.3 Zugriffe auf den globalen Speicher

Anwendung

Mit den folgenden Operationen können Sie auf den globalen, **byte- oder wortweise** organisierten Speicher über eine absolute Speicheradresse zugreifen. Die Absolutadresse ist die Summe vom BR-Register-Inhalt und der im Befehl enthaltenen Konstanten (-32768 bis 32767).

Testen und Setzen einer Belegzelle im Globalbereich

Der Zugriff einzelner CPUs auf gemeinsam genutzte Speicherbereiche kann über eine Belegzelle gesteuert werden. Jedem gemeinsam genutzten Speicherbereich wird eine Belegzelle zugeordnet, die von allen beteiligten CPUs vor jedem Zugriff getestet werden muß. Die Belegzelle enthält entweder den Wert '0' oder die Steckplatzkennung der CPU, die gerade den Speicherbereich benutzt und ihn durch **Beschreiben der Belegzelle mit '0' wieder freigeben muß**. (Beachten Sie auch die Erläuterungen zu den Operationen "Semaphor setzen/SES" und "Semaphor freigeben/SEF" in Abschnitt 3.5.5.)

Die Operation TSG bearbeitet das Testen und Setzen einer Belegzelle.

Operation	Operand	Beschreibung
TSG	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und die so adressierte Belegzelle testen und setzen

Ablauf

Als Belegtzelle wird das Low-Byte des durch BR-Register + Konstante adressierten Wortes verwendet. Falls der Inhalt des Low-Bytes '0' ist, trägt der TSG-Befehl die Steckplatzkennung in die Belegtzelle ein.

Das Testen (= Lesen) und das eventuelle Belegen (= Schreiben) bilden eine Programmeinheit, die nicht unterbrochen werden kann.

Ergebnis

Das Testergebnis ist über die Anzeigen ANZ 0 und ANZ 1 auswertbar:

ANZ 1	ANZ 0	Bedeutung
0	0	Inhalt der Belegtzelle ist '0'; die CPU trägt ihre Steckplatzkennung ein.
1	0	Die eigene Steckplatzkennung ist bereits in der Belegtzelle eingetragen.
0	1	Die Belegtzelle enthält eine fremde Steckplatzkennung.

Hinweis

Die Operation TSG muß von **allen** CPUs verwendet werden, die synchronisiert auf einen **gemeinsamen globalen Speicherbereich** zugreifen sollen.

Fehlerreaktion

Die Absolutadresse muß zwischen F 0000H und F FFFFH liegen. Ist dies nicht der Fall, so erkennt die CPU einen Transferfehler (TLAF) und ruft den **OB 32** auf. Ist der OB 32 nicht geladen, so geht die CPU mit der Fehleranzeige TLAF (USTACK) in den Stoppzustand.

**Lade und Transfer-
operationen auf den
bytwweise organisierten
globalen Speicher**

Tabelle 9-9 Operationen für Zugriffe auf den bytwweise organisierten globalen Speicher

Operation	Operand	Beschreibung
LB GB	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Byte in den AKKU-1-LL laden ^{1) 3)}
LB GW	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Wort in den AKKU-1-L laden ^{2) 3)}
LB GD	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Doppelwort in den AKKU 1 laden ³⁾
TB GB	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-LL in das so adressierte Byte transferieren
TB GW	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-L in das so adressierte Wort transferieren
TB GD	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU 1 in das so adressierte Doppelwort transferieren

¹⁾ AKKU-1-LH und AKKU-1-H werden auf '0' gesetzt.

²⁾ AKKU-1-H wird auf '0' gesetzt.

³⁾ AKKU 2_{neu} := AKKU 1_{alt}

Fehlerreaktion

Die Absolutadresse muß liegen

- zwischen F 0000H und F FFFFH (bei LB GB, TB GB),
 - zwischen F 0000H und F FFFEH (bei LB GW, TB GW)
- oder
- zwischen F 0000H und F FFFCH (bei LB GD, TB GD).

Ist dies nicht der Fall, so erkennt die CPU einen Lade-/Transferfehler (TLAF) und ruft den **OB 32** auf. Ist der OB 32 nicht geladen, so geht die CPU mit der Fehleranzeige TLAF (USTACK) in den Stoppzustand.

Lade- und Transferoperationen für den wortweise organisierten globalen Speicher

Tabelle 9-10 Operationen für Zugriffe auf den wortweise organisierten globalen Speicher

Operation	Operand	Beschreibung
LW GW	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Wort in den AKKU-1-L laden ^{1) 2)}
LW GD	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Doppelwort in den AKKU 1 laden ²⁾
TW GW	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-L in das so adressierte Wort transferieren
TW GD	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU 1 in das so adressierte Doppelwort transferieren

¹⁾ AKKU-1-H wird auf '0' gesetzt.

²⁾ AKKU 2_{neu} : = AKKU 1_{alt}

Fehlerreaktion

Die Absolutadresse muß liegen

- zwischen F 0000H und F FFFFH (bei LW GW, TW GW)

oder

- zwischen F 0000H und F FFFEH (bei LW GD, TW GD).

Ist dies nicht der Fall, so erkennt die CPU einen Lade-/Transferfehler (TLAF) und ruft den **OB 32** auf. Im AKKU 1 steht die Fehlerkennung 1A01H. Ist der OB 32 nicht geladen, so geht die CPU mit der Fehleranzeige TLAF (USTACK) in den Stoppzustand.

9.4.4 Zugriffe auf den Kachel- speicher

Anwendung

Mit den folgenden Operationen können Sie auf **byte- oder wortweise** organisierte Kacheln über eine absolute Speicheradresse zugreifen. Die Absolutadresse ist die Summe vom BR-Register-Inhalt und der im Befehl enthaltenen Konstanten (-32768 bis 32767).

Prinzipieller Ablauf des Kachelzugriffs

Der Globalbereich enthält im Bereich der Adressen F F400H bis F FBFFH ein "Fenster" zum Einblenden eines von max. 256 Speicherbereichen (= Kacheln). Eine Kachel belegt max. 2K Adressen und kann byteweise oder wortweise organisiert sein. Vor jedem Zugriff auf den Kachelbereich wird eine der 256 Kacheln durch Eintrag ihrer Kachelnummer in das **Select-Register** (Kachel-Adreßregister) ausgewählt. Der Vorgang "Beschreiben des Select-Registers und anschließender Zugriff auf den Kachelbereich" ist ununterbrechbar.

Vor jedem Zugriff (Laden/Transferieren) auf den Kachelbereich muß eine der 256 Kacheln aufgeschlagen werden. Dazu übergeben Sie die Nummer der aufzuschlagenden Kachel im AKKU-1-L; diese Nummer wird durch den Befehl ACR in das CPU-interne **Kachel-Register** eingetragen. Alle folgenden Kacheloperationen schreiben vor dem Kachelzugriff den Inhalt des Kachel-Registers in das Select-Register der entsprechenden Baugruppen auf dem S5-Bus.

Veränderung des Kachel-Registers

- Das Kachel-Register **bleibt erhalten**, wenn ein anderer Baustein aufgerufen wird.
- Wenn das Kachelregister in einem Baustein verändert wird, so bleibt sein **Wert erhalten**, wenn am Bausteinende in den aufrufen den Baustein zurückgesprungen wird.
- Das Kachelregister bleibt **nach** Einschachtelung einer anderen Programmbearbeitungsebene **erhalten**.

Aufschlagen einer Kachel

Operation	Parameter	Beschreibung
ACR		diejenige Kachel aufschlagen, deren Nummer im AKKU-1-L steht, zulässige Werte: 0 bis 255

Die Kachelnummer muß zwischen 0 und 255 liegen. Ist dies nicht der Fall, so erkennt die CPU einen Substitutionsfehler (SUF) und ruft den **OB 27** auf. Ist der OB 27 nicht geladen, so geht die CPU in den Stoppzustand.

Testen und Setzen einer Belegzelle im Kachelbereich

Der Zugriff einzelner CPUs auf gemeinsam genutzte Speicherbereiche kann über eine Belegzelle gesteuert werden. Jedem gemeinsam genutzten Speicherbereich wird eine Belegzelle zugeordnet, die von allen beteiligten CPUs vor jedem Zugriff getestet werden muß. Die Belegzelle enthält entweder den Wert '0' oder die Steckplatzkennung der CPU, die gerade den Speicherbereich benutzt und ihn durch **Beschreiben der Belegzelle mit '0' wieder freigeben muß**. (Beachten Sie auch die Erläuterungen zu den Operationen "Semaphor setzen/SES" und "Semaphor freigeben/SEF" in Abschnitt 3.5.5.)

Der Befehl TSC bearbeitet das Testen und Setzen einer Belegzelle auf der aufgeschlagenen Kachel.

Operation	Operand	Beschreibung
TSC	-32 768 bis +32 767	die angegebene Konstante zum Inhalt des BR-Registers addieren und die so adressierte Belegzelle auf der aufgeschlagenen Kachel testen und setzen

Ablauf

Als Belegzelle wird das Low-Byte des durch BR-Register + Konstante adressierten Wortes verwendet. Falls der Inhalt des Low-Bytes '0' ist, trägt der TSC-Befehl die Steckplatzkennung in die Belegzelle ein.

Das Testen (= Lesen) und das eventuelle Belegen (= Schreiben) bilden eine Programmeinheit, die nicht unterbrochen werden kann.

Ergebnis

Das Ergebnis der Operation TSC ist über die Anzeigen ANZ 0 und ANZ 1 auswertbar:

ANZ 1	ANZ 0	Bedeutung
0	0	Inhalt der Belegtzelle ist '0'; die CPU trägt ihre Steckplatzkennung ein.
1	0	Die eigene Steckplatzkennung ist bereits in der Belegtzelle eingetragen.
0	1	Die Belegtzelle enthält eine fremde Steckplatzkennung.

Hinweis

Die Operation TSC muß von **allen** CPUs verwendet werden, die **synchronisiert** auf einen **gemeinsamen globalen Speicherbereich** (Kachelbereich) zugreifen sollen.

Fehlerreaktion

Die Belegtzelle muß auch auf der entsprechenden Baugruppe und in der gemeinsam benutzten Kachel zwischen F F400H und F FBFFH liegen. Ist dies nicht der Fall, so erkennt die CPU einen Transferfehler (TLAF) und ruft den **OB 32** auf. Ist der OB 32 nicht geladen, so geht die CPU mit der Fehleranzeige TLAF (USTACK) in den Stoppzustand.

Lade- und Transferoperationen für byteweise organisierte Kacheln

Tabelle 9-11 Operationen für Zugriffe auf byteweise organisierte Kacheln

Operation	Operand	Beschreibung
LB CB	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Byte in der aufgeschlagenen Kachel in den AKKU-1-LL laden ^{1) 3)}
LB CW	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Wort in der aufgeschlagenen Kachel in den AKKU-1-L laden ^{2) 3)}
LB CD	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Doppelwort in der aufgeschlagenen Kachel in den AKKU 1 laden ³⁾

Operation	Operand	Beschreibung
Fortsetzung der Tabelle 9-11:		
TB CB	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-LL in das so adressierte Byte in der aufgeschlagenen Kachel transferieren
TB CW	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-L in das so adressierte Wort in der aufgeschlagenen Kachel transferieren
TB CD	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU 1 in das so adressierte Doppelwort in der aufgeschlagenen Kachel transferieren

¹⁾ AKKU-1-LH und AKKU-1-H werden auf '0' gesetzt.

²⁾ AKKU-1-H wird auf '0' gesetzt.

³⁾ AKKU 2_{neu} = AKKU 1_{alt}

Fehlerreaktion

Die Absolutadresse muß liegen

- zwischen F F400H und F FBFFH (bei LB CB, TB CB),
 - zwischen F F400H und F FBFEH (bei LB CW, TB CW)
- oder
- zwischen F F400H und F FBFCH (bei LB CD, TB CD).

Ist dies nicht der Fall, so erkennt die CPU einen Lade-/Transferfehler (TLAF) und ruft den **OB 32** auf. Ist der OB 32 nicht geladen, so geht die CPU mit der Fehleranzeige TLAF (USTACK) in den Stoppzustand.

Lade- und Transferoperationen für wortweise organisierte Kacheln

Tabelle 9-12 Operationen für Zugriffe auf wortweise organisierte Kacheln

Operation	Operand	Beschreibung
LW CW	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Wort in der aufgeschlagenen Kachel in den AKKU-1-L laden ¹⁾
LW CD	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Doppelwort in der aufgeschlagenen Kachel in den AKKU 1 laden ²⁾
TW CW	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-L in das so adressierte Wort in der aufgeschlagenen Kachel transferieren
TW CD	-32768 bis +32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU 1 in das so adressierte Doppelwort in der aufgeschlagenen Kachel transferieren

¹⁾ AKKU-1-H wird auf '0' gesetzt.

²⁾ AKKU 2_{neu} = AKKU 1_{alt}

Fehlerreaktion

Die Absolutadresse muß liegen

- zwischen F F400H und F FBFFH (bei LW CW, TW CW)

oder

- zwischen F F400H und F FBFEH (bei LW CD, TW CD).

Ist dies nicht der Fall, so erkennt die CPU einen Lade-/Transferfehler (TLAF) und ruft den **OB 32** auf. Ist der OB 32 nicht geladen, so geht die CPU mit der Fehleranzeige TLAF (USTACK) in den Stoppzustand.

Mehrprozessorbetrieb und -kommunikation im AG S5-155U

10

Inhalt von Kapitel 10

10.1	Mehrprozessorbetrieb	10 - 4
10.1.1	Wann verwenden Sie den Mehrprozessorbetrieb?	10 - 4
10.1.2	Welche Kommunikationsmechanismen gibt es?	10 - 4
10.1.3	Daten über Koppelmerker austauschen	10 - 5
10.1.4	Daten über Hantierungsbausteine austauschen	10 - 8
10.1.5	Was müssen Sie für den Mehrprozessorbetrieb programmieren?	10 - 9
10.1.6	Wie erstellen Sie den Datenbaustein DB 1?	10 - 9
10.1.7	Was müssen Sie über den Anlauf im Mehrprozessorbetrieb wissen?	10 - 13
10.1.8	Was Sie über den Testbetrieb wissen sollten	10 - 14
10.2	Mehrprozessorkommunikation	10 - 15
10.2.1	Einführung	10 - 15
10.2.2	Wie Sender und Empfänger identifiziert werden	10 - 16
10.2.3	Warum Daten zwischengespeichert werden.	10 - 17
10.2.4	Wie der Zwischenspeicher abgearbeitet und verwaltet wird.	10 - 18
10.2.5	Was Sie beim System-Anlauf beachten müssen	10 - 21
10.2.6	Was Sie beim Aufrufen der Kommunikations-OBs beachten müssen	10 - 22
10.2.7	Wie parametrieren Sie die Kommunikations-OBs?	10 - 23
10.2.8	Wie können Sie die Ausgangsparameter auswerten?	10 - 24
10.3	Laufzeiten der Kommunikations-OBs	10 - 31
10.4	Funktion INITIALISIEREN (OB 200)	10 - 33
10.4.1	Funktion	10 - 33
10.4.2	Aufrufparameter	10 - 35
10.4.3	Eingangsparameter	10 - 35
10.4.4	Ausgangsparameter.	10 - 38

10.5	Funktion SENDEN (OB 202)	10 - 40
10.5.1	Funktion	10 - 40
10.5.2	Aufrufparameter	10 - 40
10.5.3	Eingangsparameter	10 - 40
10.5.4	Ausgangsparameter	10 - 42
10.6	Funktion SENDE-TEST (OB 203)	10 - 45
10.6.1	Funktion	10 - 45
10.6.2	Aufrufparameter	10 - 45
10.6.3	Eingangsparameter	10 - 45
10.6.4	Ausgangsparameter	10 - 45
10.7	Funktion EMPFANGEN (OB 204)	10 - 47
10.7.1	Funktion	10 - 47
10.7.2	Aufrufparameter	10 - 47
10.7.3	Eingangsparameter	10 - 47
10.7.4	Ausgangsparameter	10 - 48
10.8	Funktion EMPFANGS-TEST (OB 205)	10 - 51
10.8.1	Funktion	10 - 51
10.8.2	Aufrufparameter	10 - 51
10.8.3	Eingangsparameter	10 - 51
10.8.4	Ausgangsparameter	10 - 51
10.9	Anwendungen	10 - 53
10.9.1	Aufruf der Sonderfunktions-OB über Funktionsbausteine	10 - 53
10.9.2	Übertragen von Datenbausteinen	10 - 60
10.9.3	Erweiterung des Koppelmerkerbereichs	10 - 66

10

Mehrprozessorbetrieb und -kommunikation im AG S5-155U

Dieses Kapitel gibt Ihnen zunächst einige Hinweise, wann Sie den Mehrprozessorbetrieb anwenden können und welcher Datenaustausch dabei möglich ist. Es gibt Ihnen Informationen darüber, was Sie als Programmierer für den Mehrprozessorbetrieb tun und beachten müssen (Abschnitt 10.1).

Schließlich erhalten Sie ausführliche Anleitungen mit Anwendungsbeispielen für den Austausch größerer Datenmengen im Mehrprozessorbetrieb (Mehrprozessorkommunikation – Abschnitte 10.2 bis 10.9).

10.1 Mehrprozessorbetrieb

Begriffsdefinition

Sie befinden sich im Mehrprozessorbetrieb, sobald Sie eine Koordinatorbaugruppe gesteckt haben, unabhängig davon, wieviele CPUs oder CPs/IPs gesteckt sind. Die CPUs müssen lückenlos gesteckt sein!

10.1.1

Wann verwenden Sie den Mehrprozessorbetrieb?

- Wenn Ihr Anwenderprogramm zu umfangreich für eine CPU ist und Speicherplatz knapp wird, so verteilen Sie Ihr Programm auf mehrere CPUs.
- Wenn ein bestimmter Teil Ihrer Anlage besonders schnell bearbeitet werden soll, so trennen Sie den betreffenden Programmteil aus dem Gesamtprogramm heraus und lassen diesen von einer eigenen "schnellen" CPU bearbeiten.
- Wenn Ihre Anlage aus mehreren Teilen besteht, die gut voneinander abzugrenzen und damit relativ eigenständig zu steuern bzw. zu regeln sind, so lassen Sie Anlagenteil 1 von CPU 1, Anlagenteil 2 von CPU 2 usw. bearbeiten.

Beachten Sie zum Mehrprozessorbetrieb unbedingt die entsprechenden Informationen in Ihrem Systemhandbuch. Sie können Ihnen besonders bei der Entscheidung helfen, welche CPUs Sie für Ihr Problem am besten einsetzen.

10.1.2

Welche Kommunikationsmechanismen gibt es?

- Für den zyklischen Austausch binärer Daten zwischen den CPUs (CPU 948, CPU 946/947, CPU 928B, CPU 928 und CPU 922) oder zwischen CPUs und Kommunikationsprozessoren stehen Ihnen die "**Koppelmerker**" zur Verfügung,
- Beim Austausch größerer Datenmengen (z. B. ganzer Datenbausteine) zwischen CPU 948, CPU 946/947, CPU 928B, CPU 928 und CPU 922 werden Sie unterstützt durch die "**Sonderfunktionen für die Mehrprozessorkommunikation**" OB 200 bis OB 205 (mehr Information hierzu finden Sie ab Abschnitt 10.2).
- Zur Kommunikation mit Intelligenten Peripheriebaugruppen (IPs) und mit Kommunikationsprozessoren (CPs) stehen Ihnen die "**Handtierungsbausteine**" zur Verfügung (diese müssen Sie getrennt bestellen).

10.1.3

Daten über Koppelmerker austauschen

Für den zyklischen Austausch binärer Daten stehen Ihnen die "Koppelmerker" zur Verfügung. Diese dienen in erster Linie zum **byteweisen** Übertragen von Informationen.

Dieser Datentransfer kann erfolgen zwischen

CPU(s) ↔ CPU(s)

CPU(s) ↔ Kommunikationsprozessor(en)

Das Systemprogramm überträgt die Koppelmerker einmal pro Zyklus. Bei einem Datentransfer zwischen CPUs werden die Koppelmerker physikalisch auf dem Koordinator zwischengespeichert.

Koppelmerker sind Merkerbytes, die transferiert werden. Sie werden für jede CPU im Datenbaustein DB 1 als Ein- oder Ausgangskoppelmerker definiert. Haben Sie z.B. das Merkerbyte 50 auf der CPU 1 als Koppelmerker**ausgang** definiert, so wird dessen Signalzustand zyklisch über den Koordinator zu der CPU übertragen, auf der das Merkerbyte 50 als Koppelmerker**eingang** definiert ist.

Hinweis

Es erfolgt **keine** Fehlermeldung, wenn das Koppelmerkerbyte zwar physikalisch vorhanden, aber nur einseitig beschrieben und nie ausgelesen wird oder umgekehrt

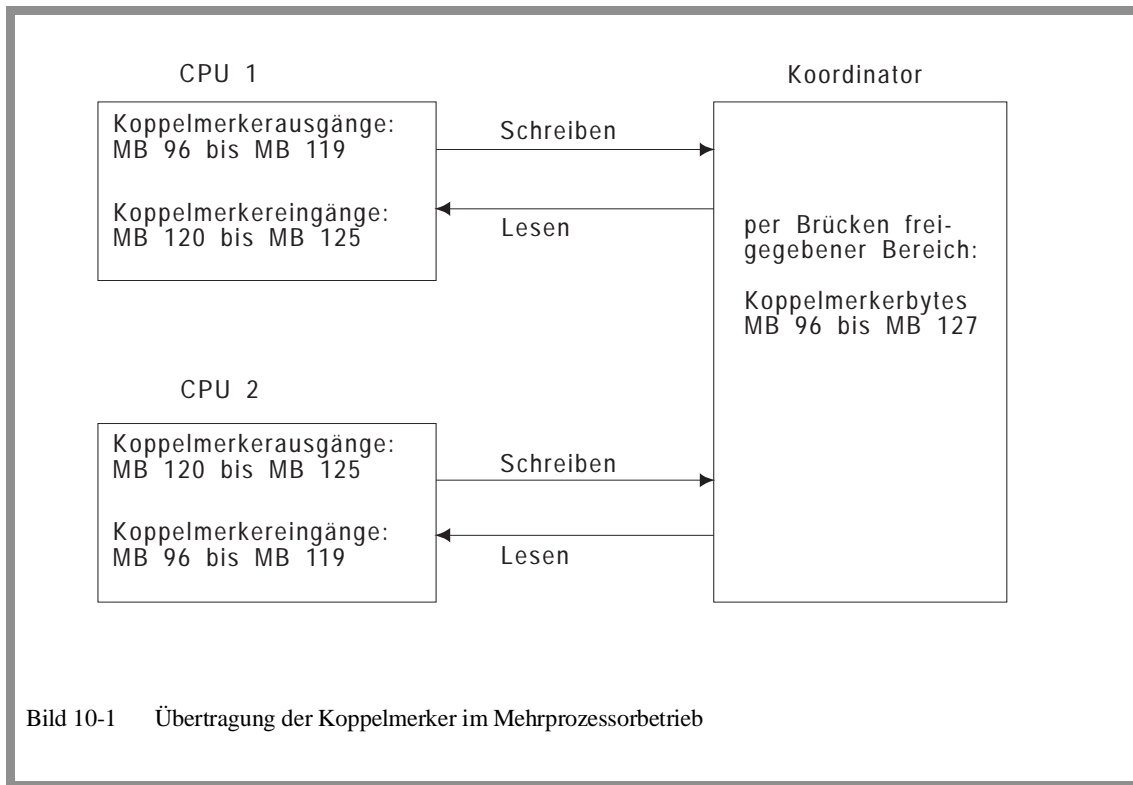
Speicherbereich

Der Speicherbereich für die Koppelmerker auf dem Koordinator und den Kommunikationsprozessoren umfaßt bei der CPU 948 die Adressen **F F200H bis F F2FFH**. Auf einer CPU/einem Kommunikationsprozessor stehen Ihnen 256 Koppelmerkerbytes zur Verfügung.

Brückeneinstellung

Um Doppelbelegungen zu vermeiden, müssen Sie (von den 256 Ihnen zur Verfügung stehenden Koppelmerkerbytes) auf den Baugruppen KOR bzw. CP eine Rangierung vornehmen. Es können Blöcke zu je 32 Bytes ein- oder ausgeblendet werden. (In Ihrem Systemhandbuch finden Sie die Information, wie die Brücken einzustellen sind.)

Beispiel



Hinweis

Als Koppelmerker dürfen Sie nur diejenigen Merkerbytes angeben, die auf dem Koordinator oder dem (den) entsprechenden CP(s) freigegeben sind. **S-Merker können Sie nicht als Koppelmerker verwenden!**

Haben Sie auf einer oder mehreren CPUs ein bestimmtes Merkerbyte als Koppelmerkereingang definiert, so muß es auf einer anderen CPU oder einem anderen CP als Koppelmerkerausgang definiert sein. Und: Ein Merkerbyte darf nur auf **einer CPU als Koppelmerkerausgang** gekennzeichnet sein; Sie können es jedoch auf z. B. drei weiteren CPUs als Koppelmerkereingang definieren!

Die auf einer CPU nicht als Koppelmerker definierten Merkerbytes können wie "normale" Merker verwendet werden!

Geben Sie im DB 1 nur die Koppelmerkerbytes an, die Sie tatsächlich benötigen: Je kleiner die Anzahl der Koppelmerkerbytes, um so kürzer die Übertragungszeit!

Datenaustausch zwischen CPUs und Kommunikationsprozessoren

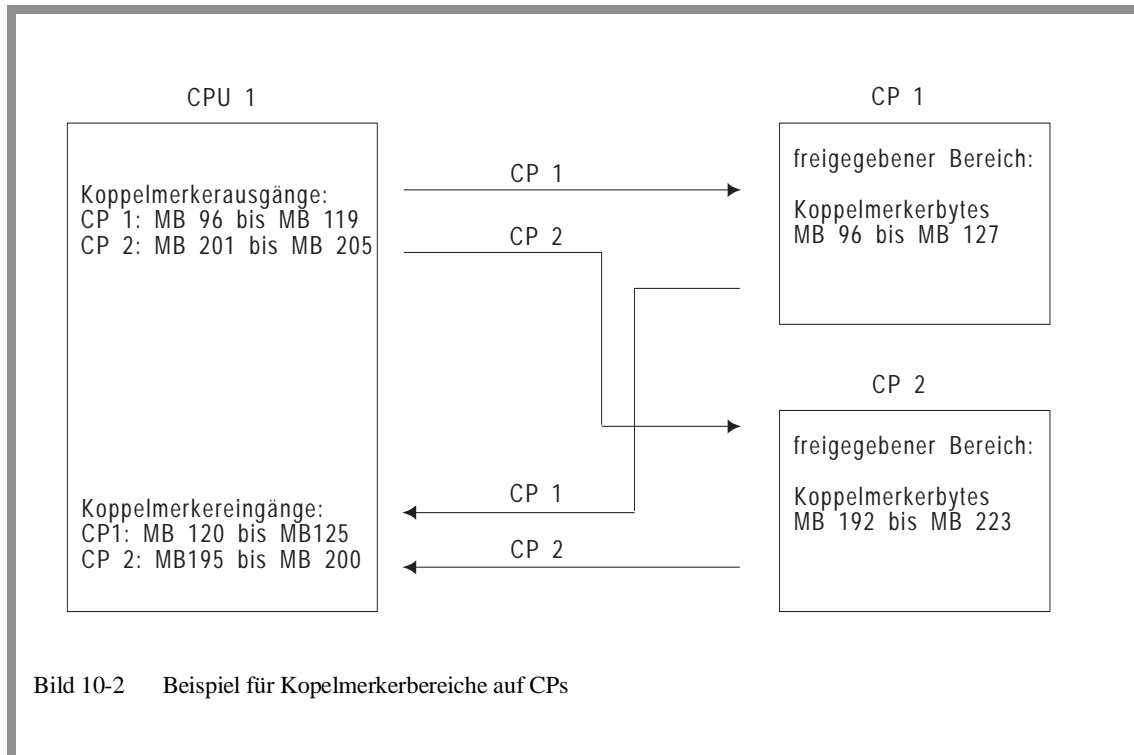
Sollen Daten zwischen einer CPU und einem Kommunikationsprozessor übertragen werden, müssen Sie die benötigte Anzahl der Koppelmerker auf dem Kommunikationsprozessor (CP) freigeben. Auch dort stehen Ihnen 256 byte zur Verfügung, die in Bereiche von 32 byte eingeteilt werden können.

Wenn von einer CPU Daten zu mehreren Kommunikationsprozessoren übertragen werden sollen, dürfen sich die auf den Kommunikationsprozessoren und dem Koordinator freigegebenen Bereiche **nicht überschneiden**, damit Adressen nicht mehrfach belegt werden.

Wenn Sie Koppelmerker **sowohl** auf dem Koordinator **als auch** auf einem oder mehreren Kommunikationsprozessoren benutzen wollen, so müssen Sie ebenfalls eine Doppeladressierung vermeiden: Teilen Sie die Koppelmerker auf dem KOR und den CPs in Bereiche von je 32 Byte ein; diejenigen Koppelmerkerbytes, die Sie auf dem Kommunikationsprozessor verwenden, blenden Sie auf dem Koordinator durch Entfernen von Brücken aus (siehe Systemhandbuch).

Auch hier gilt, daß ein bestimmtes Merkerbyte nur auf **einer** CPU als Koppelmerkerausgang definiert werden darf. Hingegen kann ein bestimmtes Merkerbyte auf mehreren CPUs als Eingangskoppelmerker gekennzeichnet sein.

Beispiel



*Übertragung der
Koppelmerker im
Mehrprozessorbetrieb*

Die im DB 1 angegebenen Koppelmerkerbytes werden am Ende jedes Programmzyklus zusammen mit der Aktualisierung des Prozeßabbildes dann übertragen, wenn die CPU die Busfreigabe erhalten hat.

Der Koordinator erteilt reihum jeder CPU das Busfreigabesignal. Dabei darf eine CPU jeweils nur **ein** Byte übertragen. Durch diese verzahnte Übertragung kann es vorkommen, daß zusammengehörige Koppelmerkerinformationen auseinandergerissen werden und anschließend mit falschen Werten gearbeitet wird.

Wenn Informationen übertragen werden sollen, die mehr als **ein** Byte umfassen, so können Sie durch eine entsprechende Parametrierung im Datenbaustein DX 0 erreichen, daß alle im DB 1 angegebenen Koppelmerker blockweise übertragen werden (siehe Kapitel 7). Solange eine CPU mit der Koppelmerkerübertragung beschäftigt ist, kann sie von einer anderen CPU nicht unterbrochen werden. Da die nächste CPU mit ihrer Übertragung warten muß, wird die zyklische Programmbearbeitung dieser CPU solange verzögert.

Beachten Sie, daß bei dieser Einstellung im DX 0 u. U. die Zykluszeit erheblich verlängert wird.

*Mehrprozessor-
kommunikation*

Für die Übertragung von Datenbausteinen, genauer: von Datenblöcken mit einer Größe von max. 64 byte (= 32 Datenwörtern), stehen Ihnen – in die CPU integriert – folgende Sonderfunktionen zur Verfügung:

- OB 200: INITIALISIEREN: Vorbesetzen
- OB 202: SENDEN: Senden eines Datenblocks
- OB 203: SENDE-TEST: Sendemöglichkeit testen
- OB 204: EMPFANGEN: Empfangen eines Datenblocks
- OB 205: EMPFANGS-TEST: Empfangsmöglichkeit testen

**10.1.4
Daten über Hantierungs-
bausteine austauschen**

Hantierungsbausteine sind mehrprozessorfähig. Eine spezielle Parametrierung des Mehrprozessorbetriebs ist bei ihnen nicht erforderlich. Nähere Informationen zu den Hantierungsbausteinen entnehmen Sie bitte dem entsprechenden Handbuch.

10.1.5

Was müssen Sie für den Mehrprozessorbetrieb programmieren?

- Damit der Koordinator die Zugriffe der einzelnen CPUs auf den Peripheriebereich koordinieren kann, **müssen Sie den Datenbaustein DB 1 programmieren**. Auch dann, wenn die CPU keine Peripherie oder Koppelmerker benutzt, muß ein (leerer) DB 1 vorhanden sein. (Mehr Informationen dazu finden Sie in Abschnitt 10.1.6)
- Der Datenbaustein DX 0 muß ebenfalls vorhanden sein. Parametrieren Sie ihn so, daß die Prozeßalarme über Eingangsbyte EB 0 abgeschaltet und die **Systeminterrupts aktiviert** sind.

10.1.6

Wie erstellen Sie den Datenbaustein DB 1?

Im Mehrprozessorbetrieb müssen Sie für jede CPU den Datenbaustein DB 1 programmieren. Dadurch legen Sie fest, mit welchen **Ein- und Ausgängen** (Byteadressen 0-127) und mit welchen **Koppelmerkerein- und -ausgängen** die jeweilige CPU arbeiten soll.

Hinweis

Nur die im DB 1 definierten Ein- und Ausgangsbytes werden bei der Aktualisierung des Prozeßabbilds berücksichtigt!

Eingabe/Änderung des DB 1

- Erstellen/Ändern am PG über DB-1-Maske
- oder
- durch Editieren des DB 1 als Datenbaustein am PG und anschließendes Übertragen zur CPU.

Hinweis

Der eingegebene bzw. geänderte DB 1 wird nur über die Anlaufart NEUSTART der CPU übernommen!

Erstellen über DB-1-Maske

1. Wählen Sie an Ihrem PG den Editor für die DB-1-Maske (siehe Bild 10-3).
2. Tragen Sie die gewünschten Werte für "Digitale Eingänge" usw. als Dezimalzahlen ein.

3. Übernehmen Sie die eingetragenen Werte durch Betätigen der Übernahmetaste am PG.
Das PG generiert danach den DB 1.
4. Übertragen Sie den DB 1 in die CPU.

Hinweis

Ein Eintrag im Parameterfeld "Zeitenblocklaenge" wird nicht berücksichtigt! Sie müssen diesen Parameter im DX 0 angeben (siehe Kapitel 7).

Beispiel für DB-1-Maske

DB 1	<u>Peripheriezuteilung:</u>
Digitale Eingaenge:	, 0, 1, 2, 3, 7, 10, , , , , , , , , , , ,
Digitale Ausgaenge:	, 0, 2, 4, 12, , , , , , , , , , , , , ,
Koppelmerker Eingaenge:	, 50, 51, 60, , , , , , , , , , , , , , , ,
Koppelmerker Ausgaenge:	, 70, 72, 100, , , , , , , , , , , , , , , ,
Zeitenblocklaenge:	, ,

Bild 10-3 PG-Maske zum Generieren des DB 1

Editieren des DB 1 als Datenbaustein

1. Schreiben Sie in die Datenwörter 0, 1 und 2 die DB-1- Anfangskennung:
 - DW 0: KH = 4D41 ('M' 'A')
 - DW 1: KH = 534B ('S' 'K')
 - DW 2: KH = 3031 ('0' '1')

2. Geben Sie anschließend (ab Datenwort 3) die einzelnen Operandenbereiche ein. Vor jedem Operandenbereich müssen Sie eine bestimmte Kennung eingeben. Die möglichen Kennwörter sind:

Kennwort für digitale Eingänge KH = DE00
 Kennwort für digitale Ausgänge KH = DA00
 Kennwort für Eingangskoppelmerker KH = CE00
 Kennwort für Ausgangskoppelmerker KH = CA00

Anschließend an das Kennwort listen Sie die Nummern der verwendeten Ein- und Ausgänge im Festpunktformat auf.

3. Schließen Sie die Einträge mit der DB-1-Endekennung "KH = EEEE" ab und übertragen Sie den DB 1 in die CPU.

Hinweis

Die Reihenfolge der Einträge ist beliebig. Beachten Sie dabei, daß das Prozeßabbild der Ein- und Ausgänge in genau **der umgekehrten Reihenfolge** aktualisiert wird, in der Sie die **Adressen im DB 1** hinterlegt haben (d. h. der letzte Eintrag zuerst).

Mehrfacheinträge gleicher Bytes, z.B. für Testzwecke, sind möglich. Auch hier beachten Sie bitte, daß das Prozeßabbild mehrfach eingetragener Bytes mehrfach aktualisiert wird.

