

SIEMENS

Automatisierungsgerät SIMATIC S5 - 101U

Programmieranleitung

Bestell-Nr.: GWA 4NEB 810 2120-01 a

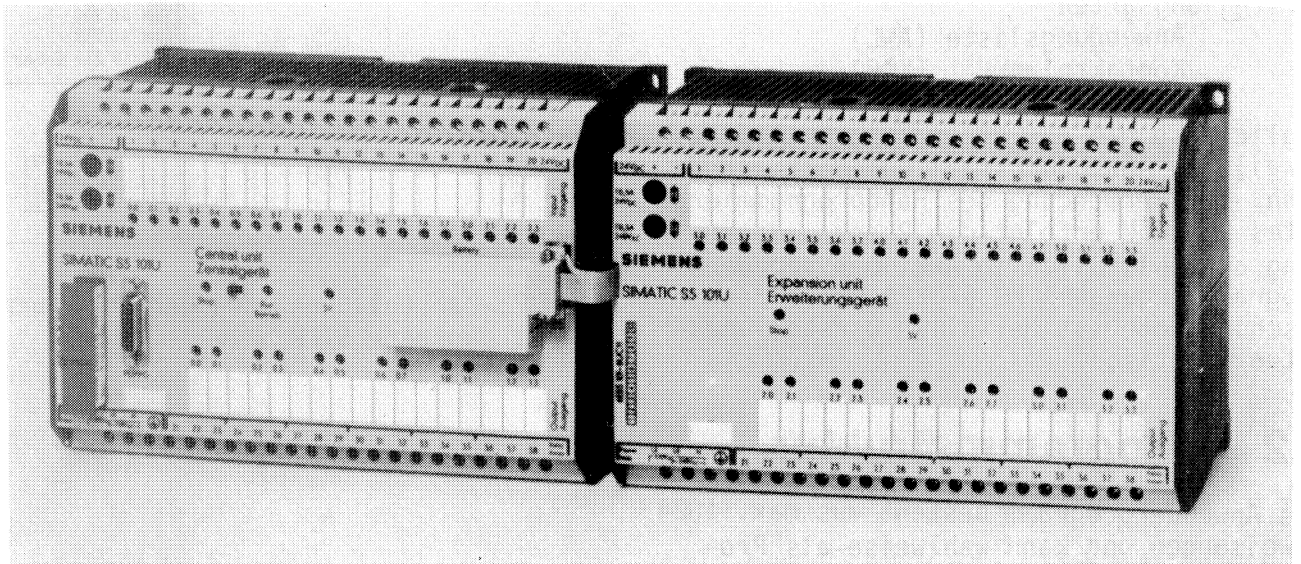


Bild 1 Automatisierungsgerät S5-101U

INHALT	Seite	INHALT	Seite
1. DIE PROGRAMMIERSPRACHE		4. INBETRIEBNAHME DES PROGRAMMES	
1.1 Programmiersprache STEP 5	1.1	4.1 Programmeingabe/-sicherung	4.1
1.2 Programmstruktur	1.1	4.2 Programmtest	4.2
2. ARBEITSWEISE DES AGS		4.2.1 Suchlauf	4.2
2.1 Programmbearbeitung	2.1	4.2.2 Signalzustandsanzeige	4.2
2.2 Betriebsarten "Betrieb", "Stop"	2.2	4.2.3 Steuern von Ausgängen/Merkern	4.2
2.3 Speicherverhalten	2.2	4.2.4 Steuern von Zeiten/Zählern	4.3
3. HINWEISE ZUR PROGRAMMERSTELLUNG		5. PROGRAMMIERBEISPIELE	
3.1 Anlaufverhalten	3.1	5.1 Grundfunktionen	
3.2 Batterieüberwachung	3.2	5.1.1 Verknüpfungsfunktionen (binär)	5.1
3.3 Remanente/nichtremanente Merker	3.2	5.1.2 Speicherfunktionen	5.4
3.4 Alarmbearbeitung	3.2	5.1.3 Lade- und Transferfunktionen	5.6
3.5 Übersetzbarkeit von KOP,FUP, AWL	3.4	5.1.4 Zeitfunktionen	5.8
3.6 Betrieb am SINEC L1-Bus	3.7	5.1.5 Zählfunktionen	5.12
		5.1.6 Vergleichsfunktionen	5.14
		5.1.7 Arithmetische Funktionen	5.17
		5.1.8 Sonstige Funktionen	5.17
		5.2 Ergänzende Funktionen	5.18
		5.2.1 Verknüpfungsfunktionen (wortw.)	5.18
		5.2.2 Umwandlungsfunktionen	5.19
		5.2.3 Schiebefunktionen	5.19
		5.2.4 Sprungfunktionen	5.20
		5.2.5 Anzeigenbildung	5.22
		6. BEFEHLSUMFANG	6.1

1. Die Programmiersprache

1.1 Programmiersprache STEP 5

Die Anwenderprogramme werden mit der Programmiersprache STEP 5 erstellt. Die Operationen der Programmiersprache STEP 5 ermöglichen die Programmierung von einfachen binären Funktionen bis hin zu komplexen digitalen Funktionen.

Bei der Programmierung sind, abhängig vom verwendeten Programmiergerät, die drei Darstellungsarten

Anwendungsliste (AWL)
Kontaktplan (KOP)
Funktionsplan (FUP)

möglich, so daß die Programmiermethode dem jeweiligen Anwendungsfall angepaßt werden kann. Bei Verwendung des Handprogrammiergerätes PG 605U erfolgt die Programmdarstellung ausschließlich in AWL. Der von den Programmiergeräten PG 670/PG 675 erzeugte Maschinencode ist bei allen Darstellungsarten identisch.

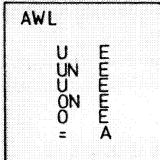
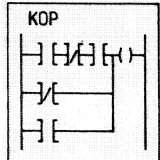
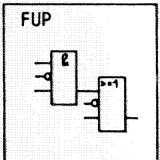
Anweisungsliste	Kontaktplan	Funktionsplan
Programmieren mit mnemotechnischen Abkürzungen der Funktionsbezeichnungen	Programmieren mit grafischen Symbolen wie Stromlaufbahn	Programmieren mit grafischen Symbolen
entspricht DIN 19 239 (Entwurf)	entspricht DIN 19 239 (Entwurf)	entspricht IEC 117-15 DIN 40 700 DIN 40 719 DIN 19 239 (Entwurf)
		

Bild 2: Darstellungsarten der Programmiersprache STEP 5

1.2 Programmstruktur

Das Anwenderprogramm besteht aus max. 1024 Anweisungen und kann wahlweise als Programmbaustein (PB) oder als Funktionsbaustein (FB) erstellt werden. **IM AG 101U sind nur PB 1 oder FB 1 ablauffähig.**

Programmbaustein

Ein Programmbaustein kann in jeder der drei Darstellungsarten AWL, KOP und FUP programmiert und dokumentiert werden. Unter Berücksichtigung bestimmter Programmierregeln (s. Kapitel 3.4) läßt sich ein Programmbaustein mit Hilfe der Programmiergeräte PG 670/PG 675 von einer Darstellungsart in die beiden anderen übersetzen.

Für Schütz-Anwender sei auf die Darstellungsart KOP hingewiesen, da der Kontaktplan dem Stromlaufplan sehr ähnlich ist.

Die Verwendung eines Programmbausteins kommt insbesondere dann in Betracht, wenn ein Bildschirmprogrammiergerät zur Verfügung steht und in einer graphischen Darstellungsart programmiert oder dokumentiert werden soll.

Hinweis: Der Befehlsumfang aus "Ergänzende Funktionen" darf im PB 1 nicht verwendet werden.

Funktionsbaustein

Ein Funktionsbaustein läßt sich ausschließlich in AWL erstellen und dokumentieren.

Durch die Verwendung von Sprungbefehlen läßt sich das Anwenderprogramm gut strukturieren und damit leistungsfähig gestalten.

Bei Verwendung von Lade- und Transferbefehlen in Verbindung mit Sprungbefehlen lassen sich kurze gleichbleibende Reaktionszeiten bei der Alarmbearbeitung realisieren (s. Kapitel 3.4).

2. Arbeitsweise des AGs

2.1 Programmbearbeitung

Das AG 101U ist ein speicherprogrammierbares Steuerungsgerät, d.h. seine Steuerungsaufgabe wird durch ein Anwenderprogramm festgelegt.

Damit das Anwenderprogramm Anweisung für Anweisung zyklisch abgearbeitet werden kann, werden vom Steuerwerk folgende Funktionen ausgeführt:

1. Bei Neustart (Netz "Aus - Ein" oder Betriebsartenschalter "Stop - Betrieb") wird das Prozeßabbild der Ausgänge* gelöscht, d.h. alle Ausgänge werden auf Null gesetzt.
2. Das Prozeßabbild der Eingänge* wird erneuert, d.h. alle Signalzustände der Eingänge werden abgefragt und in das Prozeßabbild der Eingänge geschrieben.
3. Das Anwenderprogramm (PB1 oder FB1) wird Anweisung für Anweisung abgearbeitet. Bei Abfrage des Signalzustandes der Eingänge wird auf deren Prozeßabbild zugegriffen und nicht auf die Eingänge selbst. Bei Setzen und Rücksetzen der Ausgänge durch das Anwenderprogramm wird zunächst nur das Prozeßabbild der Ausgänge überschrieben.
4. Nach Abarbeitung des Anwenderprogramms wird das Prozeßabbild der Ausgänge zu den Ausgängen übertragen.
5. Die Punkte 2, 3 und 4 werden zyklisch durchlaufen.

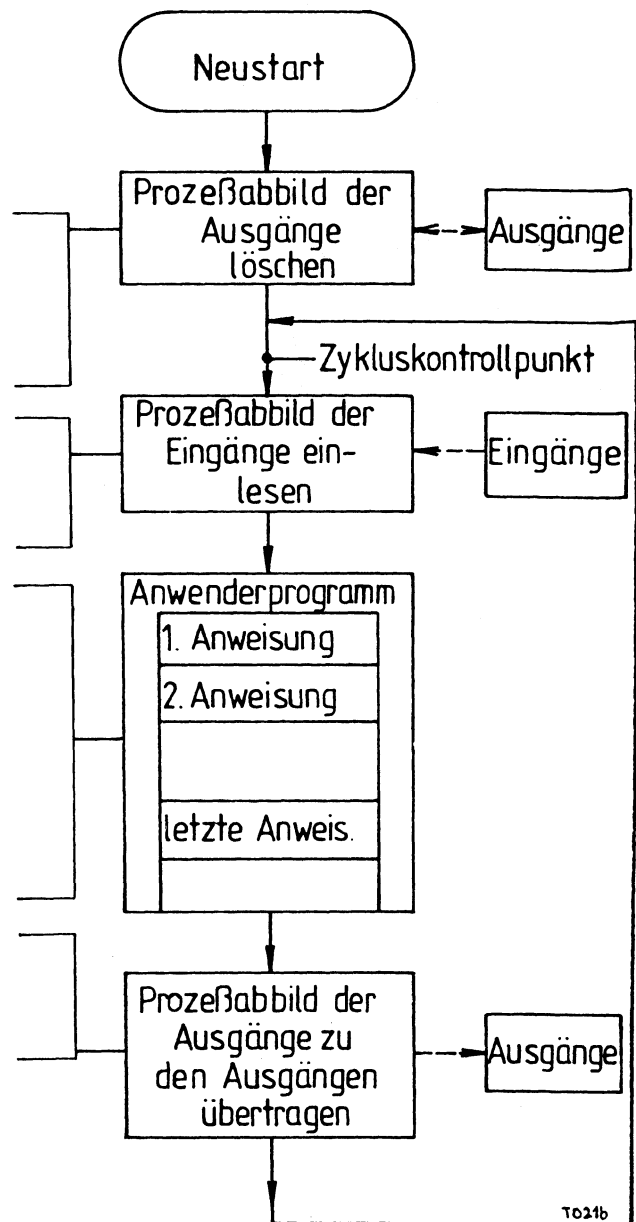


Bild 3: Arbeitsweise des AGs 101U

Ein Durchlauf von Zykluskontrollpunkt zu Zykluskontrollpunkt dauert bei 1024 Anweisungen (binär) ca. 70 ms.

Wird infolge von Programmfehlern bzw. Störungen ein Zyklus nicht innerhalb von 300ms abgeschlossen, so spricht eine interne Überwachungsschaltung an, das AG geht in den "Stop"-Zustand und schaltet alle Ausgänge stromlos.

*Prozeßabbild der Ein-/Ausgänge: Interner Speicherbereich, in dem der Signalzustand ("0" bzw. "1") der Ein- und Ausgänge hinterlegt wird.

**) Remanent, also nicht rückgesetzt*

2.2 Betriebsarten "Betrieb", "Stop"

Betriebsart "Betrieb"

In der Betriebsart "Betrieb" wird das Programm von Zykluskontrollpunkt zu Zykluskontrollpunkt zyklisch abgearbeitet. Die Betriebsart "Betrieb" wird erreicht durch

- Schalten des Betriebsartenschalters in Stellung "Betrieb".
- Anwählen der PG-Funktion "Start AG" (Betriebsartenschalter in Stellung "Betrieb").
- Netzspannungswiederkehr, wenn der Betriebsartenschalter in Stellung "Betrieb" steht und vor Netzspannungsausfall in Stellung "Betrieb" stand.

Betriebsart "Stop"

In der Betriebsart "Stop" wird das Programm nicht bearbeitet, die Ausgänge sind gesperrt. Für die Dauer des "Stop"-Zustandes behalten alle Zeit- und Zählerelemente, Merker sowie das Prozeßabbild der Ein- und Ausgänge die Werte bzw. Zustände bei, die sie im letzten Zyklus vor dem "Stop"-Zustand hatten. Bei Übergang in den "Betrieb"-Zustand werden Zeitelemente und die Zählerelemente (0..7)*rückgesetzt.

Die nicht remanenten Merker und das Prozeßabbild der Ein-/Ausgänge werden gelöscht. Die Betriebsart "Stop" wird erreicht durch:

- Schalten des Betriebsartenschalters in Stellung "Stop".
- Anwählen der PG-Funktion "Stop AG".
- Störungen oder Fehler bei der Programmbe-
arbeitung z. B. Zykluszeitüberschreitung,
Sprungfehler, vom AG nicht interpretier-
bare Befehle.

Die Ursache, die zum "Stop"-Zustand geführt hat, kann mit Hilfe der PG-Funktion "U-Stack ausgeben" abgerufen werden (s. Kapitel 4.2).