Beispiel für das Editieren des DB 1

```

DB1      FD: CPU948ST.S5D

0:      KH = 4D41;          DW 0-2:
1:      KH = 534B;          Anfangskennung
2:      KH = 3031;          für DB 1
3:      KH = DE00;          Kennwort für digitale Eingänge
4:      KF = +00000;        Eingangsbyte 0
5:      KF = +00001;        Eingangsbyte 1
6:      KF = +00002;        Eingangsbyte 2
7:      KF = +00003;        Eingangsbyte 3
8:      KF = +00007;        .
9:      KF = +00010;        Eingangsbyte 10
10:     KH = DA00;          Kennwort für digitale Ausgänge
11:     KF = +00000;        Ausgangsbyte 0
12:     KF = +00002;        Ausgangsbyte 2
13:     KF = +00004;        .
14:     KF = +00012;        Ausgangsbyte 12
15:     KH = CE00;          Kennwort für Koppelmerkereingänge
16:     KF = +00050;        Merkerbyte 50
17:     KF = +00051;        .
18:     KF = +00060;        Merkerbyte 60
19:     KH = CA00;          Kennwort für Koppelmerkerausgänge
20:     KF = +00070;        Merkerbyte 70
21:     KF = +00072;        .
22:     KF = +00100;        Merkerbyte 100
23:     KH = EEEE;          Endekennung
24:
    
```

Übernahme des DB 1

Bei NEUSTART wird der DB 1 vom Systemprogramm übernommen. Es überprüft dabei, ob die im DB 1 angegebenen Ein- und Ausgänge bzw. Koppelmerker auf entsprechenden Baugruppen quittieren. Falls nicht, geht die CPU mit DB-1-Fehler in den Stoppzustand mit langsamem Blinken der STOP-LED. Ihr Anwenderprogramm wird dann nicht bearbeitet.

Sobald Sie einen DB 1 programmiert haben und dieser durch die Anlaufart NEUSTART von der CPU übernommen worden ist, gelten folgende Regeln:

- Zugriffe auf Peripheriebaugruppen über das Prozeßabbild sind nur für die im DB 1 angegebenen Ein- und Ausgänge möglich (Operationen L.../T... ...EB, ...EW, ...ED, ...AB, ...AW, ...AD und Verknüpfungsoperationen mit Ein- und Ausgängen). Zugriffe auf nicht im DB 1 eingetragene Prozeßabbild-Adressen führen zu einem Adressierfehler.
- Direktes **Laden** von Peripheriebytes unter Umgehung des Prozeßabbildes mit den Operationen L PB/L PY, L PW, L QB, L QW ist für alle quittierenden Eingänge – unabhängig von einem Eintrag im DB 1 – möglich.
- Direkter **Transfer** (T PB/T PY, T PW) auf Bytes von 0 bis 127 ist nur für die im DB 1 angegebenen Ausgänge möglich, da beim Direkttransfer zusätzlich das Prozeßabbild beschrieben wird. Schreibende Zugriffe auf nicht im DB 1 eingetragene Peripherie-Adressen führen zu einem Adressierfehler.
- **Transfer ohne Prozeßabbild** :
Direkter Transfer auf Byteadressen >127 ist **unabhängig von einem Eintrag im DB 1** möglich.
Direkter Transfer auf Byteadressen des Erweiterten Peripheriebereichs (T QB, T QW) ist ebenfalls unabhängig von einem Eintrag im DB 1 möglich.

10.1.7

Was müssen Sie über den Anlauf im Mehrprozessorbetrieb wissen?

So starten Sie den Koordinator im Mehrprozessorbetrieb:

Ausgangszustand: Die Betriebsartenschalter aller gesteckten CPUs stehen auf RUN. Der Betriebsartenschalter des Koordinators steht auf STOP.

Hantierung: Betätigen Sie den Betriebsartenschalter am Koordinator von STOP nach RUN. (Der Anlauf des Automatisierungsgerätes im Mehrprozessorbetrieb **allein** durch das Starten des Koordinators ist nur dann möglich, wenn der Stoppzustand auch tatsächlich vom Koordinator verursacht worden ist.)

oder:

Ausgangszustand: Die Betriebsartenschalter aller gesteckten CPUs und der des Koordinators stehen auf RUN.

Hantierung: Durch die PG-Funktion "AG-Start" starten Sie die CPU, die den Stoppzustand verursacht hat, in der erwünschten Anlaufart.

Anlauf der einzelnen CPUs

Die Anlaufart der einzelnen CPUs richtet sich nun danach, ob und wie sie zwischenzeitlich im Stoppzustand bedient worden sind. Es ist damit möglich, daß einzelne CPUs einen MANUELLEN WIEDERANLAUF, andere einen NEUSTART durchführen.

Sind die CPUs in der Zwischenzeit **nicht bedient worden**, so führen sie einen **MANUELLEN WIEDERANLAUF** durch.

Hinweis

Durch die unterschiedlichen Anlaufarten können, falls das Automatisierungsgerät vorher im ZYKLUS war, über die Koppelmerker falsche Signalzustände von einer CPU an eine andere weitergegeben werden. Dies verhindern Sie durch entsprechende Programmierung der Anlauforganisationsbausteine OB 20, OB 21 und OB 22.

Durch Aufruf der Sonderfunktion OB 223 können Sie zusätzlich überprüfen, ob die Anlaufarten aller beteiligten CPUs gleich sind (siehe Kapitel 6).

*Spannungsausfall/
Spannungswiederkehr*

Bei Spannungsausfall und anschließender Spannungswiederkehr wird der Koordinator automatisch mitgestartet. In diesem Fall führen alle CPUs einen **AUTOMATISCHEN WIEDERANLAUF** bzw. einen **AUTOMATISCHEN NEUSTART** durch, je nach Einstellung im DX 0 (siehe Kapitel 7).

Der ANLAUF der einzelnen CPUs ist im Mehrprozessorbetrieb zeitlich synchronisiert, d. h. die einzelnen CPUs warten solange, bis alle anderen ihren ANLAUF beendet haben, und beginnen dann gleichzeitig ihren zyklischen Betrieb. Sie können jedoch diese Anlaufsynchro- nisation durch Einstellung im DX 0 abschalten.

**10.1.8
Was Sie über den Testbe-
trieb wissen sollten**

Gehen Sie folgendermaßen vor:

1. Stellen Sie sicher, daß die Funktion "Testbetrieb" am Koordinator freigegeben ist.
2. Betätigen Sie den Betriebsartenschalter am KOR von STOP auf TEST. Die BASP-LED erlischt.
3. Führen Sie an den CPUs, die in den RUN gehen sollen, die Bedienung für NEUSTART oder WIEDERANLAUF durch.

*Besonderheiten im
Testbetrieb*

Im Testbetrieb können Sie die gesteckten CPUs **einzel**n in Betrieb nehmen oder sie beliebig miteinander kombinieren. CPUs, die sich im Stoppzustand befinden, können dabei nicht mehr das gesamte AG blockieren.

Der ANLAUF der einzelnen CPUs ist im Testbetrieb **nicht** syn- chronisiert. Je nach Länge der Anlauforganisationsbausteine OB 20, OB 21 oder OB 22 beginnen die CPUs zu unterschiedlichen Zeit- punkten ihren zyklischen Betrieb.

Tritt bei einer CPU ein Fehler auf, so geht im Testbetrieb **nur diese** in den Stoppzustand über. Die anderen CPUs werden durch den Fehler nicht beeinflußt.



Warnung

Da im Testbetrieb keine CPU im Fehlerfall BASP ausgeben kann, muß der Testbetrieb nach abgeschlossener Inbetrieb- nahme zur Vermeidung eines kritischen oder sogar gefährlichen Anlagenzustandes unbedingt inaktiv geschaltet werden!

10.2 Mehrprozessorkommunikation

Begriffsbestimmung

Unter Mehrprozessork**ommunikation** versteht man den Austausch größerer Datenmengen (Datenbausteine) zwischen CPUs, die im Mehrprozessorbetrieb arbeiten. Für die Mehrprozessorkommunikation ist der Koordinator KOR C erforderlich.

10.2.1

Einführung

Für die Übertragung von Datenbausteinen, genauer: von Datenblöcken mit einer Größe von max. 64 byte (= 32 Datenwörtern), stehen Ihnen – in die CPU integriert – folgende Sonderfunktionen zur Verfügung:

- OB 200: INITIALISIEREN: Vorbesetzen
- OB 202: SENDEN: Senden eines Datenblocks
- OB 203: SENDE-TEST: Sendemöglichkeit testen
- OB 204: EMPFANGEN: Empfangen eines Datenblocks
- OB 205: EMPFANGS-TEST: Empfangsmöglichkeit testen

Die Sonderfunktions-OBs OB 200 und OB 202 bis OB 205 werden in den nachfolgenden Abschnitten vereinfacht "Kommunikations-OBs" genannt.

Vorausgesetzte Kenntnisse

Für die Anwendung dieser Funktionen genügen Grundkenntnisse über die Programmiersprache STEP 5 und über die Arbeitsweise von SIMATIC S5-Automatisierungsgeräten. Diese Grundkenntnisse werden in den im Literaturverzeichnis aufgeführten Schriften vermittelt.

Prinzipieller Ablauf

Um Daten zu übertragen, müssen Sie auf der Sende-CPU die Funktion SENDEN aktivieren und auf der Empfangs-CPU die Funktion EMPFANGEN.

Dabei werden aufeinanderfolgende Datenwörter eines DB- oder DX-Datenbausteins, die sich in der Sende-CPU befinden, über den Koordinator 923C zur Empfangs-CPU transportiert und dort im DB- bzw. DX-Datenbaustein mit derselben Nummer und unter derselben Datenwort-Adresse abgelegt; d. h. es handelt sich um ein "1:1"-Kopieren.

Übertragungseinheit

Die Datenmenge, welche mittels der Funktionen SENDEN bzw. EMPFANGEN in einem Block übertragen werden kann, beträgt normalerweise 32 Wörter.

Falls die Baustein-Länge (ohne Kopf) kein Vielfaches von 32 Wörtern beträgt, so werden beim letzten Block ausnahmsweise weniger als 32 Wörter übertragen.

Der Datenbaustein in der Empfangs-CPU kann länger oder kürzer sein als der Sende-Datenbaustein. Entscheidend ist, daß die von der Funktion SENDEN übertragenen Datenwörter im Empfangs-Datenbaustein existieren; andernfalls erkennt die Funktion EMPFANGEN einen Fehler.

Beispiel:

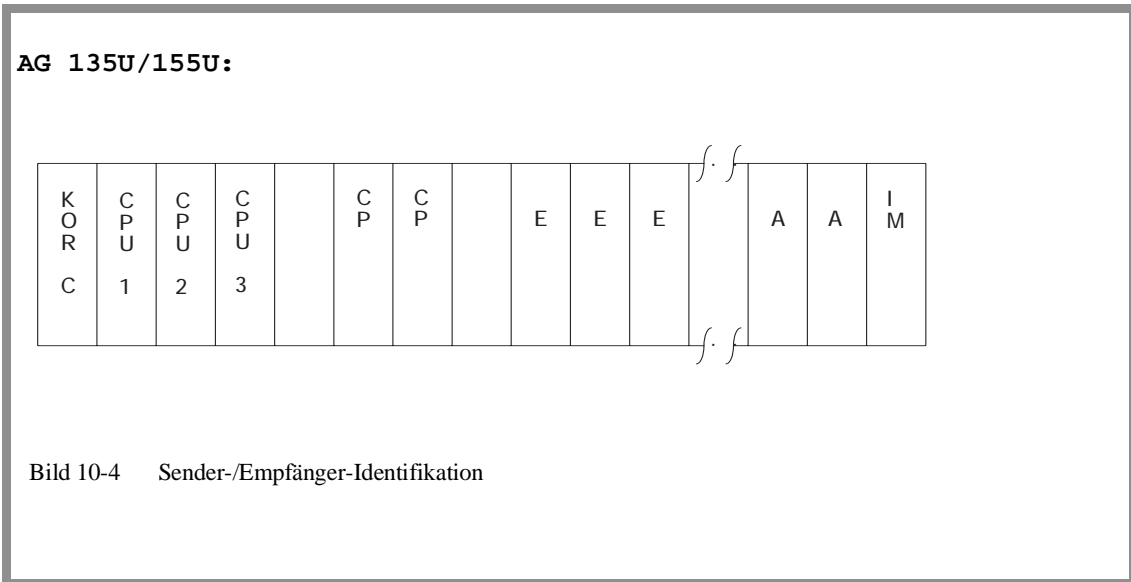
	Sende-Daten in der Sende-CPU:	Empfangs-Daten in der Empfangs-CPU:
Daten-Baustein:	DB 17	DB 17
Datenwort-Adresse:	DW 32 bis DW 63	DW 32 bis DW 63

**10.2.2
Wie Sender und Empfänger
identifiziert werden**

Jeder Datenblock, der unter den CPUs ausgetauscht wird, wird mit einer Nummer der Sende- und einer Nummer der Empfangs-CPU gekennzeichnet.

Die CPUs werden derart durchnummeriert, daß die am weitesten links gesteckte CPU die Nummer 1 erhält, die nach rechts folgenden CPUs erhalten eine jeweils um eins erhöhte Nummer.

Beispiel



10.2.3

Warum Daten zwischengespeichert werden

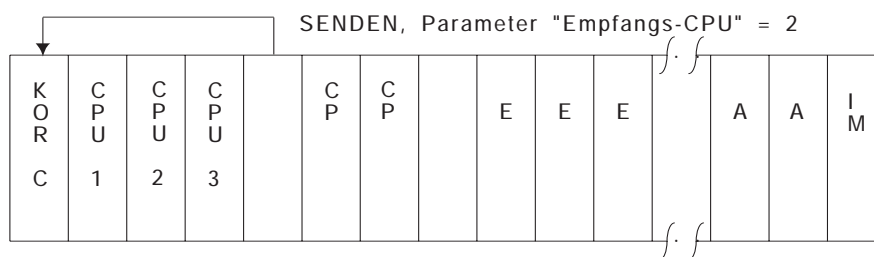
In der Regel wird der Mehrprozessorbetrieb dazu benutzt, zu bearbeitende Aufgaben auf mehrere CPUs zu verteilen. Da diese Aufgaben nicht identisch sind und zudem die Leistungsmerkmale der beteiligten CPUs unterschiedlich sein können, läuft im Mehrprozessorbetrieb die Programmbearbeitung auf den einzelnen CPUs immer **asynchron** ab. Dies bedeutet, daß die Daten einer sendenden CPU nicht sofort von der empfangenden CPU entgegengenommen werden können.

Aus diesem Grunde werden die zu übertragenden Daten im Koordinator 923C zwischengespeichert. Die Nummer der "eigenen" CPU sowie die Nummer des Empfängers beim Senden bzw. die Nummer des Senders beim Empfangen legen dabei Quelle bzw. Ziel eines Datenblocks fest.

Beispiel

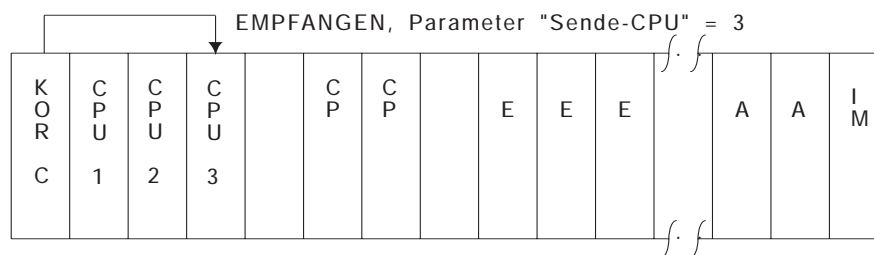
Datenübertragung von CPU 3 nach CPU 2:

1. Schritt:



Die CPU 3 legt ihre Daten im Zwischenspeicher des Koordinators ab.

2. Schritt:



Wenn CPU 2 empfangsbereit ist, kopiert sie die Daten aus dem Zwischenspeicher des Koordinators in den Ziel-DB.

10.2.4 Wie der Zwischenspeicher abgearbeitet und verwaltet wird

Abarbeitung

Dem Zwischenspeicher liegt das FIFO-Prinzip zugrunde (First in – first out, Warteschlangenprinzip). Somit ist die Empfangsreihenfolge gleich der Sendereihenfolge. Dies gilt für jede einzelne Verbindungsstrecke (gekennzeichnet durch Sende - und Empfangs-CPU) und ist von anderen Verbindungen unabhängig.

Datensicherung

Der Zwischenspeicher ist batteriegepuffert; damit ist der "Automatische Wiederanlauf nach Netzausfall" uneingeschränkt möglich. Durch einen Netzausfall während einer "laufenden" Datenübertragung gehen im Automatisierungsgerät keine Daten verloren.

Verwaltung

Die maximale Speicherkapazität des Koordinators 923C beträgt 48 Speicherblöcke mit einer festen Länge von je 32 Wörtern. Die Funktion INITIALISIEREN teilt diese Speicherblöcke den einzelnen Verbindungsstrecken zu.

Jeder **Speicherblock** nimmt genau einen **Datenblock** auf. Seine Länge kann 1 Datenwort bis 32 Datenwörtern betragen. Ein **Datenblock** wird von einer SENDEN-Funktion in einen **Speicherblock** eingetragen und von einer EMPFANGEN-Funktion wieder ausgetragen.

Die Anzahl der für eine Verbindungsstrecke vergebenen Speicherblöcke steht in direktem Zusammenhang mit den Parametern "Sende-Kapazität" (Funktion SENDEN, SENDE-TEST) und "Empfangs-Kapazität" (Funktion EMPFANGEN, EMPFANGS-TEST).

Die **Sende-Kapazität** gibt an, wieviele der für eine Verbindungsstrecke reservierten Speicherblöcke zu einem bestimmten Zeitpunkt frei sind.

Die **Empfangs-Kapazität** gibt an, wieviele der für eine Verbindungsstrecke reservierten Speicherblöcke zu einem bestimmten Zeitpunkt belegt sind.

Die Summe aus Sende- und Empfangs-Kapazität ist zu jedem Zeitpunkt gleich der Anzahl der für eine Verbindungsstrecke vergebenen Speicherblöcke.

Beispiel

Belegung des Zwischenspeichers durch eine Verbindungsstrecke:

Die Verbindungsstrecke "von CPU 3 nach CPU 2" wird initialisiert. Dabei werden ihr auf dem Zwischenspeicher des Koordinators sieben Speicherblöcke zugewiesen. Anschließend könnte z. B. folgende Datenübertragung ablaufen:

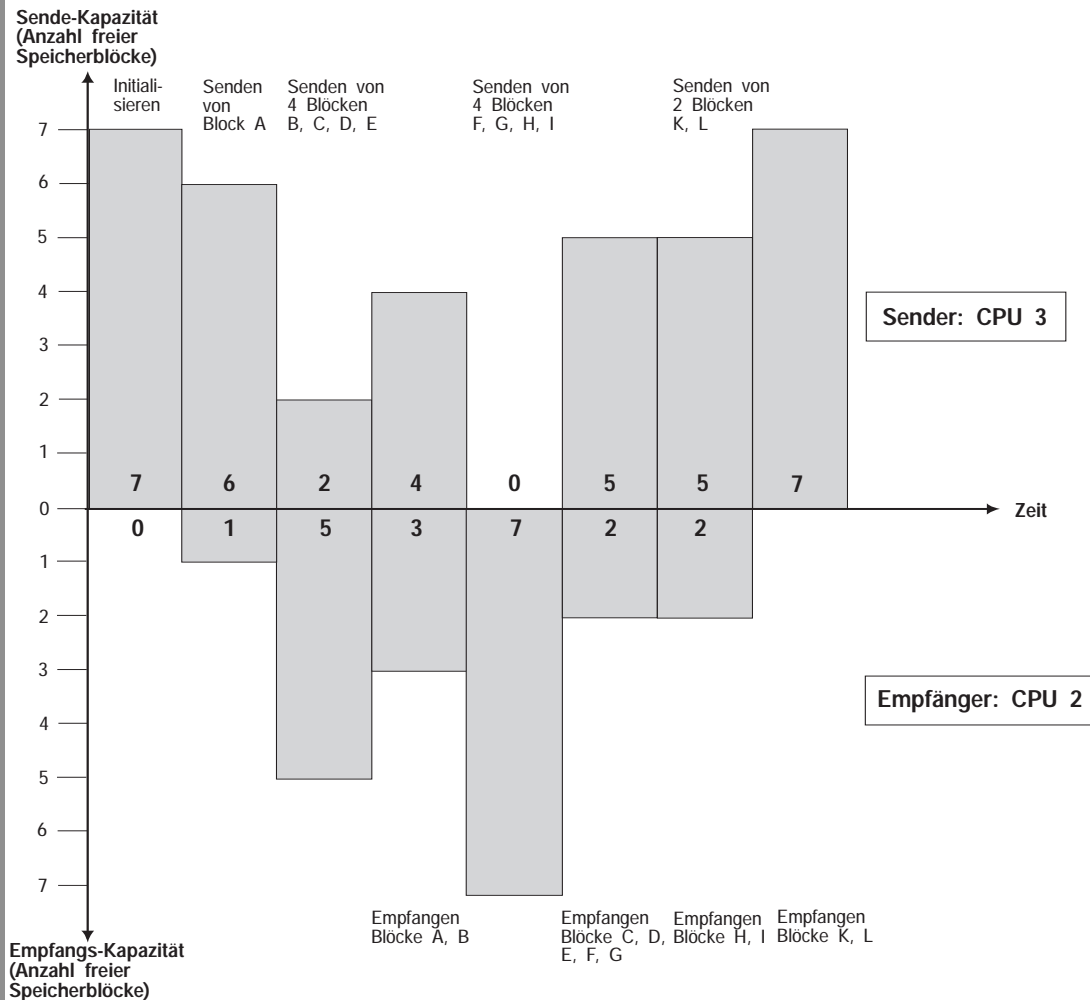


Bild 10-5 Beispiel für die Belegung des KOR-Zwischenspeichers

Senden und Empfangen von n Datenblöcken bedeutet, daß die entsprechenden Funktionen n-mal nacheinander aufgerufen werden.

Um eine einfachere Darstellung zu erreichen, wird in diesem Beispiel zunächst entweder gesendet oder empfangen. Das gleichzeitige Senden (CPU 3) und Empfangen (CPU 2) ist jedoch auch möglich und sinnvoll ("Parallelverarbeitung im Mehrprozessor-Automatisierungsgerät"). Im Beispiel werden während des Sendens der Datenblöcke K und L die Datenblöcke H und I empfangen.

Das Beispiel verdeutlicht die Warteschlangen-Organisation des Zwischenspeichers: Die zuerst gesendeten Datenblöcke (A,B,C...) werden zuerst empfangen A,B,C...).

Zusammenfassung

Das Zwischenspeichern im Koordinator KOR 923C hat die Aufgabe, die asynchronen Abläufe von Sende- und Empfangs-CPU und deren unterschiedliche Arbeitsgeschwindigkeiten auszugleichen.

Da die Kapazität des Zwischenspeichers begrenzt ist, sollte der Empfänger "häufig" und "regelmäßig" überprüfen, ob Daten gespeichert sind (Funktion EMPFANGS-TEST, Empfangs-Kapazität > 0) bzw. versuchen, gespeicherte Daten abzuholen (Funktion EMPFANGEN). Zweckmäßigerweise ist die Funktion EMPFANGEN solange wiederholt aufzurufen, bis die Empfangs-Kapazität Null wird. Ein solcher Ablauf bewirkt, daß die gesendeten Daten nicht lange zwischengespeichert bleiben, sondern dem Empfänger aktuell zur Verfügung stehen. Zusätzlich werden damit Speicherblöcke frei (die Sende-Kapazität steigt); ein Blockieren des Senders (d. h. die Sende-Kapazität ist "erschöpft", ist Null) wird verhindert.

Hinweis

Während die Empfangs-Kapazität Null den Idealzustand repräsentiert (alle gesendeten Daten vom Empfänger abgeholt), deutet die Sende-Kapazität Null auf **Projektierungsfehler** hin:

- die Funktion SENDEN wird zu häufig aufgerufen,
 - die Funktion EMPFANGEN wird zu selten aufgerufen
- oder
- der Verbindungsstrecke sind zu wenig Speicherblöcke zugewiesen. Die Kapazität des Zwischenspeichers reicht nicht aus, ein vorübergehendes Ungleichgewicht zwischen Sende- und Empfangs-Häufigkeit zu kompensieren.

10.2.5

Was Sie beim System-Anlauf beachten müssen

Die Mehrprozessor-Kommunikation erfordert, daß bei allen beteiligten CPUs der STOP-RUN-Übergang (= ANLAUF) **gleich** abläuft, d. h. entweder einheitlich NEUSTART oder einheitlich WIEDERANLAUF.

Durch entsprechende

- Bedienung (Frontschalter, Programmiergerät),
- Parametrierung (DX 0)

und/oder

- Programmierung (mittels des Sonderfunktions-Organisationsbausteins OB 223 ("Stopp bei uneinheitlicher Anlaufart im Mehrprozessorbetrieb"))

muß die **einheitliche** Anlaufart zumindest bei den an der Kommunikation beteiligten CPUs sichergestellt werden (vergl. Abschnitt - 10.1.7).

NEUSTART

Im Organisationsbaustein OB 20 (NEUSTART) ist von **nur einer** CPU mittels der Funktion INITIALISIEREN der Zwischenspeicher (im KOR 923C) einzurichten. Hierbei werden eventuell noch vorhandene Daten zerstört.

Anschließend, also noch im ANLAUF, können Sie in den einzelnen CPUs die Funktionen SENDEN, SENDE-TEST, EMPFANGEN, EMPFANGS-TEST aufrufen. Durch geeignete Programmierung müssen Sie gewährleisten, daß dies erst geschieht, nachdem die Initialisierung des Zwischenspeichers im Koordinator korrekt durchgeführt wurde. Nach Beendigung des ANLAUFs, also im RUN, wird das Anwenderprogramm **von Anfang an** bearbeitet, d. h. der erste Befehl des OB 1.

WIEDERANLAUF

In den Organisationsbausteinen OB 21 (MANUELLER WIEDERANLAUF) und OB 22 (AUTOMATISCHER WIEDERANLAUF) dürfen Sie die Funktion INITIALISIEREN **nicht** benützen. Der Aufruf der Funktionen SENDEN, SENDE-TEST, EMPFANGEN, EMPFANGS-TEST kann zu Schwierigkeiten führen; hierzu sind die Hinweise im nachfolgenden Abschnitten zu beachten.

Nach Beendigung des WIEDERANLAUFs, also im RUN, wird das Anwenderprogramm nicht von Anfang an bearbeitet, sondern an der **unterbrochenen Stelle** fortgesetzt. Die Unterbrechungsstelle kann sich z. B. innerhalb der Funktion SENDEN befinden.

10.2.6

Was Sie beim Aufrufen der Kommunikations-OBs beachten müssen

Gehen Sie folgendermaßen vor:

1. Rufen Sie die Funktion INITIALISIEREN nur im Neustart-Organisationsbaustein OB 20 auf einer CPU auf.
2. Rufen Sie die Funktionen SENDEN, SENDE-TEST, EMPFANGEN, EMPFANGS-TEST entweder **nur** innerhalb der zyklischen Programmbearbeitung oder **nur** innerhalb der zeitgesteuerten Programmbearbeitung auf.

Doppelaufruf

Abhängig von der Parametrierung des DX 0 ("Unterbrechung an Befehls Grenzen") und der Art der Programmbearbeitung (WIEDERANLAUF, Unterbrechungsbehandlung, z. B. OB 26 bei Zykluszeitfehler) ist es möglich, daß eine der Funktionen INITIALISIEREN, SENDEN, SENDE-TEST, EMPFANGEN und EMPFANGS-TEST unterbrochen wird.

Falls eine an der Unterbrechungsstelle eingeschachtelte Anwenderschnittstelle ebenfalls eine der Funktionen SENDEN, SENDE-TEST, EMPFANGEN und EMPFANGS-TEST enthält, **so erkennen diese einen unzulässigen Aufruf** (Doppelaufruf) und signalisieren Ihnen dies durch eine Fehler-Anzeige (Fehler-Nr. 67, Abschnitt 10.2.8).

Parallelverarbeitung

Nachdem Sie das Vorbesetzen des Zwischenspeichers abgeschlossen haben (Funktion INITIALISIEREN), können Sie die Funktionen SENDEN, SENDE-TEST, EMPFANGEN, EMPFANGS-TEST in beliebiger Kombination und Parametrierung in allen CPUs gleichzeitig und parallel bearbeiten lassen.

Betrachtet man eine einzelne Verbindungsstrecke (z. B. von CPU 2 nach CPU 3), so können die Funktion SENDEN (CPU 2) und die Funktion EMPFANGEN (CPU 3) gleichzeitig bearbeitet werden: Während die CPU 2 weitere Datenblöcke zum Koordinator sendet, kann die CPU 3 bereits Datenblöcke aus dem Zwischenspeicher des Koordinators abholen.

Belegte Bereiche

Die Kommunikations-OBs benötigen keinen Arbeitsbereich (z. B. für die Zwischenspeicherung von Variablen) und schlagen keine Datenbausteine auf. Sie greifen selbstverständlich auf Bereiche zu, die Parameter beinhalten, wobei nur die als Ausgangsparameter gekennzeichneten verändert werden.

Ergebnisanzeigen

Die Ergebnisanzeigen (ANZ 1/ANZ 0, VKE usw.) werden von den Kommunikations-OBs beeinflusst. Nähere Information dazu entnehmen Sie bitte dem Abschnitt 10.2.8.

Veränderung der Akkus

- CPU 922, CPU 928,
CPU 928B: Die Inhalte AKKU 1 bis AKKU 4 sowie die Registerinhalte werden von den Kommunikations-OBs nicht verändert.
- CPU 946/947,
CPU 948: Alle Registerinhalte sowie AKKU 1, 2 und 3 bleiben gleich, verändert wird lediglich der AKKU 4.

10.2.7

Wie parametrieren Sie die Kommunikations-OBs?

Es gibt bei den Kommunikations-OBs folgende Parameterarten:

- Eingangsparameter,
 - Ausgangsparameter
- und
- Aufrufparameter.

Ein- und Ausgangsparameter befinden sich in einem max. 10 byte großen **Datenfeld im M-Merkerbereich**. Das Datenfeld untergliedert sich in einen Bereich für **Eingangsparameter**, einen Bereich für **Ausgangsparameter**.

Eingangsparameter

Die Eingangsparameter legen fest, wie eine Funktion abgewickelt werden soll. Sie werden vollständig oder teilweise von den Kommunikations-OBs gelesen und ausgewertet, schreibende Zugriffe finden nicht statt.

Ausgangsparameter

Die Ausgangsparameter enthalten alle Informationen, die das aufrufende Programm über das Ergebnis eines abgegebenen Auftrags wissen muß z. B. Fehleranzeigen. Sie werden vollständig oder teilweise von den Kommunikations-OBs beschrieben, lesende Zugriffe finden nicht statt.

Hinweis

Sie können für **alle** Kommunikationsfunktionen **einen Merkerbereich mit 10 Merker-Bytes** vorsehen. Die einzelnen Funktionen benötigen jedoch eine unterschiedliche Anzahl von Bytes. Diese sind bei den Einzelfunktionen (Abschnitt 10.4ff) aufgeführt.

Aufrufparameter

Als Aufrufparameter wird bei allen Kommunikations-OBs die Nummer des ersten Merker-Bytes im Datenfeld (= Zeiger auf Datenfeld) im AKKU-1-L übergeben. Zulässige Werte dafür sind 0 bis 246.

Beispiel

Datenfeld mit Parametern der Funktion EMPFANGEN (OB 204)		
MB x + 0:	Sende-CPU	Eingangsparameter
MB x + 1:	—	nicht belegt
MB x + 2:	Anzeigenbyte	Ausgangsparameter
MB x + 3:	Empfangs-Kapazität	Ausgangsparameter
MB x + 4:	Baustein-Kennung	Ausgangsparameter
MB x + 5:	Baustein-Nummer	Ausgangsparameter
MB x + 6:	Adresse des ersten	Ausgangsparameter
MB x + 7:	empfangenen Datenwortes	Ausgangsparameter
MB x + 8:	Adresse des letzten	Ausgangsparameter
MB x + 9:	empfangenen Datenwortes	Ausgangsparameter

Dieses Beispiel verdeutlicht, daß die Nummer des ersten M-Merkerbytes im Datenfeld nicht größer als (MB) 246 sein kann und darf, da andernfalls das bis zu 10 byte große Parameterfeld die Grenzen des Merkerbereichs überschreiten würde (MB 255).

10.2.8

Wie können Sie die Ausgangsparameter auswerten?

Die Ausgangsparameter geben u. a. Hinweise, ob eine Funktion überhaupt bearbeitet werden konnte; falls nein, so zeigen sie den Grund für den Funktionsabbruch an.

Ergebnisanzeigen

Die Funktionen INITIALISIEREN, SENDEN, SENDE-TEST, EMPFANGEN, EMPFANGS-TEST beeinflussen die Ergebnisanzeigen (siehe Programmieranleitungen der jeweiligen CPUs, Allgemeine Hinweise zu den STEP-5-Operationen):

- die OV- und OS-Bits (Wortanzeigen) werden immer gelöscht,
- die OR-, STA-, ERAB-Bits (Bitanzeigen) werden immer gelöscht,
- VKE, ANZ 1 und ANZ 0 geben Auskunft, ob eine Funktion korrekt und vollständig abgearbeitet worden ist.

Tabelle 10-1 Anzeigen der Kommunikations-OBs

Anzeigen			Auswertung	Bedeutung
VKE	ANZ 1	ANZ 0		
0	0	0	SPB=	Funktion vollständig und korrekt abgearbeitet
1	0	0	SPB=	Funktion abgebrochen, Zeiger auf Datenfeld unzulässig (>246) Funktion abgebrochen aufgrund eines Initialisierungskonfliktes
1	0	1	SPB= und SPM=	Funktion abgebrochen aufgrund eines Fehlers (Fehler-Nr. 1 bis 9)
1	1	0	SPB= und SPP=	Funktion abgebrochen aufgrund einer Warnung (Warnungs-Nr. 1 oder 2)

In den nachfolgenden Ausführungen wird vorausgesetzt, daß der Zeiger auf das Datenfeld einen korrekten Wert enthält. Dann ist im ersten Byte der Ausgangsparameter die Abbruchursache in detaillierter Form hinterlegt.

Anzeigenbyte

Bit-Nr.	7	6	5	4	3	2	1	0
	W	F	I	0	Nummer			

W = 1: Warnung

F = 1: Fehler

I = 1: Initialisierungskonflikt

Nummer: - einer Warnung
 - eines Fehlers
 - eines Initialisierungskonflikts

Das erste Byte im Feld der Ausgangsparameter (Anzeigenbyte) zeigt ebenfalls an, ob eine Funktion korrekt und vollständig abgearbeitet worden ist. Der Grund eines Funktionsabbruchs wird detaillierter dargestellt als in den Ergebnisanzeigen.

Unter der oben getroffenen Vereinbarung, daß zumindest der Zeiger auf das Datenfeld einen korrekten Wert enthält, ist dieses Byte **immer** relevant.

Ist die Funktion korrekt und vollständig abgearbeitet worden, so sind alle Bits gelöscht (= 0), und alle weiteren Ausgangsparameter sind ebenfalls relevant.

Ist die Funktion mit einer Warnung abgebrochen worden (Bit-Nr. 7 = 1), so ist nur noch die Anzeige der Sende-/Empfangs-Kapazität relevant, weitere Ausgangsparameter (falls vorhanden) sind unverändert.

Ist die Funktion mit einem Fehler (Bit-Nr. 6 = 1) oder einem Initialisierungskonflikt (Bit-Nr. 5 = 1) abgebrochen worden, so sind alle weiteren Ausgangsparameter unverändert.

Auswertung des Anzeigenbytes

Die Kennungen 'W', 'F' und 'I' zeigen u.a. an, welche Bedeutung die Information "Nummer" hat.

Neben dieser bitweisen Auswertung ist es auch möglich, das gesamte Anzeigenbyte als vorzeichenlose Festpunktzahl zu interpretieren. Bei einer **byteweisen Interpretation** des Anzeigenbytes ergeben sich Nummerngruppen mit folgender Bedeutung:

Tabelle 10-2 Anzeigenbyte der Kommunikations-OBs/Nummerngruppen

Wertebereich	Bedeutung
0	Funktion korrekt und vollständig abgearbeitet
33 bis 42	Funktion abgebrochen aufgrund eines Initialisierungskonfliktes
65 bis 73	Funktion abgebrochen aufgrund eines Fehlers
129 bis 130	Funktion abgebrochen aufgrund einer Warnung

Die Fehler werden entsprechend der aufsteigenden Reihenfolge der Fehlernummern erkannt und angezeigt. Dies bedeutet, daß durchaus mehrere Fehler vorliegen können, obwohl (momentan) nur **ein** Fehler angezeigt wird. Die weiteren Fehler werden dann bei Folgeaufrufen angezeigt.

Beispiel

Die Funktion SENDEEN zeigt Fehler an und wird nicht durchgeführt. Wenn Sie nun Programm- und/oder Parameteränderungen durchführen und die Funktion SENDEEN erneut einen Fehler mit höherer Nummer als vorher anzeigt, so können Sie daraus schließen, daß Sie einen von mehreren Fehlern beseitigt haben.

Initialisierungskonflikt

Ein Initialisierungskonflikt kann nur bei der Funktion INITIALISIEREN auftreten. Er erfordert eine Änderung in der Programmierung/Parametrierung.

Initialisierungskonflikt-Nummern (bytwweise Auswertung des Anzeigenbytes):

Tabelle 10-3 Anzeigenbyte: Initialisierungskonflikt-Nummern

Anz.- Byte	Bedeutung
33	Die für die Mehrprozessorkommunikation benötigten Kacheln (Nr. 252 bis Nr. 255) sind nicht bzw. nicht vollständig vorhanden.
34	Die für die Mehrprozessorkommunikation benötigten Kacheln (Nr. 252 bis 255) sind defekt.
35	Der Parameter "Automatisch/Manuell" ist unzulässig. Unterscheiden Sie folgende Fälle: - die Kennung "Automatisch/Manuell" ist kleiner 1, - die Kennung "Automatisch/Manuell" ist größer 2.
36	Der Parameter "Anzahl CPUs" ist unzulässig. Unterscheiden Sie folgende Fälle: - die Anzahl der CPUs ist kleiner 2, - die Anzahl der CPUs ist größer 4.
37	Der Parameter "Baustein-Kennung" ist unzulässig. Unterscheiden Sie folgende Fälle: - die Baustein-Kennung ist kleiner 1, - die Baustein-Kennung ist größer 2.
38	Der Parameter "Baustein-Nummer" ist unzulässig, da es sich um einen Datenbaustein mit spezieller Bedeutung handelt. Unterscheiden Sie folgende Fälle: - falls Baustein-Kennung = 1 : DB 0, DB 1, - falls Baustein-Kennung = 2 : DX 0
39	Der Parameter "Baustein-Nummer" ist fehlerhaft, da der parametrisierte Datenbaustein nicht existiert.
40	Der Parameter "Anfangsadresse der Zuordnungsliste" ist zu groß bzw. der Datenbaustein zu kurz.

Anz.-Byte	Bedeutung
Fortsetzung der Tabelle 10-3:	
41	Die Zuordnungsliste im Datenbaustein ist nicht korrekt aufgebaut.
42	Die Summe der vergebenen Speicherblöcke ist größer als 48.

Fehler

Ein aufgetretener Fehler erfordert eine Änderung in der Programmierung/Parametrierung.

Fehler-Nummern (byteweise Auswertung des Anzeigenbytes):

Tabelle 10-4 Anzeigenbyte: Fehler-Nummern

Anz.-Byte	Bedeutung
65	Der Parameter "Empfangs-CPU" (SENDEN, SENDE-TEST) ist unzulässig. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - die Nummer der "Empfangs-CPU" ist größer 4, - die Nummer der "Empfangs-CPU" ist kleiner 1, - die Nummer der "Empfangs-CPU" ist gleich der "eigenen" Nummer.
66	Der Parameter "Sende-CPU" (EMPFANGEN, EMPFANGS-TEST) ist unzulässig. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - die Nummer der "Sende-CPU" ist größer 4, - die Nummer der "Sende-CPU" ist kleiner 1, - die Nummer der "Sende-CPU" ist gleich der "eigenen Nummer."
67	Der Aufruf des Sonderfunktions-Organisationsbausteins ist fehlerhaft (SENDEN, EMPFANGEN, SENDE-TEST, EMPFANGS-TEST). Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - Folgefehler, da die Funktion INITIALISIEREN nicht aufgerufen oder mit Initialisierungskonflikt beendet wurde, - Doppelaufruf: Der Aufruf dieser Funktion (SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGS-TEST) ist unzulässig, da in dieser CPU bereits in einer untergeordneten Bearbeitungsebene (z. B. zyklische Programmbearbeitung) eine der Funktionen INITIALISIEREN, SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGSTEST aufgerufen wurde, , - die "eigene" CPU-Nummer ist fehlerhaft (Systemdaten zerstört); nach NETZ EIN/NETZ AUS wird die CPU-Nr. vom Systemprogramm erneut erzeugt.

Anz.- Byte	Bedeutung
Fortsetzung der Tabelle 10-4:	
68	Die Verwaltungsdaten (Warteschlangenverwaltung) der angewählten Verbindungsstrecken sind fehlerhaft; den Zwischenspeicher im Koordinator 923C richten Sie mit Hilfe der Funktion INITIALISIEREN neu ein (SENDEN, EMPFANGEN, SENDE-TEST, EMPFANGS-TEST).
69	Der Parameter "Baustein-Kennung" (SENDEN) bzw. die vom Sender überlieferte Baustein-Kennung (EMPFANGEN) ist unzulässig. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - die Baustein-Kennung ist kleiner 1, - die Baustein-Kennung ist größer 2.
70	Der Parameter "Baustein-Nummer" (SENDEN) bzw. die vom Sender überlieferte Baustein-Nummer (EMPFANGEN) ist unzulässig, da es sich um einen Datenbaustein mit spezieller Bedeutung handelt. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - falls Baustein-Kennung = 1 : DB 0, DB 1 - falls Baustein-Kennung = 2 : DX 0
71	Der Parameter "Baustein-Nummer" (SENDEN) bzw. die vom Sender überlieferte Baustein-Nummer (EMPFANGEN) ist fehlerhaft. Der parametrisierte Datenbaustein existiert nicht.
72	Der Parameter "Block-Nummer" (SENDEN) ist fehlerhaft. Der Datenbaustein ist zu kurz bzw. die Block-Nummer zu groß.
73	Der Datenbaustein ist zu klein, um den vom Sender gelieferten Datenblock aufzunehmen (EMPFANGEN).

Warnung

Die Funktion konnte nicht durchgeführt werden; der Funktionsaufruf ist zu wiederholen, z. B. im nächsten Zyklus.

Warnungs-Nummern (byteweise Auswertung des Anzeigenbytes):

Tabelle 10-5 Anzeigenbyte: Warnungs-Nummern

Anz-Byte	Bedeutung
129	Die Funktion SENDEN kann keine Daten übergeben, da die Sende-Kapazität bereits beim Funktionsaufruf gleich Null war.
130	Die Funktion EMPFANGEN kann keine Daten übernehmen, da die Empfangs-Kapazität bereits beim Funktionsaufruf gleich Null war.

10.3 Laufzeiten der Kommunikations-OBs

"Laufzeit" ist die Bearbeitungszeit der Sonderfunktions-Organisationsbausteine; die Zeit(-dauer) zwischen Aufruf eines Bausteins und seinem Abschluß kann erheblich größer werden, falls er von höherpriorien Tätigkeiten unterbrochen wird (z. B. Zeitzellenaktualisierung usw.).

Tabelle 10-6 Laufzeiten der Kommunikations-OBs

Sonderfunktions-OB					
Baustein-Name	CPU 922	CPU 928	CPU 928B	CPU 946/947	CPU 948
OB 200/Initialisieren	230 ms	130 ms	130 ms	128 ms	90 ms
OB 202/Senden	806 µs (294 µs Grundlast + 16 µs/Wort); 118 µs bei Warnung	666 µs (250 µs Grundlast + 13 µs/Wort); 115 µs bei Warnung	696 µs (280 µs Grundlast + 13 µs/Wort); 145 µs bei Warnung	762 µs (426 µs Grundlast + 21 µs/Doppelwort); 243 µs bei Warnung	542 µs (220 µs Grundlast + 19 µs/Doppelwort); 110 µs bei Warnung
OB 203/ Sendetest	72 µs	50 µs	80 µs	207 µs	115 µs
OB 204/ Empfangen	825 µs (281 µs Grundlast + 17 µs/Wort); 115 µs bei Warnung	660 µs (244 µs Grundlast + 13 µs/Wort); 98 µs bei Warnung	690 µs (274 µs Grundlast + 13 µs/Wort); 128 µs bei Warnung	772 µs (421 µs Grundlast + 22 µs/Doppelwort); 243 µs bei Warnung	506 µs (218 µs Grundlast + 18 µs/Doppelwort); 132 µs bei Warnung
OB 205/ Empfangs-Test	70 µs	48 µs	78 µs	223 µs	120 µs

Die in der Tabelle 10-6 genannten Laufzeiten ergeben sich unter der Voraussetzung, daß von vier gesteckten CPUs nur diejenige CPU auf den S5-Bus zugreift, deren Laufzeiten gemessen werden. Falls weitere CPUs den Bus intensiv nutzen, steigt insbesondere beim Senden/Empfangen die Laufzeit an.

Übertragungszeit

Ein wichtiges Leistungsmerkmal einer Verbindungsstrecke (z. B. von CPU 1 nach CPU 2) ist die gesamte Datenübertragungszeit. Sie setzt sich aus den folgenden drei Komponenten zusammen:

- Sende-Dauer (siehe Laufzeit),
 - Dauer der Zwischenspeicherung (im Koordinator KOR 923C)
- und
- Empfangs-Dauer (siehe Laufzeit)

Wie lange die zu übertragenden Daten "unterwegs" sind, wird also wesentlich von der Dauer der Zwischenspeicherung und damit von der Struktur des Anwenderprogramms bestimmt (vergl. "Zwischenspeicherung der Daten").

10.4 Funktion INITIALISIEREN (OB 200)

10.4.1

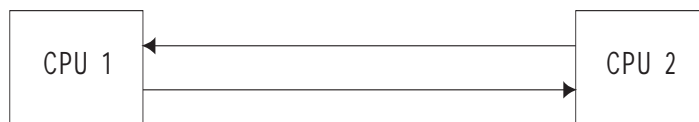
Funktion

Um von einer CPU an eine andere CPU Daten übertragen zu können, müssen diese vorübergehend zwischengespeichert werden. Die Funktion INITIALISIEREN richtet zu diesem Zweck einen Zwischenspeicher im Koordinator KOR 923C ein.

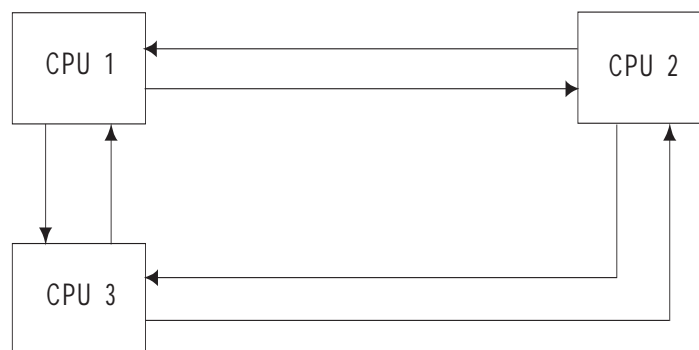
Die Speicherkapazität wird in Speicherblöcken mit einer festen Länge von 32 Wörtern eingerichtet (initialisiert).

Jeder Speicherblock nimmt genau einen Datenblock auf, dessen Länge zwischen 1 Datenwort und 32 Datenwörtern liegen kann. Ein Datenblock wird von einer SENDEN-Funktion in einen Speicherblock eingetragen und von einer EMPFANGEN-Funktion wieder ausgetragen.

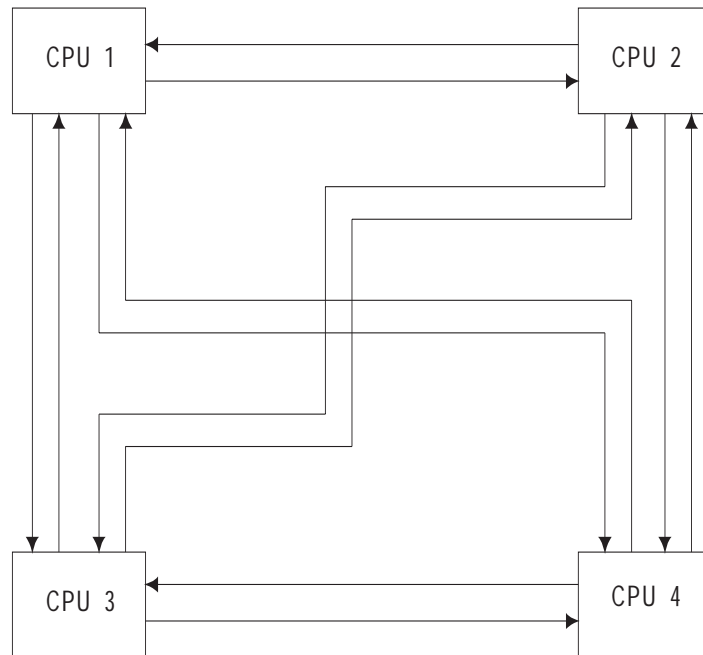
Bei zwei gesteckten CPUs ergeben sich zwei Verbindungsstrecken (Transferrichtungen, "Kanäle"):



Bei drei gesteckten CPUs ergeben sich sechs Verbindungsstrecken:



Bei vier gesteckten CPUs ergeben sich zwölf Verbindungsstrecken:



Mit der Funktion INITIALISIEREN wird festgelegt, wie die insgesamt **48** zur Verfügung stehenden Speicherblöcke den maximal zwölf Verbindungsstrecken zugeordnet werden.

D. h.: Jeder möglichen Verbindungsstrecke, gekennzeichnet durch die Parameter "Sende-CPU" und "Empfangs-CPU", steht eine bestimmte Speicherkapazität zur Verfügung.

Hinweis

Bevor auf den CPUs die Funktionen SENDEN / EMPFANGEN / SENDE-TEST / EMPFANGS-TEST aufgerufen werden dürfen, muß zuerst auf einer CPU die Funktion INITIALISIEREN aufgerufen und vollständig und fehlerfrei abgearbeitet worden sein.

Falls die Funktion INITIALISIEREN mehrfach nacheinander aufgerufen wird, so gilt die zuletzt parametrisierte Zuordnung. Während die Funktion INITIALISIEREN von einer CPU bearbeitet wird, dürfen keine weiteren Funktionen der Mehrprozessorkommunikation, also auch nicht die Funktion INITIALISIEREN, auf anderen CPUs aufgerufen werden.

10.4.2 Aufrufparameter

*Aufbau des
(Parameter-)Datenfeldes*

Vor Aufruf des OB 200 müssen Sie im Datenfeld die Eingangsparameter bereitstellen. Der OB 200 benötigt im Datenfeld 8 M-Merkerbytes für Ein- und Ausgangsparameter:

MB x + 0:	Betriebsart (Automatisch/ Manuell)	Eingangsparameter
MB x + 1:	Anzahl CPUs	Eingangsparameter
MB x + 2:	Baustein-Kennung	Eingangsparameter
MB x + 3:	Baustein-Nummer	Eingangsparameter
MB x + 4:	Anfangsadresse der Zuordnungsliste	Eingangsparameter
MB x + 5:		
MB x + 6:	Anzeigenbyte	Ausgangsparameter
MB x + 7:	Gesamt-Kapazität	Ausgangsparameter

AKKU-1-L

Bei Aufruf des OB 200 übergeben Sie in AKKU-1-L die Merkerbyte-Nr., an der das Parameterdatenfeld beginnt:

AKKU-1-LH:	0
AKKU-1-LL:	0 bis 246

10.4.3 Eingangsparameter

**Betriebsart
(Automatisch/Manuell)**

Betriebsart = 1:	Automatisch
Betriebsart = 2:	Manuell
Betriebsart = 0 oder 3 bis 255:	Unzulässig, führt zu einem Initialisierungskonflikt

Anzahl der CPUs

Dieser Parameter ist nur relevant, wenn Sie die Betriebsart "Automatisch" gewählt haben. Bei Einstellung "Automatisch" werden die zur Verfügung stehenden Speicherblöcke **gleichmäßig** entsprechend der Anzahl der CPUs wie folgt aufgeteilt:

Anzahl der CPUs	Anzahl der Verbindungsstrecken	Speicherblöcke pro Verbindungsstrecke
2	2	24
3	6	8
4	12	4
0; 1; 5 bis 255	Unzulässig, führt zu einem Initialisierungskonflikt	

*Baustein-Kennung,
Baustein-Nummer.
Adresse Zuordnungsliste*

Die Parameter sind nur relevant, wenn Sie die Betriebsart "Manuell" gewählt haben. Sie müssen dann in einem Datenbaustein eine Zuordnungsliste bereitstellen, in der nach einem festgelegten Schema die 48 zur Verfügung stehenden Speicherblöcke (oder weniger) den maximal 12 Verbindungsstrecken zugeordnet werden. Diese Funktion ist insbesondere sinnvoll, falls nicht alle CPUs in gleichem Umfang Daten miteinander austauschen.

Den CPUs, die an der Mehrprozessorkommunikation nicht teilnehmen, brauchen und sollten Sie keine Speicherblöcke zuweisen.

Mit den Parametern

- Baustein-Kennung,
- Baustein-Nummer

und

- Anfangsadresse der Zuordnungsliste

legen Sie fest, wo die Zuordnungsliste hinterlegt ist.

Baustein-Kennung

Kennung = 1:	DB-Datenbaustein
Kennung = 2:	DX-Datenbaustein
Kennung = 0 oder 3 bis 255 :	Unzulässig, führt zu einem Initialisierungskonflikt

Baustein-Nummer

Als Baustein-Nummer geben Sie die Nummer des Datenbausteins DB oder DX an, in dem die Zuordnungsliste liegt.

*Anfangsadresse der
Zuordnungsliste:*

Sie ergibt zusammen mit der Baustein-Kennung und -Nummer den Bereich (genauer: die Anfangsadresse des Bereichs) im Datenbaustein, in dem die Zuordnungsliste hinterlegt ist.

Geben Sie als Adresse der Zuordnungsliste in den Merkerbytes MB x+4 (high Byte) und MB x+5 (low Byte) die Datenwortnummer an, bei der die Zuordnungsliste beginnt.

Zuordnungsliste

Mit der Zuordnungsliste legen Sie fest, wieviele der vorhandenen 48 Speicherblöcke den einzelnen Verbindungswegen zugeordnet werden sollen.

Die Liste wird vom Systemprogramm **nicht verändert**. Sie hat folgenden Aufbau:

Tabelle 10-7 Zuordnungsliste für OB 200 (Initialisieren)

Datenwort	Format	Wert	Bedeutung
DW n + 0	KC	S1	Sender = CPU 1
DW n + 1	KY	2 , a	Empfänger = CPU 2
DW n + 2	KY	3 , b	Empfänger = CPU 3
DW n + 3	KY	4 , c	Empfänger = CPU 4
DW n + 4	KC	S2	Sender = CPU 2
DW n + 5	KY	1 , d	Empfänger = CPU 1
DW n + 6	KY	3 , e	Empfänger = CPU 3
DW n + 7	KY	4 , f	Empfänger = CPU 4
DW n + 8	KC	S3	Sender = CPU 3
DW n + 9	KY	1 , g	Empfänger = CPU 1
DW n + 10	KY	2 , h	Empfänger = CPU 2
DW n + 11	KY	4 , i	Empfänger = CPU 4
DW n + 12	KC	S4	Sender = CPU 4
DW n + 13	KY	1 , k	Empfänger = CPU 1
DW n + 14	KY	2 , l	Empfänger = CPU 2
DW n + 15	KY	3 , m	Empfänger = CPU 3

Anstelle der Kleinbuchstaben a bis m (hier fettgedruckt) sind Zahlen zwischen 0 und 48 entsprechend den zugewiesenen Speicherblöcken einzusetzen; **ihre Summe darf 48 nicht überschreiten**.

Hinweis

Der in Tabelle 10-7 angegebene Aufbau muß eingehalten werden, auch wenn weniger als vier CPUs gesteckt sind.

Beispiel

Sie haben drei CPUs gesteckt. Von CPU 2 sind sehr viele Daten an die beiden anderen zu übertragen. Diese wiederum senden nur wenige Daten zurück an CPU 2 als Rückmeldung in einem logischen Quittungsverkehr. Zwischen **CPU 1 und CPU 3** ist **kein Datenaustausch** nötig.

Die Zuordnungsliste wird im Datenbaustein DB 40 ab DW 0 abgelegt und hat folgende Parameter:

```

DB40      FD:   CPU948ST.S5D

0:      KC = S1;      Sender      CPU 1
1:      KY = 2,2;    Empfänger: CPU 2/2 Blöcke
2:      KY = 3,0;    Empfänger: CPU 3/kein Block
3:      KY = 4,0;    Empfänger: CPU 4 (nicht vorhanden)/kein Block
4:      KC = S2;      Sender:      CPU 2
5:      KY = 1,22;   Empfänger:   CPU 1/22 Blöcke
6:      KY = 3,22;   Empfänger:   CPU 3/22 Blöcke
7:      KY = 4,0;    Empfänger:   CPU 4 (nicht vorhanden)/kein Block
8:      KC = S3;      Sender:      CPU 3
9:      KY = 1,0;    Empfänger:   CPU 1/kein Block
10:     KY = 2,2;    Empfänger:   CPU 2/2 Blöcke
11:     KY = 4,0;    Empfänger:   CPU 4 (nicht vorhanden)/kein Block
12:     KC = S4;      Sender:      CPU 4 (nicht vorhanden)
13:     KY = 1,0;    Empfänger:   CPU 1/kein Block
14:     KY = 2,0;    Empfänger:   CPU 2/kein Block
15:     KY = 3,0;    Empfänger:   CPU 3/kein Block
16:
    
```

**10.4.4
Ausgangsparameter**

Anzeigenbyte

Dieses Byte informiert Sie, ob die Funktion INITIALISIEREN korrekt und vollständig abgearbeitet worden ist.

Initialisierungskonflikt

Die aufgeführten Initialisierungskonflikte werden entsprechend der aufsteigenden Reihenfolge ihrer Nummern von der Funktion erkannt und angezeigt.

Ein aufgetretener Initialisierungskonflikt erfordert eine Änderung in der Programmierung / Parametrierung.

Es können alle in Tabelle 10-3 aufgeführten Nummern im Anzeigenbyte auftreten.

<i>Fehler</i>	Eine Anzeige der Nummerngruppe "Fehler" kann bei der Funktion INITIALISIEREN nicht auftreten.
<i>Warnung</i>	Eine Anzeige der Nummerngruppe "Warnung" kann bei der Funktion INITIALISIEREN nicht auftreten.
Gesamt-Kapazität	Dieser Parameter gibt an, wieviel der zur Verfügung stehenden 48 Speicherblöcke den Verbindungsstrecken zugeordnet sind. In der Betriebsart "Automatisch" wird dieser Parameter auf jeden Fall den Wert 48 enthalten. Bei der Betriebsart "Manuell" kann der Wert kleiner sein als 48. Dies bedeutet, daß vorhandene Speicherkapazität nicht genutzt wird.

10.5 Funktion SENDEN (OB 202)

10.5.1

Funktion

Die Funktion SENDEN übergibt einen Datenblock in den Zwischenspeicher des Koordinators KOR 923C. Zusätzlich zeigt sie an, wieviele Datenblöcke noch gesendet und zwischengespeichert werden können.

10.5.2

Aufrufparameter

Aufbau des (Parameter-)Datenfeldes

Vor Aufruf des OB 202 müssen Sie im Datenfeld die Eingangsparameter bereitstellen. Der OB 202 benötigt im Datenfeld 6 M-Merkerbytes für Ein- und Ausgangsparameter:

MB x + 0:	Empfangs-CPU	Eingangsparameter
MB x + 1:	Baustein-Kennung	Eingangsparameter
MB x + 2:	Baustein-Nummer	Eingangsparameter
MB x + 3:	Block-Nummer	Eingangsparameter
MB x + 4:	Anzeigenbyte	Ausgangsparameter
MB x + 5:	Sende-Kapazität	Ausgangsparameter

AKKU-1-L

Bei Aufruf des OB 202 übergeben Sie in AKKU-1-L die Merkerbyte-Nr., an der das Parameterdatenfeld beginnt:

AKKU-1-LH:	0
AKKU-1-LL:	0 bis 246

10.5.3

Eingangsparameter

Empfangs-CPU

CPU-Nummer des Empfängers (Ziel); der zulässige Wert liegt zwischen 1 und 4, muß jedoch ungleich der "eigenen" Nummer sein.

Baustein-Kennung

Kennung = 1: DB-Datenbaustein
 Kennung = 2: DX-Datenbaustein
 Kennung = 0 oder 3 bis 255: Unzulässig, führt zu einer Fehleranzeige

Baustein-Nummer

Die Baustein-Nummer ergibt zusammen mit der Baustein-Kennung und der Block-Nummer den Bereich, dem die Sende-Daten entnommen werden (und in dem sie auf der Empfangs-CPU abgelegt werden).

Hierbei ist zu beachten, daß bestimmte Datenbausteine besondere Bedeutungen haben, beispielsweise DB 0, DB 1 oder DX 0 (siehe Programmieranleitungen der jeweiligen CPUs). Diese Datenbausteine dürfen deshalb für die hier beschriebene Datenübertragung nicht verwendet werden!

Die Verwendung dieser Bausteinnummer führt zum Abbruch der Funktion mit Fehleranzeige.

Block-Nummer

Die Block-Nummer kennzeichnet den zu Quell-Datenbereich.

Block-Nummer	Datenbereich	
	erstes Datenwort	letztes Datenwort
0	DW 0	DW 31
1	DW 32	DW 63
2	DW 64	DW 95
3	DW 96	DW 127
4	DW 128	DW 159
5	DW 160	DW 191
6	DW 192	DW 223
7	DW 224	DW 255
8	DW 256	DW 287
9	DW 288	DW 319
:	:	:
:	:	:

Folgende Fälle sind zu unterscheiden:

- **DB ist länger als Quellbereich:**
Ist der Datenbaustein ausreichend lang, so ergibt sich pro Block ein 32 Wörter großer Bereich gemäß der angeführten Tabelle.
- **DB ist zu kurz:**
Liegt das Datenbaustein-Ende innerhalb des parametrisierten Blockes, so wird im letzten Block ein Bereich mit einer Länge zwischen einem und 31 Wörtern übertragen.
- **Block liegt außerhalb des DBs:**
Ist die ermittelte erste Datenwort-Adresse eines Blocks bereits außerhalb der Datenbaustein-Länge, so wird von der Funktion SENDEN ein Fehler erkannt und angezeigt.

Beispiel

```
Ein Datenbaustein mit der Länge 80 Wörter: DW 0 bis  
DW 74, 5 Wörter sind Bausteinkopf.
```

Blocknr:	erstes Datenw.:	letztes Datenw.:	Länge:
0	DW 0	DW 31	32 Wörter
1	DW 32	DW 63	32 Wörter
2	DW 64	DW 74	11 Wörter
3 und größer	fehlerhafte Parametrierung		

10.5.4 Ausgangsparameter

Anzeigenbyte

Dieses Byte informiert Sie, ob die Funktion SENDEN korrekt und vollständig abgearbeitet worden ist.

Initialisierungskonflikt

Hat bei der Funktion SENDEN keine Bedeutung.

Fehler

Beim Aufrufen der Funktion SENDEN können folgende Fehlernummern (byteweise Auswertung des Anzeigenbytes) auftreten:

Anz.-Byte	Bedeutung
65	Der Parameter "Empfangs-CPU" ist unzulässig. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - die Nummer der Empfangs-CPU ist größer 4, - die Nummer der Empfangs-CPU ist kleiner 1, - die Nummer der Empfangs-CPU ist identisch mit der "eigenen" Nummer.
67	Der Aufruf des Sonderfunktions-Organisationsbausteins ist fehlerhaft. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - Folgefehler, da die Funktion INITIALISIEREN nicht aufgerufen oder mit Initialisierungskonflikt beendet wurde, - Doppelaufruf: Der Aufruf dieser Funktion SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGS-TEST ist unzulässig, da in dieser CPU bereits in einer untergeordneten Bearbeitungsebene (z. B. zyklische Programmbearbeitung) eine der Funktionen INITIALISIEREN, SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGS-TEST aufgerufen wurde, - die "eigene" CPU-Nummer ist fehlerhaft (Systemdaten zerstört); nach NETZ EIN/ NETZ AUS wird die CPU-Nr. vom Systemprogramm erneut erzeugt.
68	Die Verwaltungsdaten (Warteschlangenverwaltung) der angewählten Verbindungsstrecken sind fehlerhaft; der Zwischenspeicher im Koordinator 923C ist mit Hilfe der Funktion INITIALISIEREN neu einzurichten.
69	Der Parameter "Baustein-Kennung" ist unzulässig. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - die Baustein-Kennung ist kleiner 1, - die Baustein-Kennung ist größer 2.
70	Der Parameter "Baustein-Nummer" ist unzulässig, da es sich um einen Datenbaustein mit spezieller Bedeutung handelt. Unterscheiden Sie folgende Sonderfälle: <ul style="list-style-type: none"> - falls Baustein-Kennung = 1 : DB 0, DB 1 - falls Baustein-Kennung = 2 : DX 0
71	Der Parameter "Baustein-Nummer" ist fehlerhaft. Der parametrisierte Datenbaustein existiert nicht.
72	Der Parameter "Block-Nummer" ist fehlerhaft. Der Datenbaustein ist zu kurz bzw. die Block-Nummer zu groß.

Warnung

Die Funktion konnte nicht durchgeführt werden; der Funktionsaufruf ist zu wiederholen, z. B. im nächsten Zyklus.

Es kann folgende Warnungsnummer (byteweise Auswertung des Anzeigenbytes) auftreten:

Anz.-Byte	Bedeutung
129	Die Funktion SENDEN kann keine Daten übergeben, da die Sende-Kapazität (s. u.) bereits beim Funktionsaufruf gleich null war.

Sende-Kapazität

Im Parameter "Sende-Kapazität" wird angezeigt an, wieviele Datenblöcke gesendet und zwischengespeichert werden können.

10.6 Funktion SENDE-TEST (OB 203)

10.6.1

Funktion

Die Funktion SENDE-TEST ermittelt die Anzahl der freien Speicherblöcke im Zwischenspeicher des Koordinators KOR 923C. Entsprechend dieser Anzahl m kann die Funktion SENDE-TEST m -mal aufgerufen werden um m Datenblöcke zu übergeben.

10.6.2

Aufrufparameter

Aufbau des (Parameter-)Datenfeldes

Vor Aufruf des OB 203 müssen Sie im Datenfeld die Eingangsparameter bereitstellen. Der OB 203 benötigt im Datenfeld 4 M-Merkerbytes für Ein- und Ausgangsparameter:

MB $x + 0$:	Empfangs-CPU	Eingangsparameter
MB $x + 1$:	—	nicht belegt
MB $x + 2$:	Anzeigenbyte	Ausgangsparameter
MB $x + 3$:	Sende-Kapazität	Ausgangsparameter

AKKU-1-L

Bei Aufruf des OB 203 übergeben Sie in AKKU-1-L die Merkerbyte-Nr., an der das Parameterdatenfeld beginnt:

AKKU-1-LH:	0
AKKU-1-LL:	0 bis 246

10.6.3

Eingangsparameter

Empfangs-CPU

Die Nummer der "eigenen" CPU sowie die Nummer der Empfangs-CPU kennzeichnen diejenige Verbindungsstrecke, für die die Sendekapazität ermittelt wird.

10.6.4

Ausgangsparameter

Anzeigenbyte

Dieses Byte informiert Sie, ob die Funktion SENDE-TEST korrekt und vollständig abgearbeitet worden ist.

Initialisierungskonflikt

Hat bei der Funktion SENDE-TEST keine Bedeutung.

Fehler

Beim Aufrufen der Funktion SENDE-TEST können folgende Fehlernummern (byteweise Auswertung des Anzeigenbytes) auftreten:

Anz.-Byte	Bedeutung
65	Der Parameter "Empfangs-CPU" ist unzulässig. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - die Nummer der Empfangs-CPU ist größer 4, - die Nummer der Empfangs-CPU ist kleiner 1, - die Nummer der Empfangs-CPU ist identisch mit der "eigenen" Nummer.
67	Der Aufruf des Sonderfunktions-Organisationsbausteins ist fehlerhaft. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - Folgefehler, da die Funktion INITIALISIEREN nicht aufgerufen oder mit Initialisierungskonflikt beendet wurde, - Doppelaufruf: Der Aufruf dieser Funktion SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGS-TEST ist unzulässig, da in dieser CPU bereits in einer untergeordneten Bearbeitungsebene (z. B. zyklische Programmbearbeitung) eine der Funktionen INITIALISIEREN, SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGS-TEST aufgerufen wurde, - die "eigene" CPU-Nummer ist fehlerhaft (Systemdaten zerstört); nach NETZ EIN/NETZ AUS wird die CPU-Nr. vom Systemprogramm erneut erzeugt.
68	Die Verwaltungsdaten (Warteschlangenverwaltung) der angewählten Verbindungsstrecken sind fehlerhaft; der Zwischenspeicher im Koordinator 923C ist mit Hilfe der Funktion INITIALISIEREN neu einzurichten.

Warnung

Eine Anzeige der Nummerngruppe "Warnung" kann bei der Funktion SENDE-TEST nicht auftreten.

Sende-Kapazität

Im Parameter "Sende-Kapazität" wird angezeigt an, wieviele Datenblöcke gesendet und zwischengespeichert werden können.

10.7 Funktion EMPFANGEN (OB 204)

10.7.1

Funktion

Die Funktion EMPFANGEN übernimmt einen Datenblock vom Zwischenspeicher des Koordinators KOR 923C. Zusätzlich zeigt sie an, wieviele Datenblöcke noch zwischengespeichert sind und empfangen werden können.

Die Funktion EMPFANGEN sollte in einer Schleife so oft aufgerufen werden, bis alle zwischengespeicherten Datenblöcke übernommen sind.

10.7.2

Aufrufparameter

Aufbau des (Parameter-)Datenfeldes

Vor Aufruf des OB 204 müssen Sie im Datenfeld die Eingangsparameter bereitstellen. Der OB 204 benötigt im Datenfeld 10 M-Merkerbytes für Ein- und Ausgangsparameter:

MB x + 0:	Sende-CPU	Eingangsparameter
MB x + 1:	—	nicht belegt
MB x + 2:	Anzeigenbyte	Ausgangsparameter
MB x + 3:	Empfangs-Kapazität	Ausgangsparameter
MB x + 4:	Baustein-Kennung	Ausgangsparameter
MB x + 5:	Baustein-Nummer	Ausgangsparameter
MB x + 6:	Adresse des ersten	Ausgangsparameter
MB x + 7:	empfangenen Datenwortes	Ausgangsparameter
MB x + 8:	[Adresse des letzten	Ausgangsparameter
MB x + 9:		

AKKU-1-L

Bei Aufruf des OB 204 übergeben Sie in AKKU-1-L die Merkerbyte-Nr., an der das Parameterdatenfeld beginnt:

AKKU-1-LH:	0
AKKU-1-LL:	0 bis 246

10.7.3

Eingangsparameter

Sende-CPU

Der Empfangs-Baustein empfängt die Daten, die von der Sende-CPU geliefert wurden. Geben Sie die Nummer der Sende-CPU an. Der zulässige Wert liegt zwischen 1 und 4, muß jedoch ungleich der "eigenen" Nummer sein.

**10.7.4
Ausgangsparameter**

Anzeigenbyte

Dieses Byte informiert Sie, ob die Funktion EMPFANGEN korrekt und vollständig abgearbeitet worden ist.

Initialisierungskonflikt

Hat bei der Funktion EMPFANGEN keine Bedeutung.