2.3 Speicherverhalten

Das AG hat einen internen Programmspeicher, der bei Verwendung einer Pufferbatterie für mindestens 3 Jahre remanent ist. Zusätzlich stehen 2 verschiedene Speichermodule zur Verfügung, deren unterschiedliche Eigenschaften in Bild 4 dargestellt sind.

Die Speichermodule dienen zur zusätzlichen Programmsicherung bzw. als Kopiermodul, falls für mehrere AGs nur ein Speichermodul verwendet werden soll.

Bei "Netz-Ein" oder beim Umschalten des AGs auf "Betrieb" wird der Inhalt des Speichermoduls grundsätzlich in den internen Speicher kopiert und von dort aus abgearbeitet.

Speichermodul	EPROM (Lese-Speicher)	EEPROM (Schreib-Lese-Speicher)
Programm speichern	Am PG 615 (mit Adapter 984-2UA11) PG 670 (mit Adapter 984-0UA11) PG 675	Am PG 615 PG 670 (ohne Adapter 984) PG 675
Programm löschen	Nur mit einer speziellen UV-Lampe (Löschdauer: 30 min)	Direkt im AG mit obigen PGs und PG 605U (PG-Funktion: PG → AG)
Programmänderungen mittels PG	Nur Löschen des ganzen Programms möglich	möglich

Bild 4: Unterschiede zwischen EPROM- und EEPROM-Speichermodul identisch mit Bild 17 der Betriebsanleitung

3. Hinweise zur Programmerstellung

3.1 Anlaufverhalten

Ohne zusätzliche Maßnahmen im Anwenderprogramm nimmt das AG bei Einschalten der Netzspannung bzw. bei Netzspannungswiederkehr folgende Betriebszustände an:

NETZ-AUS ↓ NETZ-EIN	Aktuelle Stellung des Betriebsartenschalters des AGs bei "Netz-Ein"	Betriebsart des AGs vor "Netz-Aus"	Fehler-Kennung* gesetzt	Betriebsart nach "Netz-Ein"
	STOP BETRIEB	STOP BETRIEB	JA NEIN	STOP STOP STOP BETRIEB

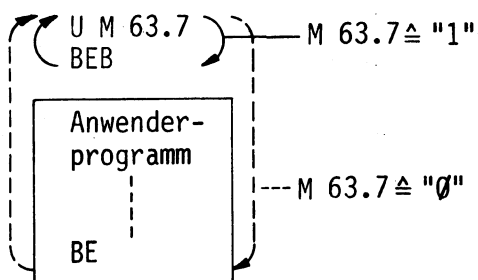
Bild 5: Automatische Betriebsarteneinstellung nach "Netz-EIN"

*Die Programmbearbeitung wurde gestört, die Ursache ist im Unterbrechungs-Stack hinterlegt.

Generelle Verhinderung eines automatischen Neustarts

Um einen automatischen Neustart nach "Netz-Ein" generell zu unterbinden, steht der MERKER M 63.7 zur Verfügung. Der Merker M 63.7 wird vom Betriebssystem des AGs bei Netzspannungswiederkehr gesetzt, wenn die Betriebsart "Betrieb" eingestellt ist und vor "Netz-Aus" eingestellt war. Um einen manuellen Anlauf zu ermöglichen, wird der Merker M 63.7 in der Betriebsart "Stop" rückgesetzt. Ein Rücksetzen durch das Anwenderprogramm ist ebenfalls möglich (z. B. in Verbindung mit einem Eingangssignal).

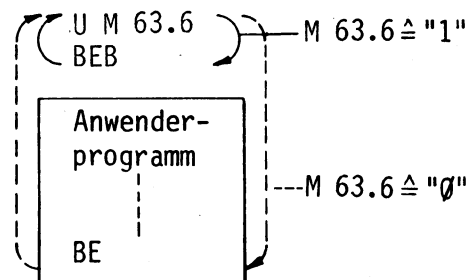
Programmierbeispiel



Verhinderung eines automatischen Neustarts bei Batterieausfall

Um einen automatischen Neustart bei Batterieausfall (remanente Merker gelöscht) zu unterbinden, steht der Merker M 63.6 zur Verfügung. Der Merker M 63.6 wird vom Betriebssystem des AGs bei Netzspannungswiederkehr gesetzt, wenn ein Ausfall bzw. das Fehlen der Pufferbatterie festgestellt wird. Voraussetzung ist, daß die Betriebsart "Betrieb" eingestellt ist. Das Rücksetzen des Merkers M 63.6 erfolgt durch "AG urlöschen" oder durch das Anwenderprogramm (z. B. in Verbindung mit einem Eingangssignal).

Programmierbeispiel



3.2 Batterieüberwachung

Zur Überwachung der Pufferbatterie ist der Merker M 63.6 vorgesehen.

Dieser wird vom Betriebssystem des AGs bei Netzspannungswiederkehr sowie im laufenden Zyklus gesetzt, wenn ein Ausfall bzw. das Fehlen der Pufferbatterie festgestellt wird.

Vorraussetzung ist, daß die Betriebsart "Betrieb" eingestellt ist.

Das Rücksetzen des Merkers M 63.6 erfolgt durch "AG urlöschen" oder durch das Anwenderprogramm. Der Anwender hat somit die Möglichkeit das Verhalten des AGs bei Batterieausfall selbst festzulegen.

3.3 Remanente/nichtremanente Merker

Das AG 101U hat insgesamt 512 Merker. Der Merkerbereich teilt sich wie folgt auf.

Remanente Merker (M 0.0...M 31.7)

- behalten bei "Netz-Ein" den letzten Zustand vor "Netz-Aus" bei (nur mit Pufferbatterie).
- behalten bei Wechsel der Betriebsart "Stop—Betrieb" ihren letzten Zustand bei (mit und ohne Pufferbatterie).
- werden bei "Netz-Ein" wie die nichtremanenten Merker rückgesetzt (ohne Pufferbatterie).
- können außer durch das Anwenderprogramm durch "AG urlöschen" rückgesetzt werden.

Durch die Verwendung der remanenten Merker kann der letzte Anlagen- oder Maschinenzustand vor Verlassen des "Betrieb"-Zustandes gespeichert werden. Bei Neustart kann die Anlage oder die Maschine an der Stelle weiterarbeiten, wo sie zum Stillstand gekommen ist.

Nichtremanente Merker (M 32.0...M 63.7)

- werden bei Wechsel der Betriebsart "Stop—Betrieb" sowie bei "Netz-Ein" generell rückgesetzt.

Die Merker M 61.0-M 62.7 sind als Koordinierungsmerker für den Betrieb am SINEC L1-Bus, die Merker M 63.0...M 63.7 als Systemmerker reserviert. Da sie durch das Betriebssystem des AGs beeinflußt werden, sind sie nicht als Merker frei verfügbar.

3.4 Alarmbearbeitung

Eine Alarmbearbeitung liegt dann vor, wenn ein vom Prozeß kommendes Signal (z. B. Not-Aus) das AG veranlaßt, die zyklische Bearbeitung des Anwenderprogramms zu unterbrechen und ein spezifisches Alarmprogramm zu bearbeiten.

Die Bearbeitung von Alarmen ist beim AG 101U ausschließlich durch das Anwenderprogramm festgelegt, so daß jeder Ein- und Ausgang zur Alarmbearbeitung benutzt werden kann.

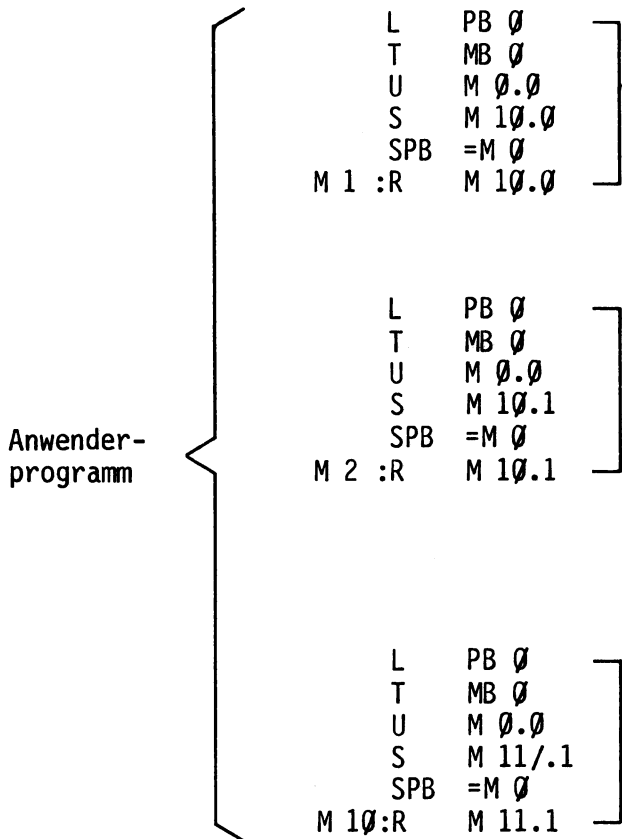
Um möglichst kurze Reaktionszeiten zu er-

reichen, werden Ein- und Ausgänge direkt, d. h. außerhalb der zyklischen Bearbeitung angesprochen. Dazu stehen die Lade-/Transferbefehle "LPB" (Eingänge) und "TPB" (Ausgänge) zur Verfügung.

Eine nahezu konstante Reaktionszeit wird erreicht, wenn das Abfragen von Eingängen, die durch den Anwender als Alarmeingänge programmiert wurden, gleichmäßig über das gesamte Anwenderprogramm verteilt wird. Wie ein Anwenderprogramm mit Alarmbearbeitung aussehen kann, zeigt Bild 6.

Aufgabenstellung: Nimmt der Eingang E 0.0 den logischen Zustand "1" an, so sollen die Ausgänge A 0.0...A 0.7 den Zustand der Merker M 3.0...M 3.7 annehmen. Um eine möglichst kurze und konstante Reaktionszeit zu realisieren, sollen 10 Alarmabfragen im Anwenderprogramm vorgesehen werden.

STEP 5-Programm in AWL

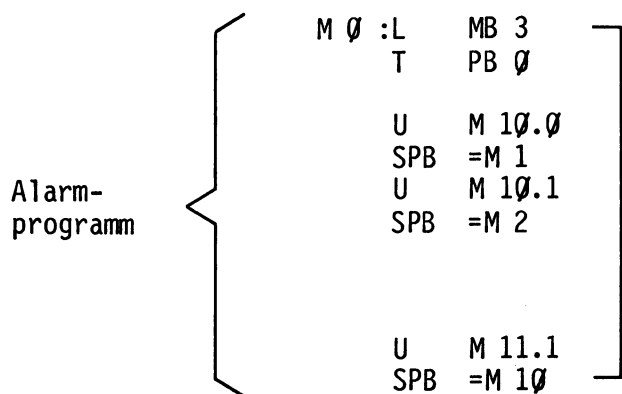


Erläuterungen

1. Alarmabfrage: Durch Laden von PB 0 wird E 0.0 direkt, d. h. unter Umgehung des Prozeßabbildes abgefragt und auf M 0.0 abgebildet. Ist M 0.0 (und damit E 0.0) gleich "1", so wird in das Alarmprogramm gesprungen. Mit M 10.0 wird die Rücksprungstelle festgelegt.

2. Alarmabfrage

10. Alarmabfrage



Das Merkerbyte MB 3 wird direkt ins Peripheriebyte PB 0, d. h. direkt zu den Ausgängen transferiert. Das Prozeßabbild der Ausgänge wird mit erneuert. Durch Abfrage der Merker M 10.0...M 11.1 (es ist nur ein Merker eingesetzt) wird das Anwenderprogramm an der zuletzt festgelegten Rücksprungstelle fortgesetzt.

Bild 6: Beispiel für ein Anwenderprogramm mit Alarmbearbeitung

3.5 Übersetzbarkeit von KOP, FUP, AWL

Allgemeines

Jede Darstellungsart der Programmiersprache STEP 5 beinhaltet eigene Eigenschaften und bestimmte Grenzen.

Daraus ergibt sich, daß ein in AWL geschriebener Programmbaustein nicht ohne weiteres in KOP oder FUP ausgegeben werden kann, und daß darüber hinaus die graphischen Darstellungsarten KOP und FUP gegebenenfalls nicht vollständig kompatibel sein können.

In anderen Worten: Die Rückübersetzbarkeit ist nicht immer gegeben.

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

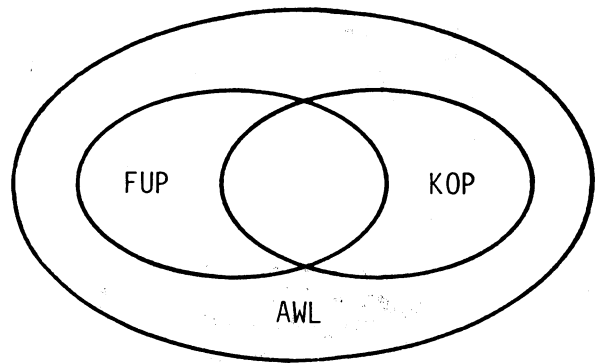


Bild 7: Umfang bzw. Begrenzungen der Darstellungsarten der Programmiersprache STEP 5

Ziel dieses Abschnittes ist die Aufstellung einiger Regeln, deren Einhaltung eine vollständige Kompatibilität der drei Darstellungsarten gewährleistet. Diese Regeln teilen sich folgendermaßen auf:

- Kompatibilitätsregeln zwischen den graphischen Darstellungsarten.
Die Einhaltung dieser Regeln ermöglicht die Eingabe einer graphischen Darstellungsart und die entsprechende Ausgabe in den übrigen Darstellungsarten.

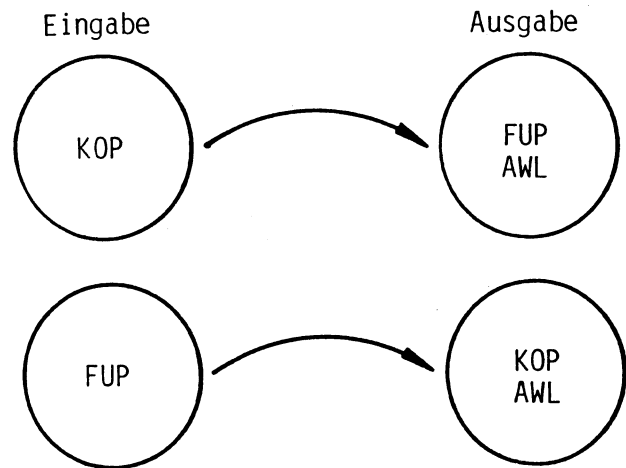


Bild 8: Graphische Eingabe

- Kompatibilitätsregeln zwischen Anweisungsliste und graphischen Darstellungsarten.
Die Einhaltung dieser Regeln gewährleistet die Eingabe in einer frei wählbaren Darstellungsart, sei sie graphisch oder nicht, und die entsprechende Ausgabe in den anderen zwei Darstellungsarten.