Fehler

Beim Aufrufen der Funktion EMPFANGEN können folgende Fehlernummern (bytwweise Auswertung des Anzeigenbytes) auftreten:

Anz.- Byte	Bedeutung
66	Der Parameter "Sende-CPU" ist unzulässig. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - die Nummer der Empfangs-CPU ist größer 4, - die Nummer der Empfangs-CPU ist kleiner 1, - die Nummer der Empfangs-CPU ist identisch mit der "eigenen" Nummer.
67	Der Aufruf des Sonderfunktions-Organisationsbausteins ist fehlerhaft. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - Folgefehler, da die Funktion INITIALISIEREN nicht aufgerufen oder mit Initialisierungskonflikt beendet wurde, - Doppelaufruf: Der Aufruf dieser Funktion SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGS-TEST ist unzulässig, da in dieser CPU bereits in einer untergeordneten Bearbeitungsebene (z. B. zyklische Programmbearbeitung) eine der Funktionen INITIALISIEREN, SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGS-TEST aufgerufen wurde, - die "eigene" CPU-Nummer ist fehlerhaft (Systemdaten zerstört); nach NETZ EIN/NETZ AUS wird die CPU-Nr. vom Systemprogramm erneut erzeugt.
68	Die Verwaltungsdaten (Warteschlangenverwaltung) der angewählten Verbindungsstrecken sind fehlerhaft; der Zwischenspeicher im Koordinator 923C ist mit Hilfe der Funktion INITIALISIEREN neu einzurichten.
69	Die vom Sender übergebene Baustein-Kennung ist unzulässig. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - die Baustein-Kennung ist kleiner 1, - die Baustein-Kennung ist größer 2.

Anz.- Byte	Bedeutung
Fortsetzung der Fehler-Nummern:	
70	Die vom Sender übergebene Baustein-Nummer ist unzulässig, da es sich um einen Datenbaustein mit spezieller Bedeutung handelt. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - falls Baustein-Kennung = 1 : DB 0, DB 1 - falls Baustein-Kennung = 2 : DX 0
71	Die vom Sender übergebene Baustein-Nummer ist fehlerhaft. Der parametrisierte Datenbaustein existiert nicht.
73	Der Datenbaustein ist zu klein, um den vom Sender gelieferten Datenblock aufzunehmen.

Warnung

Die Funktion konnte nicht durchgeführt werden; der Funktionsaufruf ist zu wiederholen, z. B. im nächsten Zyklus.

Es kann folgende Warnungsnummer (bytwweise Auswertung des Anzeigenbytes) auftreten:

Anz.- Byte	Bedeutung
130	Die Funktion EMPFANGEN kann keine Daten übernehmen, da die Empfangs-Kapazität bereits beim Funktionsaufruf gleich Null war.

Empfangs-Kapazität

Der Parameter "Empfangs-Kapazität" zeigt an, wieviel Datenblöcke noch zwischengespeichert sind und empfangen werden können.

<i>Baustein-Kennung:</i>	Kennung = 1:	DB-Datenbaustein
	Kennung = 2:	DX-Datenbaustein
	Kennung = 0 oder 3 bis 255:	Unzulässig, führt zu einer Fehleranzeige

Baustein-Nummer Baustein-Nummer des DB/DX, in dem die Daten empfangen und abgelegt worden sind (und aus dem sie in der Sende-CPU von der Funktion SENDEN entnommen worden sind).

Hierbei ist zu beachten, daß sich die Empfangs-Datenbausteine in einem Schreib-/Lese-Speicher (RAM) befinden müssen; die Verwendung von Nur-Lese-Speichern (EPROM) ist nur bei Sende-Datenbausteinen sinnvoll.

Adresse des ersten empfangenen Datenwortes Datenwortnummer innerhalb des DB/DX, in der das erste übertragene/empfangene Datenwort abgelegt wurde.

Adresse des letzten empfangenen Datenwortes Datenwortnummer innerhalb des DB/DX, in der das letzte übertragene/empfangene Datenwort abgelegt wurde.

Hinweis

Die Differenz zwischen den Adressen des ersten und des letzten übertragenen Datenwortes beträgt maximal 31, da pro Funktionsaufruf maximal 32 Wörter übertragen werden.

10.8 Funktion EMPFANGS-TEST (OB 205)

10.8.1

Funktion

Die Funktion EMPFANGS-TEST ermittelt die Anzahl belegter Speicherblöcke im Zwischenspeicher des Koordinators KOR 923C. Entsprechend dieser Anzahl m kann die Funktion EMPFANGEN m -mal aufgerufen werden um m Datenblöcke zu übernehmen.

10.8.2

Aufrufparameter

Aufbau des (Parameter-)Datenfeldes

Vor Aufruf des OB 205 müssen Sie im Datenfeld die Eingangsparameter bereitstellen. Der OB 205 benötigt im Datenfeld 4 M-Merkerbytes für Ein- und Ausgangsparameter:

MB $x + 0$:	Sende-CPU	Eingangsparameter
MB $x + 1$:	—	nicht belegt
MB $x + 2$:	Anzeigenbyte	Ausgangsparameter
MB $x + 3$:	Empfangs-Kapazität	Ausgangsparameter

AKKU-1-L

Bei Aufruf des OB 204 übergeben Sie in AKKU-1-L die Merkerbyte-Nr., an der das Parameterdatenfeld beginnt:

AKKU-1-LH:	0
AKKU-1-LL:	0 bis 246

10.8.3

Eingangsparameter

Sende-CPU

Die Nummer der "eigenen" CPU sowie die Nummer der Sende-CPU kennzeichnen diejenige Verbindungsstrecke, für die die Empfangs-Kapazität ermittelt wird.

10.8.4

Ausgangsparameter

Anzeigenbyte

Dieses Byte informiert Sie, ob die Funktion EMPFANGS-TEST korrekt und vollständig abgearbeitet worden ist.

Initialisierungskonflikt

Hat bei der Funktion EMPFANGS-TEST keine Bedeutung.

Fehler

Beim Aufrufen der Funktion EMPFANGS-TEST können folgende Fehlernummern (bytwweise Auswertung des Anzeigenbytes) auftreten:

Anz.-Byte	Bedeutung
66	Der Parameter "Sende-CPU" ist unzulässig. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - die Nummer der Sende-CPU ist größer 4, - die Nummer der Sende-CPU ist kleiner 1, - die Nummer der Sende-CPU ist identisch mit der "eigenen" Nummer.
67	Der Aufruf des Sonderfunktions-Organisationsbausteins ist fehlerhaft. Unterscheiden Sie folgende Fälle: <ul style="list-style-type: none"> - Folgefehler, da die Funktion INITIALISIEREN nicht aufgerufen oder mit Initialisierungskonflikt beendet wurde, - Doppelaufruf: Der Aufruf dieser Funktion SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGS-TEST ist unzulässig, da in dieser CPU bereits in einer untergeordneten Bearbeitungsebene (z. B. zyklische Programmbearbeitung) eine der Funktionen INITIALISIEREN, SENDEN, SENDE-TEST, EMPFANGEN oder EMPFANGS-TEST aufgerufen wurde, - die "eigene" CPU-Nummer ist fehlerhaft (Systemdaten zerstört); nach NETZ EIN/NETZ AUS wird die CPU-Nr. vom Systemprogramm erneut erzeugt.
68	Die Verwaltungsdaten (Warteschlangenverwaltung) der angewählten Verbindungsstrecken sind fehlerhaft; der Zwischenspeicher im Koordinator 923C ist mit Hilfe der Funktion INITIALISIEREN neu einzurichten.

Warnung

Eine Anzeige der Nummerngruppe "Warnung" kann bei der Funktion EMPFANGS-TEST nicht auftreten.

Empfangs-Kapazität

Der Parameter "Empfangs-Kapazität" zeigt an, wieviel Datenblöcke zwischengespeichert sind und empfangen werden können.

10.9 Anwendungen

Dieser Abschnitt erläutert Ihnen an Hand einiger Beispiele, wie Sie Ihre Mehrprozessor-Kommunikation programmieren können.

Hinweis

Wenn Sie die nachfolgend aufgeführten Funktionsbausteine verwenden und auf Ihrer CPU zusätzlich Alarme bearbeiten (z. B. durch OB 2), so achten Sie darauf, daß am Anfang einer Unterbrechungsbehandlung die "Schmiermerker" gerettet und am Ende wieder zurückgeschrieben werden.

Dies gilt auch bei der Einstellung "Unterbrechung an Bausteingrenzen", da der Aufruf der Sonderfunktions-Organisationsbausteine eine Bausteingrenze darstellt.

10.9.1

Aufruf der Sonderfunktions-OB über Funktionsbausteine

Die nachfolgend vorgestellten fünf Funktionsbausteine (FB 200 und FB 202 bis FB 205) enthalten den Aufruf des jeweiligen Sonderfunktions-Organisationsbausteins zur Mehrprozessorkommunikation (OB 200 und OB 202 bis OB 205).

Die Nummern der Funktionsbausteine sind frei gewählt und können geändert werden. Die Parameter der Sonderfunktions-OB werden bei Aufruf der Funktionsbausteine als Aktualparameter übergeben. Der direkte Aufruf der Sonderfunktions-Organisationsbausteine ist zwar lauffähiger, aber wegen der fehlenden Formalparameter schwieriger lesbar.

FB-Nummer	FB-Name	Funktion
FB 200	INITIAL	Vorbesetzen
FB 202	SENDEN	Senden eines Datenblockes
FB 203	SEND-TST	Sendemöglichkeit testen
FB 204	EMPFANG	Empfangen eines Datenblockes
FB 205	EMPF-TST	Empfangsmöglichkeit testen

Der Merkerbereich von MB 246 bis maximal MB 255 wird von den Funktionsbausteinen als Parameterfeld für die Sonderfunktions-Organisationsbausteine benutzt.

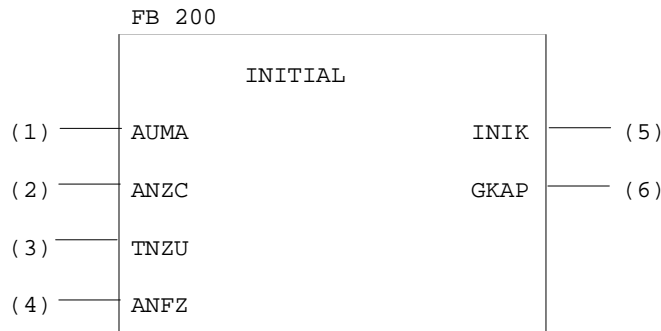
Die genaue Bedeutung der Ein- und Ausgangsparameter können Sie der Beschreibung des verwendeten Sonderfunktions-Organisationsbausteins entnehmen.

Hinweis

Es handelt sich bei den folgenden Anwendungsbeispielen um fertige Applikationen, die Sie in Ihr Programm übernehmen können.

Programmieren der Funktionsbausteine

FB 200: Vorbesetzen der Verbindungen



Parameter-Name	Bedeutung	Art	Typ	Parameter-feld
AUMA	A utomatik/ M anuell	E	BY	MB 246
ANZC	A nzahl CPUs	E	BY	MB 247
TNZU	T yp (H-Byte) und Nummer (L-Byte) des Datenbausteins, der die Z uordnungsliste enthält	E	W	MW 248
ANFZ	A nfangsadresse der Z uordnungsliste	E	W	MW 250
INIK	Initialisierungs- K onflikt	A	BY	MB 252
GKAP	G esamt- K apazität	A	BY	MB 253

Fortsetzung auf der nächsten Seite

Fortsetzung von FB 200

```

FB 200
NETZWERK 1      0000
NAME: INITIAL
BEZ   :AUMA      E/A/D/B/T/Z: E  BI/BY/W/D: BY
BEZ   :ANZC      E/A/D/B/T/Z: E  BI/BY/W/D: BY
BEZ   :TNZU      E/A/D/B/T/Z: E  BI/BY/W/D: W
BEZ   :ANFZ      E/A/D/B/T/Z: E  BI/BY/W/D: W
BEZ   :INIK      E/A/D/B/T/Z: A  BI/BY/W/D: BY
BEZ   :GKAP      E/A/D/B/T/Z: A  BI/BY/W/D: BY

0017   :L   =AUMA      Automatisch/Manuell
0018   :T   MB 246
0019   :L   =ANZC      Anzahl CPUs
001A   :T   MB 247
001B   :L   =TNZU      DB-Typ, DB-Nr.
001C   :T   MW 248
001D   :L   =ANFZ      Anfangsadresse der Zuordnungsliste
001E   :T   MW 250
001F   :
0020   :L   KB 246      F-OB:
0021   :SPA OB 200      "Initialisieren"
0022   :
0023   :L   MB 252      Initialisierungs-Konflikt
0024   :T   =INIK
0025   :L   MB 253      Gesamt-Kapazitaet
0026   :T   =GKAP
0027   :BE

```

FB 202: Senden eines Datenblocks



Parameter-Name	Bedeutung	Art	Typ	Parameter-feld
ECPU	Empfangs-CPU	E	BY	MB 246
TNDB	Typ (H-Byte) und Nummer (L-Byte) des Quell-Datenbausteins	E	W	MB 247
BLNR	Block-Nummer	E	BY	MW 249
FEWA	Fehler/Warnung	A	BY	MW 250
SKAP	Sende-Kapazität	A	BY	MB 251

FB 202

LAE=40

NETZWERK 1 0000

NAME:SENDEN

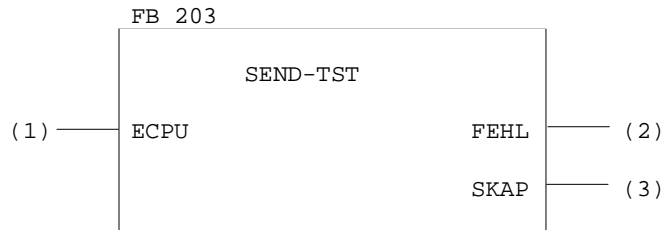
```

BEZ :ECPU E/A/D/B/T/Z: E BI/BY/W/D: BY
BEZ :TNDB E/A/D/B/T/Z: E BI/BY/W/D: W
BEZ :BLNR E/A/D/B/T/Z: E BI/BY/W/D: BY
BEZ :FEWA E/A/D/B/T/Z: A BI/BY/W/D: BY
BEZ :SKAP E/A/D/B/T/Z: A BI/BY/W/D: BY
  
```

```

0014 :L =ECPU Empfangs-CPU
0015 :T MB 246
0016 :L =TNDB DB-Typ, DB-Nr.
0017 :T MW 247
0018 :L =BLNR Block-Nummer
0019 :T MB 249
001A :
001B :L KB 246 SF-OB:
001C :SPA OB 202 "Senden eines Datenblocks"
001D :
001E :L MB 250 Fehler/Warnung
001F :T =FEWA
0020 :L MB 251 Sende-Kapazitaet
0021 :T =SKAP
0022 :BE
  
```

FB 203: Sendemöglichkeit testen



Parameter-Name	Bedeutung	Art	Typ	Parameter-Feld
ECPU	Empfangs-CPU	E	BY	MB 246
FEHL	Fehler	A	BY	MB 248
SKAP	Sende-Kapazität	A	BY	MB 249

FB 203

LAE=30

NETZWERK 1 0000

NAME:SEND-TST

BEZ :ECPU E/A/D/B/T/Z: E BI/BY/W/D: BY

BEZ :FEHL E/A/D/B/T/Z: A BI/BY/W/D: BY

BEZ :SKAP E/A/D/B/T/Z: A BI/BY/W/D: BY

000E :L =ECPU Empfangs-CPU

000F :T MB 246

0010 :

0011 :L KB 246 SF-OB:

0012 :SPA OB 203 "Sendemöglichkeit testen"

0013 :

0014 :L MB 248 Fehler

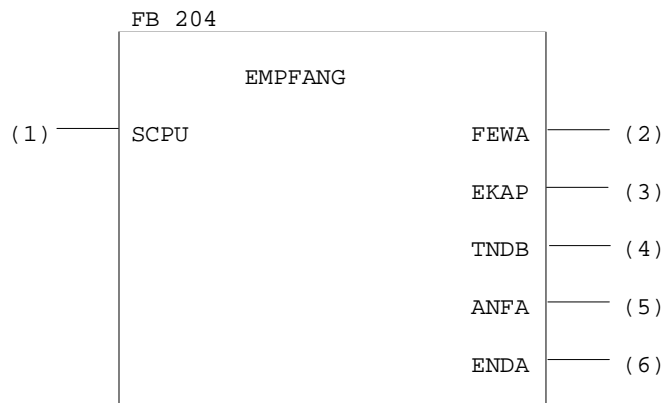
0015 :T =FEHL

0016 :L MB 249 Sende-Kapazitaet

0017 :T =SKAP

0018 :BE

FB 204: Empfangen eines Datenblocks



Parameter-Name	Bedeutung	Art	Typ	Parameter-Feld
SCPU	Sende-CPU	E	BY	MB 246
FEWA	Fehler/Warnung	A	BY	MB 248
EKAP	Empfangs-Kapazität	A	BY	MB 249
TNDB	Typ (H-Byte) und Nummer (L-Byte) des Ziel-Datenbausteins	A	W	MW 250
ANFA	Adresse des ersten empfangenen Datenwortes (A nfangsadresse)	A	W	MW 252
ENDA	Adresse des letzten empfangenen Datenwortes (E ndadresse)	A	W	MW 254

Fortsetzung auf der nächsten Seite

Fortsetzung von FB 204:

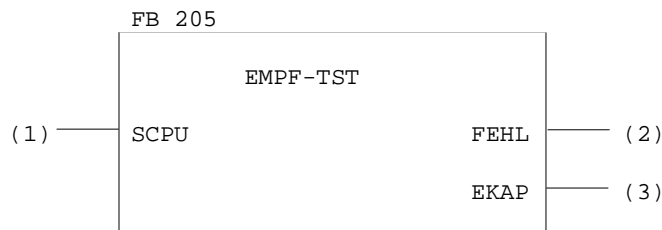
FB 204 LAE=45

```

NETZWERK 1      0000
NAME:EMPFANG
BEZ  :SCPU  E/A/D/B/T/Z: E   BI/BY/W/D: BY
BEZ  :FEWA  E/A/D/B/T/Z: A   BI/BY/W/D: BY
BEZ  :EKAP  E/A/D/B/T/Z: A   BI/BY/W/D: BY
BEZ  :TNDB  E/A/D/B/T/Z: A   BI/BY/W/D: W
BEZ  :ANFA  E/A/D/B/T/Z: A   BI/BY/W/D: W
BEZ  :ENDA  E/A/D/B/T/Z: A   BI/BY/W/D: W

0017      :L   =SCPU          Sende-CPU
0018      :T   MB 246
0019      :
001A      :L   KB 246          SF-OB:
001B      :SPA OB 204         "Empfangen eines Datenblocks"
001C      :
001D      :L   MB 248          Fehler/Warnung
001E      :T   =FEWA
001F      :L   MB 249          Empfangs-Kapazitaet
0020      :T   =EKAP
0021      :L   MW 250          DB-Typ, DB-Nr.
0022      :T   =TNDB
0023      :L   MW 252          Anfangsadresse
0024      :T   =ANFA
0025      :L   MW 254          Endadresse
0026      :T   =ENDA
0027      :BE
    
```

FB 205: Empfangsmöglichkeit testen



Parameter-Name	Bedeutung	Art	Typ	Parameter-Feld
SCPU	Sende-CPU	E	BY	MB 246
FEHL	Fehler	A	BY	MB 248
EKAP	Empfangs-Kapazität	A	BY	MB 249

Fortsetzung von FB 205:

```

FB 205                                     LAE=30

NETZWERK 1          0000
NAME:EMPF-TST
BEZ  :SCPU   E/A/D/B/T/Z: E   BI/BY/W/D: BY
BEZ  :FEHL   E/A/D/B/T/Z: A   BI/BY/W/D: BY
BEZ  :EKAP   E/A/D/B/T/Z: A   BI/BY/W/D: BY

000E      :L   =SCPU           Sende-CPU
000F      :T   MB 246
0010      :
0011      :L   KB 246           SF-OB:
0012      :SPA OB 205          "Empfangsmoeglichkeit testen"
0013      :
0014      :L   MB 248           Fehler
0015      :T   =FEHL
0016      :L   MB 249           Empfangs-Kapazitaet
0017      :T   =EKAP

```

10.9.2 Übertragen von Datenbausteinen

Mit dem Funktionsbaustein UEBT-DAT (FB 110) soll in unserem Beispiel eine parametrierbare Anzahl von Datenblöcken aus einem Datenbaustein einer CPU in den Datenbaustein gleichen Typs und gleicher Nummer einer anderen CPU übertragen werden. Die FB-Nummer (FB 110) ist zufällig gewählt und kann geändert werden.

Es wird zunächst das Programmieren des FB 110 und anschließend die Anwendung des FB 110 beschrieben.

Programmieren des FB 110

FB 110: Übertragen eines Datenbausteins

Aufgabenstellung

Der zu übertragende Datenbereich wird über die Eingangsparameter ERSB (= Nummer des ersten zu übertragenden Datenblocks) und ANZB (= Anzahl der zu übertragenden Datenblöcke) festgelegt. Ein Datenblock besteht normalerweise aus 32 Datenwörtern. Je nach Datenbaustein-Länge werden beim letzten Datenblock gegebenenfalls weniger als 32 Datenwörter übertragen.

Die Übertragung wird mit einer positiven Flanke am Starteingang STAR angestoßen. Ist anschließend der Ausgangsparameter REST gleich Null, so konnte der Funktionsbaustein UEBT-DAT alle Datenblöcke (gemäß Parameter ANZB) senden.

Fortsetzung auf der nächsten Seite

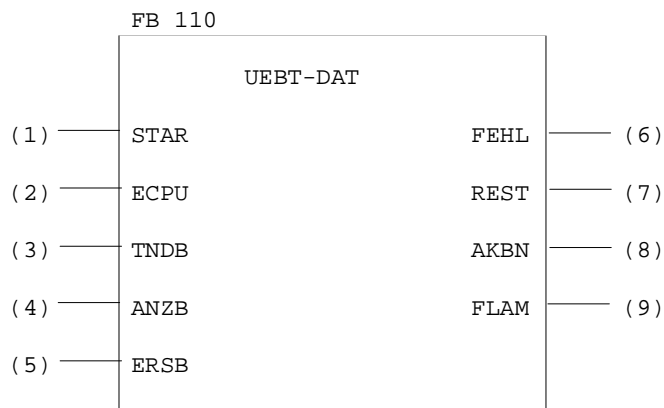
Fortsetzung 1 von FB 110:

Enthält der Ausgangsparameter REST jedoch einen Wert größer als Null, sind - z. B. im nächsten Zyklus - Folgeaufrufe erforderlich. In diesem Fall darf der gesamte Parametersatz (d.h. die Werte aller Parameter) vom Anwender (-Programm) erst dann verändert werden, wenn der Ausgangsparameter REST den Wert Null hat, d. h. wenn die Datenübertragung abgeschlossen ist.

Der mehrfache Aufruf des Funktionsbausteins UEBT-DAT mit jeweils verschiedenen Parametersätzen ist möglich. Hierbei werden verschiedene Datenbereiche gleichzeitig ("ineinander verzahnt") übertragen. Zusätzlich können die Sonderfunktions-Organisationsbausteine zur Mehrprozessorkommunikation OB 202 bis OB 205 "direkt" eingesetzt werden. Von dieser Möglichkeit wird im Anwendungsbeispiel Gebrauch gemacht.

Falls innerhalb des Funktionsbausteins UEBT-DAT die Funktion SENDEN (OB 202) nicht korrekt abgearbeitet werden konnte, wird die jeweilige Fehlernummer im Ausgangsparameter FEHL übergeben, das VKE = '1' und der Ausgangsparameter REST = '0' gesetzt.

Der Funktionsbaustein UEBT-DAT verwendet die Merkerbytes MB 246 bis MB 251 als Schmiermerker. Alle anderen Variablen, deren Wert solange von Bedeutung ist, bis nach mehrfachem Aufruf des Funktionsbausteins UEBT-DAT der Ausgangsparameter REST = '0' ist, erhalten über den Mechanismus der Formal-/Aktual-Parameter Speicherplätze zugewiesen. Dieses Verfahren ist erforderlich, damit verschiedene Datenbausteine gleichzeitig übertragen werden können.

Realisierung

Fortsetzung auf der nächsten Seite

Fortsetzung 2 von FB 110:

Parameter-Name	Bedeutung	Art	Typ
STAR	Übertragung des Datenbausteins nach positiver Flanke starten .	E	BI
ECPU	Empfangs-CPU	E	BY
TNDB	Typ (H-Byte) und Nummer (L-Byte) des zu übertragenden Datenbausteins.	E	W
ANZB	Anzahl der zu übertragenden Datenblöcke.	E	BY
ERSB	Nummer des ersten zu übertragenden Datenblocks.	E	BY
FEHL	Fehler	A	BY
REST	Anzahl der noch zu übertragenden Datenblöcke.	A	BY
AKBN ¹⁾	Aktuelle Block-Nummer	A	BY
FLAM ¹⁾	Flankenmerker	A	BI

¹⁾ Interne Zwischenmerker, nicht zur Auswertung vorgesehen.

FB 110

LAE=89

NETZWERK 1 0000

NAME:UEBT-DAT

```

BEZ  :STAR  E/A/D/B/T/Z: E    BI/BY/W/D: BI
BEZ  :ECPU  E/A/D/B/T/Z: E    BI/BY/W/D: BY
BEZ  :TNDB  E/A/D/B/T/Z: E    BI/BY/W/D: W
BEZ  :ANZB  E/A/D/B/T/Z: E    BI/BY/W/D: BY
BEZ  :ERSB  E/A/D/B/T/Z: E    BI/BY/W/D: BY
BEZ  :FEHL  E/A/D/B/T/Z: A    BI/BY/W/D: BY
BEZ  :REST  E/A/D/B/T/Z: A    BI/BY/W/D: BY
BEZ  :AKBN  E/A/D/B/T/Z: A    BI/BY/W/D: BY
BEZ  :FLAM  E/A/D/B/T/Z: A    BI/BY/W/D: BI
    
```

```

0020      :L   =ECPU           Parameterfeld fuer SF-OB 202
0021      :T   MB 246          vorbesetzen
0022      :L   =TNDB
0023      :T   MW 247
0024      :
    
```

Fortsetzung auf der nächsten Seite

Fortsetzung 3 von FB 110:

```

0025      :L  =REST          zuerst eventuell noch vorhandene
0026      :L  KB 0          Bloecke aussenden
0027      :><F
0028      :SPB =UEBT
0029      :
002A      :UN  =STAR        positive Flanke
002B      :RB  =FLAM        am START-Eingang ?
002C      :ON  =STAR
002D      :O   =FLAM
002E      :SPB =GUT
002F      :S   =FLAM
0030      :
0031      :L  =ANZB        die globalen Merker nach
0032      :T  =REST        einer postiven Flanke am
0033      :L  =ERSB        START-Eingang initialisieren
0034      :T  =AKBN
0035      :
0036      :L  =REST        solange REST ><0 ist,
0038 SCHL :L  KF+0        weiterhin versuchen,
0039      :!=F            Datenbloecke auszusenden
003A      :SPB =GUT
003B UEBT :L  =AKBN
003C      :T  MB 249
003D      :L  KB 246      SF-OB:
003E      :SPA OB 202     "Senden eines Datenblocks"
003F      :L  MB 250
0040      :SPM =FEHL      Abbruch bei Fehler
0041      :SPP =GUT      Abbruch, falls Sende-Kap. = 0
0042      :L  =AKBN      Block-Nummer
0043      :I  1          inkrementieren
0044      :T  =AKBN
0045      :L  =REST      Anzahl der verbleibenden
0046      :D  1          Bloecke dekrementieren
0047      :T  =REST
0048      :SPA =SCHL
0049      :
004A GUT  :U  M 0.0      regulaeres Programmende:
004B      :UN  M 0.0
004C      :L  KB 0      VKE = 0, FEHL = 0
004D      :T  =FEHL
004E      :BEA
004F      :
0050 FEHL :T  =FEHL      Programmende bei Fehler:
0051      :L  KB 0
0052      :T  =REST      VKE = 1, FEHL enthaelt Fehlernummer
0053      :BE

```

Anwendung des FB 110

Anwendung des FB 110 auf dem AG S5-155U

Aufgabenstellung

Die CPU 1 soll im zyklischen Anwenderprogramm die Datenbausteine DB 3 (Datenblöcke 2 bis 5) und DB 4 (Datenblöcke 1 bis 3) an die CPU 2 senden. In der CPU 2 soll ebenfalls im zyklischen Anwenderprogramm die Funktion EMPFANGEN (OB 04) aufgerufen werden.

Realisierung

Funktion	CPU 1	CPU 2
	aufgerufen in:	aufgerufen in:
Initialisieren (OB 200)	OB 20	–
Sende-Organisation (FB 1)	OB 1	–
Empfangs-Organisation (FB 2)	–	OB 1
	vorhanden:	vorhanden:
Sende-DB	DB 3; DB 4	–
Empfangs-DB	–	DB 3; DB 4

Das Anwenderprogramm im Funktionsbaustein FB 1 der CPU 1 enthält zweimal den Aufruf des Funktionsbausteins UEBT-DAT mit jeweils unterschiedlichen Parametersätzen.

Nach einer positiven Flanke am Eingang E 2.0 beginnt die Übertragung des ersten Datenbausteins DB 3. Eine positive Flanke am Eingang E 2.1 startet die Übertragung des zweiten Datenbausteins DB 4.

```

FB 1                                     LAE=yy

NETZWERK 1          0000
NAME:S-ORG
0000      :L   KB 2          an die CPU 2 ..
0001      :T   MB 0
0002      :L   KY 1,3       .. aus dem Datenbaustein DB 3
0003      :T   MW 1
0004      :L   KB 4          .. vier Datenbloecke
0005      :T   MB 3
0006      :L   KB 2          .. ab Datenblock 2 senden
0007      :T   MB 4
0008      :
    
```

Fortsetzung auf der nächsten Seite

Fortsetzung 1 des Anwendungsbeispiels:

```

0009      :SPA FB 110
000A NAME :UEBT-DAT
000B STAR :   E 2.0
000C ECPU :   MB 0
000D TNDB :   MW 1
000E ANZB :   MB 3
000F ERSB :   MB 4
0010 FEHL :   MB 5
0011 REST :   MB 6
0012 AKBN :   MB 7
0013 FLAM :   M 8.0
0014      :
0015      :
0016      :SPB =HALT           Abbruch nach Fehler
0017      :
0018      :L   KB 2           an die CPU 2 ..
0019      :T   MB 10
001A      :L   KY 1,4       .. aus dem Datenbaustein DB 4
001B      :T   MW 11
001C      :L   KB 3           .. drei Datenblöcke
001D      :T   MB 13
001E      :L   KB 1           .. ab Datenblock 1 senden
001F      :T   MB 14
0020      :
0021      :SPA FB 110
0023 NAME :UEBT-DAT
0024 STAR :   E 2.1
0025 ECPU :   MB 10
0026 TNDB :   MW 11
0027 ANZB :   MB 13
0028 ERSB :   MB 14
0029 FEHL :   MB 5
002A REST :   MB16
002B AKBN :   MB17
002C FLAM :   M 8.1
002D      :
002E      :
002F      :SPB =HALT           Abbruch nach Fehler
0030      :BEA
0031      :
0032 HALT :
0033      :                   hier erfolgt die Fehlerbehandlung
0034      :                   (z. B. Stopp, Meldungsausgabe
0035      :                   auf Drucker, ...)
0036      :
00xx      :BE

```

Fortsetzung auf der nächsten Seite

Fortsetzung 2 des Anwendungsbeispiels:

In der CPU 2 überträgt die vom FB 2 aufgerufene Funktion EMPFANGEN (OB 204) jeden ausgesendeten Datenblock in den zugehörigen Datenbaustein. Der vollständige Empfang eines Datenbausteins kann sich über mehrere Zyklen verteilen.

```

FB 2                                     LAE=yy

NETZWERK 1      0000
NAME:E-ORG
0000      :L   KB 1           Daten von CPU 1 empfangen
0001      :T   MB 246
0002      :
0003 SCHL :L   KB 246       SF-OB:
0004      :SPA OB 204       "Empfangen"
0005      :SPM =FEHL       Abbruch bei Fehler
0006      :L   MB 249       Die Funktion "Empfangen"
0007      :L   KB 0         wird solange aufgerufen,
0008      :><F             bis der Zwischenspeicher
0009      :SPB =SCHL       keine weiteren Datenblöcke
000A      :               mehr enthaelt, d. h. die
000B      :BEA             Empfangskapazitaet = 0 ist.
000C FEHL :
000D      :               hier erfolgt die Fehlerbehandlung
000E      :               (z. B. Stopp, Meldungs Ausgabe
000F      :               auf Drucker, ...)

00xx      :BE

```

10.9.3 Erweiterung des Koppelmerkerbereichs

Problemstellung

Im Automatisierungsgerät S5-135U/155U kann jedes der 256 Merkerbytes einer CPU durch Eintrag in den Datenbaustein DB 1 zum Eingangs- oder Ausgangs-Koppelmerker werden. Dadurch verringert sich jedoch die Anzahl der "normal" verwendbaren Merkerbytes. Weiterhin sind zur Übertragung eines Datensatzes (mehrere Bytes) zusätzliche Maßnahmen (Semaphor-Variable oder DX-0-Parametrierung "Koppelmerker im Block übertragen") notwendig, um zu verhindern, daß der Empfänger einen nur teilweise übertragenen Datensatz auswertet.

Lösung

Aufeinanderfolgende Datenwörter eines DB- oder DX- Datenbausteins, jeweils ab DW 0, werden als "Koppel-Datenwörter" definiert. Jede Verbindungsstrecke erhält "ihren" Datenbaustein und ist von den anderen Verbindungsstrecken völlig **unabhängig**.

Zu Beginn des Zyklus-Bausteins werden mit Hilfe der Sonderfunktions-Organisationsbausteine zur Mehrprozessor-Kommunikation die Koppel-Datenwörter empfangen. Es folgt das "reguläre" zyklische Programm, welches die empfangenen Daten auswertet und Sende-Daten erzeugt. Sie werden am Zyklusende wiederum mit Hilfe der Sonderfunktions-Organisationsbausteine zur Mehrprozessorkommunikation gesendet. So können sie von den anderen CPUs bei deren Zyklusbeginn empfangen werden.

Für jede der max. 12 möglichen Verbindungsstrecken und unabhängig von den anderen Verbindungsstrecken gilt:

- Die Sende-CPU wird nur aktiv, falls die Empfangs-CPU die "alten" Daten vollständig dem Zwischenspeicher KOR 923C entnommen hat.
- Die Empfangs-CPU wird nur aktiv, falls die Sende-CPU die "neuen" Daten vollständig im Zwischenspeicher KOR 923C abgelegt hat.

Somit steht der Empfangs-CPU entweder ein kompletter neuer Datensatz zur Verfügung oder der alte Datensatz bleibt unverändert: **Keine Mischung von "alten" und "neuen" Daten!**

Datenstruktur

Welche Datenwörter (nachfolgend Datenwortbereich genannt) von welcher CPU zu welcher CPU zu übertragen sind, ist in der Verbindungsliste (siehe Tabelle auf der nächsten Seite) beschrieben. Sie befindet sich in einem zusätzlichen Datenbaustein, welcher in allen beteiligten CPUs vorhanden sein muß.

Die Datenwortbereiche beginnen immer ab Datenwort DW 0, ihre Länge wird in Blöcken angegeben. Hierbei ist zu beachten:

- Ein kompletter Block besteht aus 32 Datenwörtern.
- Ist der letzte Block eines Sende-Datenbausteins "angeschnitten", d. h. umfaßt er zwischen einem und 31 Datenwörtern, so werden entsprechend weniger Datenwörter übertragen.
- Ist ein Sende-Datenbaustein länger als die in der Verbindungsliste angegebene Blockanzahl, so können die überzähligen Datenwörter in der entsprechenden CPU verwendet werden.
- Ist ein Empfangs-Datenbaustein länger als der empfangene Datenwortbereich, so können die überzähligen Datenwörter in der entsprechenden CPU verwendet werden.

Aufbau der Verbindungsliste

Tabelle 10-8 Verbindungsliste für die Erweiterung des Koppelmerkerbereichs

	TEILLISTE 1			TEILLISTE 2		
Verbindungs- strecke		DB-Typ	DB- Nummer			Block anzahl
von CPU 1 nach ...	DW 0	S 1		DW 16	S 1	
... CPU 2	DW 1	DW 17	2	...
... CPU 3	DW 2	DW 18	3	...
... CPU 4	DW 3	DW 19	4	...
von CPU 2 nach ...	DW 4	S 2		DW 20	S 2	
... CPU 1	DW 5	DW 21	1	...
... CPU 3	DW 6	1 ¹⁾	10 ¹⁾	DW 22	3	2 ¹⁾
... CPU 4	DW 7	DW 23	4	...
von CPU 3 nach ...	DW 8	S 3		DW 24	S 3	
... CPU 1	DW 9	DW 25	1	...
... CPU 2	DW 10	DW 26	2	...
... CPU 4	DW 11	DW 27	4	...
von CPU 4 nach ...	DW 12	S 4		DW 28	S 4	
... CPU 1	DW 13	DW 29	1	...
... CPU 2	DW 14	DW 30	2	...
... CPU 3	DW 15	DW 31	3	...
		2^{15}		2^0		2^{15}
						2^0

¹⁾ Siehe Beispiel auf der nächsten Seite

Die Verbindungsliste besteht aus zwei ähnlich strukturierten Teillisten zu je 16 Datenwörtern. Ausgehend von jeder der vier Sende-CPU's (S1, S2, S3, S4) sind zur Beschreibung jeder Verbindungsstrecke 3 Einträge vorgesehen:

- **Blockanzahl**

Die Anzahl der Blöcke legt die Größe (= Anzahl der Datenwörter) des zu übertragenden Datenwortbereiches fest. (Nicht vorhandene bzw. nicht genutzte Verbindungsstrecken werden mit Blockanzahl = 0 gekennzeichnet, ebenso bei DB-Typ und DB-Nummer.)

- **DB-Typ**

Typ des Datenbausteins, der den auszusendenden Datenwortbereich enthält.

- **DB-Nummer**

Nummer des Datenbausteins, der den auszusendenden Datenwortbereich enthält.

Diese Einträge können in der obigen Darstellung zeilenweise gelesen und ausgefüllt werden. Um beispielsweise die ersten zwei Datenblöcke aus dem Datenbaustein DB 10 von CPU 2 (S2) an die CPU 3 zu übertragen, ist folgender Eintrag notwendig:

CPU 2 (S 2) sendet ..

DW 22	3	2	DW 6	1	10
	↓	↓		↓	↓
..an	CPU 3	2 Datenblöcke aus dem		DB	10

Die Teilliste 2 ist identisch mit der für die Funktion INITIALISIEREN (OB 200) benötigten Zuordnungsliste (Betriebsart "Manuell"). Innerhalb des Datenbausteins muß die Teilliste 1 die Datenwörter 0 bis 15 und die Teilliste 2 die Datenwörter 16 bis 31 belegen. Die mit Fettschrift hervorgehobenen Einträge dürfen nicht abgeändert werden.

Programmstruktur

Auf dem Koordinator werden von einer CPU beim Anlauf durch Aufruf der Funktion INITIALISIEREN (OB 200) genau so viele Speicherblöcke pro Verbindungsstrecke reserviert, wie Datenblöcke auf dieser Strecke zu übertragen sind.

Zum Aussenden und Empfangen der Datenwortbereiche werden auf jeder CPU zwei Funktionsbausteine verwendet:

FB-Nr.	Name	Funktion
FB 100	SEND-DAT	Senden von Datenwortbereichen an die restlichen CPUs
FB 101	EMPF-DAT	Empfangen von Datenwortbereichen von den restlichen CPUs

Die FB-Nummern sind zufällig gewählt und können geändert werden.

Die Funktionsbausteine SEND-DAT und EMPF-DAT entnehmen der Verbindungsliste, welche Datenwortbereiche aus welchen Datenbausteinen auszusenden oder zu empfangen sind. Es wird immer der **gesamte** Datenwortbereich ausgesendet oder empfangen. Falls dies, wegen unzureichender Sende- oder Empfangs-Kapazität, noch nicht möglich ist, wird auf das Senden bzw. Empfangen verzichtet.

Hinweis

Dieses Beispiel (Koppelmerkererweiterung mit Hilfe der Funktionsbausteine SEND-DAT und EMPF-DAT) kann nur dann korrekt ablaufen, wenn in keiner der CPUs zusätzlich die Sonderfunktions-Organisationsbausteine zur Mehrprozessorkommunikation OB 202 bis OB 205 aufgerufen werden.

Die Funktionsbausteine SEND-DAT und EMPF-DAT enthalten die Sonderfunktions-Organisationsbausteine zur Mehrprozessorkommunikation OB 202 bis OB 205. Der zusätzliche Aufruf dieser Organisationsbausteine außerhalb von SEND-DAT / EMPF-DAT ist nicht zulässig!

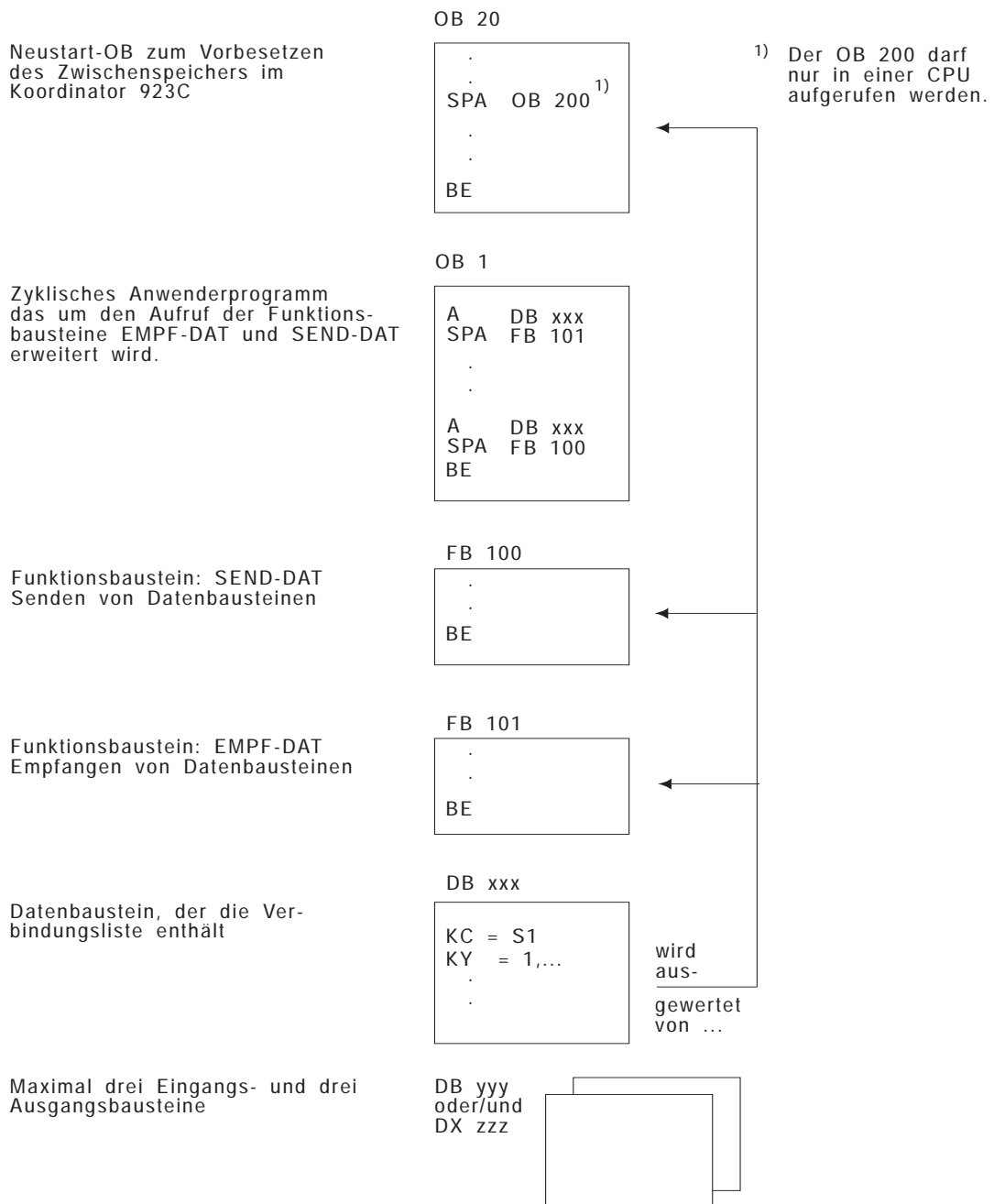


Bild 10-6 Übersicht über die in jeder CPU benötigten Bausteine

Programmieren der Funktionsbausteine

FB 100: Senden von Datenwortbereichen

Vor Aufruf des FB 100 muß der Datenbaustein aufgeschlagen werden, der die Verbindungsliste enthält. Der Funktionsbaustein SEND-DAT benötigt zum Auswerten der in der Verbindungsliste enthaltenen Informationen die Nummer der CPU, auf der er aufgerufen wird.

Falls innerhalb des Funktionsbausteins die Funktion SENDEN (OB 202) nicht korrekt abgearbeitet werden konnte, wird die jeweilige Fehler- oder Warnungsnummer im Ausgangsparameter FEWA übergeben und das VKE = 1 gesetzt. Zusätzlich enthält FEWA bei unzulässigem Eingangsparameter CPUN (CPU-Nummer) den Wert 16 (Bit-Nr. 4 = 1).

Der Funktionsbaustein SEND-DAT verwendet die Merkerbytes MB 239 bis MB 251 als Schmiermerker.



Parameter-Name	Bedeutung	Art	Typ
CPUN	Nummer der CPU, auf der der FB 100 aufgerufen wird. Zulässig sind die Nummern 1 bis 4.	D	KF
FEWA	Fehler/Warnung (siehe Funktion SENDEN / OB 202)	A	BY

FB 100

LAE=90

NETZWERK 1 0000

NAME:SEND-DAT

BEZ :CPUN E/A/D/B/T/Z: D KM/KH/KY/KC/KF/KT/KZ/KG: KF
 BEZ :FEWA E/A/D/B/T/Z: A BI/BY/W/D: BY

000B :LW =CPUN CPUN = CPUN - 1
 000C :L KB 1 Fehler falls:
 000D :-F
 000E :SPM =FEWA CPU-Nr. <1
 000F :L KB 3
 0010 :>F
 0011 :SPB =FEWA CPU-Nr >4
 0012 :TAK

Fortsetzung auf der nächsten Seite

Fortsetzung 1 von FB 100:

```

0013      :
0014      :SLW 2          CPUN = CPUN * 4
0015      :T   MB 245    Basisadresse
0016      :
0017      :L   KB 1
0018      :T   MB 244    Verbindungszaehler
0019      :
001A SCHL :L   MB 245    Basisadresse
001B      :L   MB 244    + Zaehler
001C      :+F
001D      :T   MW 240
001E      :ADD BF+16     + Offset
001F      :T   MW 242
0020      :
0021      :B   MW 242
0022      :L   DR 0      Anzahl der reservierten
0023      :T   MB 239    Bloecke = 0 ?
0024      :L   KB 0
0025      :!=F
0026      :SPB =LEER
0027      :
0028      :B   MW 242
0029      :L   DL 0      Nr. der Empfangs-CPU
002A      :T   MB 246
002B      :L   KB 246    SF-OB:
002C      :SPA OB 203    "Sendemoeglichkeit testen"
002D      :L   MB 248    Abbruch bei Fehler"
002E      :SPB =OBFE
002F      :
0030      :L   MB 249    Sende-Kapazitaet >< Anzahl
0031      :L   MB 239    der reservierten Bloecke?
0032      :><F
0033      :SPB =LEER
0034      :
0035      :L   KB 0      Blockzaehler
0036      :T   MB 249
0037      :
0038      :B   MW 240
0039      :L   DW 0      Typ und Nummer des
003A      :T   MW 247    Quell-DB
003B      :
003C UEBT :L   KB 246    SF-OB:
003D      :SPA OB 202    Senden eines Datenblocks
003E      :L   MB 250    Abbruch bei Fehler/Warnung
003F      :SPB =OBFE
0040      :
0041      :L   MB 249    Block-Nr. = Block-Nr. + 1
0042      :I   1
0043      :T   MB 249    alle Bloecke uebertragen ?
0044      :L   MB 239
0045      :<F
0046      :SPB =UEBT
0047      :

```

Fortsetzung auf der nächsten Seite

Fortsetzung 2 von FB 100:

```

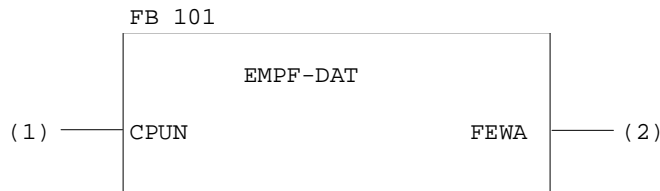
0048 LEER  :L   MB 244           Verbindungszaehler
0049      :I    1               inkrementieren
004A      :T   MB 244
004B      :L   KB  4           alle Verbindungs-
004C      :<F                    strecken bearbeitet?
004D      :SPM =SCHL
004E      :L   KB  0           regulaeres Programmende:
004F      :T   =FEWA          VKE = 0, FEWA = 0
0050      :BEA
0051      :
0052 FEWA  :L   KB 16           Programmende bei Fehler:
0053 OBFE  :T   =FEWA          VKE = 1, FEWA enthaelt
0054      :BE                    Fehler-/Warnungs-Nummer
    
```

FB 101: Empfangen von Datenwortbereichen

Vor Aufruf des FB 101 muß der Datenbaustein aufgeschlagen werden, der die Verbindungsliste enthält. Der Funktionsbaustein EMPF-DAT benötigt zum Auswerten der in der Verbindungsliste enthaltenen Informationen die Nummer der CPU, in der er aufgerufen wird.

Falls innerhalb des Funktionsbausteins die Funktion EMPFANGEN (OB 204) nicht korrekt abgearbeitet werden konnte, wird die jeweilige Fehler- oder Warnungsnummer im Ausgangsparameter FEWA übergeben und das VKE = 1 gesetzt. Zusätzlich enthält FEWA bei unzulässigem Eingangsparameter CPUN den Wert 16 (Bit-Nr. 4 = 1).

Der Funktionsbaustein EMPF-DAT verwendet die Merkerbytes MB 242 bis MB 255 als Schmiermerker.



Parameter-Name	Bedeutung	Art	Typ
CPUN	Nummer der CPU, auf der der FB 101 aufgerufen wird. Zulässig sind die Nummern 1 bis 4.	D	KF
FEWA	Fehler/Warnung (siehe Funktion EMPFANGEN / OB 204)	A	BY

Fortsetzung auf der nächsten Seite

Fortsetzung 1 von FB 101:

FB 101

LAE=88

NETZWERK 1 0000

NAME:EMPF-DAT

BEZ :CPUN E/A/D/B/T/Z: D KM/KH/KY/KC/KF/KT/KZ/KG: KF

BEZ :FEWA E/A/D/B/T/Z: A BI/BY/W/D: BY

```

000B      :LW  =CPUN                Fehler falls:
000C      :L   KB 1
000D      :<F
000E      :SPB =FEWA                CPU-Nr. <1
000F      :LW  =CPUN
0010      :L   KB 4
0011      :>F
0012      :SPB =FEWA                CPU-Nr >4
0013      :
0014      :L   KB 1                Verbindungszaehler
0015      :T   MB 242
0016      :
0017      :L   KB 16
0018      :T   MW 244                Zeiger auf Teilliste 2
0019      :
001A SUCH :L   MW 244                Teilliste 2 solange durch-
001B      :I   1                    suchen, bis der naechste
001C      :T   MW 244                Eintrag fuer die Empfangs-
001D      :B   MW 244                CPU mit der Nummer'CPUN'
001E      :L   DL 0                  gefunden ist.
001F      :LW  =CPUN
0020      :><F
0021      :SPB =SUCH
0022      :
0023      :B   MW 244
0024      :L   DR 0                Anzahl der reservierten
0025      :T   MB 243                Speicherbloেকে = 0 ?
0026      :L   KB 0
0027      :!=F
0028      :SPB =LEER
0029      :
002A      :L   MW 244                Nummer der Sende-CPU
002B      :L   KM 00000000 00001100 aus dem Zeiger auf die
002D      :UW                        Teilliste 2 bestimmen.
002E      :SRW 2
002F      :I   1
0030      :T   MB 246
0031      :
0032      :L   KB 246                SF-OB:
0033      :SPA OB 205                "Empfangsmoeglichkeit testen"
0034      :L   MB 248
0035      :SPB = OBF E                Abbruch bei Fehler
0036      :

```

Fortsetzung auf der naechsten Seite

Fortsetzung 2 von FB 101:

0037	:L	MB 249	Empfangskapazitaet = Anzahl
0038	:L	MB 243	der reservierten
0039	:><F		Speicherbloেকে ?
003A	:SPB	=LEER	
003B	:		
003C	EMPF	:L KB 246	SF-OB:
003D	:	SPA OB 204	"Empfangen eines Datenblocks"
003E	:L	MB 248	
003F	:SPM	=OBFE	Abbruch bei Fehler/Warnung
0040	:L	MB 249	bei Empfangskapazitaet = 0
0041	:L	KB 0	naechste Verbindung-
0042	:><F		strecke bearbeiten
0043	:SPB	=EMPF	
0044	:		
0045	LEER	:L MB 242	Verbindungszaehler
0046	:I	1	inkrementieren
0047	:T	MB 242	
0048	:L	KB 4	alle Verbindungsstrecken
0049	:<F		bearbeitet ?
004A	:SPM	=SUCH	
004B	:L	KB 0	regulaeres Programmende:
004C	:T	=FEWA	VKE = 0, FEWA = 0
004D	:BEA		
004E	:		
004F	FEWA	:L KB 16	Programmende bei Fehler:
0050	OBFE	:T =FEWA	VKE = 1, FEWA enthaelt
0051	:BE		Fehler-/Warnungs-Nummer

Anwendungsbeispiel**Anwendung der FB 100/101 auf dem AG S5-155U**Aufgabenstellung

Zwischen drei CPUs sollen Daten ausgetauscht werden:

- **von CPU 1 nach CPU 2:** Datenbaustein DB 3, DW 0 bis DW 127 (= 4 Blöcke)
- **von CPU 1 nach CPU 3:** Datenbaustein DX 4, DW 0 bis DW 63 (= 2 Blöcke)
- **von CPU 2 nach CPU 1
und CPU 3:** Datenbaustein DB 5, DW 0 bis DW 95 (= 3 Blöcke)

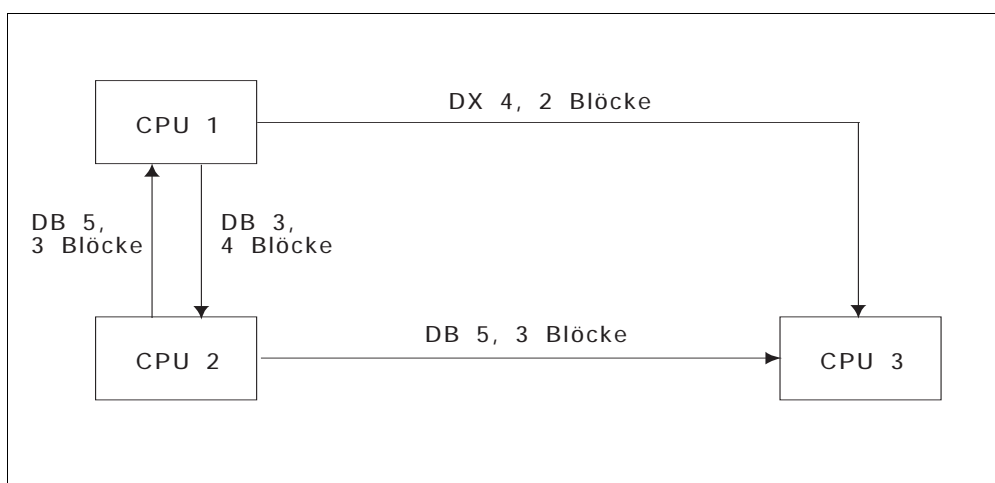


Bild 10-7 Datenaustausch zwischen 3 CPUs

Auf allen drei CPUs soll der Funktionsbaustein FB 1 die Schnittstelle zum zyklischen Anwenderprogramm bilden. Die CPU 1 soll bei NEUSTART die Funktion INITIALISIEREN (OB 200) aufrufen. Die Verbindungsliste soll im Datenbaustein DB 100 stehen.

Fortsetzung auf der nächsten Seite

Fortsetzung 1 des Anwendungsbeispiels:

Realisierung

1. Bausteine laden

Folgende Bausteine müssen zusätzlich zum OB 1 in die einzelnen CPUs geladen werden:

Funktion	CPU 1	CPU 2	CPU 3
Anlauf-OB	OB 20	—	—
Anwenderprogramm	FB 1	FB 1	FB 1
FB: SEND-DAT	FB 100	FB 100	FB 100
FB: EMPF-DAT	FB 101	FB 101	FB 101
Verbindungsliste	DB 100	DB 100	DB 100
Eingangs-DB	DB 5	DB 3	DB 5; DX 4
Ausgangs-DB	DB 3; DX 4	DB 5	—

2. Verbindungsliste erstellen

Die Verbindungsliste wird erstellt und in den Datenbaustein DB 100 eingetragen:

```

DB100                                LAE=37
                                      BLATT 1

- - Teilliste 1 - -

0:   KC = 'S1';                       von CPU 1 zu ..
1:   KY = 001,003;                     .. CPU 2 den DB 3 senden
2:   KY = 002,004;                     .. CPU 3 den DX 4 senden
3:   KY = 000,000;
4:   KC = 'S2';                       von CPU 2 zu ..
5:   KY = 001,005;                     .. CPU 1 den DB 5 senden
6:   KY = 001,005;                     .. CPU 3 den DB 5 senden
7:   KY = 000,000;
8:   KC = 'S3';
9:   KY = 000,000;
10:  KY = 000,000;
11:  KY = 000,000;
12:  KC = 'S4';
13:  KY = 000,000;
14:  KY = 000,000;
15:  KY = 000,000;
    
```

Fortsetzung 2 des Anwendungsbeispiels:

- - Teilliste 1 - -

```

16:      KC = 'S1';           von CPU 1 zu ..
17:      KY = 002,004;       .. CPU 2 vier Datenblöcke senden
18:      KY = 003,002;       .. CPU 3 zwei Datenblöcke senden
19:      KY = 004,000;
20:      KC = 'S2';           von CPU 2 zu ..
21:      KY = 001,003;       .. CPU 1 drei Datenblöcke senden
22:      KY = 003,003;       .. CPU 3 drei Datenblöcke senden
23:      KY = 004,000;
24:      KC = 'S3';
25:      KY = 001,000;
26:      KY = 002,000;
27:      KY = 004,000;
28:      KC = 'S4';
29:      KY = 001,000;
30:      KY = 002,000;
31:      KY = 003,000;

```

Die Datenwörter DW 16 bis DW 31 enthalten die für die Funktion manuelles INITIALISIEREN (OB 200) notwendige Zuordnungsliste.

3. Aufruf des OB 200 im Anlaufbaustein OB 20 für CPU 1 programmieren

Der OB 200 wird vom nachfolgend abgedruckten OB 20 der CPU 1 im Anlauf aufgerufen.

```

OB 20                                           LAE=yyABS

NETZWERK 1
0000      :L   KB 2           manuelles Initialisieren der
0001      :T   MB 246         Kacheln
0002      :
0003      :L   KY 1,100       im DB 100 ist die Zuordnungsliste
0005      :T   MW 248         ab dem Datenwort DW 16
0006      :L   KF+16         eingetragen
0008      :T   MW 250
0009      :
000A      :L   KB 246         SF-OB:
000B      :SPA OB 200         "Initialisieren"
000C      :
000D      :UN  M 252.5        Bausteinende, wenn kein
000E      :BEB               Initialisierungskonflikt
000F      :
0010      :
0011      :                  hier wird die Fehlerbehandlung im
0012      :                  Falle eines Initialisierungs-
0013      :                  konfliktes eingefügt (z.B. Stopp,
0014      :                  Meldung auf Drucker ausgeben,
                                oder ...)

00xx      :BE

```

Fortsetzung auf der nächsten Seite

Fortsetzung 3 des Anwendungsbeispiels:

4. Aufrufe der Funktionsbausteine in den FB 1 der CPUs programmieren

Auf jeder CPU wird das Anwenderprogramm um den Aufruf der Funktionsbausteine EMPF-DAT und SEND-DAT erweitert. Der abgedruckte Funktionsbaustein FB 1 ist für die CPU 1 bestimmt. Für den Ablauf auf den anderen CPUs muß lediglich der Eingangsparameter CPUN (CPU-Nummer) angepaßt werden.

```

FB 1                                                    LAE=yy

NETZWERK 1      0000
NAME:EM-SE
0000
0000      :A   DB100      Verbindungsliste DB 100
0001      :SPA FB101     Empfangen der Eingangs-
0002      :              Datenbausteine
0003 NAME :EMPF-DAT
0004 CPUN :    KF+1
0005 FEWA :    MB0
0006      :SPB =FEWA     Abbruch bei Fehler/Warnung
0007      :
0008      :
0009      :              Hier wird das zyklische Anwender-
000A      :              programm eingefuegt, das Daten aus
000B      :              den Eingangsdatenbausteinen liest
000C      :              und Daten in die Ausgangsdaten-
000D      :              bausteine eintraegt.
000E      :
000F      :
0010      :A   DB 100     Verbindungsliste DB 100
0011      :SPA FB100     Senden der Ausgangsdaten-
0012      :              bausteine
0012 NAME :SEND-DAT
0013 CPUN :    KF+1
0014 FEWA :    MB0
0015      :SPB =FEWA     Abbruch bei Fehler/Warnung
0016      :BEA
0017      :
0018 FEWA :              nach Fehler/Warnung Fehlerbehandlung
0019      :              durchfuehren (hier wird die Fehler-
001A      :              behandlung eingefuegt z.B. Stopp,
001B      :              Fehlermeldung auf Drucker oder
001C      :              Monitor ausgeben, oder ...)

00xx      :BE
    
```

PG-Schnittstellen und -Testhilfen

11

Inhalt von Kapitel 11

11.1	Übersicht	11 - 4
11.2	PG-Funktionen	11 - 5
11.2.1	Auskunft	11 - 6
11.2.2	Inbetriebnahme	11 - 7
11.2.3	Programmtest	11 - 8
11.3	Serielle Kopplung PG – AG über 1. oder 2. serielle Schnittstelle	11 - 16
11.4	Parallelbetrieb von zwei seriellen PG-Schnittstellen	11 - 17
11.4.1	Inbetriebnahme	11 - 19
11.4.2	Betrieb	11 - 19
11.4.3	Ablauf bei bestimmten Betriebsfällen	11 - 21
11.5	PG-Funktionen über S5-Bus	11 - 27
11.5.1	Anwendung	11 - 27
11.5.2	Arbeitsweise der PG-Funktionen über S5-Bus	11 - 29
11.5.3	Inbetriebnahme	11 - 31
11.5.4	Anzeige von Störungen	11 - 35

PG-Schnittstellen und -Funktionen

11

Dieses Kapitel informiert Sie darüber, wie Sie Ihr PG an eine CPU 948 koppeln können und welche Hilfen Ihnen die PG-Software bietet, um Ihr STEP-5-Programm zu testen.

Wenn Sie nur die Standard-PG-Schnittstelle (1. serielle PG-Schnittstelle) benutzen, brauchen Sie die Abschnitte 11.4 und 11.5 nicht zu lesen. Diese Abschnitte informieren Sie darüber, wie Sie über weitere Schnittstellen ein PG mit Ihrer CPU koppeln können. Sie erfahren darin ferner, was Sie bei der Benutzung der PG-Funktionen auf beiden Schnittstellen beachten müssen.

11.1 Übersicht

Das Laden und Testen Ihres Anwenderprogramms führen Sie mit Hilfe der Online-Funktionen der STEP-5-Software durch.

Dazu müssen Sie Ihre CPU mit dem PG koppeln. Für diese Kopplung stehen Ihnen folgende Schnittstellen zur Verfügung:

- Kopplung über die serielle Standard-Schnittstelle "PG - AG",
- Kopplung über die 2. serielle Schnittstelle der CPU 948,
- Kopplung über S5-Bus mit SINEC H1.

Die PG-Funktionen laufen auf den beiden seriellen Schnittstellen parallel ab, über SINEC-H1-Kopplung dagegen die PG-Funktionen nur **alternativ** zu denen auf den seriellen Schnittstellen.

Die PG-Funktionen bieten Ihnen folgende Unterstützung für die Inbetriebnahme und den Test Ihres STEP-5-Programms :

Tabelle 11-1 Hilfen für Inbetriebnahme und Test

Hilfe	Abschnitt
Auskunft	
Größe des internen RAMs und des freien Anwenderspeichers	"Speicherausbau"
Liste der geladenen Bausteine	"Bausteinliste"
Inhalt von Speicherwörtern/bytes und Peripheriebytes anzeigen	"Ausgabe Adresse"
Speicherverwaltung	
Speicher komplett löschen	"Urlöschen"
freien Speicherplatz zusammenfassen	"Speicher komprimieren"
Bausteine verwalten	"Baustein laden/löschen"
Programmtest	
CPU starten/anhalten	"Start/Stopp"
Befehlsfolge in einem Baustein testen	"Status"
einzelne Programmschritte testen	"Bearbeitungskontrolle"
Signalzustand von Prozeßvariablen anzeigen	"Status Variablen"
Ausgangssignale im Stopp ausgeben	"Steuern"
Prozeßvariable anzeigen/verändern	"Steuern Variablen"

11.2 PG-Funktionen

Hinweis

Die in diesem Abschnitt für die PG-Funktionen verwendeten Begriffe können u. U. von den Begriffen in Ihrer PG-Software abweichen.

Lesen Sie dazu bitte in Ihrem STEP-5-Handbuch nach.

Aufrufen und Bedienen

Wie Sie die einzelnen PG-Funktionen aufrufen und bedienen, lesen Sie bitte im Handbuch zu Ihrem Programmiergerät nach.

Kontrollpunkte

In der CPU werden die PG-Funktionen an definierten **Kontrollpunkten** ausgeführt. Bei der CPU 948 gibt es 4 verschiedene Kontrollpunkte. Jedem dieser Kontrollpunkte sind bestimmte Testhilfen zugeordnet.

Kontrollpunkt "Stopp"

An diesem Kontrollpunkt werden die PG-Funktionen aufgerufen, die im Betriebszustand STOP zulässig sind (z. B. "Start", "Urlöschen", "Speicher komprimieren" im Stopp).

Dieser Kontrollpunkt liegt **vor** dem Aufruf OB 39 (Baustein für die zyklische Bearbeitung im weichen STOP).

Kontrollpunkt "Zyklus"

An diesem Kontrollpunkt werden die PG-Funktionen aufgerufen, die während der zyklischen Programmbearbeitung ausgeführt werden sollen (z. B. "Speicher komprimieren" im RUN, "Stopp", "Status"). Dieser Kontrollpunkt liegt **vor** der Aktualisierung des Prozeßabbildes der Eingänge (PAE noch nicht aktualisiert).

Kontrollpunkt "Test"

Hier werden die PG-Funktionen aufgerufen, die ausgeführt werden sollen, sobald der vom Anwender vorgegebene Folgehaltepunkt erreicht ist (bei "Bearbeitungskontrolle", siehe Abschnitt 11.2.3).

Kontrollpunkt "Allgemeine Funktionen"

Diesen Kontrollpunkt gibt es sowohl im STOP als auch im RUN. Hier werden die Online-Funktionen aufgerufen, die in allen Betriebszuständen des AG ausgeführt werden können (z. B. "Baustein laden/löschen", "Status Variablen"). Im STOP liegt dieser Kontrollpunkt **vor** dem OB 39 (Baustein für die zyklische Programmbearbeitung im weichen STOP), im Zyklus **zwischen** dem Prozeßabbild der Eingänge (PAE bereits aktualisiert) und dem OB 1 (Baustein für die zyklische Programmbearbeitung im RUN).

**11.2.1
Auskunft**

Speicherausbau

Die CPU 948 wird mit zwei Speicherversionen geliefert, deren Größe Sie mit der PG-Funktion "Speicherausbau" erfragen können. Mit dieser Funktion werden dem PG folgende Daten über den Anwenderspeicher der CPU übergeben (ab PG-Software-Version V6.0 + "Delta-Diskette" für CPU 948):

- Speichergröße (640 kbyte/1664 kbyte)
- größter freier Anwender-Speicherblock
- Summe aller freien Anwender-Speicherblöcke

Die PG-Software V6.3 + "Delta-Diskette" für CPU 948 zeigt den Speicherausbau exakt an.

Bei älteren PG-Software-Versionen (z. B. V3.0 oder MT1.0) erfolgt die Darstellung bei "Speicherausbau" wie bei der CPU 946/947 (siehe Bild 11-1). Den Gesamtspeicherausbau der CPU 948 müssen Sie dann aus der Summe der Modul-Werte berechnen.

A G - I N F O R M A T I O N						SIMATIC S5 / ODS0C	
S P E I C H E R A U S B A U A G 1 5 5 U							
BAUGRUPPE	MODUL	TYP	STARTADRESSE	ENDADRESSE	LAENGE		
0	1	RAM	00000	0FFFF	64	KW	} 1)
1	1	RAM	10000	2FFFF	128	KW	
1	2	RAM	30000	4FFFF	128	KW	
1	3		MODUL LEER	ODER NICHT	GESTECKT		
2	1		MODUL LEER	ODER NICHT	GESTECKT		
2	2		MODUL LEER	ODER NICHT	GESTECKT		
2	3		MODUL LEER	ODER NICHT	GESTECKT		
			GROESSTER FREIER RAM-SPEICHERBLOCK		:	314800 WORTE	
			SUMME ALLER FREIEN RAM-SPEICHERBLOECKE		:	315664 WORTE	
F1	F2	F3	F4	F5	F6	F7	F8
AUSG ADR	SPAUS	SYSPAR	BSTACK	USTACK			ZURUECK

1) entspricht dem Speicherausbau der CPU 948-1: 320 kW = 640 kbyte

Bild 11-1 PG-Ausgabe des Speicherausbaus

Ausgabe Buch

Wenn Sie sich am PG eine Liste aller programmierten Bausteine ausgeben lassen, so wird bei der CPU 948 an Stelle der Systemprogramm-Bausteine der OB 0 angezeigt.

Die Funktion ist zulässig in den Betriebszuständen RUN, WEICHER STOP, HARTER STOP und kann auch innerhalb der Funktion "Bearbeitungskontrolle" aufgerufen werden.

Ausgabe Adresse

Mit der Funktion "Ausgabe Adresse" können Sie sich am PG den Inhalt von Speicher- und Peripherieadressen hexadezimal anzeigen lassen. Sie können alle Adressen ansprechen (RAM, S5-Bus, unbestückte Bereiche). Im Bereich des Prozeßabbildes wird kein ADF ausgelöst, im Peripheriebereich entsteht kein QVZ.

Im byteadressierbaren Bereich (Merker, Prozeßabbild) wird das High-Byte als 'FF' dargestellt.

**11.2.2
Inbetriebnahme****Urlöschen**

Sie können mit der Funktion "Löschen alle Bausteine" eine CPU vom PG aus urlöschen. Dabei wird das Urlöschen unbedingt durchgeführt (siehe Abschnitt 4.2).

Speicher komprimieren

Mit dieser Funktion werden alle gültigen Bausteine im Anwenderspeicher bündig an den Anfang des Anwenderspeichers geschoben. Lücken, die beim Löschen oder Korrigieren von Bausteinen entstehen, verschwinden. Hierzu wird jeweils ein kompletter Baustein zum Anfang des Speicherbereiches hin geschoben. Im Idealfall entsteht dadurch aus vielen kleinen Lücken eine große Lücke, in die wieder Bausteine geladen werden können.

Die Funktion kann im Betriebszustand RUN und im weichen STOP aufgerufen werden. Im Betriebszustand RUN werden DB- und DX-Datenbausteine, die länger sind als 512 Datenwörter, **nicht** verschoben. Im STOP werden **alle** Bausteine verschoben.

Damit während des Komprimierens bei einem Netz-Ausfall keine Daten verloren gehen, werden die Bausteine über einen Zwischenpuffer geschoben. Reicht dieser Puffer für eine Zwischenspeicherung nicht aus, wird mit dem Komprimieren an der nächsten Lücke im Speicher fortgefahren. Aus diesem Grund können beim Komprimieren noch Restlücken verbleiben.