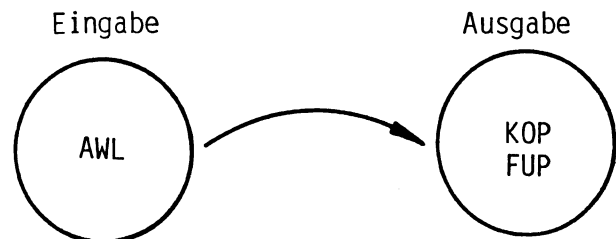


Bild 9: Eingabe in Anweisungsliste

Eingabe in KOP, Ausgabe in FUP (AWL)

Regel: Bildgrenzen für FUP nicht überschreiten.

Eine zu große Schachtelung kann im FUP zur Überschreitung der Bildgrenzen (8 Ebenen) führen.

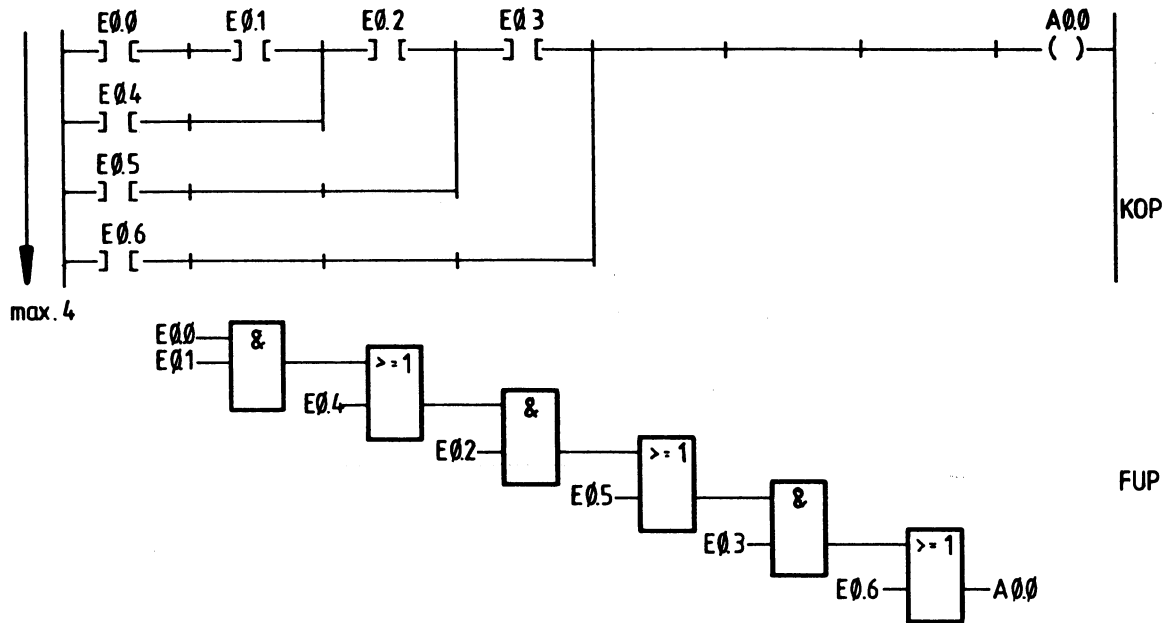


Bild 10: Beispiel einer maximalen KOP-Schachtelung zur Ausgabe in FUP

Eingabe in FUP, Ausgabe in KOP (AWL)

Regel 1: Bildgrenzen für KOP nicht überschreiten.

Eine zu große Anzahl von Eingängen an einem FUP-Kasten führt zur Überschreitung der KOP-Bildgrenze.

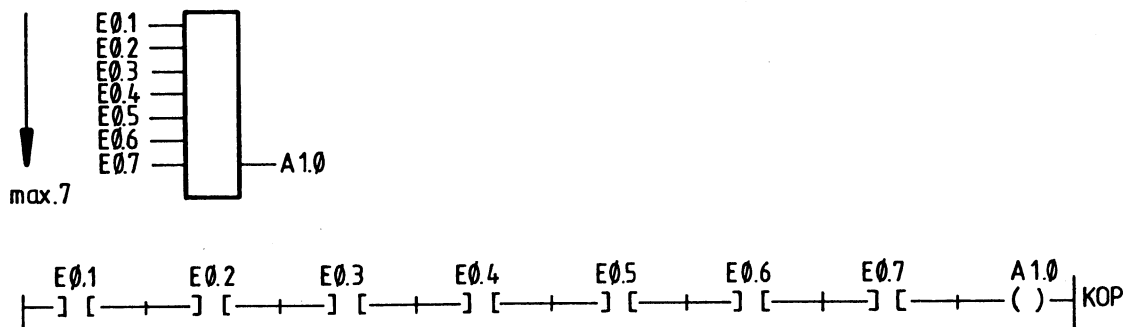


Bild 11: Beispiel eines maximalen UND-Kastenausbaues in FUP zur Ausgabe in KOP

Regel 2: Der Ausgang eines komplexen Gliedes (Speicher, Vergleicher, Zeiten und Zähler) darf nicht mit ODER weiterverknüpft werden.

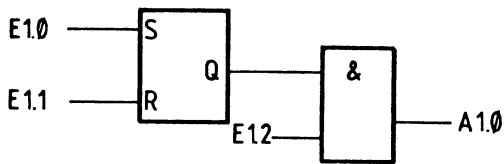


Bild 12: Beim FUP sind nach einem komplexen Glied nur UND-Kasten zulässig.

Eingabe in AWL, Ausgabe in KOP, FUP

Regel 1: Ein komplexes Glied darf keine Vorverknüpfung haben.
 Regel 2: Die Ein- und Ausgänge der komplexen Glieder sind in der selben Reihenfolge zu programmieren, wie sie am Bildschirm in graphischer Darstellungsart parametrisiert werden.
 Eine Ausnahme bilden Zeit- und Zählwert, da der entsprechende Wert vorher durch einen Ladebefehl im Akku hinterlegt werden muß.

Außerdem muß jeder unbeschaltete Ein- und Ausgang mit NOP Ø versorgt werden. Bei Zeiten und Zählern müssen deren Setzeingang und der Eingang zum Laden des Zeitwertes (TW) bzw. Zählwertes (ZW) gemeinsam abgeschaltet sein.

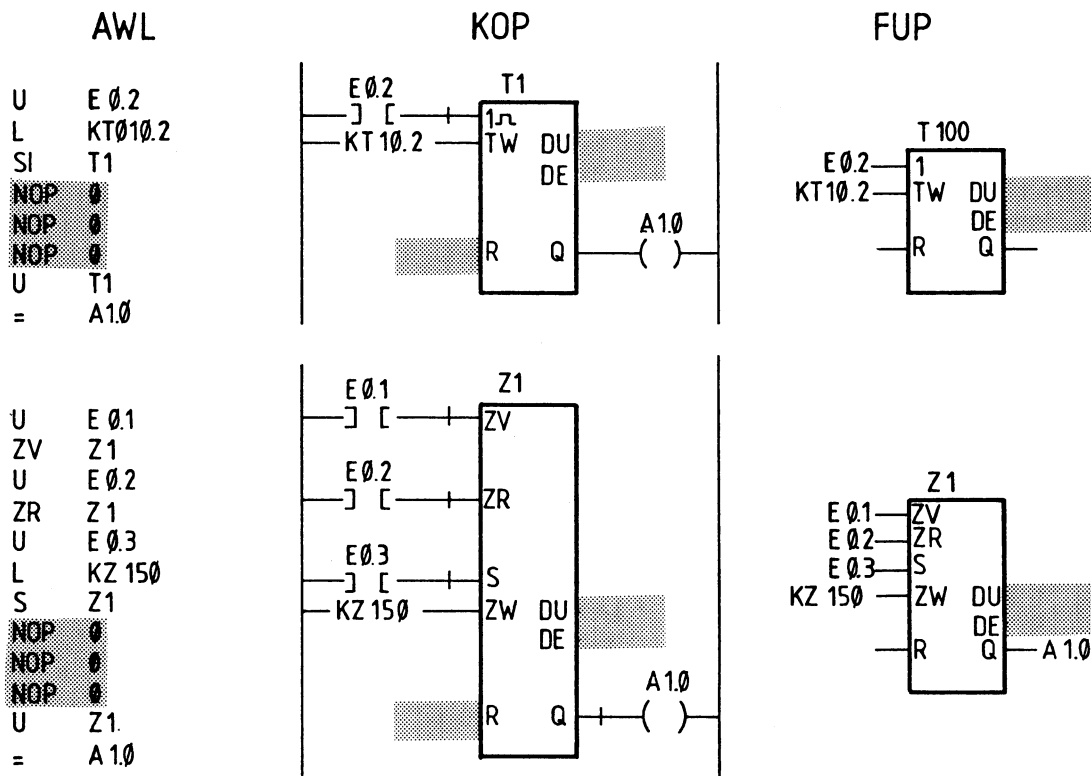


Bild 13: Beispiel zur Versorgung unbeschalteter Ein-/Ausgänge bei Zeiten und Zählern.

3.6 Betrieb am SINEC L1-Bus

Der SINEC L1-Bus dient zur Kopplung von AGs im unteren Kommunikationsleistungsbe- reich und arbeitet nach dem "Master-Slave- Verfahren".

Master ist der Kommunikationsprozessor CP 530, Slaves können die CPUs aller klei- nen AGs sein. Jeder Slave erhält eine Slave-Nummer, unter der er angesprochen wird. Daten können ausgetauscht werden zwischen dem Master und max. 30 Slaves, sowie zwischen Slaves untereinander.

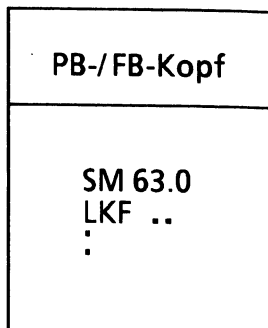
Beim AG 101U werden die Slave-Nummer, die Koordinierungsmerker und das Empfangs- und das Sendefach wie folgt festgelegt:

Neben den eigentlichen Daten werden Steue- rungs- und Sicherungsinformationen mit übertragen, die dem STEP 5-Anwenderpro- gramm über ein Koordinierungs-Merkerwort zugänglich gemacht werden.

Die eigentlichen Daten werden in einem Empfangsfach und einem Sendefach hinter- legt, auf das der Anwender mit Lade-/Trans- ferbefehlen zurückgreifen kann.

Slave-Nummer

Die Slave-Nummer wird zusammen mit einer Kennung am Anfang des Anwenderprogramms (PB 1/FB 1) hinterlegt.

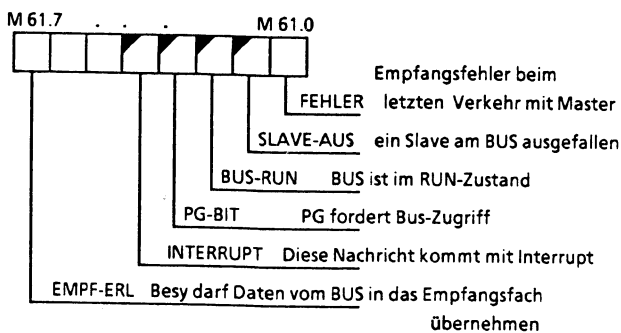


Kennung
Slave-Nr. 1...30

Bei PG-BUS ist Slave Nr. identisch zur PG-BUS Nr. ! keine weitere Parametrierung nötig

Koordinierungs-Merkerbyte EMPFANG (KME)

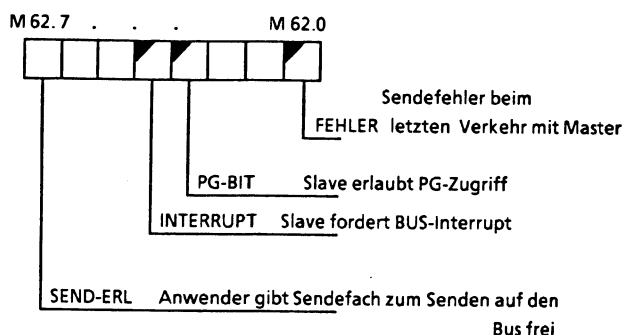
Als KME wird das Merkerbyte MB 61 benutzt.



Bit vom Bus-Master

Koordinierungs-Merkerbyte SENDEN (KMS)

Als KMS wird das Merkerbyte MB 62 benutzt.



Bit für Bus-Master

Die Koordinierungsmerker werden vom Betriebssystem des AG beeinflusst und sind daher nicht als Merker frei verfügbar.

Empfangsfach

Das Empfangsfach liegt im Datenbaustein DB 1 (DW 40...DW 72) und ist wie folgt aufgebaut:

DL 40	LÄNGE der Nettodaten (0...64)	DR 40	QUELL-SLAVE-NR. (0...30) ¹⁾	DW 40	} Netto- daten
DL 41	1. DATUM	DR 41	2. DATUM	DW 41	
DL 42	3. DATUM	DR 42	4. DATUM	DW 42	
.	
.	
.	
.	
DL 71	61. DATUM	DR 71	62. DATUM	DW 71	
DL 72	63. DATUM	DR 72	64. DATUM	DW 72	

1) Slave-Nr. 0 $\hat{=}$ Master

Sendefach

Das Sendefach liegt im Datenbaustein DB 1 (DW 88...DW 112) und ist wie folgt aufgebaut:

DL 80	LÄNGE der Nettodaten (0...64)	DR 80	ZIEL-SLAVE-NR. (0...30) ¹⁾	DW 80	} Netto- daten
DL 81	1. DATUM	DR 81	2. DATUM	DW 81	
DL 82	3. DATUM	DR 82	4. DATUM	DW 82	
.	
.	
.	
.	
DL 111	61. DATUM	DR 111	62. DATUM	DW 111	
DL 112	63. DATUM	DR 112	64. DATUM	DW 112	

1) Slave-Nr. 0 $\hat{=}$ Master

Umfassende Informationen über den SINEC L1-Bus sind der Betriebsanleitung (4NEB 811 0545) und der Programmieranleitung (4NEB 811 0546) für Bussystem SINEC L1 zu entnehmen.

4. Inbetriebnahme des Programmes

Zur Programmeingabe und zum Programmtest stehen die Handprogrammiergeräte PG 605U/ PG 615 und die Bildschirmprogrammiergeräte PG 670 / PG 675 zur Verfügung.

An den Programmiergeräten PG 670 und PG 675 sind in Verbindung mit dem AG 101U folgende Voreinstellungen notwendig:
 PG 670: 150 AK 130 W
 PG 675: 150 S NEIN

4.1 Programmeingabe / -sicherung

Vor der Programmeingabe ist "AG urlöschen" durchzuführen. Damit werden

- der interne Programmspeicher des AGs
- das binäre Prozeßabbild der Ein-/Ausgänge
- sämtliche Merker
- Fehlerkennungen und Unterbrechungsursachen gelöscht.