Beachten Sie dazu Abschnitt 8.3.

Baustein übertragen

Mit dieser Funktion übertragen Sie neue oder schon vorhandene Code- und Datenbausteine in den Anwenderspeicher der CPU.

Ist ein Baustein bereits im Anwenderspeicher der CPU vorhanden, so wird er als ungültig markiert und der neue erhält die Gültigmarkierung.

Baustein löschen

Mit dieser Funktion erklären Sie Code- und Datenbausteine im Anwenderspeicher der CPU für ungültig.

Der Platz, an dem diese Bausteine gelegen haben, wird freigegeben und kann neu benutzt werden.

**11.2.3
Programmtest**

Start/Stop

Die PG-Bedienung entspricht der manuellen Bedienung.

Mit dem Aufruf der PG-Funktion "Stopp" im Betriebszustand RUN bringen Sie das Automatisierungsgerät in den Stoppzustand.

In den Steuerbits ist PGSTP angekreuzt. Bei Mehrprozessorbetrieb ist bei den anderen CPU das Steuerbit HALT gesetzt.

Sie verlassen den Betriebszustand WEICHER STOP mit NEUSTART oder WIEDERANLAUF. Im Einzelprozessorbetrieb verläßt die CPU den Stoppzustand. Im Mehrprozessorbetrieb wird zunächst die Anlaufart registriert (Steuerbit NEUDF oder WIEDF ist gesetzt), die CPU bleibt aber im Stopp. Mit der nachfolgenden Bedienung "Systemstart" können Sie das Automatisierungsgerät starten. Dies entspricht der Bedienung des Koordinators (Schalter auf RUN).

Mit der PG-Funktion "Start" können Sie im Mehrprozessorbetrieb nacheinander bei allen CPUs den gewünschten Anlauf durchführen und erst bei der letzten CPU das AG starten.

Status Baustein

Mit Hilfe der Funktion "Status" testen Sie an beliebiger Stelle im Anwenderprogramm zusammenhängende Befehlsfolgen (STEP-5-Operationen) in einem Baustein.

Zu jedem ausgeführten Befehl im Baustein werden die aktuellen Signalzustände der Operanden, die AKKU-Inhalte, das VKE usw. am Programmiergerät ausgegeben (Schrittbetrieb). Auch die Parametrierung von Funktionsbausteinen kann auf diese Weise getestet werden (Blockbetrieb):

Angezeigt werden die Signalzustände der Aktualoperanden.

Funktion aufrufen und Haltepunkt vorgeben

Wenn Sie die Funktion "Status" am PG aufrufen und Bausteinart und Bausteinnummer (evtl. mit Schachtelreihenfolge und Suchbegriff) des zu testenden Bausteins eingeben, so geben Sie damit einen sog. "Haltepunkt" vor.

Bei Aufruf der Funktion während der Programmbearbeitung im RUN wird die Programmbearbeitung solange fortgesetzt, bis der durch den vorgegebenen Haltepunkt markierte Befehl in der richtigen Schachtelreihenfolge erreicht ist. Danach werden die überwachten Befehle jeweils bis zur Befehlsgrenze ausgeführt und die Ergebnisse der Befehlsbearbeitung am PG ausgegeben.

Hinweis

Die Ergebnisse der Befehlsbearbeitung werden nicht in jedem folgenden Durchlauf ausgegeben.

Einschachtelungen und Unterbrechungen

Eine durch einen Haltepunkt markierte Befehlsfolge wird vollständig durchlaufen, auch wenn zwischendurch an einer Befehlsgrenze eine andere Programmbearbeitungsebene (z. B. ein Fehler- oder Alarm-OB) eingeschachtelt und bearbeitet wird. Dabei können Sie erkennen, ob Daten durch eine eingeschachtelte Programmbearbeitung verändert wurden.

Führt in einer eingeschachtelten Programmbearbeitungsebene eine Unterbrechungsursache die CPU in den STOP, so werden im Stoppzustand die Daten bis zu demjenigen Befehl ausgegeben, der als letzter vor der Einschachtelung ausgeführt worden ist. Die Daten der restlichen Befehle werden mit '0' aufgefüllt (auch SAZ = 0).

Die Funktion "Status" ist zulässig in den Betriebszuständen RUN, ANLAUF (OB 20, OB 21 und OB 22) und im WEICHEN STOP (**nur** OB 39).

Operationen B MW/B DW

Wenn bei laufender Funktion "Status Baustein" der Cursor genau auf die Folgeoperation zu B MW bzw. B DW positioniert wird, erscheint auf dem PG die Meldung "Anweisung nicht bearbeitet".

Abhilfe:

Vermeiden Sie, den Cursor auf die Folgeoperation zu B MW bzw. B DW zu positionieren.

*Anwendung bei älteren
PG-Software-Versionen*

Wenn Sie bei älteren PG-Softwareversionen in der PG-Funktion "Status Baustein" den Cursor schnell bewegen, entsteht dabei pro Cursor-Bewegung (Zeile) eine Wartezeit von ca. 3 bis 5 s.

Abhilfe:

Brechen Sie mit der Abbruch-Taste den Status ab, positionieren Sie den Cursor neu und setzen Sie den Status fort. Dabei entstehen keine Wartezeiten.

Bearbeitungskontrolle

Mit der Funktion "Bearbeitungskontrolle" testen Sie an beliebiger Stelle im Anwenderprogramm einzelne Programmschritte. Dazu halten Sie die Programmbearbeitung an und lassen die CPU dann einen Befehl nach dem anderen bearbeiten. Zu jedem ausgeführten Befehl werden die aktuellen Signalzustände von Operanden, die AKKU-Inhalte, das VKE usw. am Programmiergerät ausgegeben.

*Testhilfe aufrufen und
1. Haltepunkt vorgeben*

Ein Aufruf der Testhilfe ist in den Betriebszuständen RUN und WEICHER STOP zulässig. Beim Aufruf geben Sie Baustein**art** und Baustein**nummer** (evtl. mit Schachtelreihenfolge) des zu testenden Bausteins an und markieren am PG den ersten Befehl, dessen Daten ausgegeben werden sollen. Damit geben Sie den 1. Haltepunkt vor.

Wenn Sie den 1. Haltepunkt während der **Programmbearbeitung** vorgeben, setzt die CPU die Programmbearbeitung solange fort, bis der durch den vorgegebenen Haltepunkt markierte Befehl erreicht ist. Der Befehl wird bis zur Befehlsgrenze ausgeführt. (Die Befehle BMW und BDW werden **einschließlich** des substituierten Befehls bearbeitet.) Jetzt prüft die CPU, ob die aktuelle Baustein-Schachtelungsreihenfolge mit der von Ihnen angegebenen übereinstimmt. Stimmt die Reihenfolge nicht überein, setzt die CPU die Programmbearbeitung fort.

Wird der vorgegebene Haltepunkt nicht erreicht (z. B. weil die CPU im STOP ist oder aufgrund einer Dauerschleife im Anwenderprogramm), erscheint am PG die Meldung "Anweisung wird nicht bearbeitet". Die Funktion und der vorgegebene Haltepunkt bleiben jedoch weiterhin aktiv.

Ist die Schachtelungstiefe hingegen korrekt, wird die Befehlsausgabe gesperrt (Leuchtdiode BASP = an) und die Daten des bearbeiteten Befehls werden am PG ausgegeben. Die CPU wartet daraufhin auf die weitere PG-Bedienung.

Operationen B MW/B DW

Bei laufender Funktion "Bearbeitungskontrolle" läßt sich der Cursor nicht über die Folgeoperation zu B MW/B DW bewegen.

Abhilfe:

Brechen Sie die Funktion ab, überspringen Sie die o. g. Operationsfolge und setzen die neuen Haltepunkt hinter die Folgeoperation zu B MW bzw. B DW.

*Aufruf der Testhilfe im
WEICHEN STOP*

Auch im Stoppzustand können Sie die Funktion "Bearbeitungskontrolle" aufrufen und einen ersten Haltepunkt vorgeben. Die CPU bleibt weiterhin im Stoppzustand. Sie können jetzt sowohl einen NEU-START als auch einen MANUELLEN WIEDERANLAUF durchführen. Die CPU führt die Programmbearbeitung bis zum markierten Befehl aus und verfährt dann wie oben.

*Funktion fortführen und
weiteren Haltepunkt
vorgeben*

Ausgangspunkt: Die CPU hat den 1. Haltepunkt bearbeitet.

Um die Funktion fortzuführen, haben Sie zwei Möglichkeiten.

1. Sie geben einen **Folgehaltepunkt** vor:

Der vorgegebene Haltepunkt wird um einen Befehl verschoben. Die CPU setzt die Programmbearbeitung um diesen einen Befehl fort. Wenn der Befehl bis zur Befehlsgrenze bearbeitet ist, gibt die CPU Daten aus und wartet wieder auf die weitere Bedienung. Wird der Folgebefehl jedoch in einer eingeschachtelten Programmbearbeitungsebene erreicht, setzt die CPU die Programmbearbeitung fort. Der Folgehaltepunkt bleibt weiterhin vorgegeben.

Hinweis

Im Stoppzustand können Sie **keinen** Folgehaltepunkt vorgeben!

2. Sie geben einen **neuen** Haltepunkt vor:

Sie geben am PG einen beliebigen anderen Befehl im gleichen oder in einem anderen Baustein vor. Die CPU setzt die Programmbearbeitung fort, bis sie den neuen Haltepunkt erreicht. Der Befehl wird bis zur Befehlsgrenze bearbeitet. Dann gibt die CPU die Daten aus.

Haltepunkt zurücknehmen

Ist ein vorgegebener Haltepunkt noch nicht erreicht, so haben Sie die Möglichkeit, diesen nachträglich zurückzunehmen, indem Sie am PG die Abbruchtaste betätigen. Danach können Sie einen neuen Haltepunkt vorgeben oder "Bearbeitungskontrolle Ende" aufrufen.

Funktion abbrechen

Durch Aufrufen von "Bearbeitungskontrolle Ende" können Sie während der Programmbearbeitung und im Stoppzustand die Funktion abbrechen. Die CPU geht in den weichen STOP (bzw. bleibt im STOP). Die STOP-LED blinkt langsam. In den Steuerbits wird BEARBE angekreuzt. Anschließend ist ein NEUSTART erforderlich. Tritt während der Funktion "Bearbeitungskontrolle" ein Schnittstellenfehler auf (Unterbrechung am PG-Kabel), wird die Funktion abgebrochen wie oben.

Einschachtelungen bei "Unterbrechbarkeit an Befehlsgrenzen"

Bei aufgerufener Funktion "Bearbeitungskontrolle" können andere Programmverarbeitungsebenen eingeschachtelt werden, wenn die Betriebsart "Unterbrechbarkeit an Befehlsgrenzen" eingestellt ist..

Wenn der Befehl am Haltepunkt bearbeitet ist und an dieser Stelle eine andere Programmverarbeitungsebene aufgerufen ist (z. B. ein Fehler-OB oder ein Alarm-OB), so wird diese zunächst vollständig bearbeitet und erst danach die Bearbeitung am Folgehaltepunkt fortgesetzt.

Hinweis

Die Daten werden an der Befehlsgrenze gelesen und dort ausgegeben. Alle zugehörigen Einschachtelungen sind noch nicht bearbeitet.

Das Ablaufprinzip der Funktion "Bearbeitungskontrolle" zeigt Bild 11-2.

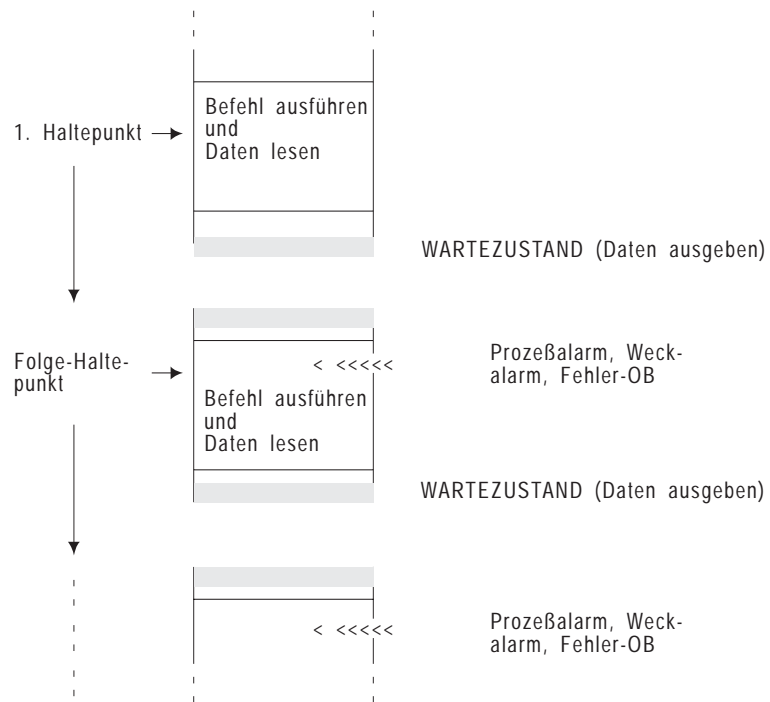


Bild 11-2 Testablauf bei "Bearbeitungskontrolle"

Hinweis

Wenn der Befehl am Haltepunkt bearbeitet ist und eine Einschachtelung angefordert ist, haben Sie die Möglichkeit, einen Haltepunkt auf einen Befehl der Einschachtelung zu setzen. So können Sie z. B. bei einem Befehl, der einen QVZ auslöst, direkt danach den QVZ-Fehler-OB beobachten.

Unterbrechungen

- Programmbearbeitung → Stoppzustand:

Tritt während der Programmbearbeitung eine Unterbrechung auf (z. B. Mehrprozessor-Stopp, Peripherie unklar, STOP-Schalter, Fehler-OB nicht programmiert usw.) und der vorgegebene Haltepunkt ist noch nicht erreicht, so geht die CPU in den Stoppzustand. Wenn danach ein Anlauf (NEUSTART oder MANUELLER WIEDERANLAUF) durchgeführt wird, bleibt die Funktion "Bearbeitungskontrolle" weiterhin aufgerufen; der Haltepunkt ist weiterhin vorgegeben.

- Befehlsbearbeitung am Haltepunkt → Stoppzustand:

Wenn während der Befehlsbearbeitung am Haltepunkt oder Folgehaltepunkt Stopp-Bedingungen auftreten (STOP-Schalter, STEP-5-Befehl STS, Fehler-OB nicht programmiert), so geht die CPU unmittelbar in den weichen STOP und übergibt dort die Daten. Wird im Stoppzustand kein neuer Haltepunkt vorgegeben, ist nach einem Anlauf die Funktion "Bearbeitungskontrolle" weiterhin aufgerufen.

Während die Funktion "Bearbeitungskontrolle" aufgerufen ist, können grundsätzlich weitere Funktionen am PG aufgerufen werden. Zulässig ist der Aufruf von

- Ausgabe USTACK
- Ausgabe BSTACK
- Baustein laden
- Baustein lesen
- Baustein löschen
- Bausteinliste ausgeben
- Steuern Variablen
- Steuern

In seltenen Fällen kann es dabei passieren, daß die Funktion beendet wird und sich die CPU danach im STOP befindet.

Status Variablen

Mit der Testhilfe "Status Variablen" können Sie sich die aktuellen Signalzustände bestimmter Operanden (Prozeßvariablen) anzeigen lassen.

Wenn ein Kontrollpunkt erreicht ist, wird der zu diesem Zeitpunkt aktuelle Signalzustand der gewünschten Prozeßvariablen ausgegeben. Sie können alle Prozeßvariablen angeben (Eingänge, Ausgänge, Merker, Zeitzellen, Zähler und Datenwörter). Im Bereich des Prozeßab-bilds wird kein ADF ausgelöst.

Ablauf im RUN

Läuft die Funktion im Betriebszustand RUN, so werden bei Erreichen des Kontrollpunktes die Signalzustände der Operanden abgefragt und angezeigt. Eingänge werden aus dem **Prozeßabbild** gelesen. Solange die Funktion nicht abgebrochen wird, werden die Signalzustände zyklisch aktualisiert.

Hinweis

Wird der Systemkontrollpunkt nicht erreicht, erfolgt keine Ausgabe der Signalzustände (z. B. bei einer lang andauernden Programmschleife im Anwenderprogramm)!

Ablauf im WEICHEN STOP

Wenn die Funktion im weichen STOP läuft, werden die Signalzustände der Operanden ausgegeben, wie sie am Systemkontrollpunkt vorliegen. Wichtig ist dabei, daß die Eingänge direkt von der Peripheriebaugruppe abgefragt und ausgegeben werden.

Steuern

Mit Hilfe der Funktion "Steuern" können Sie die Ausgangsbytes des Automatisierungsgerätes auf einen gewünschten Signalzustand setzen.

Hinweis

Die Funktion "Steuern" ist nur im **Stoppzustand** (weicher STOP oder innerhalb der Funktion "Bearbeitungskontrolle") zulässig!

Ablauf der Funktion

Bei Aufruf der Funktion im Stoppzustand wird die Befehlsausgabesperre aufgehoben (BASP = aus). Die **gesamte** digitale Peripherie wird gelöscht (Wert '0'). Während des Löschens der Peripherie ist die Funktion nicht unterbrechbar. Eventuell auftretender Quittungsverzug (QVZ) wird dabei ignoriert.

Die Peripherieausgänge werden byteweise gesteuert.

Im Mehrprozessorbetrieb können Sie **alle** Peripherieausgänge steuern (unabhängig von einer Peripheriezuteilung im DB 1).

Ein evtl. auftretender Quittungsverzug wird erkannt, während die Ausgänge geändert werden (PG-Meldung "Signalformer nicht vorhanden").

Beenden der Funktion

Sie beenden die Funktion durch Drücken der Abbruchtaste am PG. Die Befehlsausgabesperre ist wieder aktiv (Leuchtdiode BASP = an). Die Funktion wird außerdem beendet, wenn die CPU zwischen dem Aufruf der Funktion und dem eigentlichen Steuern in den Betriebszustand RUN übergeht.

Steuern Variablen

Mit der Testhilfe "Steuern Variablen" können Sie die Werte von Operanden (Prozeßvariablen) im Prozeßabbild anschauen und verändern. Dies ist im RUN, im weichen STOP und innerhalb der Funktion "Bearbeitungskontrolle" möglich. Sie können alle Prozeßvariablen (Eingänge, Ausgänge, Merker, Zeiten, Zähler und Datenwörter) angeben.

Besonderheiten

Die Änderung wird am nächsten Kontrollpunkt wirksam, d. h. unabhängig vom Systemkontrollpunkt (Zyklusanfang oder -ende).

Beachten Sie, daß die gesteuerten Werte nachträglich überschrieben werden können (z. B. durch das Anwenderprogramm oder durch die Prozeßabbildaktualisierung)!

Hinweis

Das PG steuert die Prozeßvariablen byteweise.

Wenn Sie **mehrere Operanden** steuern, dann werden die geänderten Bytes nacheinander über mehrere Zyklen verteilt im Speicher geändert.

11.3 Serielle Kopplung PG – AG über 1. oder 2. serielle Schnittstelle

Für die serielle Kopplung PG – AG existieren folgende Möglichkeiten:

- Direkte Verbindung CPU – über die Standard-Steckleitung,
- Verbindung zum PG über den Koordinator KOR C . Dabei wird das PG über die Steckleitung mit dem Koordinator verbunden. Damit ist die 1. serielle Schnittstelle nicht mehr benutzbar.
- Verbindung zum PG über einen PG-Multiplexer 757. Die zulässigen Steckverbindungen können Sie dem Systemhandbuch 135U/155U /2/ entnehmen.
- Verbindung zum PG über SINEC H1/L2/L1 und "Affenschaukel"; dabei können auch KOR C oder PG-Multiplexer zwischengeschaltet werden.

11.4 Parallelbetrieb von zwei seriellen PG-Schnittstellen

Die zweite Schnittstelle der CPU 948 (SI 2) können Sie ebenso wie die erste Schnittstelle als **PG-Schnittstelle** nutzen.

Um Ihr PG über diese Schnittstelle anzukoppeln zu können, müssen Sie zusätzlich zu Ihrer CPU 948 das PG-Schnittstellenmodul bestellen (die Bestell-Nummer dazu finden Sie im Systemhandbuch 135U/155U /2/).

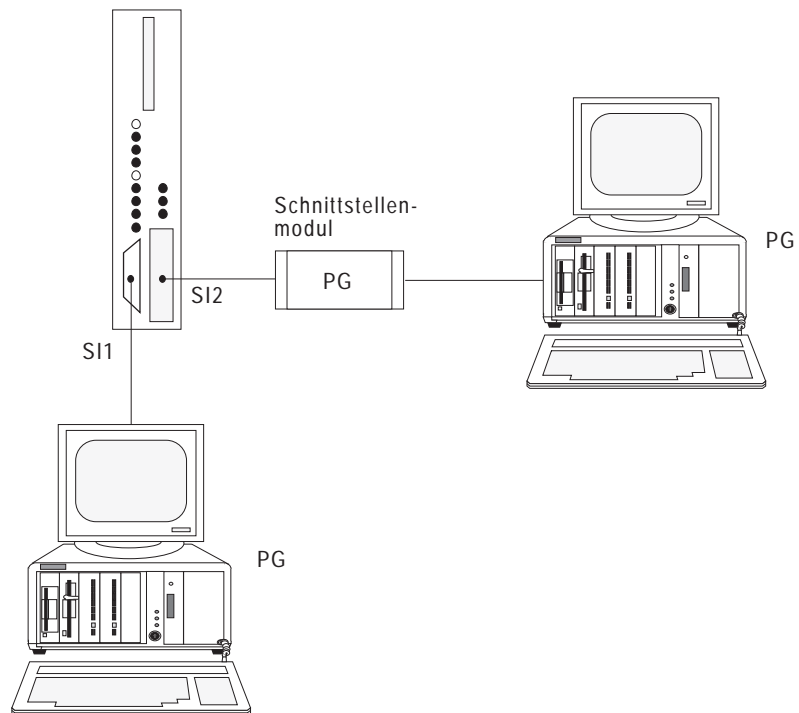


Bild 11-3 Zweite Schnittstelle als PG-Schnittstelle nutzen

Sämtliche PG-Funktionen stehen Ihnen auf beiden Schnittstellen zur Verfügung. In den folgenden Abschnitten finden Sie nur die Informationen, die Sie benötigen, wenn Sie gleichzeitig an beiden Schnittstellen mit Programmiergeräten oder OPs arbeiten.

Konfigurations-Beispiele

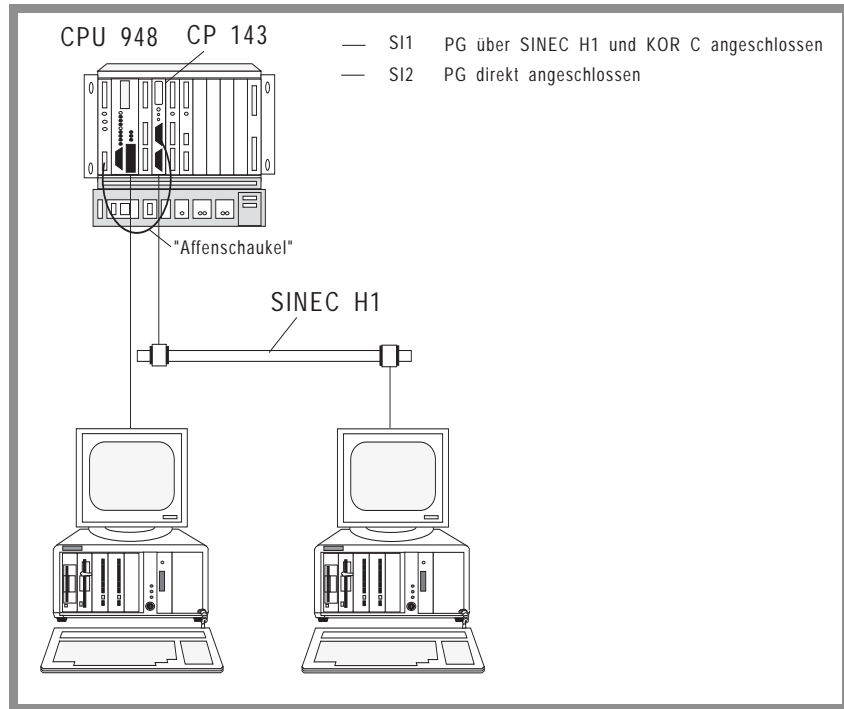


Bild 11-4 1. Konfigurations-Beispiel

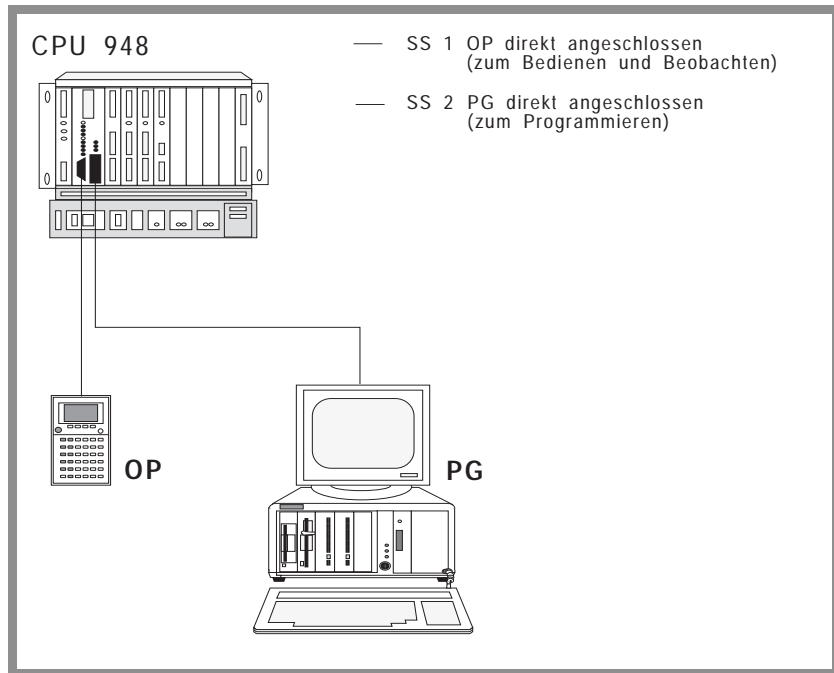


Bild 11-5 2. Konfigurations-Beispiel

**11.4.1
Inbetriebnahme**

Um die zweite Schnittstelle der CPU 948 als PG-Schnittstelle in Betrieb zu nehmen, gehen Sie in folgenden Schritten vor:

Schritt	Aktion
1	Bauen Sie das PG-Modul in die CPU 948 ein. (lesen Sie die Anleitungen dazu bitte im Anhang nach
2	Schließen Sie das PG an die serielle Schnittstelle SI2 an.

**11.4.2
Betrieb**

Wenn Sie die 2. serielle Schnittstelle als PG-Schnittstelle einsetzen, steht auf jeder Schnittstelle der volle Funktionsumfang der Standard-PG-Schnittstelle zur Verfügung. Dies gilt solange, wie die einzelnen Funktionen sich gegenseitig nicht beeinflussen, d. h. sequentiell nacheinander aufgerufen werden.

Zum Verständnis der Ausnahmen ist es sinnvoll, die PG-Funktionen in drei Gruppen zu unterteilen:

Gruppe	Kennzeichen
kurzlaufende Funktionen	Funktionen, die einen Auftrag ausführen und anschließend beendet sind. (z. B. "Übertragen", "Löschen" usw.)
langlaufende Funktionen	Funktionen, die eine feste Anzahl von Aufträgen bearbeiten: - "Steuern", - "Bearbeitungskontrolle".
zyklische Funktionen	Die Funktionen führen einen Auftrag immer wieder aus, bis sie vom Anwender beendet werden: - "Status Baustein", - "Status Variablen", - "Steuern Variablen".



Vorsicht

Bei langlaufenden und zyklischen Funktionen müssen Sie den Aufruf dieser Funktionen auf beiden PGs koordinieren.

In der nachfolgenden Tabelle sind die Funktionspaare aufgelistet, die Sie **nicht parallel** betreiben können.

Tabelle 11-2 Funktionen, die nicht parallel auf zwei PGs ablaufen

Am ersten PG läuft die Funktion:	Diese Funktion ist am zweiten PG nicht möglich:
"Steuern"	jede Funktion
"Bearbeitungskontrolle"	jede Funktion
eine "Status"-Funktion	"Steuern"
eine "Status"-Funktion	"Bearbeitungskontrolle"
eine "Status"-Funktion	"Urlöschen"
"Status" auf nicht bearbeiteten oder langlaufenden Baustein	jede Funktion

Wenn Sie dies mißachten, meldet das zweite PG einen Fehler, z. B.: *"AS-Funktion gesperrt: laufende Funktion"*.

Die gleiche Fehlermeldung oder *"Zeitüberlauf bei Datenaustausch mit AG"* erscheint, wenn die CPU 948 gerade Funktionen des anderen PGs bearbeitet, welche den Zugriff Ihres PG innerhalb der Zeitüberwachung verhindern. Ihre Eingabe wird dann abgewiesen. Wiederholen Sie Ihre Eingabe, nachdem die Funktionen des anderen PGs abgeschlossen sind.

Hinweis

Durch unterschiedlichen Leistungs- und Funktionsumfang sind Zeitüberwachung und Fehlermeldeverhalten nicht bei allen PGs und OP gleich.

Wenn Sie gleichzeitig an beiden PGs die Funktion "Speicherausbau" aktivieren, kann es zu falschen Anzeigen kommen.



Vorsicht

Wenn Sie gleichzeitig online an beiden PGs Bausteine eingeben, korrigieren oder löschen, so müssen Sie beachten, daß die Bausteine nicht vor dem Zugriff durch das jeweils andere PG geschützt sind.

"Status" auf nicht angesprungenen Baustein oder "Status" im STOP blockiert die andere Schnittstelle für alle Funktionen.

**11.4.3
Ablauf bei bestimmten
Betriebsfällen**

**Paralleler Betrieb bei
kurzlaufenden Funktionen**

Wenn Sie an beiden Schnittstellen gleichzeitig mit Programmiergeräten arbeiten, versuchen beide PGs, unabhängig voneinander Ihre jeweiligen Funktionen auszuführen. Solange sie ihre Aufträge zeitlich versetzt an die CPU übertragen, werden diese in der Reihenfolge ihres Eintreffens nacheinander bearbeitet.

Es kann aber vorkommen, daß die CPU 948 zwei Aufträge entweder gleichzeitig erhält oder einen Auftrag vom zweiten PG erhält, während noch ein Auftrag des ersten PGs läuft.

Da die gleichzeitige Bearbeitung nicht möglich ist, werden diese beiden Aufträge nacheinander bearbeitet; allerdings sind die Wartezeiten für den zweiten Auftrag meistens so gering, daß sie für den Anwender kaum in Erscheinung treten.

Es ergibt sich somit für gleichzeitige Aufträge folgender Ablauf:

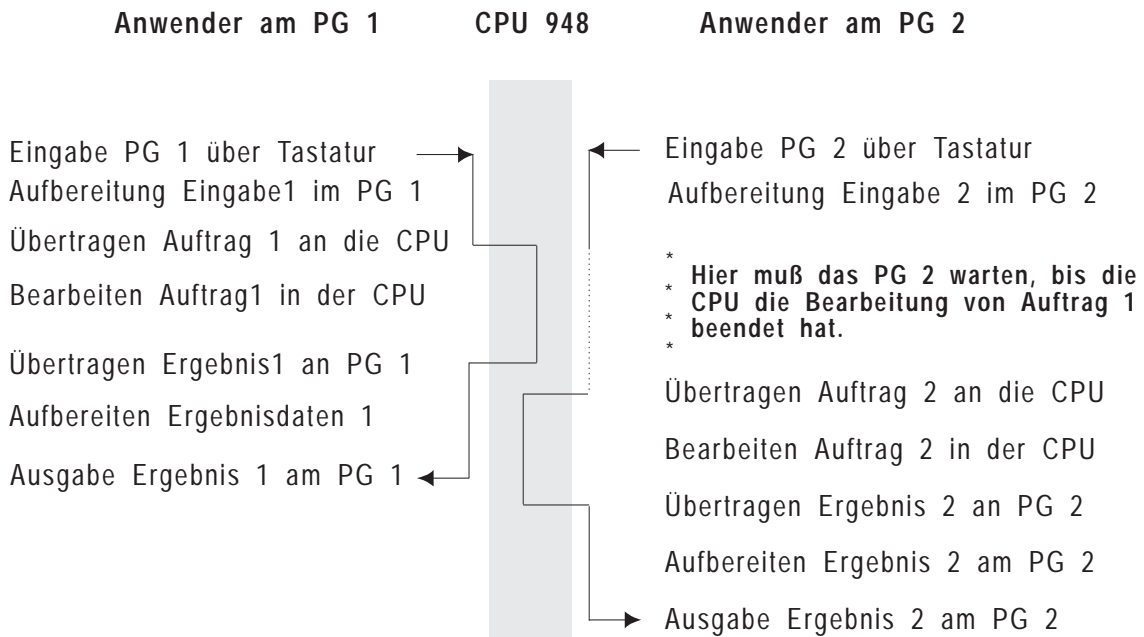


Bild 11-6 Zeitlicher Ablauf bei gleichzeitigen Aufträgen

Aus diesem Ablauf ist zu ersehen, daß Sie zwar an beiden PGs unabhängig voneinander arbeiten können, daß aber eine gegenseitige Beeinflussung auftritt.

So kann es passieren, daß beide PG gleichzeitig denselben Baustein bearbeiten oder über ein PG ein Baustein gelöscht wird, den das andere PG gerade bearbeitet.

Sie müssen also bei jeder Bearbeitung berücksichtigen, inwieweit Sie das Verhalten des anderen PG beeinflussen.

Paralleler Betrieb bei langlaufenden Funktionen

Die langlaufenden Funktionen "Steuern" und "Bearbeitungskontrolle" können keine andere Funktion unterbrechen und können auch von keiner anderen Funktion unterbrochen werden. Sie dürfen damit auch nicht parallel ausgeführt werden, d. h. sie werden wie ein Standardauftrag als ein Block behandelt.

Paralleler Betrieb bei zyklischen Funktionen

Zyklische Funktionen können parallel sowohl zu zyklischen als auch zu kurzlaufenden Funktionen ausgeführt werden. Als Beispiel wird hier der Standardablauf für die Funktion "Status Variablen" dargestellt.

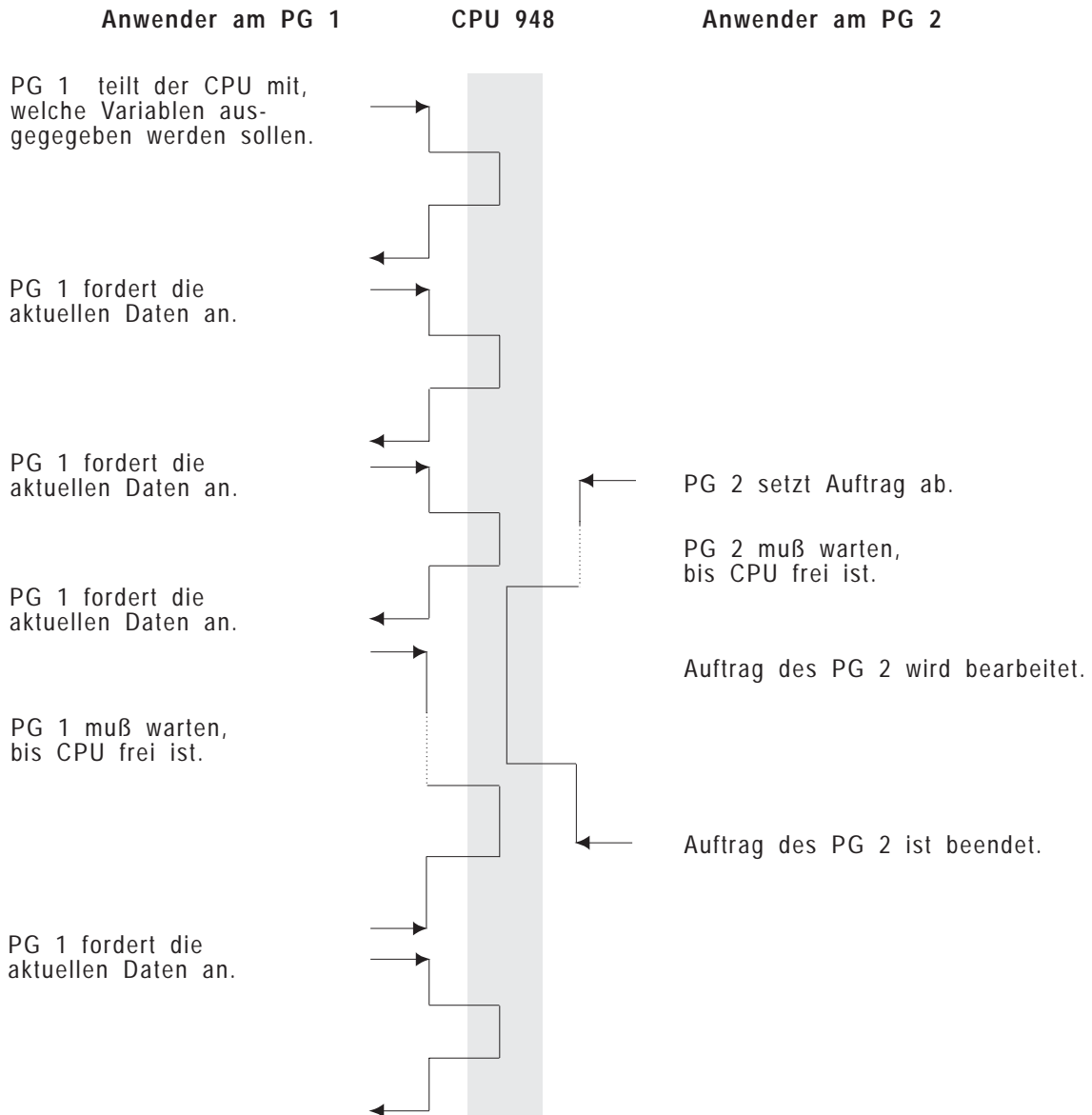


Bild 11-7 Typischer zeitlicher Ablauf einer zyklischen Funktion mit paralleler kurzlaufender Funktion

Um einem zweiten PG die Möglichkeit zu bieten, auch einen Auftrag an die CPU zu senden, wird die Statusfunktion zwischen zwei Anforderungen unterbrochen und nach dem eingeschobenen Auftrag wieder fortgesetzt. Da die unterbrechende Funktion zu ihrer Bearbeitung CPU-Leistung benötigt, muß die gesamte CPU-Systemleistung auf beide Funktionen aufgeteilt werden, z. B. werden bei der Funktion "Status Variablen" die auszugebenden Daten etwas langsamer aktualisiert.

Bei gleichzeitigem Arbeiten an beiden PGs ergibt sich der in Bild 11.8 dargestellte Ablauf.

Dieser gilt auch für den Fall, daß auf beiden PGs zyklische Funktionen ablaufen; die beiden PGs wechseln sich dann beim Zugriff auf die CPU ab.

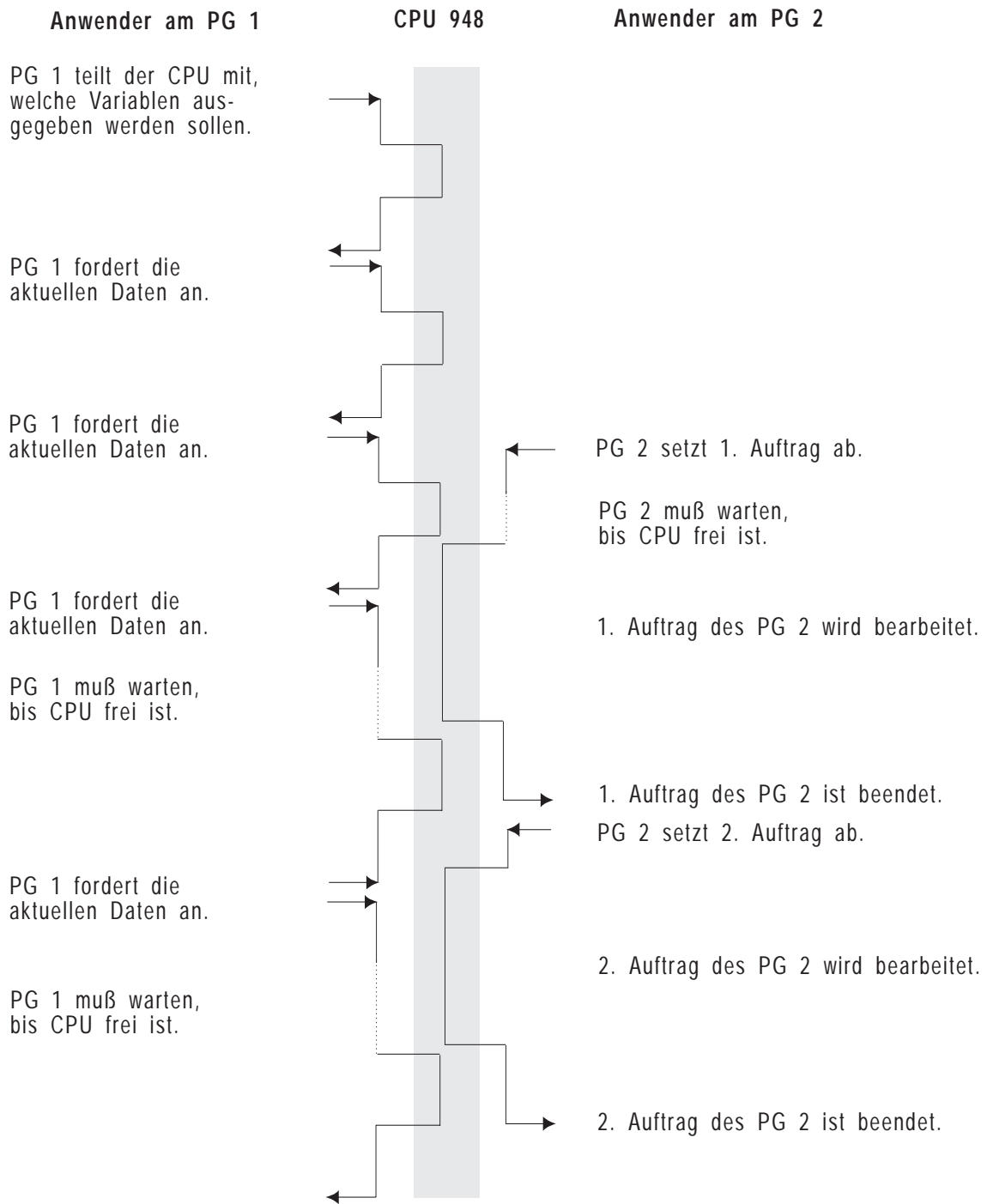


Bild 11-8 Zeitlicher Ablauf zweier paralleler zyklischer Funktionen

Besonderheit bei zyklischen Funktionen auf beiden PG

Blockiert die unterbrechende Funktion die CPU 948 ("Status" auf einen Baustein, der nicht ausgeführt wird), so wird die unterbrochenen Funktion ebenfalls blockiert. Sie wird erst dann wieder fortgesetzt, wenn die unterbrechende Funktion beendet wird.

Bei gleichzeitigem Arbeiten an beiden PGs ergibt sich dann folgender Ablauf:

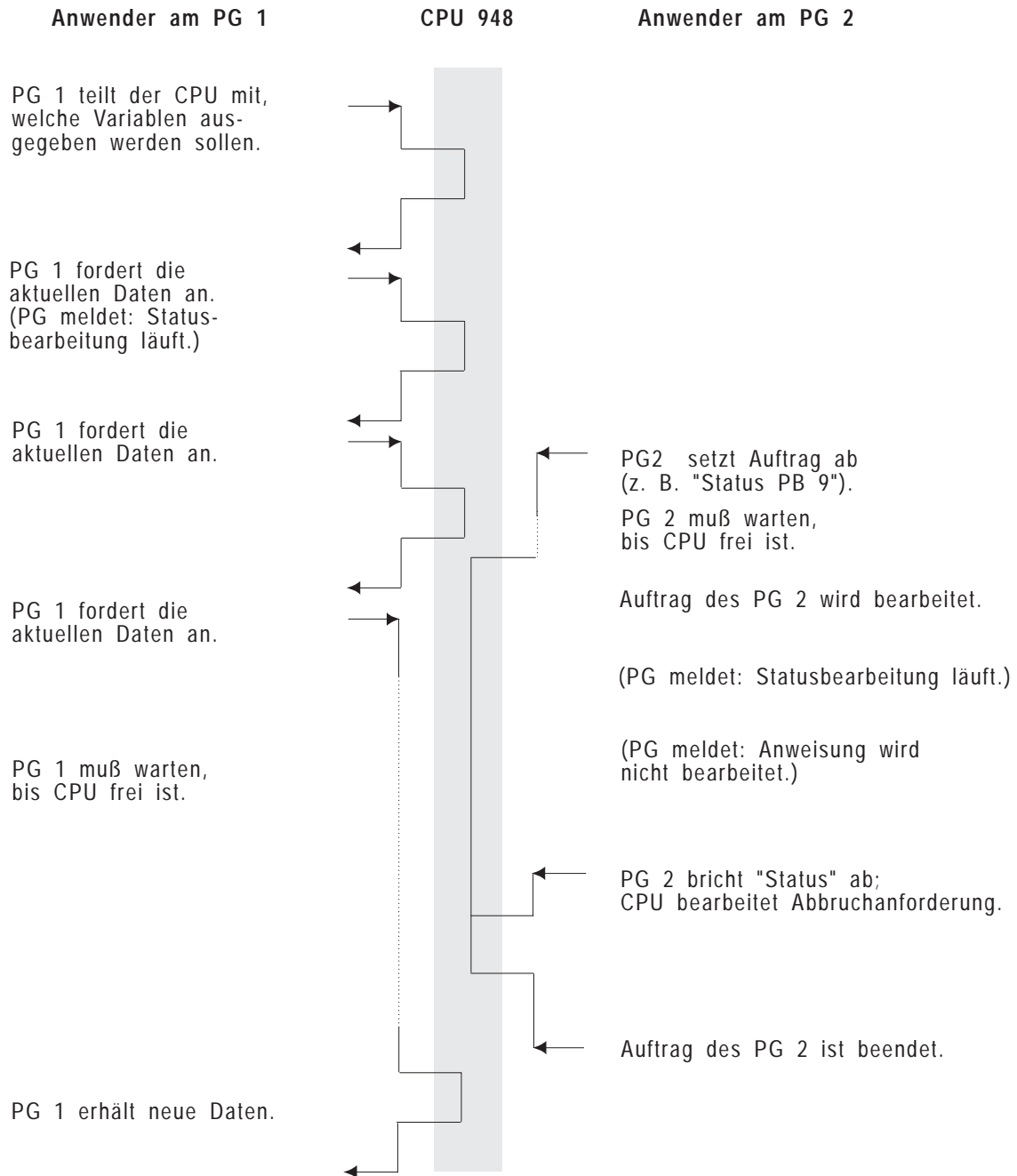


Bild 11-9 Zeitlicher Ablauf, wenn eine Funktion die CPU 948 blockiert

Allgemeine Hinweise

Wird an **einer** Schnittstelle "Status Variablen", "Steuern Variablen" (bei der Statusanzeige) oder "Status" ausgegeben und an der **anderen** Schnittstelle "Speicher komprimieren", "Baustein löschen" oder "Baustein übertragen" durchgeführt, so kann es zu einer falschen Anzeige in der Statusanzeige kommen.

11.5 PG-Funktionen über S5-Bus

11.5.1

Anwendung

Die PG-Funktionen über S5-Bus ermöglichen das Laden und Steuern der über SINEC H1 vernetzten Automatisierungsgeräte S5-155U mit CPU 948 von PG 7xx. Mit den PG-Funktionen über S5-Bus kann die CPU 948 bis zu achtmal schneller geladen werden als über die PG-Schnittstelle. Die Geschwindigkeit hängt dabei von der Länge der zu übertragenden Bausteine ab.

Sie können die PG-Funktionen über S5-Bus auch im Mehrprozessorbetrieb benutzen.

Die PG-Funktionen über S5-Bus sind Bestandteil des Systemprogramms der CPU 948.



Vorsicht

Die PG-Funktionen über S5-Bus dürfen **nur alternativ** zu den PG-Funktionen über die 1. und 2. serielle Schnittstelle verwendet werden. Bei einigen Funktionen kann es bei gleichzeitigem/verschachteltem Ablauf zu Daten-/Bausteinverfälschungen kommen.

Bei der CPU 948 sind die Schnittstellennummern ≥ 232 für die PG-Funktionen über S5-Bus reserviert. Sie dürfen diese Nummern nicht beliebig für Hantierungsbausteine benutzen!

Bild 11-10 zeigt ein Konfigurationsbeispiel für Mehrprozessorbetrieb.

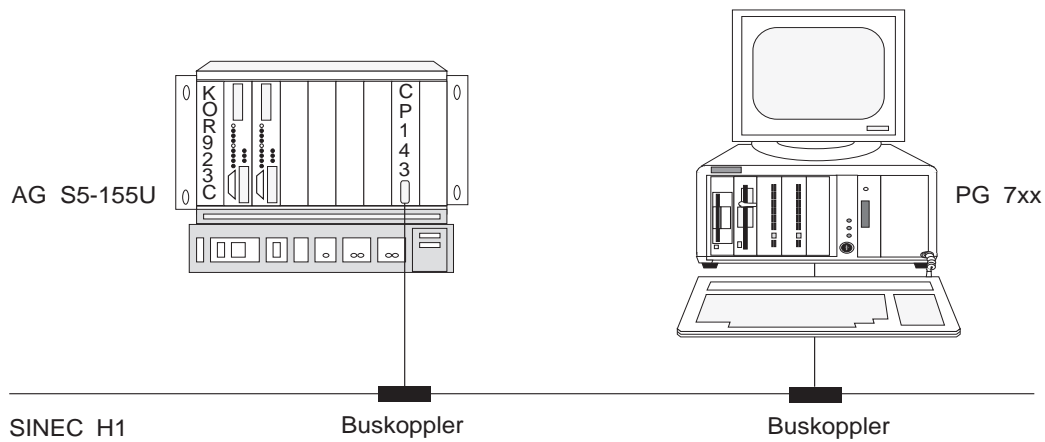


Bild 11-10 **Mehrprozessorbetrieb** mit einem CP 143
(2 x CPU 948, 1 x CP 143)

*keine Parametrierung
der CPU*

Für die PG-Funktionen über S5-Bus sind **in der CPU 948** keine Parametrierungen erforderlich.

*Technische
Voraussetzungen*

Die PG-Funktionen über S5-Bus mit CPU 948 setzen eine Vernetzung des PGs und des AGs über SINEC-H1 voraus.

Dazu benötigen Sie:

- ein PG 7xx mit SINEC-H1-Anschluß und mit installierter STEP-5-Software Version 6.3 (ST) bzw. 6.0 (MT) + Deltadiskette CPU 948.
- im AG (Zentralgerät oder Erweiterungsgerät EG 185) einen Kommunikationsprozessor CP 143 ab Ausgabestand 06 (Firmwareversion 3.0) mit gewählter Basis-Schnittstellennummer 232 (die Basis-Schnittstellennummer stellen Sie in der Hardware über Steckbrücken und im SYSID mittels COM 143 ein).

11.5.2 Arbeitsweise der PG-Funktionen über S5-Bus

Benutzung von Kacheln

Der CP 143 besitzt für die Kommunikation mit den CPUs vier Kacheln (Schnittstellen). Wenn Sie die PG-Funktionen über S5-Bus nicht benutzen, stehen alle Kacheln für die Kommunikation über Hantierbausteine (HTBs) zur Verfügung.

Bei der Benutzung der S5-Bus-Funktionen werden die Kacheln des CP in zwei Kacheln für Anwender-HTB- und zwei Kacheln für PG-Funktionen aufgeteilt.

Die Kacheln für Anwender-HTB können Sie wie bisher für SINEC-H1-Anwendungen verwenden. Beachten Sie dabei jedoch die in Abschnitt 11.5.3 aufgeführten Besonderheiten.

Die Kacheln für PG-Funktionen werden vom CP 143 und der CPU 948 für die PG-Funktionen über S5-Bus genutzt und stehen daher für die Kommunikation über Hantierungsbausteine nicht mehr zur Verfügung.

Schnittstellenummern (SSNR)

Die PG-Funktionen über S5-Bus werden im CP 143 automatisch aktiviert, wenn Sie bei der Parametrierung des CPs als **Basis-Schnittstellenummer 232 oder 236** einstellen (Steckbrücken **und** SYSID!). Sie belegen die Schnittstellenummern 232 bis 239. Die Schnittstellenummern 240 bis 247 sind für spätere Erweiterungen (z. B. CPs mit 8 Kacheln/Schnittstellen) reserviert.

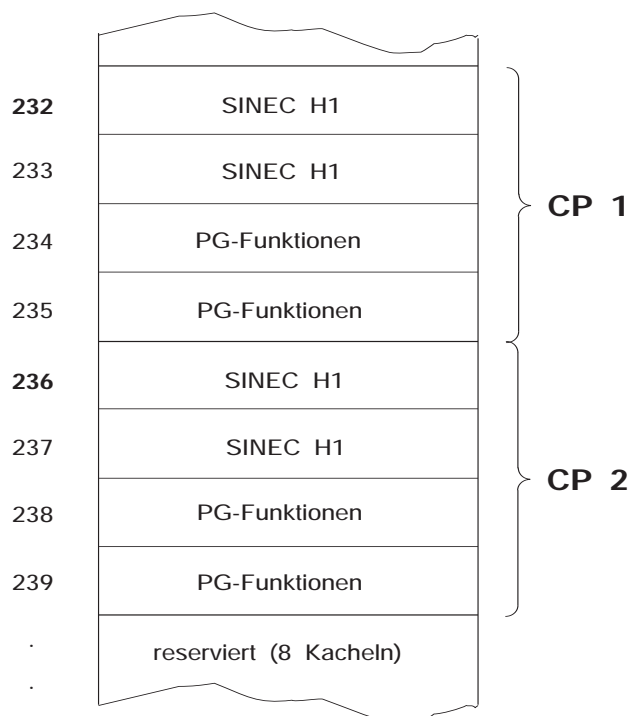


Bild 11-11 Schnittstellenbelegung der PG-Funktionen über S5-Bus

Parametrieren des CP 143

Wie Sie den CP 143 parametrieren, lesen Sie bitte im Handbuch zum CP 143 /6/ nach.



Vorsicht

Die Schnittstellennummern 232ff und 236ff dürfen beim CP 143 im Betrieb mit anderen SIMATIC-CPU's nicht parametriert werden. Im Betrieb der CPU 948 mit anderen CP's dürfen die Schnittstellennummern 232 bis 247 nicht uneingeschränkt benutzt werden.

Mehrprozessorbetrieb

Die PG-Funktionen über S5-Bus können bei der CPU 948 auch im Mehrprozessorbetrieb benutzt werden.

Mit **einem** CP 143 können im AG S5-155U **zwei CPU's** (948) die Online-Funktionen ausnutzen. Der CP 143 kann auch im Erweiterungsgerät (EG 185) eingesetzt werden.

Im Mehrprozessorbetrieb nutzt die **CPU 1 die Kachel mit SSNR 234**, die **CPU 2 die Kachel mit SSNR 235**.

Wird ein zweiter CP 143 gesteckt und entsprechend parametriert, so ist dieser für die Online-Funktionen über S5-Bus mit CPU 3 und CPU 4 (mit SSNR 238 und 239) reserviert.

**11.5.3
Inbetriebnahme**

Bei der Inbetriebnahme müssen Sie folgende Alternativen beachten:

Der CP 143 wird ausschließlich für PG-Funktionen benutzt

Wenn der CP 143 ausschließlich für die PG-Funktionen über S5-Bus eingesetzt werden soll, sind außer der SINEC-H1-Parametrierung keine weiteren Parametrierungen erforderlich.

Nach NETZ EIN sind die PG-Funktionen über S5-Bus auf der CPU 948 immer verfügbar, ohne daß zuvor der CP 143 mit dem HTB SYNCHRON (FB 125) synchronisiert wurde. Der Betriebsartenschalter der CPU muß jedoch auf RUN stehen.

Eine "leere" CPU 948 kann ohne URLÖSCHEN über den S5-Bus in Betrieb genommen werden.

Die CPU 948 synchronisiert nach NETZ EIN automatisch die ihr zugeordneten Kacheln für die PG-Funktionen über S5-Bus auf dem CP 143.

Für die Inbetriebnahme sind folgende Bedienschritte erforderlich:

Schritt	Aktion	
1	Stellen Sie am CP 143 die Schnittstellenummer (SSNR) ein (Steckbrücken): Wählen Sie dazu die SSNR entsprechend der vorliegenden Hardware-Konfiguration, wie unten aufgeführt. Beachten Sie dabei auch die Erläuterungen in /6/.	
	mögliche Hardware-Konfiguration	zugehörige SSNR des/der CP 143
	1 x CPU 948, 1 x CP 143	232
	1 x CPU 948, 2 x CP 143	232 am 1. CP, 236 am 2. CP
	2 x CPU 948, 1 x CP 143	232
	3 x CPU 948, 2 x CP 143	232 am 1. CP, 236 am 2. CP
	4 x CPU 948, 2 x CP 143	232 am 1. CP, 236 am 2. CP
2	Stecken Sie den CP 143 in das AG 155U (das AG muß dabei spannunglos sein!).	
3	Schließen Sie das PG an die PG-Schnittstelle des CP 143 an und laden Sie das Programm COM.	
4	Stellen Sie im SYSID des CP 143 mit COM 143 die in Schritt 1 eingestellte Schnittstellenummer und die Ethernetadresse ein.	

Schritt	Aktion
5	<p>Laden Sie die Parametrierdaten in den CP 143: Die Parametrierdaten des CP 143 können Sie ablegen</p> <ul style="list-style-type: none"> - in einem EPROM-Modul oder - im RAM des CP 143. <p>Sie können die Parametrierdaten über die serielle Schnittstelle des PG 7xx übertragen. Die einzelnen Bedienschritte zum Laden der Parametrierdaten des CP 143 finden Sie in /6/.</p>
6	<p>Führen Sie an der CPU Urlöschen durch, schalten Sie die Netzspannung am AG aus und wieder ein.</p>
7	<p>Editieren Sie in der Busanwahlmaske von STEP 5 den Pfad zur CPU 948.</p>
8	<p>Wählen Sie in der Voreinstellungsmaske von STEP 5 den Pfad über SINEC H1/CP 143 zur CPU 948.</p>

Die PG-Funktionen über S5-Bus sind nach diesen Aktionen einsatzbereit. Sie können nun Ihr Anwenderprogramm laden und ablaufen lassen bzw. testen.

Alternativbetrieb über serielle PG-Schnittstelle

Zu einem Zeitpunkt bearbeitet die CPU 948 nur eine PG-Funktion. Wenn Sie versuchen, bei laufender Bearbeitung einer PG-Funktion von einem zweiten PG über serielle PG-Schnittstelle weitere PG-Funktion zu aktivieren, wird an diesem PG eine Meldung ausgegeben, z. B. "AS-Funktion gesperrt: laufende Funktion".

Hinweise

Beim Abbrechen einer PG-Funktion durch Betätigen des Betriebsartenschalters an der CPU von RUN nach STOP oder durch einen Fehler entstehen bei der Kommunikation über SINEC H1 systembedingte Wartezeiten > 15 s.

Bei Fehlbedienung am PG (z. B. Ausschalten des PGs bei laufender PG-Funktion) kann es vorkommen, daß beim Pfadaufbauen die Funktion "Anwahl" wiederholt werden muß.

Der CP 143 wird für PG-Funktionen und Kommunikation über SINEC H1 benutzt

Wenn Sie den CP 143 neben den PG-Funktionen über S5-Bus auch für Kommunikation über SINEC H1 benutzt werden soll, müssen Sie zusätzlich zu der Parametrierung gemäß Alternative "Der CP 143 wird ausschließlich für PG-Funktionen benutzt" noch weitere Einstellungen und Besonderheiten beachten.

Gehen Sie bei der Inbetriebnahme folgendermaßen vor:

Schritt	Aktion
1 bis 8	Die Schritte 1 bis 8 sind identisch mit der Alternative "Der CP 143 wird ausschließlich für PG-Funktionen benutzt" aufgeführten.
9	<p>Programmieren Sie in den Anlauf-OBs OB 20 und OB 22 jeweils den Aufruf des HTB SYNCHRON (FB 125), damit bei MANUELLEM/AUTOMATISCHEM NEUSTART und AUTOMATISCHEM WIEDERANLAUF der CP 143 für die SINEC-H1-Kommunikation synchronisiert wird.</p> <p>Der HTB SYNCHRON sollte nur dann aufgerufen werden, wenn die Schnittstelle tatsächlich benutzt wird, da dadurch auch die Verbindung zum PG abgebaut wird und anschließend vom PG aus manuell wieder aufgebaut werden muß.</p>

Kacheln für Kommunikation über Hantierungsbausteine benutzen

Das folgende Bild zeigt, wie die PG-Funktionen über S5-Bus die Kacheln des CP 143 belegen. Die freien Kacheln für Anwender-HTB können von der CPU 1 bis 4 für Kommunikationszwecke über SINEC H1 benutzt werden.

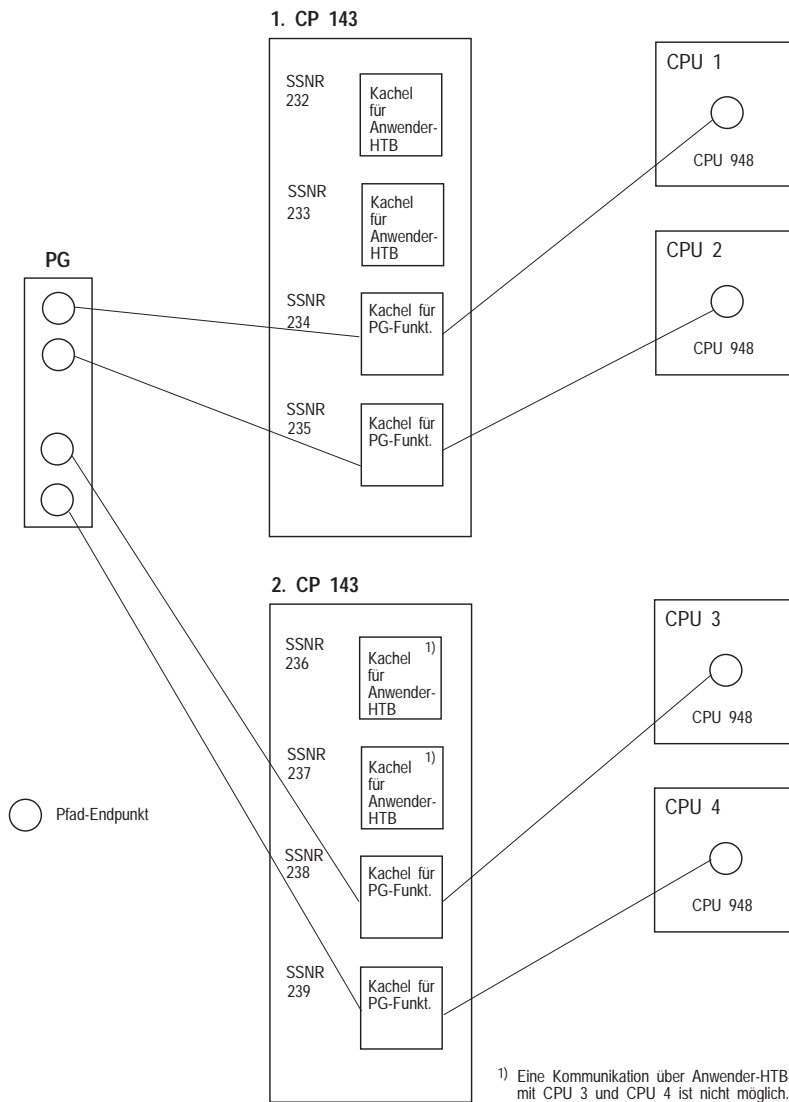


Bild 11-12 Pfade zwischen PG und CPU 948 sowie Belegung der Kacheln des CP 143

Besonderheiten bei Kommunikation über Kachel für Anwender-HTB

Durch die Synchronisation des CP 143 für Kommunikationszwecke (Aufruf des FB 125 mit SSNR 232, 233 bzw. 236, 237) werden vom CP 143 die bestehenden Verbindungen abgebaut.

Die Pfade müssen erneut aufgebaut werden. Dabei entstehen im PG (auch bei Betätigen der Abbruchtaste) Wartezeiten. Die Pfadanwahl muß am PG wiederholt werden.

Wegen dieses Verhaltens können Sie die PG-Funktion "**Status Baustein**" über S5-Bus **nicht auf die Anlauf-OBs** anwenden, wenn Sie Kacheln für Anwender-HTB für die Kommunikation benutzen. Sie sollten deshalb den FB 125 (HTB SYNCHRON) nur im NEUSTART bzw. im Anlauf nach NETZ EIN (NEUSTART oder WIEDERANLAUF) aufrufen.

**11.5.4
Anzeige von Störungen**

Jede der max. vier CPUs (CPU 948), bei der die PG-Funktionen über S5-Bus aktiviert werden, hinterlegt in ihrem BS- und BT-Bereich Anzeigen, wenn bei den PG-Funktionen über S5-Bus ein Fehler auftritt.

Diese Anzeigen bestehen aus einem Parametrierfehlerbyte (PAFE) für jede mögliche Verbindung und einem Anzeigenwort (ANZW) für die Anzeige des aktuellen Ablaufs der Sende- und Empfangsbausteine. Diese Anzeigen entsprechen weitgehend denen der Handtierungsbausteine.

BS 50

Im Systemdatenwort BS 50 (Adresse E F032H) sind die PAFE-Anzeigen abgelegt, die beim Synchronisieren der PG-Funktionen aufgetreten sind.

Auswerten von PAFE in BS 50 Das PAFE-Byte steht immer im im High-Byte des BS 50.

CPU-Nr.	BS 50 High-Byte	BS 50 Low-Byte
1	PAFE SSNR 234	-
2	PAFE SSNR 235	-
3	PAFE SSNR 238	-
4	PAFE SSNR 239	-

*Bedeutung der
PAFE-Anzeigen*

Es werden alle Fehler angezeigt, die im Zusammenspiel mit dem CP 143 auftreten. Es können folgende PAFE-Anzeigen vorkommen:

PAFE-Wert	Bedeutung
00H	keine Fehler
71H	Schnittstelle (Kachel) nicht vorhanden
81H	Schnittstelle unklar
91H	Schnittstelle überlastet
A1H	Schnittstelle von einer anderen CPU belegt
B1H	Auftragsnummer oder Blockgröße unzulässig (FB SYNCHRON)
C1H	Schnittstelle reagiert nicht bzw. nicht rechtzeitig
D1H	sonstige Schnittstellenfehler, auch Fehleranzeige für den CP

Bedeutung der Anzeige 71H:

Die Anzeige 71H bedeutet, daß die Kachel nicht vorhanden ist. Die Inbetriebnahme der PG-Funktionen über S5-Bus kann bei Auftreten dieser Anzeige nicht erfolgen. Überprüfen Sie in diesem Fall die Schnittstellenbelegung des CP 143: Es müssen die Schnittstellennummern 232 ff bzw. 236ff eingestellt sein (in Steckbrücke **und** SYSID!).

BT-Bereich

Nur wenn die Kacheln für PG-Funktionen vorhanden sind und die Verbindung zum CP 143 hergestellt ist, wird im BT-Bereich der CPU 948 ein aus 16 Wörtern bestehender Informationsblock mit folgendem Aufbau angelegt:

Hinweis

Solange keine Verbindung zum CP 143 hergestellt ist (PAFE = 71), z. B. weil keine Kachel mit SSNR 232ff bzw. 236ff vorhanden ist, werden im BT-Bereich **keine Zusatzinformationen** abgelegt.

Adresse

E F2E8H	Datenrichtung von CPU zum PG für SS 234 bzw. 238		BT 232
E F2E9H	Datenrichtung von CPU zum PG für SS 235 bzw. 239		BT 233
E F2EAH	reserviert		BT 234
E F2EBH	reserviert		BT 235
E F2ECH	Datenrichtung von PG zur CPU für SSNR 234 bzw. 238		BT 236
E F2EDH	Datenrichtung von PG zur CPU für SSNR 235 bzw. 239		BT 237
E F2EEH	reserviert		BT 238
E F2EFH	reserviert		BT 239
E F2F0H	reserviert	PAFE 234	BT 240
E F2F1H	reserviert	reserviert	BT 241
E F2F2H	reserviert	PAFE 235	BT 242
E F2F3H	reserviert	reserviert	BT 243
E F2F4H	reserviert	PAFE 238	BT 244
E F2F5H	reserviert	reserviert	BT 245
E F2F6H	reserviert	PAFE 239	BT 246
E F2F7H	reserviert	reserviert	BT 247

Hinweis

Der BT-Bereich wird beim URLÖSCHEN **zurückgesetzt!**

Wenn Sie die PG-Funktionen über S5-Bus einsetzen, wird der BT-Bereich wie oben beschrieben belegt und steht dann nicht mehr für andere Programme (z. B. Standard-FBs) zur Verfügung. Bitte beachten Sie dies bei der Projektierung Ihrer Anlage.

ANZW

ANZW enthält den aktuellen Zustand der Sende- und Empfangsbausteine. Die einzelnen Bits von ANZW haben folgende Bedeutung:

High-Byte	
Bit-Nr.	Belegung
15	nicht belegt
14	
13	
12	
11	
10	
9	
8	
Low-Byte	
7	nicht belegt
6	Datenübernahme fertig
5	Datenübergabe fertig
4	1: Fehler
3	Auftrag fertig mit Fehler
2	Auftrag fertig ohne Fehler
1	0: SEND freigegeben ¹⁾ 1: SEND gesperrt
0	0: RECEIVE gesperrt ¹⁾ 1: RECEIVE freigegeben

¹⁾ spezifisch für PG-Funktionen über S5-Bus

Anhang

12

Inhalt von Kapitel 12

Anhang 1: Brückeneinstellung für Systeminterrupts.	12 - 4
Anhang 2: Ein- und Ausbau des PG-Moduls	12 - 5
Anhang 3: Technische Daten der CPU 948, CPU 946/947 und CPU 928B	12 - 7
Anhang 4: Ergebniskennungen einiger Sonderfunktions-OBs in AKKU 1	12 - 10

12

Anhang

Dieses Kapitel gibt Ihnen einige zusätzliche Informationen zur CPU 948 wie Brückeneinstellungen für die Systeminterrupts, Hinweise zum Ein- und Ausbau des PG-Moduls, Laufzeitenvergleich mit CPU 946/947 und CPU 928B sowie Ergebnisanzeigen von einigen Sonderfunktions-OBs.

Anhang 1: Brückeneinstellung für Systeminterrupts

Für die interruptgesteuerte Programmbearbeitung mit der CPU 948 stehen Ihnen 4 Systeminterrupts zur Verfügung:

- INT A/B/C/D (abhängig vom Steckplatz der CPU, siehe Systemhandbuch /2/),
- INT E,
- INT F
und
- INT G.

Die Interrupts, die Sie nutzen wollen, müssen Sie mit Hilfe der beiliegenden Steckbrücken freigeben. Der Brückenstecker befindet sich auf der Grundplatte oberhalb des Schachtes für die Memory Card. Die genaue Lage entnehmen Sie dem folgenden Bild:

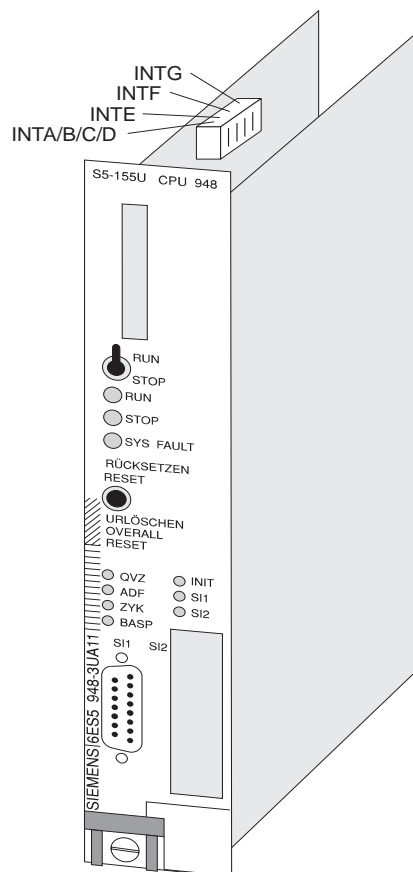


Bild 12-1 Lage des Brückensteckers

Anhang 2: Ein- und Ausbau des PG-Moduls

Wenn Sie ein PG-Modul einsetzen wollen, müssen Sie dieses zuerst (außerhalb des Zentralgerätes) in die CPU einbauen.