Nach der Programmeingabe wird das Programm vom PG-Speicher zum AG-Speicher übertragen. Ein Übertragen ist nur im "Stop"-Zustand des AGs möglich.

Nach dem Übertragen ist das Programm im PG-Speicher verloren und muß zur Programmkorrektur in den PG-Speicher zurückgeholt werden* (Ausgabe FB 1/PB 1). Ein Sichern des Programmes in einem EPROM-Modul ist

am PG 615 (mit Adapter),
 am PG 670 (mit Adapter 984)

und am PG 675 möglich.

Zum Sichern eines Programmes in einem EEPROM-Modul ist nach Bild 14 zu verfahren.

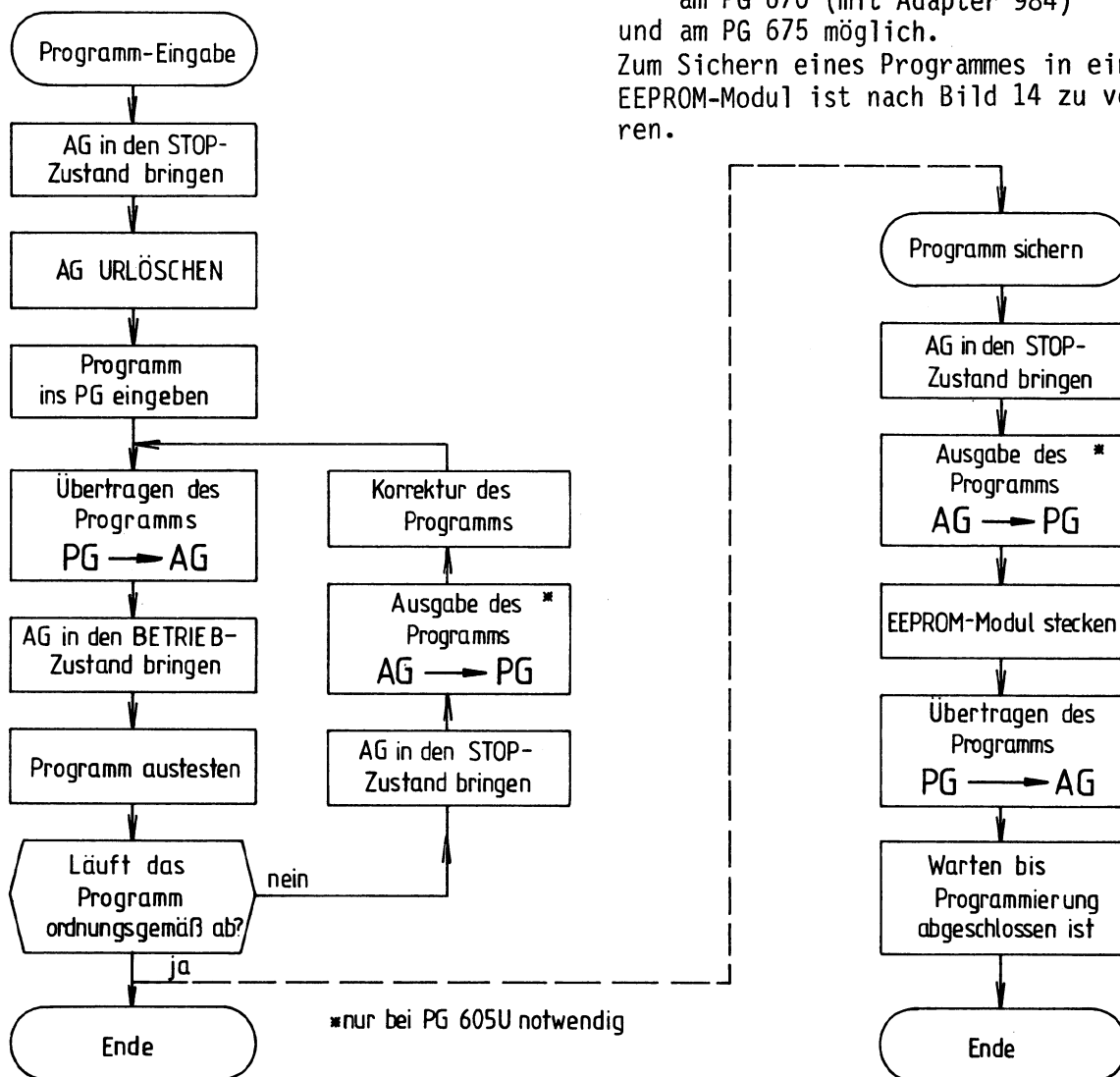


Bild 14: Schematische Darstellung einer Programmeingabe mit anschließender Sicherung des Programmes in einem EEPROM-Speichermodul

4.2 Programmtest

Störungen, die das AG in den "Stop"-Zustand führen, können mit Hilfe des "U-Stack" identifiziert werden (siehe Betriebsanleitung Kap. 4.2).

Zum Auffinden von logischen Fehlern in der Programmbearbeitung stehen folgende Testfunktionen zur Verfügung.

4.2.1 Suchlauf

Zum schnelleren Auffinden von Stellen im Anwenderprogramm steht die PG-Funktion "Suchlauf" zur Verfügung.

Damit können z. B. in der Testphase alle Stellen in einem Programm zur Anzeige gebracht werden, die einen bestimmten Operanden enthalten, z. B. einen Ausgang der nicht erwartungsgemäß arbeitet.

Suchbegriffe können sein:

- Anweisungen z. B. UE 1.0
- Operanden z. B. E 1.0
- Marken (nur bei FB 1)
- Adressen

Suchläufe sind im Zusammenhang mit folgenden Funktionen wichtig:

- Eingabe/Korrektur
- Ausgabe
- Programmabhängige Signalzustandsanzeige

Nähere Angaben sind den Bedienungsanleitungen der Programmiergeräte zu entnehmen.

4.2.2 Signalzustandsanzeige

Zum Ausgeben der Signalzustände binärer und digitaler Operanden stehen folgende PG-Funktionen zur Verfügung:

Direkte Signalzustandsanzeige

Mit Hilfe dieser Testfunktion kann der Zustand eines beliebigen Operanden am Zykluskontrollpunkt (Kap. 2.1) beobachtet werden.

Programmabhängige Signalzustandsanzeige

Mit Hilfe dieser Testfunktion können der Signalzustand eines Operanden und das Verknüpfungsergebnis (VKE) zum Zeitpunkt der Bearbeitung der angewählten Anweisung beobachtet werden.

4.2.3 Steuern von Ausgängen / Merkern

Mit der Funktion "Steuern" können binäre und digitale Operanden im "Betrieb"-Zustand des AGs gezielt beeinflusst werden.

Die gewünschten Zustände der Operanden werden am PG byteweise eingegeben und zum AG übertragen.

Es können gesteuert werden

- AB 0...AB 3
- MB 0...MB 63

Auf diese Weise können bei der Inbetriebnahme Ausgänge ohne Anwenderprogramm gezielt gesteuert und es kann die richtige Verdrahtung von Stellgeräten, Leuchtmeldern etc. kontrolliert werden.

Beim "Steuern" während eines laufenden Anwenderprogramms werden die eingegebenen Zustände der Operanden einmalig ins AG übernommen und die Programmbearbeitung wird mit diesen Zuständen fortgesetzt. Dies bedeutet, daß die eingegebenen Zustände der Operanden ggf. vom laufenden Programm verändert werden können.

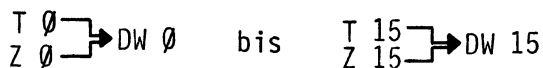
4.2.4 Steuern von Zeiten/Zählern

Das Steuern von Zeiten und Zählern ist nur in Verbindung mit dem PG 605U in beiden Betriebsarten des AGs möglich.

PG 615
OP 393

Programmierung (ab Version 1.1)

Zur Aufnahme der Preset-Werte für Zeiten und Zähler sind die Datenworte DW 0...DW 15 im Datenbaustein DB 1 reserviert. Im Anwenderprogramm wird bei Starten von Zeiten und Zählern auf das zugehörige Datenwort Bezug genommen.

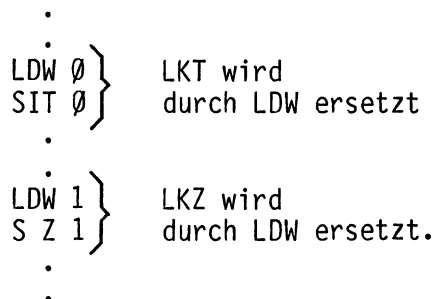


Sind Zeiten und Zähler zu steuern, so sind unterschiedliche Nummern zu verwenden, z.B.

T 0...T 7 DW 0...DW 7
Z 8...Z 15 DW 8...DW 15

Es sind maximal 16 Zeiten und Zähler in Summe steuerbar.

Folgender Programmaufbau ist im AG erforderlich:



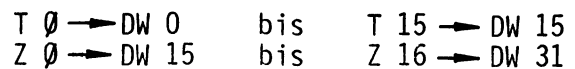
Steuern per Programmiergerät

Das einer Zeit oder einem Zähler zugeordnete Datenwort wird über die PG-Funktion "Direkte Signalzustandsanzeige" mit dem Preset-Wert besetzt. Der zur Aufnahme der Preset-Werte vorgesehene Datenbaustein ist am PG wählbar, beim AG 101U jedoch fest vorgegeben (DB 1).

Weitere Informationen sind der Bedienungsanleitung des PG 605U (6ES5 998-OUP11) Kapitel "PROGRAMM-TEST-Steuern von Zeiten und Zählern" zu entnehmen.

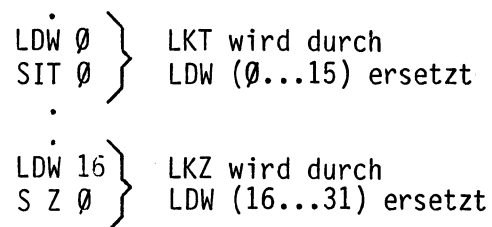
Programmierung (ab Version 1.2)

Zur Aufnahme der Preset-Werte für Zeiten und Zähler sind die Datenworte DW 0...DW 31 im Datenbaustein DB 1 reserviert. Im Anwenderprogramm wird bei Starten von Zeiten und Zählern auf das zugehörige Datenwort Bezug genommen.



Es sind je 16 Zeiten und 16 Zähler steuerbar.

Folgender Programmaufbau ist im AG erforderlich:



Es ist dafür zu sorgen, daß beim Wechsel der Betriebsart "Stop" → "Betrieb" ein sinnvoller Zeitwert in den benutzten Datenwörtern steht. Zu empfehlen ist, die Zeiten/Zähler im "Stop"-Zustand des AGs mit den Vorab-Preset-Werten zu steuern.

5. Programmierbeispiele

5.1 Grundfunktionen

Grundfunktionen können im FB 1 und im PB 1 programmiert werden.

5.1.1 Verknüpfungsfunktionen (binär)

UND-Verknüpfung

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U E 1.1 U E 1.3 U E 1.7 = A 1.0 </pre>		

Am Ausgang A 1.0 erscheint Signalzustand "1", wenn alle Eingänge gleichzeitig den Signalzustand "1" aufweisen.

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

Am Ausgang A 1.0 erscheint Signalzustand "0", wenn mindestens einer der Eingänge den Signalzustand "0" aufweist.

ODER-Verknüpfung

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

Am Ausgang A 1.2 erscheint Signalzustand "1", wenn mindestens einer der Eingänge den Signalzustand "1" aufweist.

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

Am Ausgang A 1.2 erscheint Signalzustand "0", wenn alle Eingänge gleichzeitig den Signalzustand "0" aufweisen.

UND-vor-ODER-Verknüpfung

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U E 1.5 U E 1.6 O U E 1.4 U E 1.3 = A 1.1 </pre>		

Am Ausgang A 1.1 erscheint Signalzustand "1", wenn mindestens eine UND-Verknüpfung erfüllt ist.

Am Ausgang A 1.1 erscheint Signalzustand "0", wenn keine UND-Verknüpfung erfüllt ist.

ODER-vor-UND-Verknüpfung

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> O E 1.0 O U E 1.1 U(O E 1.2 O E 1.3) = A 1.1 </pre>		

Am Ausgang A 1.1 erscheint Signalzustand "1", wenn Eingang E 1.0 oder E 1.1 und einer der Eingänge E 1.2 bzw. E 1.3 Signal "1" führen.

Am Ausgang A 1.1 erscheint Signalzustand "0", wenn Eingang E 1.0 Signal "0" führt und die UND-Verknüpfung nicht erfüllt ist.

ODER-vor-UND-Verknüpfung

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U(O E1.4 O E1.5) U(O E2.0 O E2.1) = A2.0 </pre>		

Am Ausgang A 2.0 erscheint Signalzustand "1", wenn beide ODER-Verknüpfungen erfüllt sind.

Am Ausgang A 2.0 erscheint Signalzustand "0", wenn mindestens eine ODER-Verknüpfung nicht erfüllt ist.

Abfrage auf Signalzustand "0"

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U E1.5 UNE1.6 = A2.0 </pre>		

Am Ausgang A 2.0 erscheint Signalzustand "1" nur dann, wenn der Eingang E 1.5 den Signalzustand "1" und der Eingang E 1.6 den Signalzustand "0" führt.

5.1.2 Speicherfunktionen

RS-Speicherglied für speichernde Signalausgabe

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	U E 1.7 S A 1.5 U E 1.4 R A 1.5 NOP Ø*		

Signalzustand "1" am Eingang E 1.7 bewirkt das Setzen des Speichergliedes.

Wechselt der Signalzustand am Eingang E 1.7 nach "0", so bleibt dieser Zustand erhalten, d. h. das Signal wird gespeichert.

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

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

Bei gleichzeitigem Anliegen des Setzsignals (Eingang E 1.7) und des Rücksetzsignals (Eingang E 1.4) ist die zuletzt programmierte Abfrage (hier UE 1.4) während der Bearbeitung des übrigen Programms wirksam, d. h. der Ausgang A 1.5 wird vorrangig rückgesetzt.

*NOP Ø ist nur erforderlich, wenn Programm in KOP oder FUP am PG 670/PG 675 dargestellt werden soll. Beim Programmieren im Kontaktplan oder Funktionsplan werden diese NOP Ø-Operationen selbsttätig vergeben.

RS-Speicherglied mit Merkern

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	U E 1.3 R M 1.7 U E 1.6 S M 1.7 U M 1.7 = A 1.4		

Signalzustand "1" am Eingang E 1.6 bewirkt das Setzen des Speichergliedes.

Wechselt der Signalzustand am Eingang E 1.6 nach "0", so bleibt dieser Zustand erhalten, d. h. das Signal wird gespeichert.

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

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

Bei gleichzeitigem Anliegen des Setzsignals (Eingang E 1.6) und des Rücksetzsignals (Eingang E 1.3) ist die zuletzt programmierte Abfrage (hier UE 1.6) während der Bearbeitung des übrigen Programms wirksam, d. h. der Merker M 1.7 wird vorrangig gesetzt.

Nachbildung eines Wischkontaktes (Flankenauswertung)

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U E 1.7 UNM 4.0 = M 2.0 U M 2.0 S M 4.0 UNE 1.7 R M 4.0 NOP 0 </pre>		

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

Der Merker M 2.0 wird zurückgesetzt. Der Merker M 2.0 führt also während eines einzigen Programmdurchlaufs Signalzustand "1".

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

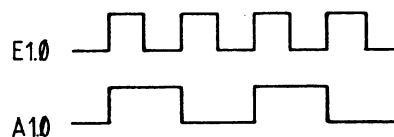
*Verknüpfungsergebnis

Binäruntersetzer (T-Kippglied)

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U E 1.0 UNM 1.0 S A 1.0 U E 1.0 U M 1.0 R A 1.0 NOP 0 UNE 1.0 U A 1.0 S M 1.0 UNE 1.0 UNA 1.0 R M 1.0 NOP 0 </pre>		

Wenn am Eingang E 1.0 der Übergang von "0" nach "1" (ansteigende Flanke) eintritt, dann wechselt der Ausgang A 1.0 seinen Zustand. Bei einer fallenden Flanke am Eingang ändert sich der Ausgang nicht.

Belegt man den Eingang mit einer festen Frequenz, so erscheint am Ausgang die halbe Eingangsfrequenz.



5.1.3 Lade- und Transferfunktionen

Lade- und Transferoperationen

Operation	Parameter	Funktion
L <input type="checkbox"/>		Laden
<input type="checkbox"/> ↑↑ E B	0 bis 5	eines Eingabebytes (vom PAE ¹⁾)
E W	0 bis 4	eines Eingaworts (vom PAE)
A B	0 bis 3	eines Ausgabebytes (vom PAA ²⁾)
A W	0 bis 2	eines Ausgabeworts (vom PAA)
M B	0 bis 63	eines Merkerbytes
M W	0 bis 62	eines Merkerworts
D R	1 bis 255	eines Datums (rechtes Byte)
D L	1 bis 255	eines Datums (linkes Byte)
D W	1 bis 255	eines Datums (Wort)
P B	0 bis 5	eines Peripheriebytes der Digital-Eingaben bzw. -Ausgaben (unter Umgehung des PAE)
T	0 bis 15	eines Zeitwerts (dual)
C T	0 bis 15	(BCD-codiert)
Z	0 bis 15	eines Zählers (dual)
C Z	0 bis 15	(BCD-codiert)
K M ³⁾	beliebiges Bitmuster (16 Bits)	einer Konstanten als Bitmuster
K H ³⁾	0 bis FFFF	einer Konstanten im Hexa-Code
K F ³⁾	- 32768 bis + 32767	einer Konstanten als Festpunktzahl
K Y ³⁾	0 bis 255 für jedes Byte	einer Konstanten, 2 Byte
K C ³⁾	2 beliebige alpha-numerische Zeichen	einer Konstanten, 2 ASCII-Zeichen
K T ³⁾	0.0 bis 999.3	eines Zeitwertes (Konstante)
K Z ³⁾	0 bis 999	eines Zählwertes (Konstante)

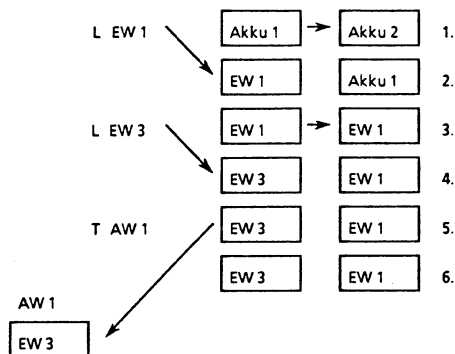
Hinweis:

Das Automatisierungsgerät hat für Vergleichs- und Rechenoperationen sowie für Digitalverknüpfungen zwei Akkumulatoren (16 Bit).

Laden heißt, daß der Inhalt von Akku 1 nach Akku 2 umgeladen und daß dann dem Operanden der Ladeoperation entsprechend der Akku 1 neu geladen wird.

Nach zwei Ladeoperationen kann man daher zum Beispiel mit Vergleichsoperationen Aussagen über den Inhalt der Akkumulatoren bekommen.

Beispiel:



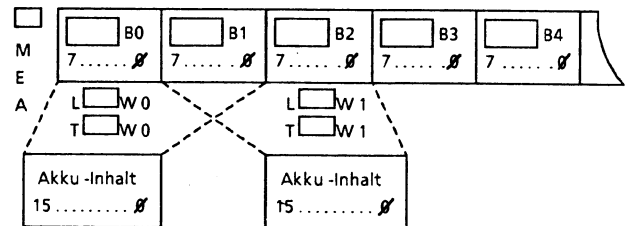
Die Transferoperation transferiert immer den Inhalt von Akku 1 zu dem bei der Transferoperation angegebenen Operanden. Der Inhalt wird dabei nicht verändert.

Operation	Parameter	Funktion
T <input type="checkbox"/>		Transferieren
<input type="checkbox"/> ↑↑ E B	0 bis 5	eines Eingabebytes (vom PAE ¹⁾)
E W	0 bis 4	eines Eingaworts (vom PAE)
A B	0 bis 3	eines Ausgabebytes (vom PAA ²⁾)
A W	0 bis 2	eines Ausgabeworts (vom PAA)
M B	0 bis 63	eines Merkerbytes
M W	0 bis 62	eines Merkerworts
D R	1 bis 255	eines Datums (rechtes Byte)
D L	1 bis 255	eines Datums (linkes Byte)
D W	1 bis 255	eines Datums (Wort)

Die Lade- und Transferoperationen sind absolute Befehle, das heißt, sie werden unabhängig vom Verknüpfungsergebnis ausgeführt.

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

Beim Laden/Transferieren von MW, EW, AW gilt folgender Zusammenhang zwischen Akku-Inhalt und den zu einem Wort gehörenden Bytes:



Beim Laden eines MB, EB, AB, PB wird das Byte stets in das LOW-Byte des AKKU geladen. In das HIGH-Byte des AKKU wird 0 geschrieben.

Beim Transferieren eines MB, EB, AB, PB wird stets das LOW-Byte des AKKU transferiert.

- 1) PAE Prozeßabbild für Eingänge
- 2) PAA Prozeßabbild für Ausgänge
- 3) 4 Byte-Befehl, wobei der Opcode in Byte 0/1 und die Konstante in Byte 2/3 steht.

Laden und Transferieren eines Zeitwertes (siehe auch Zeit- und Zähloperationen)

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 2.0 L KT150.0 SI T 10 NOP 0 L T 10 T MW 20 NOP 0 NOP 0 </pre>		

Der Ausgang DU vom Zeitglied wurde, bei der graphischen Eingabe, mit MW 20 belegt.

Das Programmiergerät hinterlegt daraufhin selbsttätig den entsprechenden Lade- und Transferbefehl im Anwenderprogramm. So wird der Inhalt der mit T 10 adressierten Speicherzelle in den Akkumulator (AKKU 1) geladen.

Anschließend wird der Akkumulatorinhalt (AKKU 1) in das MW 20 transferiert.

Am MW 20 sieht man in diesem Beispiel die Zeit T 10 dualcodiert mitlaufen.

Die Ausgänge DU und DE sind digitale Ausgänge. An Ausgang DU (DE) steht der Zeitwert dual-codiert (BCD-codiert mit Zeitraster) an.

5.1.4 Zeitfunktionen

Bei Netzspannungs-Wiederkehr werden die Zeitglieder neu gestartet.

Impuls

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> UE 2.0 L KT10.2 SIT 1 NOP 0 NOP 0 NOP 0 UT 1 = A 1.0 </pre>		

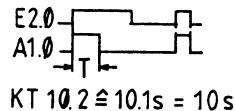
Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis "1" bleibt das Zeitglied unbeeinflusst.

Das Zeitglied wird mit dem angegebenen Wert (10) geladen. Die Zahl rechts vom Punkt gibt das Zeitraster an:
 $0 \hat{=} 0.01 \text{ s}$ $2 \hat{=} 1 \text{ s}$
 $1 \hat{=} 0.1 \text{ s}$ $3 \hat{=} 10 \text{ s}$

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

Die Ausgänge DU und DE sind digitale Ausgänge. Am Ausgang DU (DE) steht der Zeitwert dual-codiert (BCD-codiert) an.

Die Abfragen UT bzw. OT liefern Signalfzustände "1", solange die Zeit läuft.



Verlängerter Impuls

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> UE 2.0 L KT10.2 SV T1 NOP 0 NOP 0 NOP 0 UT 1 = A 1.0 </pre>		

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



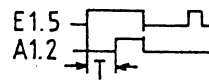
Bei Verknüpfungsergebnis "0" bleibt das Zeitglied unbeeinflusst.

Die Abfragen UT oder OT liefern Signalfzustand "1", solange die Zeit läuft.

Einschaltverzögerung

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 1.5 L KT9.2 SE T 3 NOP 0 NOP 0 NOP 0 UT 3 = A 1.2 </pre>		

Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis "1" bleibt das Zeitglied unbeeinflusst.



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

Die Abfrage von UT bzw. OT liefert Signalzustand "1", wenn die Zeit abgelaufen und das Verknüpfungsergebnis am Eingang noch ansteht.

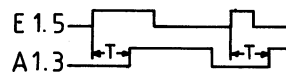
Das Zeitglied wird mit dem angegebenen Wert (9) geladen. Die Zahl rechts vom Punkt gibt das Zeitraster an:
 $0 \hat{=} 0.01 \text{ s}$ $2 \hat{=} 1 \text{ s}$
 $1 \hat{=} 0.1 \text{ s}$ $3 \hat{=} 10 \text{ s}$

Die Ausgänge DU und DE sind digitale Ausgänge. Am Ausgang DU (DE) steht der Zeitwert dual-codiert (BCD-codiert) an.

Speichernde Einschaltverzögerung

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 1.5 L KT9.2 SS T4 UE 1.6 RT 4 NOP 0 NOP 0 UT 4 = A 1.3 </pre>		

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



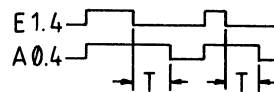
Bei Verknüpfungsergebnis "0" bleibt das Zeitglied unbeeinflusst.

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 wurde.

Ausschaltverzögerung

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre> U E 1.4 L KT100.0 SA T5 NOP 0 NOP 0 NOP 0 U T5 = A 0.4 </pre>		

Bei Verknüpfungsergebnis "0" und erstmaliger Bearbeitung wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis "0" bleibt das Zeitglied unbeeinflusst.



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

1 sec (s)

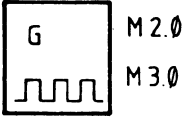
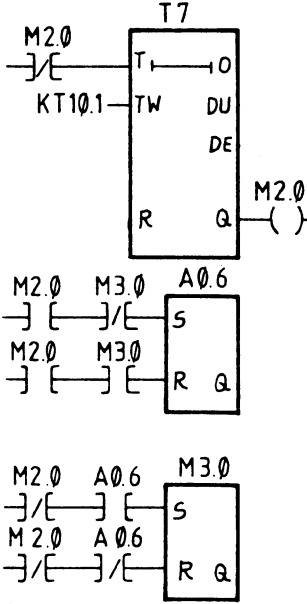
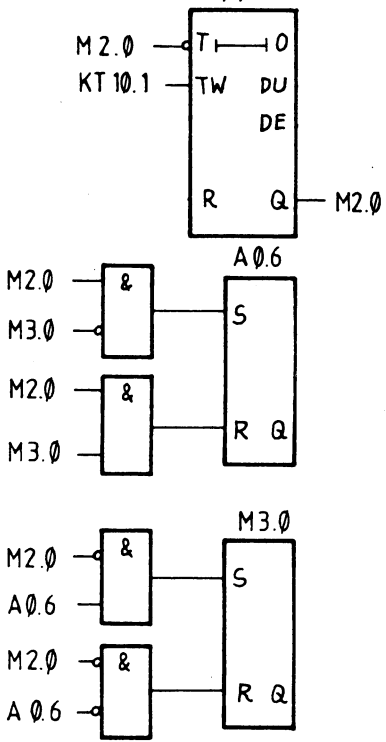
Das Zeitglied wird mit dem angegebenen Wert geladen. Die Zahl rechts vom Punkt gibt das Zeitraster an:

0 $\hat{=}$ 0.01 s 2 $\hat{=}$ 1 s
 1 $\hat{=}$ 0.1 s 3 $\hat{=}$ 10 s

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

Die Ausgänge DU und DE sind digitale Ausgänge. Am Ausgang DU (DE) steht der Zeitwert dual-codiert (BCD-codiert) an.

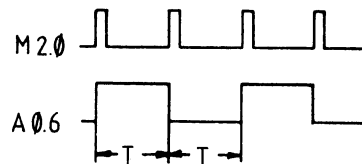
Taktgeber (Taktgenerator)

Vorlage	STEP-5-Darstellung Anweisungs- liste	Kontaktplan	Funktionsplan
 <p>A 0.6</p>	<pre> UNM 2.0 LKT 10.1 SET 7 NOP 0 NOP 0 NOP 0 UT 7 = M 2.0 UM 2.0 UNM 3.0 SA 0.6 UM 2.0 UM 3.0 UM 3.0 RA 0.6 NOP 0 UNM 2.0 UA 0.6 SM 3.0 UNM 2.0 UNA 0.6 RM 3.0 NOP 0 </pre>		

Ein Taktgeber kann durch ein selbsttaktendes Zeitglied, dem ein T-Kippglied (Binär-
untersetzer) nachgeschaltet ist, aufgebaut
werden.

Mit dem Merker M 2.0 wird die Zeit T 7
nach jedem Ablauf neu gestartet, d.h. der
Merker M 2.0 führt nach jedem Ablauf der
Zeit für einen Zyklus den Signalzustand
"1".

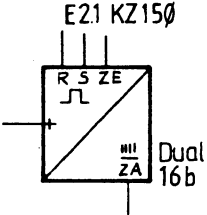
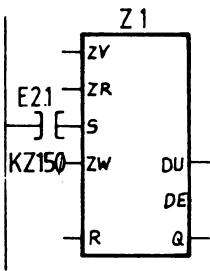
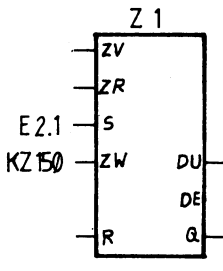
Diese Impulse des Merker M 2.0 wirken auf
das nachfolgende T-Kippglied, so daß am
Ausgang A 0.6 eine Impulsfolge mit dem
Tastverhältnis 1 : 1 erscheint. Die Peri-
odendauer dieser Impulsfolge ist doppelt so
groß wie der Zeitwert T des selbsttaktenden
Zeitgliedes.



5.1.5 Zählfunktionen

Bei Netzspannungs-Wiederkehr werden die Zähler ~~0...15~~ neu gestartet.
~~0...15~~
 8...15

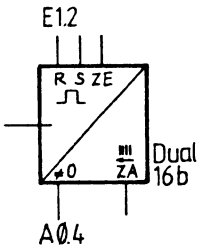
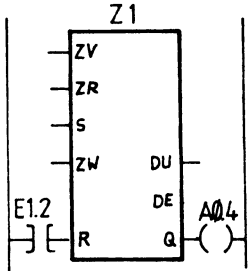
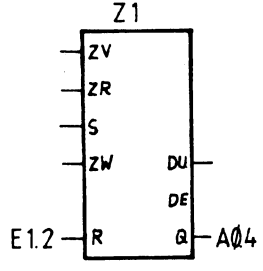
Zähler setzen

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	<pre> U E 2.1 L KZ150 S Z 1 </pre>		

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

In dem obigen Beispiel beträgt der Anfangswert des Zählers 150. Die Ausgänge DU und DE sind digitale Ausgänge. Am Ausgang DU (DE) steht der Zählerwert dual-codiert (BCD-codiert).

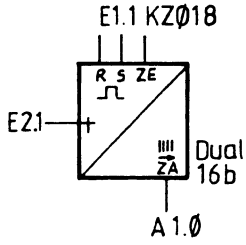
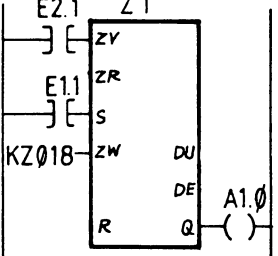
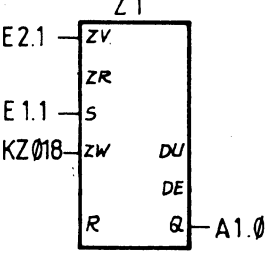
Zähler zurücksetzen

Vorlage	STEP-5-Darstellung		
	Anweisungsliste	Kontaktplan	Funktionsplan
	<pre> U E 1.2 R Z 1 U Z 1 = A 0.4 </pre>		

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

Bei Verknüpfungsergebnis "0" bleibt der Zähler unbeeinflusst.

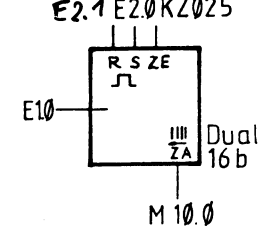
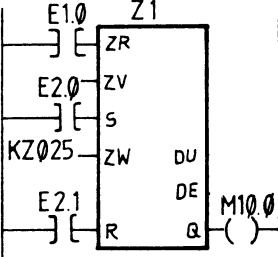
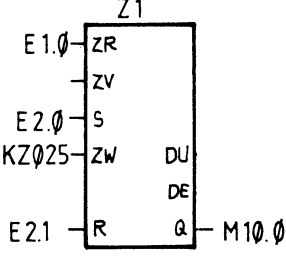
Vorwärts zählen

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U E 2.1 ZVZ 1 NOP 0 U E 1.1 LKZ 018 S Z 1 NOP 0 NOP 0 NOP 0 U Z 1 = A 1.0 </pre>		

Der Wert des adressierten Zählers wird um 1 erhöht. Die Funktion ZV wird nur bei einer positiven Flanke (von "0" nach "1") der vor ZV programmierten Verknüpfung ausgeführt.

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

Rückwärts zählen

Vorlage	STEP-5-Darstellung		Funktionsplan
	Anweisungs- liste	Kontaktplan	
	<pre> U E 1.0 ZRZ 1 NOP 0 U E 2.0 LKZ 025 S Z 1 U E 2.1 R Z 1 NOP 0 NOP 0 U Z 1 = M 10.0 </pre>		

Der Wert des adressierten Zählers wird um 1 erniedrigt. Die Funktion ZR wird nur bei einer positiven Flanke (von "0" nach "1") der vor ZR programmierten Verknüpfung ausgeführt.

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

5.1.6 Vergleichsfunktionen

Vergleich auf gleich

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>L EB 0 L EB 1 !=F =A1.0</pre>		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Das Vergleichsergebnis wird durch die Anzeigebits ANZO und ANZ1 angezeigt.

für	ANZ1	ANZO	VKE
EB 0 = EB 1	0	0	1
EB 0 < EB 1	0	1	0
EB 0 > EB 1	1	0	0

Beim Vergleich wird die Zahlendarstellung der Operanden (Festpunktrechnung) berücksichtigt.

Nach einem Vergleich auf gleich kann mit der Sprungfunktion (bei VKE = 1) SPZ = ... zu einer "Marke" (+ 127 Wörter) gesprungen werden.

VKE = Verknüpfungsergebnis

Vergleich auf ungleich

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>L EB 0 L EB 1 ><F =A1.1</pre>		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Das Vergleichsergebnis wird durch die Anzeigebits ANZO und ANZ1 angezeigt.

für	ANZ1	ANZO	VKE
EB 0 = EB 1	0	0	0
EB 0 < EB 1	0	1	1
EB 0 > EB 1	1	0	1

Beim Vergleich wird die Zahlendarstellung der Operanden (Festpunktrechnung) berücksichtigt.

Nach einem Vergleich auf ungleich kann mit der Sprungfunktion (bei VKE = 1) SPN = ... zu einer "Marke" (+ 127 Wörter) gesprungen werden.

Vergleich auf größer

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>L EB 0 L EB 1 > F = A1.2</pre>		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Das Vergleichsergebnis wird durch die Anzeigebits ANZ0 und ANZ1 angezeigt.

für	ANZ1	ANZ0	VKE
EB 0 = EB 1	0	0	0
EB 0 < EB 1	0	1	0
EB 0 > EB 1	1	0	1

Beim Vergleich wird die Zahlendarstellung der Operanden (Festpunktrechnung) berücksichtigt.

Nach einem Vergleich auf größer kann mit der Sprungfunktion (bei VKE = 1) SPP = ... zu einer "Marke" (+ 127 Wörter) gesprungen werden.

Vergleich auf kleiner

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>L EB 0 L EB 1 < F = A1.4</pre>		

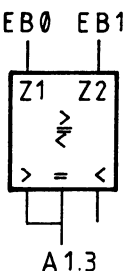
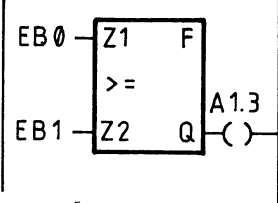
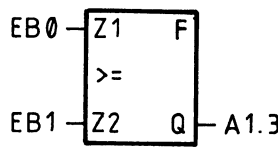
Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Das Vergleichsergebnis wird durch die Anzeigebits ANZ0 und ANZ1 angezeigt.

für	ANZ1	ANZ0	VKE
EB 0 = EB 1	0	0	0
EB 0 < EB 1	0	1	1
EB 0 > EB 1	1	0	0

Beim Vergleich wird die Zahlendarstellung der Operanden (Festpunktrechnung) berücksichtigt.

Nach einem Vergleich auf kleiner kann mit der Sprungfunktion (VKE = 1) SPM = ... zu einer "Marke" (+ 127 Wörter) gesprungen werden.

Vergleich auf größer-gleich

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>L EB0 L EB1 >=F =A1.3</pre>		

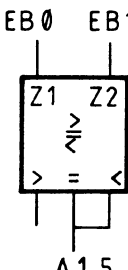
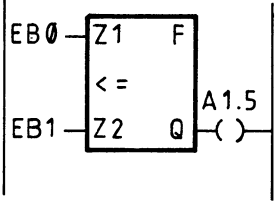
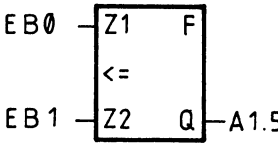
Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Das Vergleichsergebnis wird durch die Anzeigebits ANZO und ANZ1 angezeigt.

für	ANZ1	ANZO	VKE
EB 0 = EB 1	0	0	1
EB 0 < EB 1	0	1	0
EB 0 > EB 1	1	0	1

Beim Vergleich wird die Zahlendarstellung der Operanden (Festpunktrechnung) berücksichtigt.

Nach einem Vergleich auf größer-gleich kann mit der Sprungfunktion (VKE = 1) SPB = ... zu einer "Marke" (+ 127 Wörter) gesprungen werden.

Vergleich auf kleiner-gleich

Vorlage	STEP-5-Darstellung		
	Anweisungs- liste	Kontaktplan	Funktionsplan
	<pre>L EB0 L EB1 <=F =A1.5</pre>		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen. Das Vergleichsergebnis wird durch die Anzeigebits ANZO und ANZ1 angezeigt.

für	ANZ1	ANZO	VKE
EB 0 = EB 1	0	0	1
EB 0 < EB 1	0	1	1
EB 0 > EB 1	1	0	0

Beim Vergleich wird die Zahlendarstellung der Operanden (Festpunktrechnung) berücksichtigt.

Nach einem Vergleich auf kleiner-gleich kann mit der Sprungfunktion (VKE = 1) SPB = ... zu einer "Marke" (+ 127 Wörter) gesprungen werden.

5.1.7 Arithmetische Funktionen

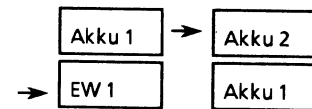
Die arithmetischen Operationen können nur in der Anweisungsliste dargestellt werden.

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

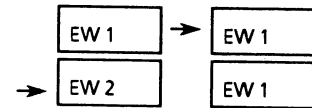
Operation	Parameter	Funktion
+	F	Addieren (Akku 1 + Akku 2)
-	F	Subtrahieren (Akku 2 - Akku 1)

Durch zwei Ladeoperationen können die beiden Akkus 1 und 2 entsprechend den Operanden der Ladeoperationen geladen werden. Anschließend lassen sich die Inhalte beider Akkus addieren oder subtrahieren.

Beispiel: L EW 1

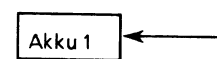


L EW 2



-F

EW 1 - EW 2 = Ergebnis



Beispiel	AWL	Erläuterung
Von der Zahl +127 soll das rechte Byte des Datenwortes 85 subtrahiert werden und das Ergebnis im linken Byte des Datenwortes 85 abgespeichert werden.	L KF +127	Die konstante Festpunktzahl +127 wird in den Akku 1 geladen, gleichzeitig wird der alte Inhalt von Akku 1 in den Akku 2 geschoben.
	L DR 85	Das rechte Byte des Datenwortes 85 wird in den Akku 1 geladen und die Festpunktzahl +127 in den Akku 2 geschoben.
	-F	Der Inhalt von Akku 1 wird vom Akku 2 subtrahiert, und das Ergebnis in Akku 1 hinterlegt.
	T DL 85	Der Inhalt von Akku 1 (Ergebnis) wird ins linke Byte des Datenwortes 85 transferiert.

	127
DR 85	74
Ergebnis-F	53

Hinweis:

Bei Überschreitung des Zahlbereiches (-32768 bis +32767) ist das Ergebnis der Operation undefiniert (OV = "1")

5.1.8 Sonstige Funktionen

Die folgenden Operationen sind nur in der Anweisungsliste darstellbar.

Operation	Parameter	Funktion
S T P		Stop
N O P 0		Nulloperation (alle Bits gelöscht)
N O P 1		Nulloperation (alle Bits gesetzt)
B L D	0 bis 255	Bildaufbaubefehl

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

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

Der Bildaufbaubefehl bestimmt die Einteilung von Programmteilen in Segmente innerhalb eines Bausteins.

5.2 Ergänzende Funktionen

Ergänzende Funktionen können nur im FB 1 programmiert werden.

5.2.1 Verknüpfungsfunktionen (wortw.)

Operation	Beschreibung	
UW	UND-Verknüpfung digital von Akku 1 und Akku 2	Durch zwei Ladeoperationen können der Akku 1 und Akku 2 entsprechend den Operanden der Ladeoperationen geladen werden. Anschließend lassen sich die Inhalte beider Akkus digital verknüpfen.
OW	ODER-Verknüpfung digital von Akku 1 und Akku 2	
XOW	Exklusiv-ODER-Verknüpfung digital von Akku 1 und Akku 2	

Beispiel	AWL	Erläuterungen
<p>Die Hexzahl 3F84_H soll mit dem Eingangswort 1 der Eingänge EW 1 UND verknüpft werden. Das Ergebnis soll im Ausgangswort 0 der Ausgänge (AW 10) gezeigt werden.</p> <p style="text-align: right;"> 3F84_H EW 1 4793_H Ergebnis UW 0780_H </p>	<p>L KH 3F84</p> <p>L EW 1</p> <p style="border: 1px solid black; padding: 2px;">UW</p> <p>T AW 0</p>	<p>Die Hexzahl 3F84 wird in den Akku 1 geladen gleichzeitig wird der alte Inhalt von Akku 1 in den Akku 2 geschoben. Das Eingangswort 1 (EW 1) wird in den Akku 1 geladen und die Hexzahl in den Akku 2 geschoben. Der Inhalt von Akku 1 wird mit Akku 2 digital UND verknüpft und das Ergebnis in Akku 1 hinterlegt. Der Inhalt (Ergebnis) von Akku 1 wird zum Ausgangswort 0 transferiert.</p>
<p>Die beiden Bitmuster 0101 1110 1000 1011</p> <p>und 0111 0001 0111 1100 sollen miteinander ODER verknüpft werden.</p> <p style="text-align: right;"> 0101 1110 1000 1011 0111 0001 0111 1100 Erg.OW 0111 1111 1111 1111 </p>	<p>L KM 01...11</p> <p>L KM 01...00</p> <p style="border: 1px solid black; padding: 2px;">OW</p> <p>T MW 13</p>	<p>Das erste Bitmuster wird in den Akku 1 geladen, gleichzeitig wird der alte Inhalt von Akku 1 in den Akku 2 geschoben. Das zweite Bitmuster wird in den Akku 1 geladen und das erste Bitmuster in den Akku 2 geschoben.</p> <p>Der Inhalt von Akku 1 wird mit Akku 2 digital ODER verknüpft und das Ergebnis in Akku 1 hinterlegt. Der Inhalt (Ergebnis) von Akku 1 wird ins Merkerwort 13 (MW 13) transferiert.</p>
<p>Das Eingangswort 0 soll mit dem Datenwort 12 auf Gleichheit überprüft werden. Die ungleichen Bits der Worte sollen im Ausgangswort 0 gezeigt werden.</p> <p style="text-align: right;"> DW 12 EA83_H EW 0 68C5_H Ergebnis XOW 8246_H </p>	<p>L DW 12</p> <p>L EW 0</p> <p style="border: 1px solid black; padding: 2px;">XOW</p> <p>T AW 0</p>	<p>Das Datenwort 12 wird in den Akku 1 geladen, gleichzeitig wird der alte Inhalt von Akku 1 in den Akku 2 geschoben. Das Eingangswort 0 wird in den Akku 1 geladen und das Datenwort 12 in den Akku 2 geschoben.</p> <p>Der Inhalt von Akku 1 wird mit Akku 2 digital EXOR verknüpft und das Ergebnis in Akku 1 hinterlegt. Der Inhalt (Ergebnis) von Akku 1 wird zum Ausgangswort 0 transferiert.</p>

5.2.2 Umwandlungsfunktionen

Operation	Beschreibung
KEW	1er-Komplement
KZW	2er-Komplement
	Der im Akkumulator 1 stehende Wert wird umgewandelt. Das Ergebnis kann im Akkumulator weiter verarbeitet werden.

Beispiel	AWL	Erläuterung
Der Inhalt des Datenwortes 64 soll Bit für Bit invertiert werden und im Datenwort 78 abgelegt werden. DW 64 EA83 _H DW 78 157C _H	L DW 64 KEW T DW 78	Lade Datenwort 64 in den Akku 1. Einerkomplement des Inhalt von Akku 1. Ergebnis befindet sich im Akku 1. Transferiere den Inhalt von Akku 1 ins Datenwort 78.
Der Inhalt des Datenwortes 42 ist als Festpunktzahl zu interpretieren und mit umgekehrten Vorzeichen ins Datenwort 35 abzulegen. DW 42 +51 DW 35 -51	L DW 42 KZW T DW 35	Lade Datenwort 42 in den Akku 1. Zweierkomplement des Inhalt von Akku 1. Ergebnis befindet sich im Akku 1. Transferiere den Inhalt von Akku 1 ins Datenwort 35.

5.2.3 Schiebefunktionen

Operation	Beschreibung
SLW 0 bis 15	Schieben links
SRW 0 bis 15	Schieben rechts
	Der Parameter dieser Anweisung gibt die Anzahl der Bitstellen an, um die der Inhalt des Akkumulators 1 nach links (SLW) bzw. nach rechts (SRW) verschoben wird. Die beim Schieben freiwerdenden Bitstellen werden mit Nullen aufgefüllt.

Beispiel	AWL	Erläuterung
Bei der Hexzahl 14AF _H im Datenwort 1 sollen die letzten 4 Bit wegfallen, und die sich daraus ergebene neue Hexzahl 014A _H soll im Datenwort 3 abgespeichert werden.	L DW 1 SR W 4 T DW 3	Lade Datenwort 1 in den Akku 1. Schiebe den Inhalt von Akku 1 vier Bitstellen nach rechts. Die beim Schieben frei werdenden Bitstellen werden mit Nullen aufgefüllt. Transferiere den Inhalt von Akku 1 in das Datenwort 3.

Hinweis:

Die Schiebefunktionen werden unabhängig von Bedingungen ausgeführt. Das zuletzt hinausgeschobene Bit kann mit Sprungfunktionen abgefragt werden.

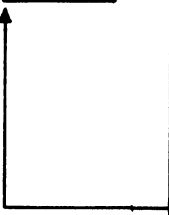
Mit SPZ kann gesprungen werden, wenn das Bit "0" ist und mit SPN oder SPP wenn das Bit "1" ist.

5.2.4 Sprungfunktionen

Das Sprungziel für unbedingte und bedingte Sprünge wird symbolisch angegeben (maximal 4 Zeichen). Bei PG 605U/PG 615U können die Sprungmarken (M0...M99) vergeben werden. 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 über Segmentgrenzen hinweg sind nicht zulässig.

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

Operation	Beschreibung
SPA = <input type="text"/>	<p>Sprung unbedingt</p> <p>Der unbedingte Sprung wird unabhängig von Bedingungen ausgeführt.</p>
SPB = <input type="text"/>	<p>Sprung bedingt</p> <p>Der bedingte Sprung wird ausgeführt, wenn das Verknüpfungsergebnis "1" ist. Bei Verknüpfungsergebnis "0" wird die Anweisung nicht ausgeführt und das Verknüpfungsergebnis auf "1" gesetzt.</p>
SPZ = <input type="text"/>	<p>Sprung, wenn Akkumulatorinhalt Null (Zero)</p> <p>Der Sprung wird ausgeführt, wenn der Akkumulatorinhalt Null ist. Ist der Akkumulatorinhalt nicht Null, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.</p>
SPN = <input type="text"/>	<p>Sprung, wenn Akkumulatorinhalt ungleich Null</p> <p>Der Sprung wird ausgeführt, wenn der Akkumulatorinhalt ungleich Null ist. Ist der Akkumulatorinhalt Null, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.</p>
SPP = <input type="text"/>	<p>Sprung, wenn Akkumulatorinhalt positiv</p> <p>Der Sprung wird ausgeführt, wenn der Akkumulatorinhalt größer als Null ist. Ist der Akkumulatorinhalt Null oder kleiner Null, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.</p>
SPM = <input type="text"/>	<p>Sprung, wenn Akkumulatorinhalt negativ</p> <p>Der Sprung wird ausgeführt, wenn der Akkumulatorinhalt kleiner Null ist. Ist der Akkumulatorinhalt Null oder größer Null, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.</p>
SPO = <input type="text"/> 	<p>Sprung bei Überlauf (Overflow)</p> <p>Der Sprung wird ausgeführt, wenn ein Überlauf vorliegt. Liegt kein Überlauf vor, wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert.</p> <p>Symboladresse einsetzen (max. 4 Zeichen)</p>

Beispiel	AWL	Erläuterung
Ist kein Eingang des Eingangswort 1 gesetzt, so wird zur Marke "AN 1" gesprungen	ANO:L EW1 <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px 0;">:SPZ=AN1</div>	Das Eingangswort 1 wird in den Akku 1 geladen. Ist Akkuinhalt 1 gleich Null, so wird zur Marke "AN 1" gesprungen, ansonsten wird die nächste Anweisung (UE 1.Ø) bearbeitet.
Stimmen Eingangswort 1 und Ausgangswort 0 nicht überein, so wird zur Marke "AN 0" zurückgesprungen. Stimmen EW 1 und AW 3 überein, so wird EW 1 mit dem Datenwort 12 verglichen. Ist EW 1 größer oder kleiner DW 12 so wird zur Marke "Ziel" gesprungen.	:U EWØ AN1:L EW1 :L AWO :XOW <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px 0;">:SPP=ANO</div> :L EW1 :L DW12 :>< F <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px 0;">:SPB=Ziel</div> . . . Ziel:U E12.2	Vergleich von Eingangswort 1 und Ausgangswort 3. Bei Ungleichheit sind einzelne Bits im Akku 1 gesetzt. Ist der Akku 1 nicht null, so wird zur Marke "AN 0" zurückgesprungen, sonst nächste Anweisungen bearbeiten. Das Eingangswort 1 wird mit dem Datenwort 12 auf größer oder kleiner verglichen. Bei größer oder kleiner wird VKE "1" gesetzt. Ist VKE = "1", so wird zur Marke "Ziel" gesprungen. Ist VKE = "0", so wird die nächste Anweisung bearbeitet.

5.2.5 Anzeigenbildung

Das Steuerwerk des Automatisierungsgeräts SIMATIC S5 101U besitzt 3 Anzeigen:

- o Anz 1 Wert positiv
- o Anz 0 Wert negativ
- o OVF Überlauf (Overflow)

Die Anzeigen werden mit den Vergleichsoperationen, den Rechenoperationen, den Schiebeoperationen und einigen Umwandlungsoperationen beeinflusst.

Anzeigenbildung bei Vergleichsoperationen

Die Ausführung der Vergleichsoperationen führt zum Setzen der Anzeigen Anz 0 und Anz 1. Die Überlaufanzeige wird nicht beeinflusst.

Die beiden, an einem Vergleich beteiligten Operanden wurden wie folgt definiert:

```

Beispiel :L DW20 (1. Operand)
          :L DW21 (2. Operand)
          :!=F (Vergleichsoperation)
    
```

Datenwort DW 20 ist in diesem Beispiel der 1. Operand, Datenwert DW 21 oder 2. Operand

	Anzeigen		Sprungfunktionen
	Anz 1	Anz 0	
Der 1.Operand ist gegenüber dem 2. Operanden:			Es werden folgende Sprungfunktionen ausgeführt:
gleich	0	0	SPZ
kleiner	0	1	SPN, SPM
größer	1	0	SPN, SPP

Hinweis:

Die Vergleichsoperationen beeinflussen das Verknüpfungsergebnis. Bei erfüllter Bedingung ist das VKE = 1. So kann die bedingte Sprungoperation SPB nach einer Vergleichsoperation eingesetzt werden.

Anzeigenbildung bei Rechenoperationen

Die Ausführung der Rechenoperationen führt zum Setzen aller Anzeigen. Dieses wiederum richtet sich nach dem Akkumulatorinhalt (dem Ergebnis der Rechenoperation).

Anzeigenbildung bei Festpunkt-Arithmetik:

	Anzeigen			Sprungfunktionen
	Anz 1	Anz 0	OV	
Das Ergebnis nach Ausführung der Rechenoperation ist:				Es werden folgende Sprungfunktionen ausgeführt:
< -32768	1	0	1	SPN, SPP, SPO
-32768 bis -1	0	1	0	SPN, SPM
0	0	0	0	SPZ
+1 bis +32767	1	0	0	SPN, SPP
> +32767	0	1	1	SPN, SPM, SPO
(-)65536*	0	0	1	SPZ, SPO

*) Ergebnis der Rechnung: -32768 -32768

Anzeigenbildung bei Digitalverknüpfungen

Die Digitalverknüpfungen führen zum Setzen der Anzeigen Anz 0 und Anz 1. Die Überlaufanzeige wird nicht beeinflusst.

Das Setzen der Anzeigen hängt vom Inhalt des Akkumulators nach der Bearbeitung der Operation ab:

	Anzeigen		Sprungfunktionen
	Anz 1	Anz 0	
Der Inhalt des Akkumulators ist:			Es werden folgende Sprungfunktionen ausgeführt:
Null (KH = 0000)	0	0	SPZ
nicht Null	1	0	SPN, SPP

Anzeigenbildung bei Schiebeoperationen

Die Ausführung der Schiebeoperationen führt zum Setzen der Anzeigen Anz 0 und Anz 1. Die Überlaufanzeige wird nicht beeinflusst.

Das Setzen der Anzeigen hängt vom Zustand des zuletzt hinausgeschobenen Bits ab.

	Anzeigen		Sprungfunktionen
	Anz 1	Anz 0	
Der Wert des zuletzt hinausgeschobenen Bits ist:			Es werden folgende Sprungfunktionen ausgeführt:
"0"	0	0	SPZ
"1"	1	0	SPN, SPP

Anzeigenbildung bei Umwandlungsfunktionen

Die Bildung des Zweierkomplements (KZW) führt zum Setzen aller Anzeigen. Dieses richtet sich nach dem Ergebnis der Umwandlungsfunktion:

	Anzeigen			Sprungfunktionen
	Anz 1	Anz 0	OV	
Das Ergebnis nach der Umwandlungsfunktion ist:				Es werden folgende Sprungfunktionen ausgeführt:
< -32768	1	0	1	SPN, SPP, SPO
-32768 bis -1	0	1	0	SPN, SPM
+1 bis +32767	1	0	0	SPN, SPP
> +32767	0	1	1	SPN, SPM, SPO
(-)65536*	0	0	1	SPZ, SPO

*) Ergebnis der Umwandlung von KH = 0000

Hinweis:

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

6. Befehlsumfang

Verknüpfungsfunktionen, binär

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begren- zend	Wort Anzeigen werden beein- flußt			Bearbeitungs- zeit max.µs	Funktion
		Wort 0		Wort 1				ANZ1	ANZO	OV		
		B0	B1	B2	B3							
UND-Verknüpfung mit												
U E	∅.∅ bis 2.3 3.∅ bis 5.3	C8	00	-	-	N	N	-	-	-	65	Abfrage eines Einganges auf Signalzustand "1"
U A	∅.∅ bis 1.3 2.∅ bis 3.3	C8	80	-	-	N	N	-	-	-	65	Abfrage eines Ausganges auf Signalzustand "1"
U M	∅.∅ bis 63.7	88	00	-	-	N	N	-	-	-	74	Abfrage eines Merkers auf Signalzustand "1"
U T	∅ bis 15	F9	00	-	-	N	N	-	-	-	69	Abfrage einer Zeit (Time) auf Signalzustand "1"
O Z	∅ bis 15	B9	00	-	-	N	N	-	-	-	64	Abfrage eines Zählers auf Inhalt " > 0"
UN E	∅.∅ bis 2.3 3.∅ bis 5.3	E8	00	-	-	N	N	-	-	-	66	Abfrage eines Einganges auf Signalzustand "0"
UN A	∅.∅ bis 1.3 2.∅ bis 3.3	E8	80	-	-	N	N	-	-	-	66	Abfrage eines Ausganges auf Signalzustand "0"
UN M	∅.∅ bis 63.7	A8	00	-	-	N	N	-	-	-	75	Abfrage eines Merkers auf Signalzustand "0"
UN T	∅ bis 15	FD	00	-	-	N	N	-	-	-	70	Abfrage einer Zeit (Time) auf Signalzustand "0"
UN Z	∅ bis 15	BD	00	-	-	N	N	-	-	-	65	Abfrage eines Zählers auf Inhalt "=0"
ODER-Verknüpfung mit												
O E	∅.∅ bis 2.3 3.∅ bis 5.3	C8	00	-	-	N	N	-	-	-	65	Abfrage eines Einganges auf Signalzustand "1"
O A	∅.∅ bis 1.3 2.∅ bis 3.3	C8	80	-	-	N	N	-	-	-	65	Abfrage eines Ausganges auf Signalzustand "1"
O M	∅.∅ bis 63.7	88	00	-	-	N	N	-	-	-	74	Abfrage eines Merkers auf Signalzustand "1"
O T	∅ bis 15	F9	00	-	-	N	N	-	-	-	71	Abfrage einer Zeit (Time) auf Signalzustand "1"
O Z	∅ bis 15	B9	00	-	-	N	N	-	-	-	64	Abfrage eines Zählers auf Inhalt " > 0"
ON E	∅.∅ bis 2.3 3.∅ bis 5.3	E8	00	-	-	N	N	-	-	-	66	Abfrage eines Einganges auf Signalzustand "0"
ON A	∅.∅ bis 1.3 2.∅ bis 3.3	E8	80	-	-	N	N	-	-	-	66	Abfrage eines Ausganges auf Signalzustand "0"
ON M	∅.∅ bis 63.7	A8	00	-	-	N	N	-	-	-	76	Abfrage eines Merkers auf Signalzustand "0"
ON T	∅ bis 15	FD	00	-	-	N	N	-	-	-	72	Abfrage einer Zeit (Time) auf Signalzustand "0"
ON Z	∅ bis 15	BD	00	-	-	N	N	-	-	-	67	Abfrage eines Zählers auf Inhalt "=0"

+ relative Adresse

+ Bitadresse

)		BF	00	-	-	N	J	-	-	-	63	Klammer zu
U(BA	00	-	-	N	J	-	-	-	66	UND-Verknüpfung von Klammerausdrücken
O(BB	00	-	-	N	J	-	-	-	69	ODER-Verknüpfung von Klammerausdrücken
O		FB	00	-	-	N	J	-	-	-	48	ODER-Verknüpfung von UND-Funktionen

Es können bis zu 6 Klammerebenen programmiert werden.

Speicherfunktion

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begren- zend	Wort Anzeigen werden beein- flußt			Bearbeitungs- zeit max.,µs	Funktion	
		Wort 0		Wort 1				ANZ1	ANZO	OV			
		B0	B1	B2	B3								
Setzen													
S	E	∅.∅ bis 2.3 3.∅ bis 5.3	D0	00	-	-	J	J	-	-	-	70	eines Einganges (im PAE)
S	A	∅.∅ bis 1.3 3.∅ bis 3.3	D0	80	-	-	J	J	-	-	-	70	eines Ausganges (im PAA)
S	M	∅.∅ bis 63.7	90	00	-	-	J	J	-	-	-	74	eines Merkers
Rücksetzen													
R	E	∅.∅ bis 2.3 3.∅ bis 5.3	F0	00	-	-	J	J	-	-	-	70	eines Einganges (im PAE)
R	A	∅.∅ bis 1.3 2.∅ bis 3.3	F0	80	-	-	J	J	-	-	-	70	eines Ausganges (im PAA)
R	M	∅.∅ bis 63.7	B0	00	-	-	J	J	-	-	-	74	eines Merkers
Zuweisen													
=	E	∅.∅ bis 2.3 3.∅ bis 5.3	D8	00	-	-	N	J	-	-	-	66	eines Einganges (im PAE)
=	A	∅.∅ bis 1.3 2.∅ bis 3.3	D8	80	-	-	N	J	-	-	-	66	eines Ausganges (im PAA)
=	M	∅.∅ bis 63.7	98	00	-	-	N	J	-	-	-	72	eines Merkers
<div style="border: 1px solid black; padding: 5px; display: inline-block; margin: 5px;">+ relative Adresse</div> <div style="border: 1px solid black; padding: 5px; display: inline-block; margin: 5px;">+ Bitadresse</div>													
Zeit- und Zählfunktionen													
Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begren- zend	Wort Anzeigen werden beein- flußt			Bearbeitungs- zeit max.,µs	Funktion	
		Wort 0		Wort 1				ANZ1	ANZO	OV			
		B0	B1	B2	B3								
SI	T	∅ bis 15	34	00	-	-	J	J	-	-	-	134	Starten einer Zeit als Impuls
SV	T	∅ bis 15	1C	00	-	-	J	J	-	-	-	120	Starten einer Zeit als verlängerter Impuls
SE	T	∅ bis 15	24	00	-	-	J	J	-	-	-	136	Starten einer Zeit als Einschaltverzögerung
SS	T	∅ bis 15	2C	00	-	-	J	J	-	-	-	120	Starten einer Zeit als speichernde Einschaltverzögerung
SA	T	∅ bis 15	14	00	-	-	J	J	-	-	-	136	Starten einer Zeit als Ausschaltverzögerung
R	T	∅ bis 15	3C	00	-	-	J	J	-	-	-	64	Rücksetzen einer Zeit
S	Z	∅ bis 15	5C	00	-	-	J	J	-	-	-	114	Setzen eines Zählers
R	Z	∅ bis 15	7C	00	-	-	J	J	-	-	-	66	Rücksetzen eines Zählers
ZV	Z	∅ bis 15	6C	00	-	-	J	J	-	-	-	82	Vorwärtszählen eines Zählers
ZR	Z	∅ bis 15	54	00	-	-	J	J	-	-	-	89	Rückwärtszählen eines Zählers

+ relative Adresse

Lade- und Transferfunktionen

Operation	Parameter	Maschinencode (hexadezimal)				Ab- hängig von VKE	VKE begren- zend	Wort Anzeigen werden beein- flußt			Bearbeitungs- zeit max. µs	Funktion
		Wort 0		Wort 1				ANZ1	ANZ0	OV		
		B0	B1	B2	B3							
											Laden	
L EB	Ø bis 5	4A	00	-	-	N	N	-	-	-	61	eines Eingabebytes (vom PAE)
L EW	Ø bis 4	52	00	-	-	N	N	-	-	-	65	eines Eingabewortes (vom PAE)
L AB	Ø bis 3	4A	80	-	-	N	N	-	-	-	61	eines Ausgabebytes (vom PAA)
L AW	Ø bis 2	52	80	-	-	N	N	-	-	-	65	eines Ausgabewortes (vom PAA)
L MB	Ø bis 63	0A	00	-	-	N	N	-	-	-	61	eines Merkerbytes
L MW	Ø bis 62	12	00	-	-	N	N	-	-	-	67	eines Merkerwortes
L DL	Ø bis 255	22	00	-	-	N	N	-	-	-	65	eines Datums (linkes Byte)
L DR	Ø bis 255	2A	00	-	-	N	N	-	-	-	66	eines Datums (rechtes Byte)
L DW	Ø bis 255	32	00	-	-	N	N	-	-	-	78	eines Datums (Wort)
L T	Ø bis 15	02	00	-	-	N	N	-	-	-	67	eines Zeitwertes
L Z	Ø bis 15	42	00	-	-	N	N	-	-	-	68	eines Zählwertes
L PB	Ø bis 5	72	00	-	-	N	N	-	-	-	116	eines Peripheriebytes der Eingaben
LC T	Ø bis 15	0C	00	-	-	N	N	-	-	-	124	eines Zeitwertes (BCD-codiert)
LC Z	Ø bis 15	4C	00	-	-	N	N	-	-	-	122	eines Zählwertes (BCD-codiert)
+ relative Adresse												
L KC	2 alphanum. Zeichen	30	10	00	00	N	N	-	-	-	63	2 ASCII-Zeichen
L KM	Bitmuster (16 Bit)	30	80	00	00	N	N	-	-	-	63	als Bitmuster
L KH	Ø bis FFFF	30	40	00	00	N	N	-	-	-	63	im Hexa-Code
L KF	Ø bis $(2^{16}-1)$	30	04	00	00	N	N	-	-	-	63	als Festpunktzahl
L KY	Ø bis 255, jedes Byte	30	20	00	00	N	N	-	-	-	63	2 Byte
L KT	Ø bis 999.3	30	02	00	00	N	N	-	-	-	63	als Zeitwert
L KZ	Ø bis 999	30	01	00	00	N	N	-	-	-	63	als Zählwert

+ Konstante (1 Wort)

Lade- und Transferfunktionen (Fortsetzung)

Operation	Parameter	Maschinencode (hexadezimal)				Ab- hängig von VKE	VKE begren- zend	Wort Anzeigen werden beein- flußt			Bearbeitungs- zeit max.µs	Funktion	
		Wort 0		Wort 1				ANZ1	ANZO	OV			
		B0	B1	B2	B3								
Transferieren													
T	EB	∅ bis 5	4B	00	-	-	N	N	-	-	-	55	zu einem Eingabebyte (im PAE)
T	EW	∅ bis 4	53	00	-	-	N	N	-	-	-	58	zu einem Eingabewort (im PAE)
T	AB	∅ bis 3	4B	80	-	-	N	N	-	-	-	55	zu einem Ausgangsbyte (im PAE)
T	AW	∅ bis 2	53	80	-	-	N	N	-	-	-	58	zu einem Ausgabewort (im PAA)
T	MB	∅ bis 63	0B	00	-	-	N	N	-	-	-	52	zu einem Merkerbyte
T	MW	∅ bis 62	13	00	-	-	N	N	-	-	-	61	zu einem Merkerwort
T	DR	∅ bis 255	2B	00	-	-	N	N	-	-	-	57	zu einem Datum (rechtes Byte)
T	DL	∅ bis 255	23	00	-	-	N	N	-	-	-	54	zu einem Datum (linkes Byte)
T	DW	∅ bis 255	33	00	-	-	N	N	-	-	-	61	zu einem Datum (Wort)
T	PB	∅ bis 3	73	00	-	-	N	N	-	-	-	90	zu einem Peripheriebyte der Ausgaben mit Nachführen des PAA
+ relative Adresse													
Vergleichsfunktionen													
Operation	Parameter	Maschinencode (hexadezimal)				Ab- hängig von VKE	VKE begren- zend	Wort Anzeigen werden beein- flußt			Bearbeitungs- zeit max.µs	Funktion	
		Wort 0		Wort 1				ANZ1	ANZO	OV			
		B0	B1	B2	B3								
Vergleich-Festpunktzahlen													
I	=F		21	80	-	-	N	N	J	J	-	87	auf gleich
	><F		21	60	-	-	N	N	J	J	-	87	auf ungleich
	>F		21	20	-	-	N	N	J	J	-	87	auf größer
	>=F		21	A0	-	-	N	N	J	J	-	87	auf größer gleich
	<F		21	40	-	-	N	N	J	J	-	87	auf kleiner
	<=F		21	C0	-	-	N	N	J	J	-	87	auf kleiner-gleich
Arithmetische Funktionen													
Operation	Parameter	Maschinencode (hexadezimal)				Ab- hängig von VKE	VKE begren- zend	Wort Anzeigen werden beein- flußt			Bearbeitungs- zeit max.µs	Funktion	
		Wort 0		Wort 1				ANZ1	ANZO	OV			
		B0	B1	B2	B3								
	+F		79	00	-	-	N	N	J	J	J	58	Festpunkt-Addition
	-F		59	00	-	-	N	N	J	J	J	54	Festpunkt-Subtraktion
Bausteinaufrufe													
Operation	Parameter	Maschinencode (hexadezimal)				Ab- hängig von VKE	VKE begren- zend	Wort Anzeigen werden beein- flußt			Bearbeitungs- zeit max.µs	Funktion	
		Wort 0		Wort 1				ANZ1	ANZO	OV			
		B0	B1	B2	B3								
	BE		65	00			N	J	-	-	-	42	Bausteinende
	BEB		05	00			J	J	-	-	-	46	Bausteinende bedingt
	BEA		65	01			N	J	-	-	-	42	Bausteinende absolut

Sonstige Funktionen

Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort Anzeigen werden beeinflusst			Bearbeitungszeit max.µs	Funktion
		Wort 0		Wort 1				ANZ1	ANZO	OV		
		B0	B1	B2	B3							
NOP 0		00	00			N	N	-	-	-	40	Nulloperation (alle Bits gelöscht)
NOP 1		FF	FF			N	N	-	-	-	40	Nulloperation (alle Bits gesetzt)
STP		70	03			N	N	-	-	-	44	Stop
												Segmentende für Programmierung in Anweisungsliste.
BLD		10	00			N	N	-	-	-	40	Dient zum Bildaufbau im Kontaktplan für das Programmiergerät
Verknüpfungsfunktionen, wortweise (ergänzende Funktionen)												
UW		41	00			N	N	X	X	-	55	UND-Verknüpfung von AKKU 1 und AKKU 2
OW		49	00			N	N	X	X	-	55	ODER-Verknüpfung von AKKU 1 und AKKU 2
XOW		51	00					X	X	-	55	Exklusiv-ODER-Verknüpfung von AKKU 1 mit AKKU 2
Umwandlungsfunktionen (ergänzende Funktionen)												
Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort Anzeigen werden beeinflusst			Bearbeitungszeit max.µs	Funktion
		Wort 0		Wort 1				ANZ1	ANZO	OV		
		B0	B1	B2	B3							
KEW		01	00	-	-	N	N	-	-	-	42	1-er-Komplementbildung (Festpunkt)
KZW		09	00	-	-	N	N	J	J	J	62	2-er-Komplementbildung (Festpunkt)
Schiebefunktionen (ergänzende Funktionen)												
Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort Anzeigen werden beeinflusst			Bearbeitungszeit max.µs	Funktion
		Wort 0		Wort 1				ANZ1	ANZO	OV		
		B0	B1	B2	B3							
SLW	Ø bis 15	61	00	-	-	N	N	J	J	-	49	Schieben links (16 bit)
SRW	Ø bis 15	69	00	-	-	N	N	J	J	-	47	Schieben rechts (16 bit)
				+ Schiebezahl								+ 10 x Schiebezahl
Sprungfunktionen (ergänzende Funktionen)												
Operation	Parameter	Maschinencode (hexadezimal)				Abhängig von VKE	VKE begrenzend	Wort Anzeigen werden beeinflusst			Bearbeitungszeit max.µs	Funktion
		Wort 0		Wort 1				ANZ1	ANZO	OV		
		B0	B1	B2	B3							
SPA =	Symboladresse	2D	00	-	-	N	N	-	-	-	55	Sprung unbedingt
SPB =	Symboladresse	FA	00	-	-	J	J	-	-	-	62	Sprung bedingt (Sprungbedingung: VKE)
SPZ =	Symboladresse	45	00	-	-	N	N	-	-	-	62	Sprung bedingt (Sprungbedingung: ANZ 1, ANZ 0)
SPN =	Symboladresse	35	00	-	-	N	N	-	-	-	62	Sprung bedingt (Sprungbedingung: ANZ 1, ANZ 0)
SPP =	Symboladresse	15	00	-	-	N	N	-	-	-	62	Sprung bedingt (Sprungbedingung: ANZ 1, ANZ 0)
SPM =	Symboladresse	25	00	-	-	N	N	-	-	-	62	Sprung bedingt (Sprungbedingung; ANZ 1, ANZ 0)
SPO =	Symboladresse	0D	00	-	-	N	N	-	-	-	57	Sprung bedingt (Sprungbedingung: OV)
				+ Sprungweite								