Vorsicht

Schalten Sie die Stromversorgung des Automatisierungsgerätes aus, bevor Sie die CPU ziehen.

Einbauen

Hinweis

Die Brücken am PG-Modul sind bei der Auslieferung bereits richtig eingestellt. Sollten sich nach dem Einbau Schwierigkeiten im Betrieb ergeben, so vergleichen Sie die Brückeneinstellung mit der Darstellung im Systemhandbuch /2/.

Bauen Sie Ihr PG-Modul in folgenden Schritten ein:

Schritt	Handlung
1	Schalten Sie die Stromversorgung Ihres AG aus.
2	Ziehen Sie die CPU aus dem Zentralgerät.
3	Lösen Sie die beiden Schrauben, mit denen die Abdeckung des Modulschachts auf der CPU befestigt ist, und entfernen Sie die Abdeckung.
4	Stecken Sie das PG-Modul durch die Frontplatte in die Steckerverbindung (Bauelemente in dieselbe Richtung wie bei der CPU).
5	Befestigen Sie das Modul mit den beiden Schrauben, mit denen die Abdeckung befestigt war.
6	Stecken Sie die CPU in das Zentralgerät.
7	Schalten Sie die Stromversorgung Ihres AG wieder ein.

Ausbauen

Bauen Sie Ihr PG-Modul in folgenden Schritten aus:

Schritt	Handlung
1	Schalten Sie die Stromversorgung Ihres AG aus.
2	Ziehen Sie die CPU aus dem Zentralgerät.
3	Lösen Sie die beiden Verriegelungsschrauben des Moduls und ziehen Sie das Modul aus dem Schacht.
4	Bauen Sie ein anderes Modul ein (wie zuvor beschrieben) oder verschließen Sie den Modulschacht mit der zugehörigen Abdeckung. Verwenden Sie die Befestigungsschrauben des Moduls.
5	Stecken Sie die CPU in das Zentralgerät.
6	Schalten Sie die Stromversorgung Ihres AG wieder ein.

Hinweis

Durch die Verschraubung des Schnittstellenmoduls mit der CPU werden Störpulse über die Schirmmasse der CPU abgeleitet. Sie dürfen die CPU nur mit geschlossenem Modulschacht (Abdeckung oder Modul) betreiben.

Anhang 3: Technische Daten der CPU 948, CPU 946/947 und CPU 928B

Operation / Bearbeitung	CPU 948	CPU 928B
typische Befehlsausführungszeiten für Bitbefehle:		
mit M, E, A D Formaloperand	0,18 μ s 0,7 μ s 0,91 μ s	0,57 μ s 3,4 μ s 2,4 μ s
typische Befehlsausführungszeiten für Wortbefehle:		
- Ladeoperationen L MB (Byte) L MW (Wort) L MD (Doppelwort)	0,18 μ s 0,5 μ s 0,71 μ s	0,81 μ s 0,9 μ s 1,6 μ s
- Festpunktarithmetik - Gleitpunktarithmetik	0,55 ... 3,8 μ s 3,3 ... 6,3 μ s	0,9 ... 10,4 μ s 9,1 ... 15,6 μ s
zyklische Programmbearbeitung (Einzelprozessorbetrieb)		
Grundlast bei Aufruf OB 1/FB 0:	65/- μ s	104/106 μ s
Zuschlag für die Prozeßabbildaktualisierung in Abhängigkeit von der Anzahl der E/A-Bytes (n) mit $0 < n \leq 128$	$n \leq 64$: 64 μ s + n * 2,3 μ s $n > 64$: 92 μ s + n * 2,3 μ s	E: 14 μ s + n * 1,1 μ s A: 5 μ s + n * 4,1 μ s
Zuschlag für die Koppelmerkerübertragung in Abhängigkeit von der Anzahl der Koppelmerker (n) mit $0 < n \leq 256$	$n \leq 64$: 64 μ s + n * 2,1 μ s $n > 64$: 92 μ s + n * 2,1 μ s	E: 14 μ s + n * 1,4 μ s A: 5 μ s + n * 4,3 μ s
Zuschlag für Zeitzellenbearbeitung in Abhängigkeit von der Zeitenblocklänge (ZBL) ZBL =0 ZBL #0 n = Anzahl der laufenden Zeitzellen (Raster: 10 ms)	alle 10 ms 11,6 μ s 11,6 μ s + ZBL * 0,32 μ s	alle 10 ms 10 μ s 16 μ s + ZBL * 0,2 μ s (keine Unterschiede zwischen laufenden und nicht laufenden Zeitzellen)
alarmgesteuerte Programmbearbeitung		
Verlängerung der Zykluszeit durch Einschachtelung eines leeren OB 2 (ohne STEP-5-Operationen) an einer Bausteingrenze	262 μ s	300 μ s

Operation / Bearbeitung	CPU 948	CPU 928B
Reaktionszeit	175 µs	270 µs
zeitgesteuerte Programmbearbeitung		
Verlängerung der Zykluszeit durch Einschachtelung eines leeren OB 13 (ohne STEP-5-Operationen) an einer Befehlsgrenze	287 µs	310 µs für den ersten Weckalarm-OB 170 µs für jeden weiteren, zum gleichen Zeitpunkt fälligen Weckalarm-OB
Zeittakt für den Aufruf des zeitgesteuerten Programms (Weckalarme OB 10 bis OB 18)	variabler Grundtakt von 1 ... 255 ms. Angabe bezogen auf 10 ms: 10, 20, 50, 100 200, 500 ms, 1, 2, 5 s oder 10, 20, 40, 80, 160, 320, 640 ms, 1,28, 2,56 s	10, 20, 50, 100, 200, 500 ms, 1, 2, 5 sec
Auflösungszeiten für uhrzeitgesteuerten Weckalarm (OB 9)	minütlich, stündlich, täglich, wöchentlich, monatlich, jährlich, einmalig	minütlich, stündlich, täglich, wöchentlich, monatlich, jährlich, einmalig
Auflösungszeit für Verzögerungsalarm (OB 6)	1 ms	1 ms
Zykluszeitüberwachung		
Voreinstellung einstellbar zwischen triggerbar	200 ms 1 ... 2550 ms ja	150 ms 1 ... 13000 ms ja
Speichergrößen		
Größe des Anwenderspeichers (in K byte)	640 oder 1664	64
Größe des Speichers für Datenbausteine (DB-RAM, in K byte)	–	ca. 46,6
Zeit- und Zählzellen, Merker		
Anzahl der Zeit- und Zählzellen	je 256	je 256

Operation / Bearbeitung	CPU 948	CPU 928B
Anzahl der Merker	2048 Merker + 32768 S-Merker	2048 Merker + 8192 S-Merker

Anhang 4: Ergebniskennungen einiger Sonderfunktions-OBs in AKKU 1

Bytekennungen in AKKU-1-LL

Die Bytekennungen werden zum Teil von mehreren SF-OBs gemeinsam benutzt. Ihre Bedeutung ist daher abhängig vom aufgerufenen OB.

SF-OB	AKKU-1-LL	Bedeutung
OB 124	01H	Funktion wurde korrekt bearbeitet
	45H 47H 4DH	Fehler: Bausteinart nicht zulässig Baustein nicht vorhanden Online-Funkt. SPEICHER KOMPRIMIEREN tätig
	8DH 8EH	Warnung Konflikt mit einer Online-Funktion (außer "Speicher komprimieren") 10-ms-Wartezeit noch nicht abgelaufen
OB 125	01H	Funktion wurde korrekt bearbeitet
	42H 43H 44H 45H 4DH	Fehler: Baustein bereits vorhanden Speicherplatz reicht nicht aus Bausteinlänge ist nicht zulässig Bausteinart nicht zulässig Online-Funkt. SPEICHER KOMPRIMIEREN tätig
	8DH 8EH	Warnungen: Konflikt mit einer Online-Funktion (außer "Speicher komprimieren") 10-ms-Wartezeit noch nicht abgelaufen
OB 126	01H	Funktion wurde korrekt bearbeitet
	02H 03H	Fehler: Funktions-Nr. ist unzulässig Zeiger in AKKU-1-L (Merker-Nr.) ist unzulässig
	04H	Bausteintyp/-nummer ist unzulässig bzw. Baustein DB/DX ist nicht vorhanden
	05H	im angegebenen Datenwort des Datenbausteins steht nicht das 1. Kennwort (falsche DW-Nr.) oder die Adreßliste enthält ein falsches Kennwort
	06H	Adreßlisten-Nr. ist unzulässig
	07H	Aufruf der Funktion ist in der aktuellen Programmbearbeitungsebene unzulässig

SF-OB	AKKU-1-LL	Bedeutung
OB 223	01H 02H 03H 04H	Anlaufarten sind gleich interner Systemfehler Anlaufarten sind ungleich Einzelprozessorbetrieb: ein Vergleich der Anlaufarten ist nicht möglich
OB 254/ 255	01H	Funktion wurde korrekt bearbeitet
	41H	Fehler: Bausteinkopf auf der Memory Card ungültig
	43H	Speicherplatz reicht nicht aus
	48H	Quelldatenbaustein nicht vorhanden
	4AH	Bausteinummer oder -Typ unzu- lässig/Quell-DB
	4BH	Bausteinummer oder -Typ unzu- lässig/Ziel-DB
	4CH	Zieldatenbaustein ist im Anwender- speicher bereits vorhanden
	4DH	Online-Funktion SPEICHER KOMPRIMIEREN tätig
	4EH	es ist keine Memory Card gesteckt
8DH	Warnungen: Konflikt mit einer Online-Funktion (außer SPEICHER KOMPRIMIEREN)	
8EH	10-ms-Wartezeit noch nicht abgelaufen	

Wortkennungen in AKKU-1-L

Die Wort-Ergebniskennungen werden (bis auf eine Ausnahme) jeweils nur einmal benutzt. Die nachfolgende Tabelle ist daher nach Kennungswerten sortiert.

Kennung in AKKU-1-L	Anzeige von SF-OB:	Bedeutung
8D01H 8D02H	OB 141	unzulässige Funktions-Nr.in AKKU-2-L ¹⁾ eines der reservierten Bits in AKKU 1 ist '1' ¹⁾
8E01H 8E02H 8EFFH	OB 142	unzulässige Funktions-Nr.in AKKU-2-L ¹⁾ eines der reservierten Bits (Nr. 4 bis 15) in AKKU 1 ist '1' ¹⁾ falsche Betriebsart (z. B. wenn der Verzögerungsalarm gesperrt werden soll und der DX 0 den Parameter "Prozeßalarme über EB 0 = ein" enthält)
8F01H 8F02H	OB 143	unzulässige Funktions-Nr.in AKKU-2-L ¹⁾ eines der reservierten Bits in AKKU 1 ist '1' ¹⁾
9601H 960FH 9611H 9612H 9613H 9614H 9615H 9621H 9622H 9623H 9624H 9625H 9626H 9627H 9628H 9629H	OB 150	Datenbaustein nicht geladen Mehrfachaufruf des Bausteins unzulässige Funktions-Nr. Adreßbereichs-Typ unzulässig Datenbaustein-Nr. unzulässig "Nummer des ersten Datenfeldwortes" unzulässig Datenbausteinlänge < 4 Wörter Jahresangabe im Datenfeld unzulässig Monatsangabe im Datenfeld unzulässig Monatstagangabe im Datenfeld unzulässig Wochentagangabe im Datenfeld unzulässig Stundenangabe im Datenfeld unzulässig Minutenangabe im Datenfeld unzulässig Sekundenangabe im Datenfeld unzulässig 1/100 Sekunde im Datenfeld ungleich 0 Stunden-Format ungleich Einstellung bei OB 151

¹⁾ im AKKU-2-L befindet sich der fehlerhafte Wert

Kennung in AKKU-1-L	Anzeige von SF-OB:	Bedeutung
9701H 970FH 9710H 9711H 9712H 9713H 9714H 9715H 9721H 9722H 9723H 9724H 9725H 9726H 9727H 9728H 9729H 972AH	OB 151	Datenbaustein nicht geladen Mehrfachaufruf des Bausteins falsche Betriebsart ("Prozeßalarme über EB 0 = ein") unzulässige Funktions-Nr. Adreßbereichs-Typ unzulässig Datenbaustein-Nr. unzulässig "Nummer des ersten Datenfeldwortes" unzulässig Datenbausteinlänge < 4 Wörter Jahresangabe im Datenfeld unzulässig Monatsangabe im Datenfeld unzulässig Monatstagangabe im Datenfeld unzulässig Wochentagangabe im Datenfeld unzulässig Stundenangabe im Datenfeld unzulässig Minutenangabe im Datenfeld unzulässig Sekundenangabe im Datenfeld unzulässig 1/100 Sekunde im Datenfeld ungleich 0 Stunden-Format ungleich Einstellung bei OB121/OB 150 Auftragsart unzulässig
990FH 9910H 9911H 9921H	OB 153	Mehrfachaufruf des Bausteins falsche Betriebsart ("Prozeßalarme über EB 0 = ein") unzulässige Funktions-Nr. Verzögerungszeit unzulässig
B401H B410H B411H	OB 180	Es ist kein Datenbaustein aufgeschlagen Die Verschiebezahl V ist kein Vielfaches von 16 a) Die Verschiebezahl V ist zu groß, das Bausteinende würde durch die neue Position des Fensters überschritten. b) Die Verschiebezahl V ist negativ
B501H B502H B503H	OB 181	Baustein nicht vorhanden falsche Bausteinnummer falsche Bausteinkennung

Kennung in AKKU-1-L	Anzeige von SF-OB:	Bedeutung
B601H B60FH B611H B612H B613H B621H B622H B623H B624H B625H B626H B627H B628H B629H B62AH B62BH B62CH	OB 182	Datenbaustein nicht geladen Bausteinmehrfachaufruf Beschreibung des Datenfeldes fehlerhaft Adreßbereich-Typ unzulässig Datenbaustein-Nr. unzulässig "Nummer des ersten Datenfeldwortes" unzulässig "Quell-Datenbaustein-Typ" unzulässig "Quell-Datenbaustein-Nr." unzulässig "Nr. des 1. zu übertragenden Datenwortes im Quell-DB" unzulässig Länge Quelldatenbaustein im Bausteinkopf < 5 Wörter "Ziel-Datenbaustein-Typ" unzulässig "Ziel-Datenbaustein-Nr." unzulässig "Nr. des 1. zu beschreibenden Datenwortes im Ziel- DB" unzulässig Länge Zieldatenbaustein im Bausteinkopf < 5 Wörter "Anzahl zu übertragende Datenwörter" unzulässig (= 0 oder > 4091) Quelldatenbaustein zu kurz Zieldatenbaustein zu kurz
F001H F00FH F101H F102H F103H F104H F105H F106H F107H F108H F109H	OB 121	unzulässige Funktions-Nr. Mehrfachaufruf des Bausteins Jahresangabe unzulässig Monatsangabe unzulässig Tagesangabe unzulässig Wochentagsangabe unzulässig Stundenangabe unzulässig Minutenangabe unzulässig Sekundenangabe unzulässig 1/100 ...1/10 Sekunde unzulässig Stundenformat ungleich OB-151-Einstellung
F001H	OB 122	unzulässige Funktions-Nr.

Verzeichnisse

13

Inhalt von Kapitel 13

Abkürzungsverzeichnis.....	A - 1
Stichwortverzeichnis.....	Index - 1

Abkürzungsverzeichnis

Abkürzungen

(Die Erklärung der speziellen Abkürzungen des USTACK finden Sie in Abschnitt 5.4)

ADF	Adressierfehler
AG	Automatisierungsgerät
AKKU-1 (2, 3, 4)-L	Low-Wort im Akkumulator 1 (2, 3, 4), 16 bit
AKKU-1 (2, 3, 4)-H	High-Wort im Akkumulator 1 (2, 3, 4), 16 bit
AKKU-1 (2, 3, 4)-LL	Low-Byte des Low-Wort im Akkumulator 1 (2, 3, 4), 8 bit
AKKU-1 (2, 3, 4)-LH	High-Byte des Low-Wort im Akkumulator 1 (2, 3, 4), 8 bit
ANZ 1, ANZ 0	Wort-Anzeigen, codiert ("Flags")
AWL	Anweisungsliste
ANZW	Anzeigenwort
BASP	Befehlsausgabe sperren (Signal an S5-Bus)
BCD	Binär codierte Dezimalzahl
BR	Basisadreibregister
BSTACK	Bausteinstack
CP	Kommunikationsprozessor
CPU	central processing unit, Zentralprozessor, Zentralbaugruppe (ZBG)
DB	Datenbaustein
DBA	Datenbaustein-Anfangsadresse (im Register 6)
DBL	Datenbaustein-Länge (im Register 8)
DX	Erweiterter Datenbaustein
EG	Erweiterungsgerät
EPROM	erasable programmable read only memory (UV-löschbarer, programmierbarer Nur-Lese-Speicher)
ERAB	Erstabfrage (Bit-Anzeige)
FB	Funktionsbaustein
FUP	Funktionsplan
FX	Erweiterter Funktionsbaustein
IM	Interface Modul (Schnittstellenmodul)
INT	(System-)Interrupt
IP	Intelligente Peripheriebaugruppe

KB	Aufruf eines nicht geladenen Codebausteins
KDB	Aufschlagen eines nicht geladenen DB/DX-Datenbausteins
KOP	Kontaktplan
KOR	Koordinatorbaugruppe
KZU	Klammerstack-Überlauf
LED	light-emitting diode (Anzeige-Element)
NAU	Netzausfall
OB	Organisationsbaustein
OR	Oder (Bit-Anzeige)
OS	Overflow speichernd (Wortanzeige)
OV	Overflow (Wortanzeige)
PA	Prozeßabbild
PAA	Prozeßabbild der Ausgänge
PAE	Prozeßabbild der Eingänge
PAFE	Parametrierfehler-Byte
PARE	Parityfehler
PB	Programmbaustein
PEU	Ausfall der Versorgungsspannung vom Erweiterungsgerät
PG	Programmiergerät
QVZ	Quittungsverzug
RAM	random-access memory (Schreib-Lese-Speicher)
SAZ	STEP-Adreßzähler
SB	Schrittbaustein
SPU	Betriebssystemprozessor
STA	Status (Bitanzeige)
STS	Stoppanweisung
SUF	Substitutionsfehler
STUEB	BSTACK-Überlauf
STUEU	USTACK-Überlauf
TLAF	Transfer- oder Ladefehler
USTACK	Unterbrechungsstack
VKE	Verknüpfungsergebnis
ZG	Zentralgerät
ZYK	Zyklusfehler

Stichwortverzeichnis

A

ADF (Adressierfehler)	5-26
AG-Identifikationsblock	8-42
Akku-Schreibweise	6-5
Akkumulatoren	3-15, 6-26
Aktualoperanden von Funktionsbausteinen	2-25, 2-29, 3-52, 3-55
Alarmer sperren	6-12, 6-29
Alarmer verzögern	6-32, 6-35
alarmgesteuerte Programmbearbeitung	1-6, 4-30, 4-39
ANLAUF	
allgemein	3-11, 4-16
auslösen	4-17, 4-19
Unterbrechungen	4-28
Anlaufarten vergleichen	6-64
Anwenderprogramm	1-7, 1-9
Aufgaben	1-9
Bearbeitung	3-4, 3-11
Siehe Programm	
Speicherung	1-10
Anwenderschnittstellen	
für den Anlauf	4-24
für Interrupts	4-42
für Prozeßalarmer	4-39
für uhrzeitgesteuerten Weckalarmer	4-33
für Verzögerungsalarmer	4-32
für Weckalarmer	4-36
für zyklische Programmbearbeitung	4-29
Anwenderspeicher	1-14, 3-10
Organisation	8-10
ANZ 1 und ANZ 0	
Siehe Ergebnisanzeigen	
Arbeitsweise der CPU	1-5
arithmetische Operationen	3-31
AWL (Anweisungsliste)	2-4

B

BA-/BB-Bereich	8-14
BASP-Signal	4-26 - 4-27
Baustein	
allgemein	2-6
Bausteinadreßliste	3-8, 8-13
Bausteinaufrufe	2-16, 3-8, 3-32
Bausteinkennung	2-36
Bausteinkopf	2-13, 2-24, 8-12
Bausteinnummer	2-12, 2-26, 2-36, 3-33
Bausteinrumpf	2-13, 2-24, 2-36

Bausteintypen	2-12, 2-26
Bausteinorkopf	2-14, 2-35
Formaloperanden (Bausteinparameter)	2-27
Schachtelung von Bausteinen	3-5, 3-8
Bausteine erzeugen	6-17
Bausteine löschen	6-14
BCD-codierte Zahl	2-11
Bearbeitungsoperationen	3-66
Befehlsumfang	2-4
Betriebsarten	7-4
Bibliotheksnummer	2-36
Bildaufbau-Operation	3-33
BR-Register	9-22
BS-/BT-Bereich	8-15
Belegung des BS-Bereiches	8-16
BSTACK (Bausteinstack)	5-5
Ausgabe	5-6
auswerten	5-7

C

CP 143	11-28
parametrieren	11-30
CPU-Typ und -Kennung	8-44

D

Datenbaustein DB 0	2-41, 3-8
Datenbaustein DB 1	2-41
erstellen	10-9
Datenbaustein DB 2	2-41
Datenbaustein DX 0	2-41, 7-4
Aufbau	7-5
Parameter	7-8
Datenbaustein DX 1	2-41
Datenbausteine	
allgemein	1-13
Datenbausteine (DB/DX)	
allgemein	2-13, 2-35
Aufbau	2-35
aufschlagen	2-38
erzeugen	3-33
Gültigkeitsbereich	2-38
kopieren/duplizieren	6-65
programmieren	2-37
testen	6-57
variabler Datenzugriff auf DB/DX	6-53
von Memory Card kopieren	6-65
Datenbereich	6-59
Datenwort	1-13, 2-35 - 2-36, 2-39
Datum	6-38

DBA (Datenbaustein-Anfangsadresse)	9-11	GRAPH 5	2-5
DBL (Datenbaustein-Länge)	9-13	Grundoperationen	2-4, 3-19
Dekrementieren	3-66	Gültigkeitsbereich	
Dezimalzahlen	2-8	eines aufgeschlagenen Datenbausteins	
Dualzahlen	2-8		2-40
E		H	
Echtzeituhr	8-32	hot restart	
Eignung der CPU 948	1-4	erweiterter automatischer Wiederanlauf	4-2
Einschachtelung			
von Programmbearbeitungsebenen	4-6		
Ergänzende Operationen	2-4, 3-49	I	
Ergebnisanzeigen	3-16, 3-20	Inkrementieren	3-66
allgemein	3-16	Interrupts	
ANZ 1 und ANZ 0	3-18, 3-61	Brückeneinstellung	4-41, 12-4
OR	3-17	Interrupts	
OS	3-17	allgemein	4-41
OV	3-17		
STA	3-17, 3-20	K	
VKE	2-7, 3-17, 3-20	Kachelbereich/Kachelspeicher	
Exponent		Belegzelle	9-30
Siehe Gleitpunktzahl		Kachelbereich/Kachelspeicher	9-29
		Kacheln	
		Zugriff auf Kacheln	9-29
F		KB (Aufruf nicht geladener Code-Baust.)	5-24
FEDBX (Fehler bei E DB/EX DX)	5-32	KDB (Aufschlagen nicht geladener	
Fehler bei Selbsttest	5-33	DB/DX)	5-24
Fehleranalyse	5-8	Kommunikations-OBs	10-22
Fehlerbehandlung		Anzeigenbyte	10-25
über Organisationsbausteine	5-20	Laufzeiten	10-31
Fehlerinformationen	5-5	Parameter	10-23
Fehlerursachen	5-23	Kommunikationsprozessoren (CPs)	10-7
Fehlervermeidung	5-4	Kontrollpunkte	11-5
Festpunktzahlen	2-9	KOP (Kontaktplan)	2-4
Formaloperanden		Koppelmerker	
von Funktionsbausteinen	2-25, 3-52	allgemein	3-13, 10-5
Funktionsbausteine (FB/FX)		Brückeneinstellung/Rangierung	10-5
allgemein	2-13, 2-23	Datenaustausch über Koppelmerker	10-5
Aufbau	2-24	Übertragung der Koppelmerker	10-8
aufrufen und parametrieren	2-25, 2-28	Korrigieren von Bausteinen	2-15
programmieren	2-25		
Standard-Funktionsbausteine	2-23, 2-33	L	
FUP (Funktionsplan)	2-4	Ladeoperationen	3-21, 3-55
		LED-Anzeige	4-12 - 4-13, 5-5
G		Lokaler Speicher	
Gleitpunktzahlen	2-9	allgemein	9-4
Globaler Speicher		Zugriff	9-24
allgemein	9-4		
Zugriff	9-25		

		Sonderfunktions-OB	2-22, 6-4
		zum Steuern des Anlaufverhaltens	2-20
		organisatorische Operationen	3-59
M		OS (Overflow speichernd)	
M-Merker	1-12, 10-23	Siehe Ergebnisanzeigen	
Mantisse		OV (Overflow)	
		Siehe Ergebnisanzeigen	
Mehrprozessorbetrieb			
Anlauf	10-13		
Datenaustausch über HTBs	10-8		
Datenaustausch zwischen CPUs und			
CPs	10-7		
Kommunikationsmechanismen	10-4		
Mehrprozessorkommunikation	6-62		
programmieren	10-9		
Testbetrieb	10-14		
Mehrprozessorkommunikation	6-62		
Ablauf	10-15		
Anwendungsbeispiele	10-53		
Betriebsarten	10-35		
Daten empfangen	10-47		
Daten senden	10-40		
initialisieren	10-33		
Parallelverarbeitung	10-22		
Sender-/Empfänger-Identifikation	10-16		
Übertragungseinheit	10-15		
Zuordnungsliste	10-37		
Zwischenspeicherung von Daten	10-17		
Memory Card	3-10		
Merker			
Mehrfachverwendung	4-44		
N			
NAU (Netzspannungsausfall)	4-19, 4-28		
NEUSTART	4-21		
Null-Operation	3-33		
O			
Operanden-Substitution	3-68		
Operandenbereiche	1-11		
Operandenteil	2-7		
Operationscode	2-6		
Operationsteil	2-6		
OR (Oder)			
Siehe Ergebnisanzeigen			
Organisationsbausteine (OB)			
allgemein	2-12, 2-16		
für Anwenderschnittstellen	2-18		
für die Bearbeitung von Fehlern	2-20		
für die Kommunikation im			
WEICHEN STOP	2-20		
		P	
		P-Bereich	
		Siehe Peripherie	
		Parallelbetrieb von seriellen PG-Schnittst.	11-17
		Inbetriebnahme	11-19
		kurzlaufende Funktionen	11-19, 11-21
		langlaufende Funktionen	11-19, 11-22
		zyklische Funktionen	11-19, 11-22
		Parameter	2-6
		Parametrieren DX 0	1-8
		PARE (Parityfehler)	5-28
		Peripherie	
		Adreßraumteilung	8-8
		Baugruppen	1-11
		P-Bereich	1-11
		Q-Bereich	1-11
		PG-Funktionen	
		Aufrufen und Bedienen	11-5
		Übersicht	11-4
		PG-Funktionen über S5-Bus	11-27
		Anzeigen	11-35
		Benutzung von Kacheln	11-29
		Inbetriebnahme	11-31
		Konfigurations-Beispiel	11-28
		Mehrprozessorbetrieb	11-30
		Schnittstellennummern	11-29
		Technische Voraussetzungen	11-28
		PG-Maske	
		für DB-1-Generierung	10-10
		für DX-0-Parametrierung	7-14
		PG-Modul	11-17
		ausbauen	12-6
		einbauen	12-5
		PG-Software	1-18
		Priorität	1-6, 4-5, 4-7, 4-31, 4-40, 4-42, 7-10
		Programm	
		Programm-Organisation	3-5
		Speicherung	3-10
		Programmbausteine (PB)	2-12, 2-16
		Programmbearbeitungsebenen	4-4
		allgemein	4-6, 6-32
		Programmiersprache	
		C mit S5-C-Compiler	1-18
		GRAPH 5	1-18

SCL	1-18	serielle Kopplung PG - AG	11-16
STEP 5	1-18, 2-4	Softwareschutz	8-35
Programmierwerkzeuge	1-18	Sonderfunktionen	
Prozeßabbild		allgemein	6-4
aktualisieren	4-27	Aufruf	6-5
allgemein	1-11, 3-13	Fehler bei der Bearbeitung von	
der Ausgänge (PAA)	1-5, 1-11	Sonderfunkt.	6-6
definieren/übertragen	6-20	Schnittstellen zu Sonderfunktionen	6-5
der Eingänge (PAE)	1-5, 1-11	SPEICHER KOMPRIMIEREN	2-15
Prozeßalarme		Speicherblöcke transferieren	9-19
freigeben	3-72, 4-43	Speicheroperationen	3-20, 3-52
sperrern	3-72	Speicherorganisation	9-4
Prozeßalarme über Eingangsbyte EB 0		Speicherzugriffe	
allgemein	4-39	allgemein	9-4
Anwenderschnittstellen	4-39	über das BR-Register	9-22
PROZESSALARM/INTERRUPT	4-27	über Adresse in AKKU 1	9-8
		Sprungoperationen	3-59
		STA (Status)	
		Siehe Ergebnisanzeigen	
		Standard-Funktionsbausteine	
		Siehe Funktionsbausteine	
		STEP-5-Operationen	3-15
		Steuerbits	5-5, 5-9 - 5-10
		STOP	
		Betriebszustand	4-9
		Stopp-Operation	3-33
		Struktur des Speicherbereichs	8-4 - 8-6
		Strukturierte Programmierung	2-5
		SUF (Substitutionsfehler)	5-28
		System-RAM	8-6
		Systemdaten	8-15
		Systemdatenwörter	
		Bitbelegung	8-18
		Systeminterrupt	
		siehe Interrupt	12-4
		Systeminterrupts	4-30, 4-41, 7-4, 7-9
		Siehe auch Interrupts	
		Systemoperationen	2-4, 3-59, 9-4
		Systemprogramm	1-7
		Systemzeit	6-8, 6-38
		T	
		Test der Adreßleitungen	5-36
		Test der Hardwareuhr	5-35
		Test der Zykluszeitüberwachung	5-36
		Test des Anwenderspeichers	5-35
		Test des BASP-Signals	5-35
		Test des Bausteinodes	5-36
		Test des Systemprogrammcodes	5-36
		TLAF (Lade- und Transferfehler)	5-29
		Transferoperationen	3-21
Q			
Q-Bereich			
Siehe Peripherie			
QVZ (Quittungsverzug)	5-25		
R			
Reaktionen			
auf Alarme	3-12		
auf Fehler	5-23		
bei nicht geladenen (Fehler-)OBs	5-21		
auf Fehler	2-20, 3-12		
Reaktionszeit	4-44		
Rechenoperationen	3-57		
RUN			
allgemein	4-27		
S			
S-6 (Kommunikationsfehler)	5-32		
S-Merker	1-12		
Schachtelungstiefe	3-9		
Schiebeoperationen	3-61		
Schmiermerker	10-53		
Schnittstelle			
zum Systemprogramm	1-8, 1-10, 2-18		
Schrittbausteine (SB)	2-12, 2-16		
Selbsttest	5-34		
Ein-/Ausschalten einzelner Tests	5-37		
Fehlerbehandlung	5-38		
Fehlerinformation	5-39		
Steuerbits	8-40		
Semaphore	3-75		

		Einstellen der Anzahl	5-37
U		Zeitwert	3-27
		Zklusüberwachungszeit	5-27
Uhrzeit	6-38	Zugriffe auf DW	9-15
uhrzeitgesteuerter Weckalarm	4-27, 4-31, 4-33, 6-43	Zuordnungsliste	2-7, 2-24
Umwandlungsoperationen	3-63	ZYK (Zykluszeitfehler)	5-27
Unterbrechungsereignisse	3-14	zykische Weckalarme	4-27
Unterebene	4-7	Zyklische Kommunikation	4-9
URLÖSCHEN	4-14	zyklische Programmbearbeitung	1-5, 1-16, 3-4, 3-11, 4-28
USTACK	4-5	zyklische Weckalarme	4-31, 4-35
USTACK (Unterbrechungsstack)		ZYKLUS	3-11, 4-28
Anzeigen	5-15	Anwenderschnittstelle OB 1	4-29
Ausgabe	5-6, 5-9	Programmbearbeitungsebene	4-27
Fehlerinformation	5-5	Unterbrechungsstellen	4-29
Informationen im USTACK	5-15	Zyklusüberwachungszeit	3-13, 6-63, 7-8 - 7-11
Inhalt	5-14	Zykluszeit	3-12
		aktuelle	8-20
		verbrauchte	8-41
		Zykluszeitüberwachung	3-13
V			
Vergleichsoperationen	3-32		
Verknüpfungen	3-50		
binäre	3-19		
digitale	3-50		
Verzögerungsalarm	4-27, 4-31 - 4-32, 6-50		
VKE (Verknüpfungsergebnis)			
Siehe Ergebnisanzeigen			
Voreinstellung			
des Systemprogramms	1-8		
modifizieren	7-4		
Voreinstellung modifizieren	1-8		
Vorgehen beim Programmieren	1-15		
W			
Weckalarme	4-27, 7-8		
Weckfehler	4-37		
WEFES/WEFEH (Weckfehler)	5-30		
WIEDERANLAUF	4-21		
Z			
Zahlendarstellung	2-8		
Zähler Z	1-13		
Zählwert	3-28		
Zeit- und Zähloperationen	3-26, 3-53		
Zeitauftrag	4-33, 6-43		
Zeiten T	1-13		
zeitgesteuerte Programmbearbeitung	1-6, 4-31		
Zeitscheibe	5-34		
Berechnen der Zeitscheibe	5-37		

An
Siemens AG
A&D AS E 81
Östliche Rheinbrückenstr. 50
76181 Karlsruhe

Absender:

Ihr Name: _____
Ihre Funktion: _____
Ihre Firma: _____
Straße: _____
Ort: _____
Telefon: _____

Bitte kreuzen Sie Ihren zutreffenden Industriezweig an:

- | | |
|----------------------------------------------|----------------------------------------------------|
| <input type="checkbox"/> Automobilindustrie | <input type="checkbox"/> Pharmazeutische Industrie |
| <input type="checkbox"/> Chemische Industrie | <input type="checkbox"/> Kunststoffverarbeitung |
| <input type="checkbox"/> Elektroindustrie | <input type="checkbox"/> Papierindustrie |
| <input type="checkbox"/> Nahrungsmittel | <input type="checkbox"/> Textilindustrie |
| <input type="checkbox"/> Leittechnik | <input type="checkbox"/> Transportwesen |
| <input type="checkbox"/> Maschinenbau | <input type="checkbox"/> Andere _____ |
| <input type="checkbox"/> Petrochemie | |



Vorschläge und Anmerkungen zur Anwenderdokumentation

Ihre Anmerkungen und Vorschläge helfen uns, die Qualität und Benutzbarkeit unserer Dokumentation zu verbessern. Bitte füllen Sie diesen Fragebogen bei der nächsten Gelegenheit aus und senden Sie ihn an Siemens zurück.

Geben Sie bitte bei den folgenden Fragen Ihre persönliche Bewertung mit Werten von 1 = gut bis 5 = schlecht an.

1. Entspricht der Inhalt Ihren Anforderungen?
2. Sind die benötigten Informationen leicht zu finden?
3. Sind die Texte leicht verständlich?
4. Entspricht der Grad der technischen Einzelheiten Ihren Anforderungen?
5. Wie bewerten Sie die Qualität der Abbildungen und Tabellen?

Falls Sie auf konkrete Probleme gestoßen sind, erläutern Sie diese bitte in den folgenden Zeilen:
