

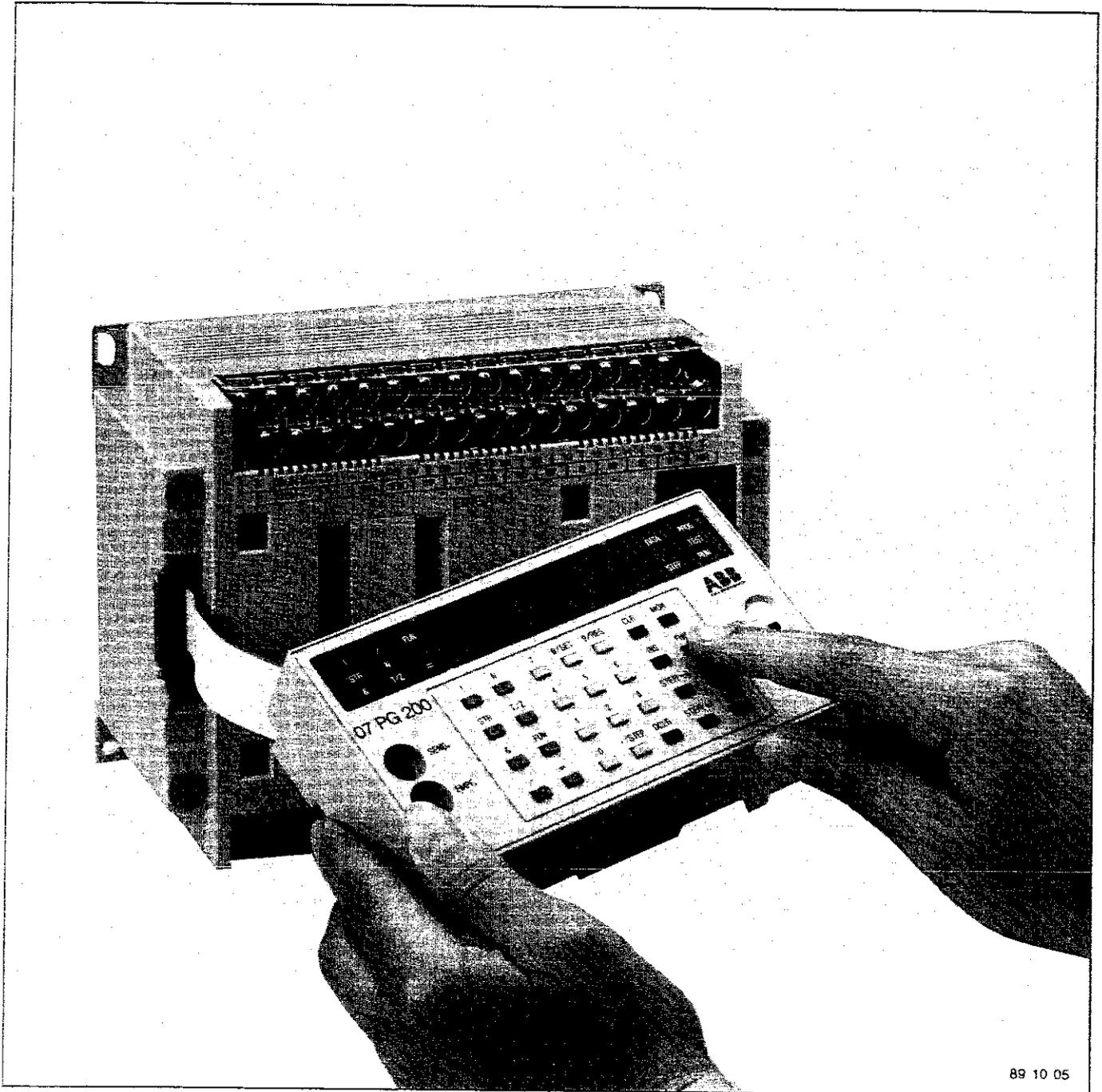
Systembeschreibung

ABB Procontic K200

Speicherprogrammierbare
Steuerungsfamilie in
Kompaktbauweise

Software

Bestell-Nr.
GATS 1313 02 R1001



89 10 05

ABB Schalt- und Steuerungstechnik GmbH

ABB
ASEA BROWN BOVERI

Vorschriften

Vorschriften für das Errichten von Anlagen:

Außer den grundlegenden Bestimmungen für das Errichten von Starkstromanlagen VDE 0100 und für die Bemessung der Kriech- und Luftstrecken VDE 0110 gilt für die Ausrüstung von Starkstromanlagen mit elektrischen Komponenten VDE 0160 in Verbindung mit VDE 0660, Teil 500. Für Steuerungen von Be- und Verarbeitungsmaschinen ist zusätzlich VDE 0113 zu beachten. Bei der Anordnung von Betätigungselementen in der Nähe berührungsgefährlicher Teile ist VDE 0106, Teil 100, maßgebend.

Der Anwender hat sicherzustellen, daß das Gerät sowie die zugehörigen Komponenten nach diesen Bestimmungen montiert werden. Die jeweils gültigen Sicherheitsbestimmungen, z. B. Unfallverhütungsvorschriften, Gesetz über technische Arbeitsmittel, sind auch für die angeschlossenen Maschinen und Anlagen einzuhalten.

ABB Procontic-Geräte sind nach der VDE-Bestimmung 0160 gebaut. Der nach Abschnitt 5.5.1 dieser VDE-Bestimmung geforderte Schutz gegen direktes Berühren ist durch den Anwender, z. B. Einbau in einen Schaltschrank, sicherzustellen. ABB Procontic-Geräte sind für den Einsatz entsprechend Isolationsgruppe A nach VDE 0110 ausgelegt. Sind bei der Anwendung erhebliche Verschmutzungen zu erwarten, dann ist der Einbau in ein Gehäuse entsprechender Schutzart vorzunehmen.

Verkaufs- und Lieferbedingungen

Die allgemeinen Lieferbedingungen für Erzeugnisse und Leistungen der Elektro-Industrie gelten in Verbindung mit den Allgemeinen Verkaufsbedingungen.

Änderungen der Konstruktionen, Abbildungen, Größen, Gewichte, Preise usw. bleiben vorbehalten.

Erfüllungsort für Lieferungen und Zahlungen sowie alleiniger Gerichtsstand ist Heidelberg.

ABB Schalt- und Steuerungstechnik GmbH

Inhaltsverzeichnis

1	Übersicht über die ABB Procontic K200-Sprache	1- 1	2.7	UND- vor ODER-Funktion	2- 3
1.1	Adreßbelegung der Ein-/Ausgänge (E/As), Merker, Sonderfunktionen, Zeit- und Zählglieder	1- 1	2.8	ODER- vor UND-Funktion	2- 3
1.2	Sprachstruktur (Semantik)	1- 2	2.9	Exklusiv-ODER-Glied	2- 3
1.3	Elemente der Sprache (Syntax)	1- 3	2.10	Äquivalenz-Glied	2- 4
1.3.1	Operationsteil	1- 3	2.11	Ansteuerung mehrerer Ausgänge (Ausgangsduplizierung)	2- 4
1.3.2	Operandenteil	1- 3	3	Speicherfunktionen	3- 1
1.3.3	Operandenadresse	1- 3	3.1	Speicher für Ausgänge oder Merker - dominierend löschend	3- 1
1.3.4	Zusammenfassung der Programmierbefehle	1- 3	3.2	Speicher für Ausgänge oder Merker dominierend setzend	3- 1
1.3.5	Sonderfunktionen	1- 4	3.3	FUN 45 (Speicher mit dynamischem Setzeingang)	3- 2
1.3.6	Verarbeitungszeiten der einzelnen Befehle	1- 5	4	Flankenauswertung	4- 1
1.4	Verknüpfungszeichen	1- 5	4.1	Wischimpuls durch 0-1-Flanke am Eingang 02 ausgelöst	4- 1
1.5	Vorzeichen	1- 6	4.2	Wischimpuls durch 1-0-Flanke am Eingang 01 ausgelöst	4- 1
1.6	Zusammenfassende Beispiele für die Anwendung der Programmierbefehle I, IN, &, &N, /, /N, =, =N	1- 6	4.3	Stromstoßrelais	4- 2
1.7	Verknüpfung paralleler und serieller Blöcke mit STR und STRN	1- 6	5	Zeitfunktionen	5- 1
1.8	Eingänge	1- 7	5.1	Einschaltverzögerer	5- 1
1.9	Ausgänge	1- 7	5.2	Ausschaltverzögerer	5- 1
1.10	Variable T - Zeitglied	1- 8	5.3	Ein-, Ausschaltverzögerer mit einem Zeitglied	5- 2
1.11	Variable Z - Zähler	1- 9	5.4	Ein-, Ausschaltverzögerer mit zwei Zeitgliedern	5- 2
1.12	Zusammenfassende Beispiele für T/Z Zeiten/Zähler	1-10	5.5	Blocker	5- 3
1.13	Merker nicht gepuffert (128 Stück)	1-10	5.6	Blocker mit Signalabbruch	5- 3
1.14	Merker gepuffert (248 Stück)	1-11	5.7	Oszillator mit einem Zeitglied	5- 4
1.15	FUN 02 (Setzen/Rücksetzen) und FUN 03 (RS-Speicher)	1-12	5.8	Oszillator mit zwei Zeitgliedern	5- 4
1.16	FUN 04 (MC-Setzen) und FUN 05 (MC-Rücksetzen)	1-13	5.9	Hinweis zur Verwendung von Zeitgliedern	5- 4
1.17	Sprungbefehle FUN 06 und FUN 07	1-13	6	Zähler	6- 1
1.18	FUN 00 (Impulserzeugung) und FUN 99 (Programmende)	1-14	6.1	Zähler	6- 1
2	Verknüpfungsfunktionen	2- 1	6.2	Dezimalzähler, vorwärts	6- 1
2.1	UND-Funktion	2- 1	6.3	Dezimalzähler, rückwärts	6- 2
2.2	ODER-Funktion	2- 1	6.4	Dezimalzähler, vorwärts/rückwärts 2 Dekaden	6- 2
2.3	Zusammenfassung von Verknüpfungen	2- 1	6.5	BCD-Zähler vorwärts	6- 4
2.4	NICHT-Funktion	2- 2	6.6	BCD-Zähler rückwärts	6- 4
2.5	NAND-Funktion	2- 2	6.7	BCD-Zähler vorwärts/rückwärts (FUN 40)	6- 5
2.6	NOR-Funktion	2- 2	6.8	BCD-Zähler - setzen	6- 6

7	Register	7- 1	10	Rechenbefehle	10- 1
7.1	Schieberegister 16 bit (FUN 47)	7- 1	10.1	FUN 11 - Addition	10- 1
7.2	Siloregister	7- 2	10.2	FUN 12 - Subtraktion	10- 2
8	Code-Umsetzer	8- 1	10.3	FUN 13 - Multiplikation	10- 2
8.1	Code-Umsetzer, BCD in 1 aus 10	8- 1	10.4	FUN 14 - Division	10- 3
8.2	Code-Umsetzer, 1 aus 10 in BCD	8- 1	10.5	FUN 24 - FUN 25, Umwandlung BNR-BCD und BCD-BNR	10- 4
8.3	Code-Umsetzer, BCD in 1 aus 8	8- 2	10.6	Status des Übertragungsmerkers (Carry Flag) bei den Rechenbefehlen	10- 4
8.4	Code-Umsetzer, 1 aus 8 in BCD	8-2			
8.5	Code-Umsetzer, DUAL in 1 aus 16	8- 3	10.7	Programmbeispiel für eine arithmetische Gleichung	10- 5
8.6	FUN 24 - FUN 25, Umwandlung Binär (Hex) in BCD und umgekehrt	8- 3	10.8	3-Punkt-Regler mit Hysterese	10- 5
9	Vergleicher/Arithmetikbefehle	9- 1	10.9	Ein- und Auslesen von Analogwerten	10- 6
9.1	Vergleicher auf Gleichheit	9- 1	10.10	Addition Konstante (BCD) zu Analog- wert - Ausgabe des neuen Wertes als Analogwert	10- 7
9.2	Größer-Kleiner-Gleich-Vergleicher (4 bit)	9- 2	10.11	Addition BCD-Wert zu Analogwert - Ausgabe des Wertes auf Analogaus- gang bzw. Ausgabe im BCD-Format	10- 7
9.3	Größer-Kleiner-Gleich Vergleicher (2 bit)	9- 2	10.12	Begrenzung eines Analogwertes auf eine obere Grenze (z. B. 8 V)	10- 8
9.4	Allgemeines über Arithmetikbefehle	9- 3	10.13	Begrenzung eines Analogwertes auf eine obere und untere Grenze (Obergrenze 8 V, Untergrenze 2 V)	10- 8
9.5	FUN 0. - Ladekonstante	9- 3			
9.6	FUN 7. - Vergleich der Konstanten auf \geq mit Inhalt des Arithmetik- Registers	9- 4	11	Meldesteuerungen	11- 1
9.7	FUN 8. - Vergleich der Konstanten auf = mit Inhalt des Arithmetik- Registers	9- 4	11.1	Meldung mit Dauerlicht	11- 1
9.8	FUN 9. - Vergleich der Konstanten auf < mit Inhalt des Arithmetik- Registers	9- 4	11.2	Neuwertmeldung mit Einfachblinklicht	11- 1
9.9	Zusammenfassendes Beispiel für die Funktionen FUN 7., FUN 8. und FUN 9.	9- 5	11.3	Neuwertmeldung mit Doppelblinklicht	11- 2
9.10	Status des Übertragungsmerkers (Carry Flag) bei den Vergleichs- befehlen	9- 6	11.4	Erstwertmeldung mit Einfach- Quittierung nach DIN 19 235	11- 3
			11.5	Erst- und Neuwertmeldung mit Zwei- fach-Quittierung nach DIN 19 235	11- 4

12	Wortbefehle 16 Bit	12- 1	14	Programmierung	14- 1
12.1	Funktion 30 (Lade Wort), Funktion 32 (Ausgabe Wort)	12- 1	14.1	Konzept der Programmierung	14- 1
12.2	Funktion 31 (Lade Wort von T/Z), Funktion 32 (Ausgabe Wort)	12- 1	14.2	Vorgehensweise bei der Programmierung	14- 2
12.3	Funktion 30 (Lade Wort), Funktion 33 (Lade (T/Z))	12- 1	14.3	Anwenderprogrammspeicher	14- 2
12.4	Funktion 36 (Lade Inhalt schneller Zähler), Funktion 32 (Ausgabe Wort)	12- 1	14.4	Applikationsbeispiele	14- 3
12.5	Funktion 31 (Lade von T/Z), Funktion 34 (Vergleiche Wort auf \leq)	12- 2	15	Formulare zur Projektierung	15- 1
12.6	Funktion 31 (Lade von T/Z), Funktion 35 (Vergleiche mit T/Z \leq)	12- 2		Eingänge, Ausgänge	15- 2
13	Sonderfunktionen	13- 1		Sonderfunktionen, Zeiten, Zähler	15- 4
13.1	Sonderfunktionen 770 bis 777	13- 1		Merker	15- 5
				Haftmerker	15- 6
				Anweisungsliste	15- 8

1 Übersicht über die ABB Procontic K200-Sprache

1.1 Adreßbelegung der Ein-/Ausgänge (E/As), Merker, Sonderfunktionen, Zeit- und Zählglieder

Bezeichnung	Nummer	Gerätetyp	
Eingänge Grundausbau- baustufe	000-007 010-013	KR220	
	010-017	KR228	
	020-027	KR240	
	030-037 040-047	KR264	
Ausgänge Grundausbau- baustufe	050-057	KR220	
	060-063	KR228	
	060-067	KR240	
	070-077	KR264	
Eingangser- weiterungen	100-107 110-117 120-127	EA240	
	130-137 140-147	EA264	
Ausgangser- weiterungen	150-157 160-167	EA240	
	170-177	EA264	
Merker nicht gepuffert 128 Stück	200-207 300-307 210-217 310-317 220-227 320-327 230-237 330-337 240-247 340-347 250-257 350-357 260-267 360-367 270-277 370-377		
	Merker gepuffert 248 Stück	400-407 600-607 410-417 610-617 420-427 620-627 430-437 630-637 440-447 640-647 450-457 650-657 460-467 660-667 470-477 670-677 500-507 700-707 510-517 710-717 520-527 720-727 530-537 730-737 540-547 740-747 550-557 750-757 560-567 760-767 570-577	

Bezeichnung	Nummer	Gerätetyp
Spezielle Funktionen 8 Stück	770-777	
Zeitglieder 40 Stück	T00-T07 T10-T17 T20-T27 T30-T37 T40-T47	
Rückwärts- zähler 24 Stück	Z50-Z57 Z60-Z67 Z70-Z77	

Hinweis: Die Maximale Ausbaustufe des Systems (80 E, 48 A) ist nur durch den Einsatz der 07 KR 264 zu erreichen. Bei Einsatz kleinerer Grundausbau-stufen reduziert sich die Zahl der Ein-/Ausgänge entsprechend.

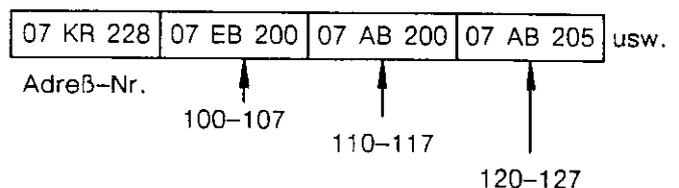
Beispiel: Grundausbaustufe 07 KR 228: 16 E, 12 A
Erweiterung 07 EA 264: 40 E, 24 A
Gesamt: 56 E, 36 A

Grundausbaustufe 07 KR 220: 12 E, 8 A
und E/A-Module, max. 40 E, 24 A
entspricht insgesamt max. 52 E, 32 A
Realisierbar mit einer Grundausbaustufe
(07 KR 220) plus 5 EB XXX und 3 AB XXX
(XXX = 200 oder 205).

Anmerkung ①:

Werden statt des Erweiterungsmoduls 07 EA 240 oder 07 EA 264 E/A-Module der Typen 07 EB 200, 07 EB 205, 07 AB 200 oder 07 AB 205 eingesetzt, so werden die Adreßnummern der Ein-/Ausgänge durch die Bestückung festgelegt.

Beispiel:



Das erste Modul neben der Grundausbaustufe belegt den Adreßbereich 100-107, das zweite Modul den Adreßbereich 110-117 usw., gleichgültig ob es sich

um ein Eingangs- oder Ausgangsmodul handelt. Nicht durch Hardware belegte Ausgangskanäle können im Programm als zusätzliche Merker angesprochen werden.

Beispiel: Bei Einsatz der Grundausbaustufe 07 KR 228 stehen die Ausgangsabbilder 064-077 und 100-177 zusätzlich als Merker (nicht gepuffert) zur Verfügung.

Bemerkung: Die Adressierung der ABB Procontic K200 erfolgt oktäl. Es sind nur Ziffern 0-7, 10-17, 20-27 usw. erlaubt.

Anmerkung ②:

Die Pufferung erfolgt über einen eingebauten Goldkondensator. Die Pufferzeit beträgt 14 Tage.

1.2 Sprachstruktur (Semantik)

Die ABB Procontic K200-Programmiersprache ist anwenderfreundlich orientiert. Sie beschreibt einen steuerungstechnischen Funktionsablauf in einer für den Anwender verständlichen Form.

Die Programmiersprache ist speziell auf Steuerungsaufgaben zugeschnitten.

Das Programm setzt sich aus Programmsätzen zusammen. Ein Programmsatz besteht aus Programmworten (Anweisungen).

Grundstruktur:

Wort	(Anweisung) Kleinste sinnvolle Untergliederung eines Programms.
------	---

Satz Wort Wort ... Wort	Ein Satz setzt sich aus mehreren Worten zusammen.
----------------------------	---

Satzstruktur:

Satz					
Abfrageteil			Zuweisungsteil		
!11	& N01	.../ 12	= 50	= 60	... = 52
Wort 1	Wort 2	... Wort n	Wort n+1	Wort n+2	... Wort n+m

Ein Satz besteht zwingend aus:

- Satzanfang "!" (WENN)
- mindestens einem Wort im Abfrageteil
- mindestens einem Wort im Zuweisungsteil (mit Zuweisungszeichen "=" (DANN))

Wortstruktur:

Wort	
Operationsteil	Operandenteil

Im Operationsteil stehen Verknüpfungs- und Vorzeichen, er ist also die Verarbeitungsanweisung für den Operanden, bzw. Aufruf von SW-Funktionen.

Im Operandenteil stehen Variablen (Operanden)-Kennzeichen und Operandenadressen, SW-Funktionsnummern und Zeit-Zählerwerte.

ABB Procontic K200-Wort		
Operations- teil	Operan- denkenn- zeichen	Operan- den- Adresse

Ein ABB Procontic K200-Wort kann aus folgenden Zeichen aufgebaut sein:

Verknüp- fungs- zeichen	Vor- zeichen
&	N

Variable	Adresse (E/A oder Merker)
	01

Zuwei- sungs- zeichen	Vor- zeichen
=	N

Variable	Adresse (E/A oder Merker)
	205

Zuwei- sungs- zeichen
=

Variable	Nr.	Wert
T	05	035

Funktion
FUN

Nr. SW- Funktion	Operan- den- Adresse
40	50

Kleinstmöglicher Satz:

Wort 1 ! ---	Wort 2 = ---
-----------------	-----------------

Ein Satz kann aus beliebig vielen Worten - begrenzt durch den Programmspeicherplatz - bestehen. Mehrfachzuweisungen sind möglich.

1.3 Elemente der Sprache (Syntax)

1.3.1 Operationsteil

Verknüpfungszeichen		Bedeutung
!	WENN	Satzanfang
&	UND	Konjunktion
/	ODER	Disjunktion
=	DANN	Zuweisung
STR	STORE	Speichern
FUN	FUNKTION	Zuweisung

Vorzeichen		
N	NICHT	Negation

1.3.3 Operandenadresse

00 – 047, 100 – 147	für Eingaben
50 – 077, 150 – 177	für Ausgaben
T00 – T47	für Zeitwerke T
T50 – T77	für Zähler Z
200 – 377	für Merker nicht gepuffert
400 – 767	für Merker gepuffert
770 – 777	spezielle Funktionen
00 – 99	Software Funktions Nr.

1.3.2 Operandenkennzeichen

Operandenkennzeichen	Bedeutung
T	Zeitwerk
Z	Zähler

1.3.4 Zusammenfassung der Programmierbefehle

	Befehl	Name	Funktion
1	!	Wenn	wird immer zu Beginn einer Anweisung geschrieben
2	! N	Wenn nicht	
3	STR	Store	Zum Zwischenspeichern von Verzweigungen
4	STR N	Store nicht	
5	&	Und	Und-Verknüpfung von Kontakten (Reihenschaltung)
6	& N	Und nicht	
7	/	Oder	Oder-Verknüpfung von Kontakten (Parallelschaltung)
8	/ N	Oder nicht	
9	& STR	Und store	Bildet die logische Summe von in Reihe geschalteten Zweigen
10	/ STR	Oder store	
11	=	Dann	Bildet die logische Summe von parallel geschalteten Zweigen
12	= N	Dann nicht	
13	FUN 0	Ladekonstante (Konst)	Verknüpfungsergebnis an Ausgang oder Merker geben
14	FUN 00	Ladekonstante (Konst)	
15	FUN 02	Impulserzeugung (IMP)	Lade Konstante nach Register
16	FUN 03	Setzen/ Rücksetzen (S/R)	Zum Erzeugen von Impulsen aus Dauersignalen; positive Flanken- auswertung (z. B. Setzen eines Speichers)
17	FUN 04	R/S – Speicher (R/S-Sp)	Befehl zum Setzen/Rücksetzen von Ausgängen und Merkern
18	FUN 05	MC-setzen (MCS)	R/S – Speicher für Ausgänge oder Merker
19	FUN 06	MC-rücksetzen (MCR)	
20	FUN 07	Befehle zur Vereinfachung komplexer Verknüpfungen	Befehle zur Vereinfachung komplexer Verknüpfungen
21	FUN 7	Anfang des Sprungbefehls (SPR-A)	Sprungbefehl auf Adresse, ab der das Programm nicht abgearbeitet werden soll
22	FUN 8	Ende des Sprungbefehls (SPR-E)	Das Ende des Programmteils wird aufgerufen, welches übersprungen wird
23	FUN 9	Register \geq Konstante (\geq Konst)	Register \geq Konstante, dann „1“-Signal nach Ausgang/Merker 1 bit (16 bit)
24	FUN 11	Register = Konstante (= Konst)	Inhalt Register = Konstante, dann „1“-Signal nach Ausgang/Merker 1 bit (16 bit)
		Register < Konstante (< Konst)	Inhalt Register < Konstante, dann „1“-Signal nach Ausgang/Merker 1 bit (16 bit)
		Addieren (ADD)	Addiere zum Inhalt des Arithmetik-Registers eine Zahl und übernehme das Ergebnis ins AR *) (16 bit)

	Befehl	Name	Funktion
25	FUN 12	Subtrahieren (SUB)	Subtrahiere vom Inhalt des AR eine Zahl und übernehme das Ergebnis ins AR (16 bit)
26	FUN 13	Multiplizieren (MULT)	Multipliziere den Inhalt des AR mit einer Zahl und übernehme das Ergebnis ins AR (16 bit)
27	FUN 14	Dividieren (DIV)	Dividiere den Inhalt des AR durch eine Zahl und übernehme das Ergebnis ins AR (16 bit)
28	FUN 22	Ausgabe Wort (=W)	Lade Wort (Wert) von AR nach Ausgänge oder Merkern (16 bit)
29	FUN 24	Bin --> BCD (BIN-BCD)	Wandle Zustand des AR von Bin in BCD um
30	FUN 25	BCD --> Bin (BCD-BIN)	Wandle Zustand des AR von BCD in Bin um
31	FUN 30	Wort laden (W LAD)	Lade Wort von E/A oder Merkern in AR (16 Bit)
32	FUN 31	T/Z laden (T/Z LAD)	Lade Wort (Wert) von T/Z in Register (16 Bit)
33	FUN 32	Ausgabe Wort (=Wc)	Lade Wort von AR nach Ausgänge oder Merkern (16 Bit)
34	FUN 33	Wort nach T/Z (=T/Z)	Lade Wort von AR nach T/Z (16 Bit)
35	FUN 34	Vergleiche Inhalt AR (W VGL)	Vergleiche AR mit E/A oder Merkern (16 Bit). Wenn der Wert der E/A oder Merker größer oder gleich dem Wert des AR ist, wird ein FLAG gesetzt. Im anderen Fall bleibt das FLAG gelöscht.
36	FUN 35	Vergleiche Inhalt AR (T/Z VGL)	Vergleiche AR mit Wert T/Z (16 Bit). Ist der Wert der Zeiten/Zähler größer oder gleich dem Wert des AR, so wird ein FLAG gesetzt. Im anderen Fall bleibt das FLAG gelöscht.
37	FUN 36	HZ-laden (HZ Zähl)	Lade Inhalt Schneller Zähler in AR (16 Bit)
38	FUN 40	V-R-Zähler (V/R-Z)	Vor/Rückwärtszähler (4 Dekaden) BCD kodiert
39	FUN 45	dyn. Speicher (Dyn.SP)	Funktion zum Aufbau von Speichern mit dynamischen Eingängen (Rücksetzpriorität)
40	FUN 47	Schieberegister (Regist.)	Schieberegister (16 Bit)
41	FUN 99	Ende	Kennzeichnung Ende Programm
42	T/Z	Zeitglied/Zähler	0 – 1 Verzögerer oder Rückwärtszähler

*) AR: Arithmetik Register

Anmerkung: Die Funktionen FUN 0 bis FUN 36 bezeichnet man als wortverarbeitende Funktionen (16 Bit-Befehle).

1.3.5 Sonderfunktionen

Softwaremäßige Sonderfunktion:

Lfd. Nr.	Nummer	Funktion
1	770	Alle Ausgänge Null setzen
2	771	Rücksetzen aller gepufferten Werte (Merker, Zeiten, Zähler)
3	772	Zyklus-Zeit Oszillator (1 Zyklus Impuls, 1 Zyklus Pause)
4	773	0,1 s Oszillator
5	774	1 s Oszillator
6	775	0,01 s Oszillator
7	776	1 min. Oszillator
8	777	Ein Impuls nach Starten des Programms (Zyklus-Zeit)

1.3.6 Verarbeitungszeiten der einzelnen Befehle:

Nr.	Bezeichnung	Zeit in μ s	
		min.	max.
1	!, IN		3
2	&, &N		3
3	/, /N		3
4	STR, STRN		5
5	&STR		4,5
6	/STR		4,5
7	=		3
8	= N		5,5
9	= T/Z	22	56
10	= T/Z N	22	56,5
11	FUN 0		34,5
12	FUN 00	7,5	14,5
13	FUN 02	5	74
14	FUN 03	11	74
15	FUN 04		5
16	FUN 05		3
17	FUN 06		3,5
18	FUN 07		1,5
19	FUN 7		36,5
20	FUN 8		35
21	FUN 9		34
22	FUN 11		190,5
23	FUN 12		202,0
24	FUN 13		529,5
25	FUN 14		629,0
26	FUN 22		241,0
27	FUN 24		149,0
28	FUN 25		101,5
29	FUN 30	153	157
30	FUN 31		19
31	FUN 32	225	227
32	FUN 33	10,5	34,5
33	FUN 34	150,5	159,0
34	FUN 35	23	30
35	FUN 36		14,5
36	FUN 40	41,5	513
37	FUN 45	11	18
38	FUN 47	38,5	146,5
39	FUN 99	169,5	365
40	Übertragungszeit im Monitorbetrieb	250	940

1.4 Verknüpfungszeichen

! WENN, Satzanfang

„WENN“ steht grundsätzlich am Satzanfang. Es leitet den Abfrageteil des Satzes ein.

& UND, Konjunktion

„UND“ steht zwischen zwei Variablen und verknüpft diese konjunktiv.

Beispiel:

Programm:

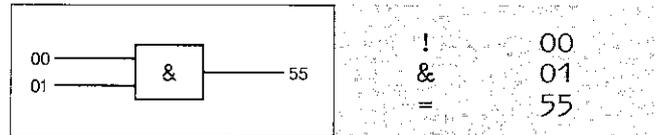


Bild 1.1

/ ODER, Disjunktion

„ODER“ steht zwischen zwei Variablen und verknüpft diese disjunktiv.

Beispiel:

Programm:

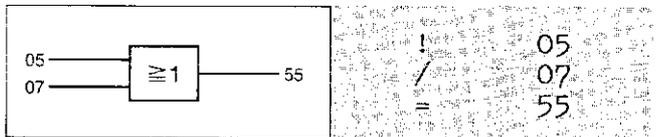
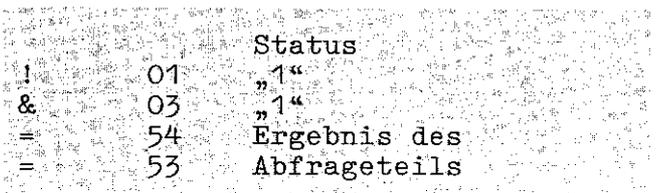


Bild 1.2

= DANN, Zuweisung

„DANN“ ist die Kennzeichnung für die Zuweisung eines Satzes. Es weist der hinter ihm stehenden Variablen das Ergebnis der logischen Verknüpfung des Abfrageteils zu. Mehrfachzuweisungen sind möglich. Dabei erhält jedes Wort des Zuweisungsteils das Ergebnis des Abfrageteils zugeordnet.

Programm-Beispiel:



1.5 Vorzeichen

N NICHT, Negation

„NICHT“ invertiert den Status einer Abfrage oder Zuweisung.

Beispiel: Programm

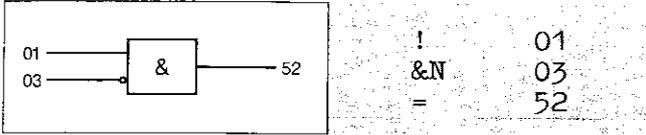


Bild 1.3

Beispiel: Programm



Bild 1.4

1.6 Zusammenfassende Beispiele für die Anwendung der Programmierbefehle !, !N, &, &N, /, /N, =, =N

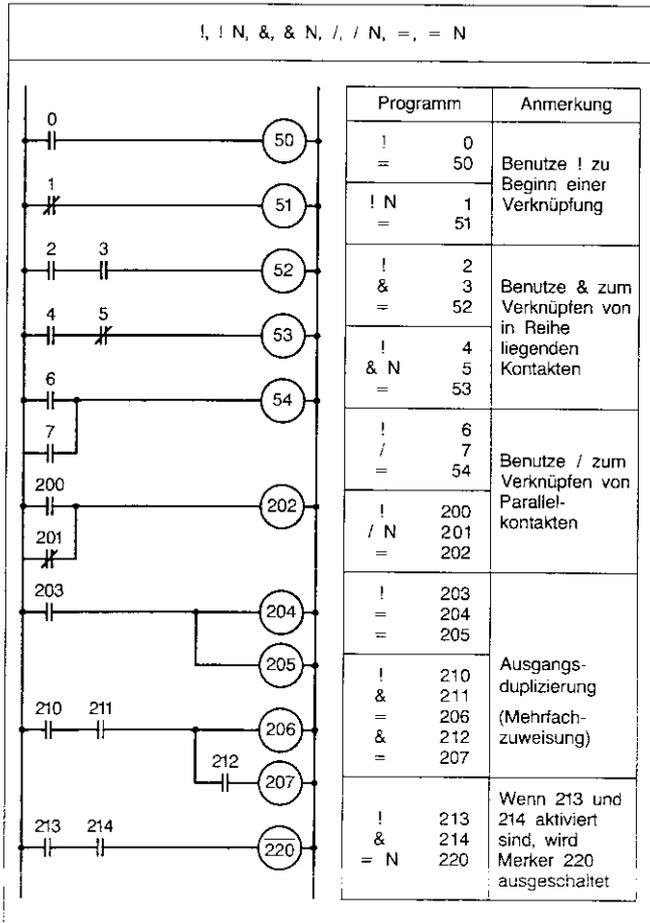


Bild 1.5 (a)

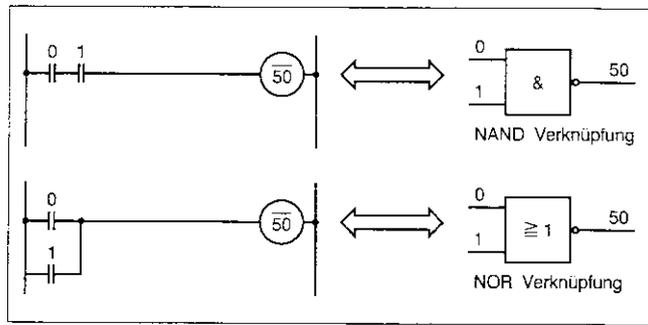


Bild 1.5

1.7 Verknüpfung paralleler und serieller Blöcke mit STR oder STRN

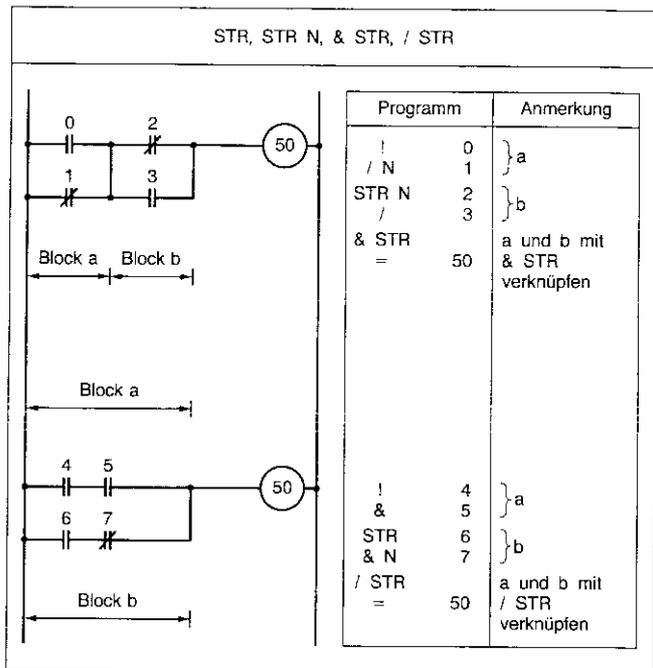


Bild 1.6

- a) Nachdem die Schaltung in Blöcke aufgeteilt ist, wird der erste Block (in obigen Beispielen jeweils Block a) wie gewohnt verknüpft. Zu Beginn der folgenden Blöcke (in obigen Beispielen folgt Block b) wird jeweils STR oder STRN geschrieben.

Wie die Blöcke miteinander verknüpft werden, ist bei dieser Eingabe noch nicht entschieden. Erst vor der Ausgabezuweisung wird durch die Eingabe von & STR bzw. / STR festgelegt, ob es sich um eine &- bzw. /-Verknüpfung der Blöcke handeln soll, d. h. bei Reihenschaltung wird mit & STR und bei Parallelschaltung mit / STR abgeschlossen.

- b) & STR oder / STR kann maximal siebenmal nacheinander geschrieben werden
- c) Ein Syntax-Fehler wird erscheinen, wenn nach einem STR oder STRN im Programm kein & STR oder / STR folgt.

Dies gilt nicht für Zähler, Zeiten, Schieberegister, Speicher FUN03 und FUN45.

1.8 Eingänge

Eingänge

Adreßbereich: 000 ... 047
100 ... 147

Mögliche Verknüpfungs- und Vorzeichenkombinationen:

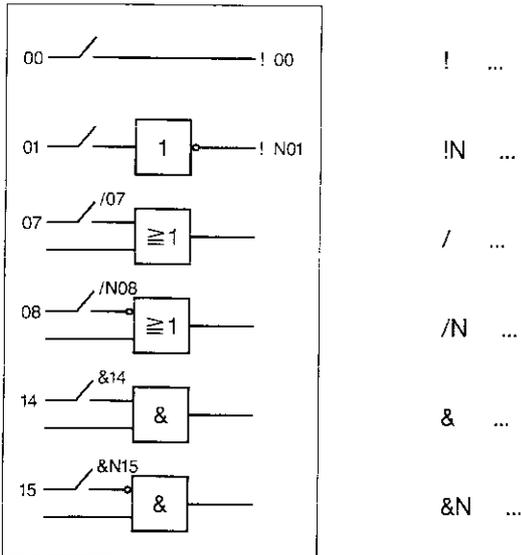


Bild 1.7: Darstellung von Eingangsverknüpfungen durch binäre Funktionen

Achtung: Ein Programmwort mit dem durch die Adreß-Nr. gekennzeichneten Eingangskanal bewirkt die Abfrage auf seinen Status („0“ oder „1“).

Die Variable Eingang darf nicht im Zuweisungsteil eines Satzes stehen (Eingänge können nur verarbeitet, nicht aber durch Programm direkt beeinflusst werden).

1.9 Ausgänge

Ausgänge

Adreßbereich: 050 ... 077
150 ... 177

Mögliche Verknüpfungs- und Vorzeichenkombinationen:

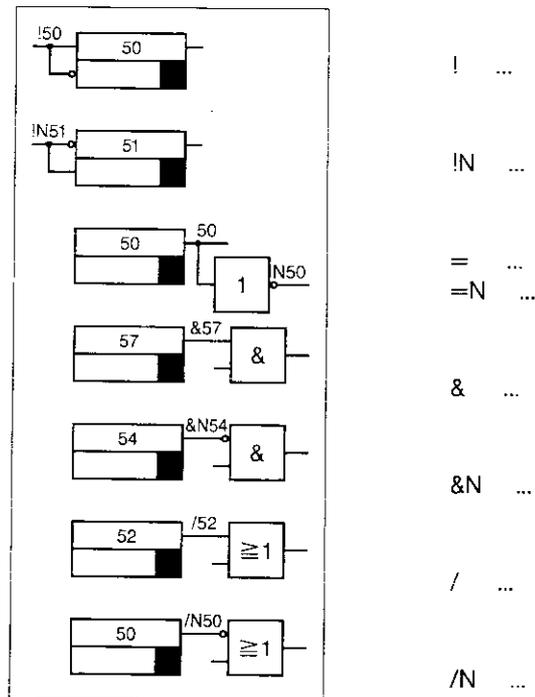


Bild 1.8: Darstellung von Ausgangsverknüpfungen durch binäre Funktionen

Achtung: Die Variable Ausgang kann im Abfrageteil und im Zuweisungsteil eines Satzes stehen.

Im Abfrageteil bewirkt ein Programmwort mit der Variablen Ausgang die Abfrage des durch die Adreß-Nr. gekennzeichneten Ausgangskanals auf seinen Status („0“ oder „1“).

Im Zuweisungsteil sind zwei Verwendungsformen möglich:

a) Nicht speichernd (Ansprache = ... , =N...)
Dem durch Adreß-Nr. gekennzeichneten Ausgangskanal wird das Ergebnis des Abfrageteils zugewiesen.

Ein nicht speichernder Ausgang darf in einem Programm nur einmal zugewiesen werden.

b) Speichernder Ausgang (Ansprache z.B. über „FUN 02“, „FUN 03“ = ...)
Bei erfüllter Satzbedingung wird der Ausgangskanal auf „1“ gesetzt und erst bei erfüllter Löschbedingung wieder zurückgesetzt.

Achtung: Ein Ausgang darf im Programm nur einmal zugewiesen werden.

1.10 Variable T - Zeitglied

T Zeitglied

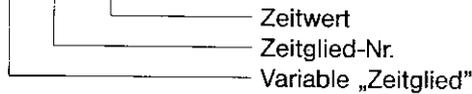
Adreßbereich Zeitwertbereich

T00 ... T47	0,01 ... 9,99 s
	0,1 ... 99,9 s
	1 ... 999 s

Die Zeitglieder (anzugsverzögert) werden mit der Variablen T und einer zweidekadischen Nummer (T00 bis T47) programmiert, d. h. 40 Zeitglieder können definiert werden, da die Adressierung der PROCONTIC K200 oktal erfolgt.

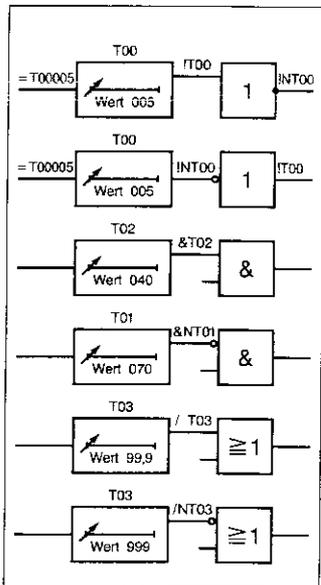
Die Programmierung der dreidekadischen Zeit erfolgt mit Dezimalpunkt wenn notwendig.

Beispiel: T 03 100



In dem gegebenen Beispiel wurde das Zeitglied Nr. 03 mit 100 s programmiert.

Mögliche Verknüpfungs- und Vorzeichenkombinationen:



!	T...
=	T...
=N	T...
IN	T...
&	T...
&N	T...
/	T...
/N	T...

Bild 1.9: Darstellung von Zeitverknüpfungen durch binäre Funktionen

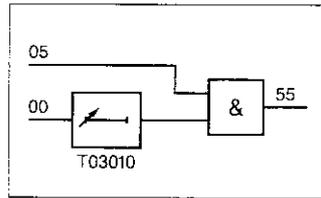
Achtung: Die Zeitglieder sind als „0-1“-Verzögerer aufgebaut, die Zeitwerte werden per Software am Gerät eingestellt.

Die Rücksetzzeit einer Zeitfunktion ist kleiner als die Verarbeitungszeit eines Wortes. Die eingestellte Zeit läuft ab, wenn „T“ im Zuweisungsteil steht und die Ansteuerungsbedingung länger als die Verzögerungszeit ansteht. Die Abfrage nach „T“ ergibt den Wert „0“, solange die Ansteuerungsbedingung für „T“ nicht erfüllt ist oder die Zeit abläuft.

Die Abfrage nach „T“ ergibt den Wert „1“, wenn die Zeit abgelaufen und die Ansteuerungsbedingung noch erfüllt ist.

Bei Unterbrechung der Ansteuerungsbedingung während des Ablaufs der Verzögerungszeit wird das Zeitglied zurückgesetzt und wird mit Erfüllung der Ansteuerbedingung erneut gestartet.

Programm



```
!      00
=T 03 010 *)
!      05
&T    03
=      55
```

Bild 1.10

*) Zeitglied 03 mit 10 s

Anwendung von Zeitwerken

Mehrfachausnutzung

Eine Mehrfachausnutzung von Zeitwerken ist möglich, wenn

- derselbe Zeitwert in einem Programm mehrmals benötigt wird
- die einzelnen Ansteuerungsbedingungen disjunktiv (ODER) auf eine Zuweisung verknüpft sind
- jede Zeitabfrage mit zugehörigen Ansteuerungsbedingungen konjunktiv (UND) verknüpft ist
- eine gleichzeitige Ansteuerung durch mehrere Bedingungen ausgeschlossen ist.

Beispiel:

```
!      00
/N     05
=T 01 050 (Zuweisung)
!      00
& T01 (Abfrage)
=      52

!N     05
& T01 (Abfrage)
=      53
```

Achtung: Mehrfaches Zuweisen von Zeitfunktionen ist nicht zulässig.

Die Einsatzmöglichkeiten von Zeitwerken können erweitert werden durch:

- Aneinanderreihen von Zeiten (Wertvergrößerung) (s. Softwaremodule)
- Verwendung als „1-0“-Verzögerer, Blocker usw. (s. Softwaremodule)

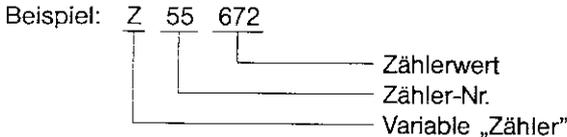
1.11 Variable Z – Zähler

Z Zähler

Adreßbereich Zählwertbereich

Z50 ... Z77 001 ... 999
Z70 ... Z77 001 ... 9999

Die Rückwärtszähler werden mit der Variablen Z und einer zweidekadischen Nummer (Z50 bis Z77) programmiert, d.h. 24 Zählglieder können definiert werden, da die Adressierung der PROCONTIC K200 oktal erfolgt. Der dreidekadische Zählwert kann im Zählbereich zwischen 0 und max. 999 programmiert werden.



In dem gegebenen Beispiel wurde der Zähler Nr. 55 mit dem Zähleranfangsstand 672 programmiert.

Mögliche Verknüpfungs- und Vorzeichenkombinationen:

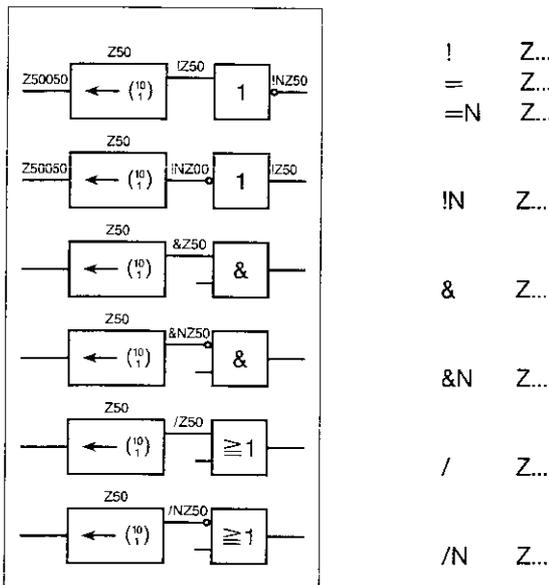


Bild 1.11: Darstellung von Zählerverknüpfungen durch binäre Funktionen

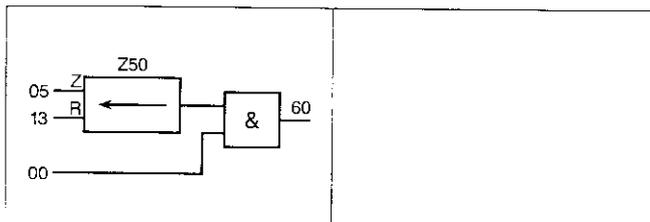


Bild 1.12

Achtung: Die Zähler sind als Rückwärtszähler aufgebaut, die Zählerwerte werden bei der Zuweisung der Zähler im Programm eingegeben. Bei Abfrage der entsprechenden Zähler wird der Ausgang oder Merker „1“, wenn der Zählerwert Null erreicht.

Der Zählerwert wird bei Spannungsausfall gespeichert. Mit dem Befehl STR... wird der Zähler gelöscht und auf den Ausgangswert gesetzt.

Beispiel:

```
! 05 (Zähleingang)
STR 13 (Rücksetzeingang)
=Z 50051 (Zähler 50, Wert 51)
! Z50 (Abfrage, ob Z50
& 00 Null ist)
= 60 (Ausgang 60)
```

Aufruf der vierdekadischen Rückwärtszähler
Die Zähler Z70 – Z77 können als Rückwärtszähler mit 4 Dekaden programmiert werden. Dabei geht man folgendermaßen vor:

Man unterschlägt bei der Programmierung die Zehnerstelle (7) und programmiert nach der Zähler-Nr. (0-7) einen Dezimalpunkt.

Beispiel: Zähler 74 soll mit dem Wert 1350 programmiert werden.

Lösung: =Z4.1350

Soll der Zähler 74 dagegen mit einer dreidekadischen Zahl programmiert werden, z.B. 256 so ist zu programmieren:

Lösung: = Z74256

Soll der Zählerausgang abgefragt werden, so wird bei beiden Beispielen mit Zähler 70 bis 77 abgefragt.

Beispiel: Zähler 75 soll mit 3560 programmiert werden. Nach Erreichen der Zahl Null, soll Ausgang 50 gesetzt werden.

Programm:

```
! 773 (Zähleingang)
STR 3 (Rücksetzeingang)
=Z 5.3560 (Zähler 75, Wert 3560)
! Z 75 (Wenn Zähler 75 = Null)
= 50 (Setze Ausgang 50)
```

1.12 Zusammenfassende Beispiele für T/Z Zeiten/Zähler

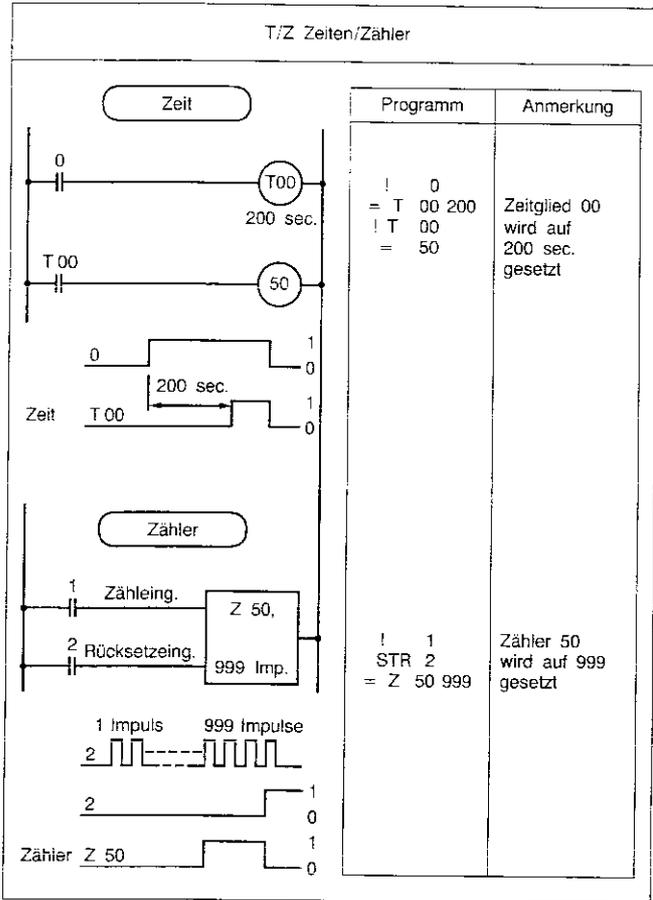


Bild 1.13

1.13 Merker nicht gepuffert (128 Stück)

Merker (nicht gepuffert)

Adreßbereich: 200 ... 377

Mögliche Verknüpfungs- und Vorzeichenkombinationen:

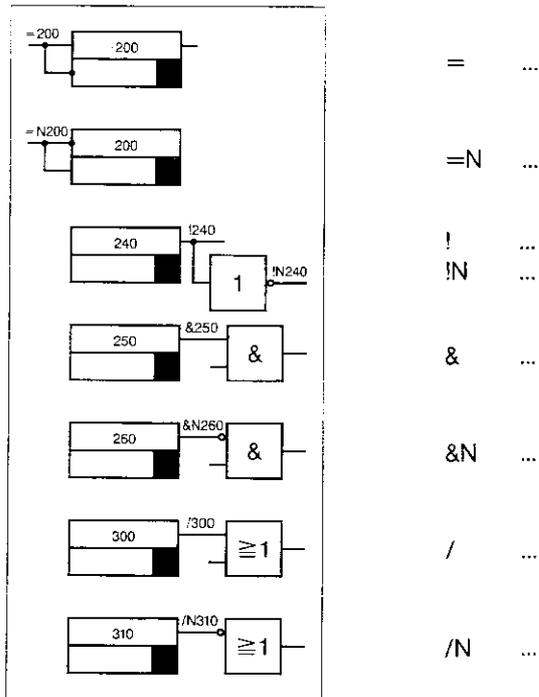


Bild 1.14: Darstellung von Merkerverbindungen durch binäre Funktionen

Merker sind Hilfsfunktionen. Sie dienen in speichernder oder nicht speichernder Form zur Zusammenfassung eines Programmteils oder zur Zustandsspeicherung (z.B. mit FUN 02, FUN 03).

Beispiel:



Im Zuweisungsteil wird der Merker 250 auf „1“-Signal gesetzt, wenn im Abfrageteil die Verknüpfungen erfüllt sind, bzw. „0“-Signal gesetzt, wenn sie nicht erfüllt sind.

Im Zuweisungsteil sind zwei Verwendungsformen möglich:

- a) Nicht speichernd (Ansprache = 200, =N 250)
 Dem durch die Adresse gekennzeichneten Merker wird der jeweilige Wert des Abfrageteils zugewiesen.

Ein nicht speichernder Merker kann in einem Programm nicht mehrfach zugewiesen werden. Etwaige Mehrfachzuweisungen müssen in einer ODER-Verknüpfung zusammengefaßt werden.

Beispiel:

```
!      05
/      02
/N     20
=     200
```

- b) Speichernd (Ansprache mit FUN 02, FUN 03)

Bei erfüllter Setzbedingung wird der Merker auf „1“-Signal gesetzt und erst bei erfüllter Löschbedingung wieder zurückgesetzt. Speichernde Merker dürfen im Programm nicht mehrfach zugewiesen werden.

Beispiele:

```
!      00
&     T02
FUN   02
=     200

!      04
&     06
FUN   02
=N    200
```

Wenn im Abfrageteil die Bedingungen erfüllt sind, wird durch FUN 02 der Merker 200 im Zuweisungsteil auf „1“-Signal gesetzt, (1. Satz). Im 2. Satz wird er zurückgesetzt.

Achtung: Ein Merker (z.B. 205) darf im Programm nur in einer der beiden Verwendungsformen vorkommen.

1.14 Merker gepuffert (248 Stück)

Merker gepuffert

Adreßbereich: 400 ... 767

Mögliche Verknüpfungs- und Vorzeichenkombinationen:

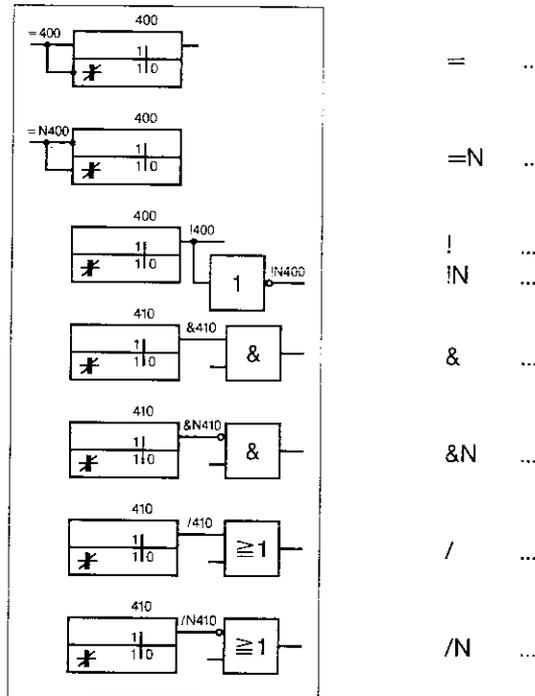


Bild 1.15: Darstellung gepufferter Merker-Verknüpfungen durch binäre Funktionen

Merker sind Hilfsfunktionen. Sie dienen in speichernder (z.B. mit FUN 02, FUN 03) oder nicht speichernder Form zur Zusammenfassung eines Programmteils oder zur Zustandsspeicherung. Gepufferte Merker behalten ihre Information auch nach Speisespannungsausfall oder Abschaltung der Speisespannung.

Beispiel:

```
!      00      (Setzeingang)
STR    02      (Rücksetzeingang)
FUN    03
=     400
!     400
=     50
```

Das Beispiel stellt einen dominierend rücksetzenden Speicher mit Haftverhalten dar. Wenn Eingang 00, dann setze Haftmerker 400, dann setze Ausgang 50. Wenn Eingang 02, dann rücksetze 400 und 50.

Im Zuweisungsteil sind 2 Verwendungsformen möglich:

- a) Nicht speichernd (Ansprache mit = 400, =N 400)
 Dem durch die Adresse gekennzeichneten gepufferten Merker wird der jeweilige Wert des Abfrageteils zugewiesen.
 Ein nicht speichernder Merker kann in einem Programm nur einmal zugewiesen werden. Er kann aber beliebig oft abgefragt werden.
 Etwaige Mehrfachzuweisungen müssen in einer ODER-Verknüpfung zusammengefaßt werden.

Beispiel:

```
!      05
/      200
/N     50
=     400
```

- b) Speichernd (Ansprache mit FUN 02, FUN 03)
 Bei erfüllter Setzbedingung wird der Merker auf „1“-Signal gesetzt und erst bei erfüllter Löschbedingung wieder zurückgesetzt. Speichernde, gepufferte Merker dürfen im Programm nicht mehrfach zugewiesen werden.

Beispiel:

```
!      00 (Setzeingang)
STR    02 (Rücksetzeingang)
FUN    03 (Funktion Speicher,
           dominierend rücksetzend)
=     400 (Merker 400)
```

Achtung: Ein gepufferter Merker (z.B. 405) darf im Programm nur in einer der beiden Verwendungsformen vorkommen.

1.15 FUN 02 (Setzen/Rücksetzen) und FUN 03 (RS-Speicher)

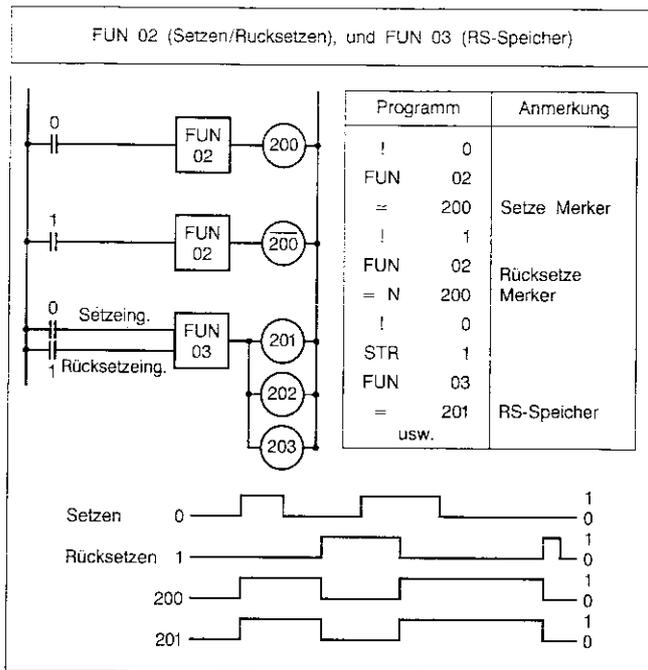


Bild 1.16

- a) FUN 02 ist zum setzen oder rücksetzen eines Ausgangs oder Merkers. Anhand des Ablaufs (siehe Bild) soll dies erklärt werden.
- Wenn an Eingang 0 „1“-Signal ansteht, wird Ausgang 200 gesetzt.
 - Liegt an Eingang 1 „0“-Signal, so geschieht nichts, d.h. der Ausgang bleibt gesetzt.
 - Liegt an Eingang 0 „0“-Signal, so geschieht ebenfalls nichts.
 - Wenn an Eingang 1 ein „1“-Signal ansteht, so wird Ausgang 200 rückgesetzt.

b) FUN 03 ist ein RS-Speicher.

Achtung: Die Ausgänge können bei oben aufgeführten Funktionen dupliziert werden.

FUN 02 und FUN 03 haben Rücksetzpriorität

Programm mit WENN – DANN Struktur

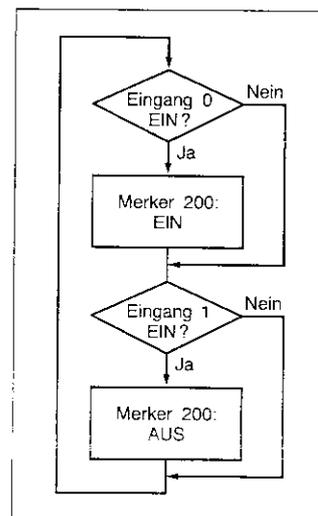


Bild 1.17

1.16 FUN 04 (MC-Setzen) und FUN 05 (MC-Rücksetzen)

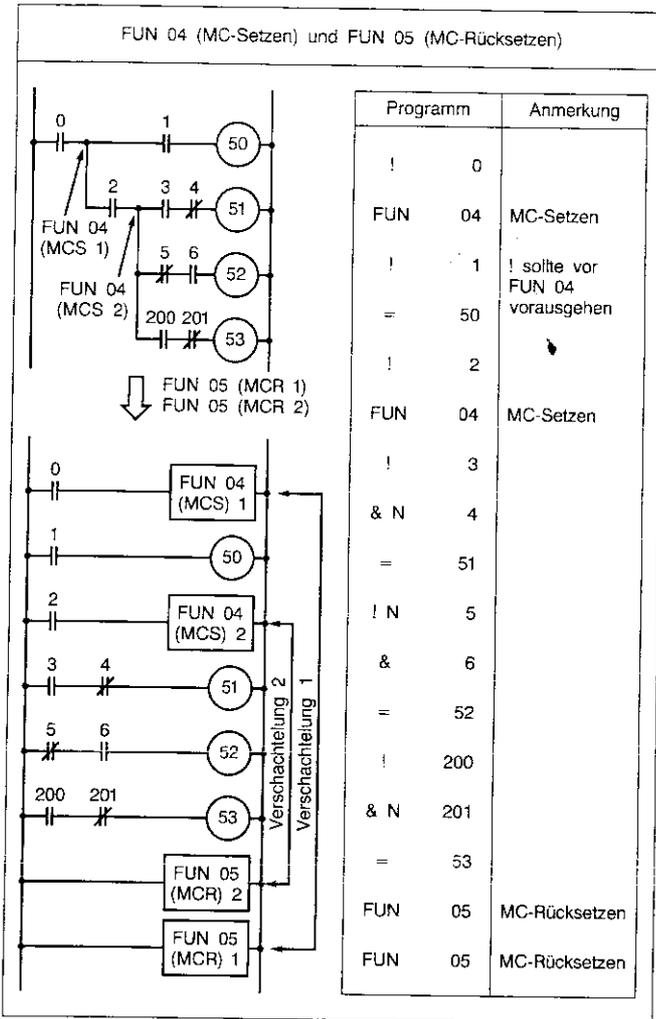


Bild 1.18

- a) FUN 04 und FUN 05 müssen paarweise eingesetzt werden, ansonsten antwortet das System mit Syntax-Fehler.
- b) Die maximale Verschachtelungstiefe ist 3. Bei einer Verschachtelung größer als 3 wird ebenfalls mit einem Syntax-Fehler geantwortet.

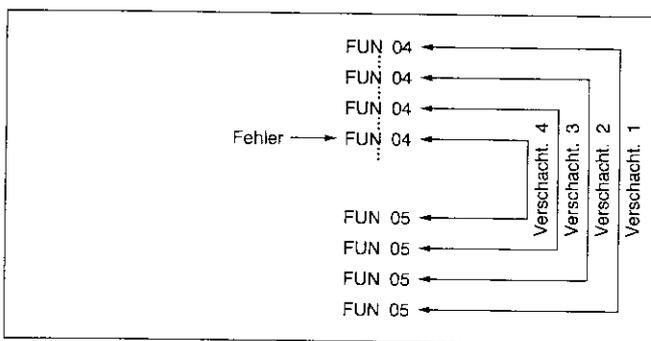


Bild 1.19

1.17 Sprungbefehle FUN 06 und FUN 07

FUN 06 Sprung nach Adresse

FUN 06 steht im Zuweisungsteil eines Satzes. Bei erfüllter Abfragebedingung werden alle hinter FUN 06 aufgeführten Programmschritte bis zur Funktion FUN 07 nicht bearbeitet. Bei nicht erfüllter Abfragebedingung werden die zwischen FUN 06 und FUN 07 enthaltenen Programmschritte wie normal abgearbeitet.

Achtung: Alle zwischen FUN 06 und FUN 07 zugewiesenen Variablen werden nicht mehr verändert, wenn die Abfragebedingung für FUN 06 erfüllt ist, d.h. alle bei Wegfall der Abfragebedingung gesetzten Variablen bleiben gesetzt.

FUN 07 Sprung Ende

FUN 07 steht allein, ohne weitere Verknüpfung oder Zuweisung.

Ein mit FUN 06 eingeleiteter Sprungbefehl muß mit FUN 07 abgeschlossen werden.

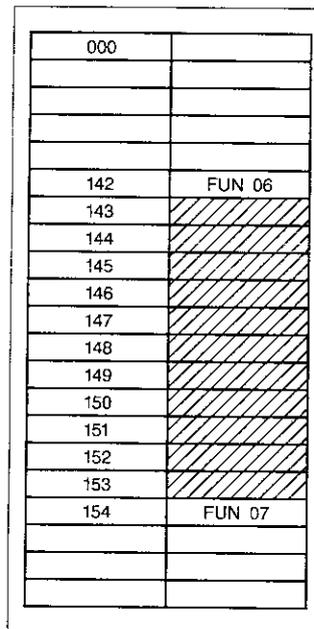


Bild 1.20

Beispiel:

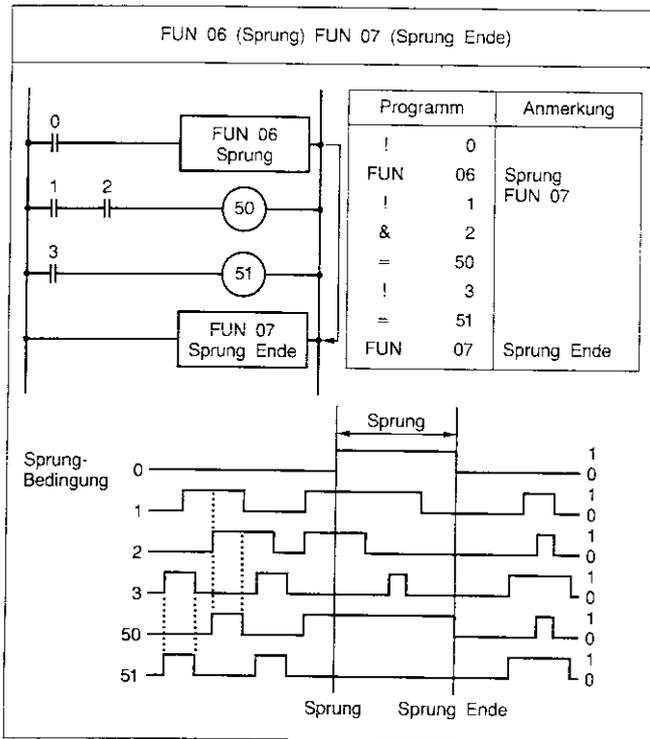


Bild 1.21

a) FUN06 und FUN07 müssen immer paarweise benutzt werden, ansonsten entsteht ein Syntax-Fehler.

Eine Verschachtelung in einem Programmteil ist zu programmieren, wie im Bild dargestellt.

b) Wenn der Sprungbefehl aktiviert ist, wird das Programm zwischen FUN06 und FUN07 nicht ausgeführt und die Ausgänge oder Merker bleiben in dem Zustand, in dem sie vor Aktivierung des Sprungbefehls waren.

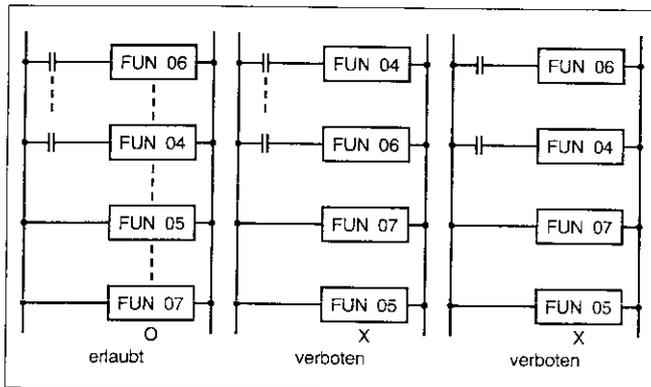


Bild 1.22

1.18 FUN 00 (Impulserzeugung) und FUN 99 (Programmende)

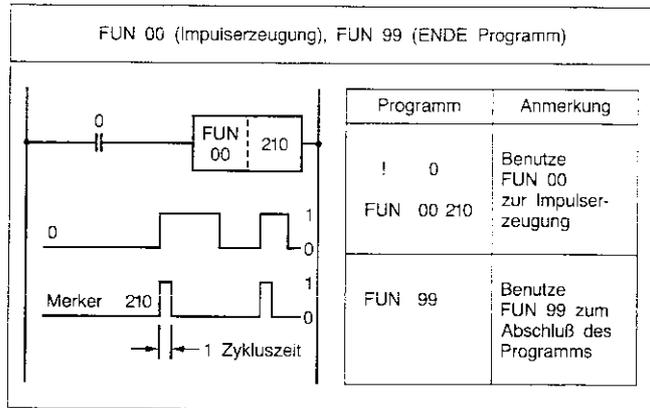


Bild 1.23

a) Wenn aus Dauersignalen Impulse erzeugt werden sollen (z.B. zum Setzen eines Speichers), so wird dies mit FUN00 realisiert.

b) FUN99 wird zum Kennzeichnen des Programmendes eingesetzt. Das Programmende braucht aber nicht programmiert zu werden.

FUN99 kann an beliebiger Stelle im Programm eingefügt werden, um z.B. bei der Inbetriebnahme Programmteile auszutesten. Die Abarbeitung des Programmes wird dadurch beschleunigt, da der nach FUN99 folgende Programmteil nicht bearbeitet wird (s. Bild).

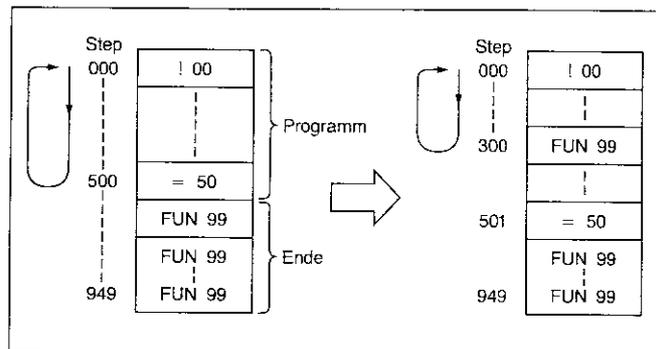


Bild 1.24

2 Verknüpfungsfunktionen

2.1 UND-Funktion

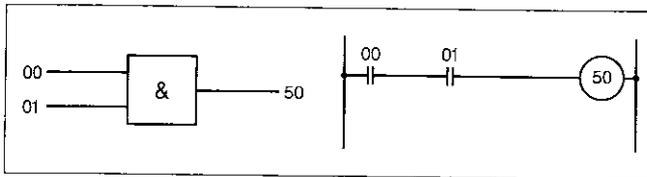


Bild 2.1

!	00	Satzanfang	Eingang 00 = „1“?
&	01	UND	Eingang 01 = „1“?
=	50	Zuweisung	Ausgang 50 = „1“

Bei diesem Beispiel werden die Eingänge 00 und 01 auf Signalzustand „1“ abgefragt und nach UND verknüpft. Das Ergebnis dieser Abfrage wird dem Ausgang 50 zugewiesen.

00	01	Ausgang 50
0	0	0
0	1	0
1	0	0
1	1	1

2.2 ODER-Funktion

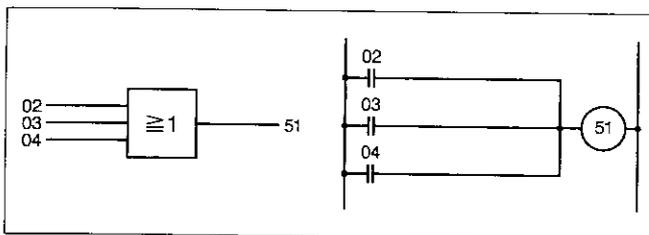


Bild 2.2

!	02	Satzanfang	Eingang 02 = „1“?
/	03	ODER	Eingang 03 = „1“?
/	04	ODER	Eingang 04 = „1“?
=	51	Zuweisung	Ausgang 51 = „1“

Bei diesem Beispiel werden die Eingänge 02 bis 04 auf Signalzustand „1“ abgefragt und nach ODER verknüpft. Das Ergebnis dieser Abfrage wird dem Ausgang 51 zugewiesen.

02	03	04	Ausgang 51
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

2.3 Zusammenfassung von Verknüpfungen

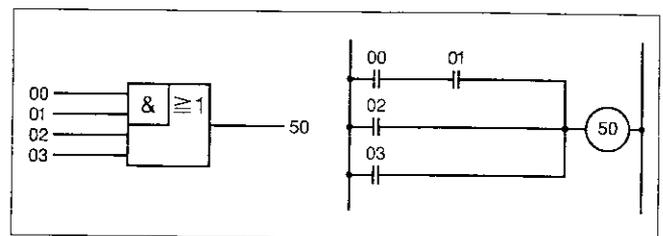


Bild 2.3

!	00	Satzanfang	Eingang 00 = „1“?
&	01	UND	Eingang 01 = „1“?
/	02	ODER	Eingang 02 = „1“?
/	03	ODER	Eingang 03 = „1“?
=	50	Zuweisung	Ausgang 50 = „1“

00	01	02	03	Ausgang 50
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Achtung: Wenn in den parallelen Zweigen (z.B. 02) mehrere Kontakte in Reihe liegen, muß mit dem STR Befehlen verknüpft werden.

2.4 NICHT-Funktion

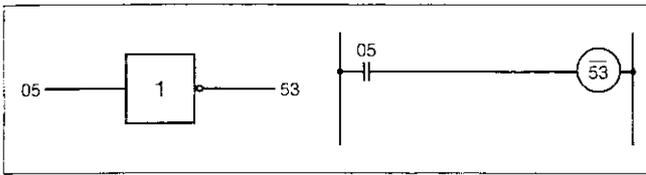


Bild 2.4

!	05	Satzanfang	Eingang 05 = „1“?
=N	53	Zuweisung	Ausgang 53 = „0“

05	Ausgang 53
1	0
0	1

Alle bisher beschriebenen Abfragen liefern den Status „1“ bei „1“-Signal und den Status „0“ bei „0“-Signal an den Eingängen. Diese Abfragen werden in einem Relais-Schaltplan „Schließer“ genannt.

Es werden aber in allen heutigen Steuerungen Abfragen benötigt, die den Status „1“ bei „0“-Signal an den Eingängen bzw. den Status „0“ bei „1“-Signal an den Eingängen liefern. Diese Abfragen entsprechen bei dem Relais-Ersatzschaltbild der Funktion „Öffner“. In der Programmiersprache übernimmt das Zeichen „N“ diese Funktion.

Anstelle von

!	05
=N	53

kann mit der gleichen Bedeutung

!N	05
=	53

geschrieben werden (De Morgan).

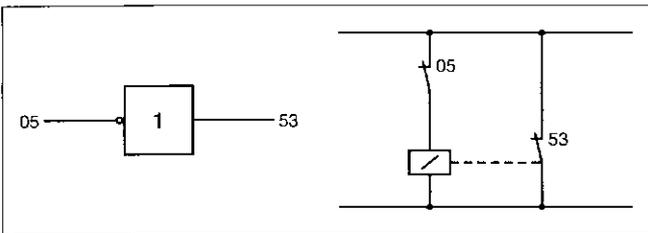


Bild 2.5

2.5 NAND-Funktion

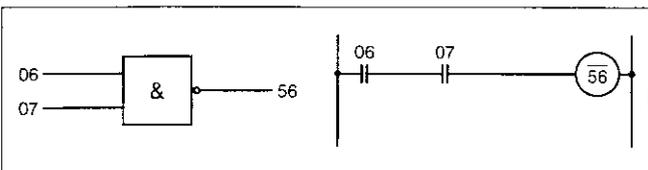


Bild 2.6

!	06	Satzanfang	Eingang 06 = „1“?
&	07	UND	Eingang 07 = „1“?
=N	56	Zuweisung	Ausgang 56 = „0“

Bei diesem Beispiel werden die Eingänge 06 und 07 auf Signalzustand „1“ abgefragt und nach UND verknüpft. Das Ergebnis dieser Abfrage wird dem Ausgang 56 NEGIERT zugewiesen.

06	07	Ausgang 56
0	0	1
0	1	1
1	0	1
1	1	0

Anstelle von

!	06
&	07
=N	56

kann mit der gleichen Bedeutung

!N	06
/N	07
=	56

geschrieben werden (De Morgan).

2.6 NOR-Funktion

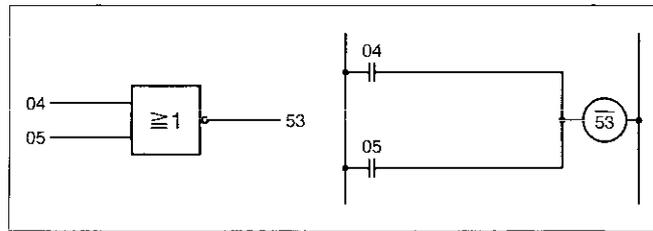


Bild 2.7

!	04	Satzanfang	Eingang 04 = „1“?
/	05	ODER	Eingang 05 = „1“?
=N	53	Zuweisung	Ausgang 53 = „0“

Bei diesem Beispiel werden die Eingänge 04 und 05 auf Signalzustand „1“ abgefragt und nach ODER verknüpft. Das Ergebnis dieser Abfrage wird dem Ausgang 53 NEGIERT zugewiesen.

04	05	Ausgang 53
0	0	1
0	1	0
1	0	0
1	1	0

Anstelle von

!	04
/	05
=N	53

kann mit der gleichen Bedeutung

!N	04
&N	05
=	53

geschrieben werden (De Morgan).

2.7 UND- vor ODER-Funktion

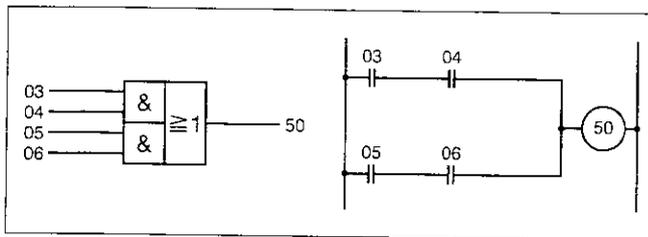


Bild 2.8

!	03	Satzanfang	Eingang 03 = „1“?
&	04	UND	Eingang 04 = „1“?
STR	05	ODER	Eingang 05 = „1“?
&	06	UND	Eingang 06 = „1“?
/	STR	/STR	Verknüpfung nach ODER
=	50	Zuweisung	Ausgang 50 = „1“

Diese UND- vor ODER-Funktion (disjunktive Verknüpfung) wird mit dem STR-Befehl eingeleitet (zwischen gespeichert). Vor der Ausgangszuweisung wird mit /STR nach ODER verknüpft.

03	04	05	06	Ausgang 50
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	1
0	0	1	0	0
1	0	1	0	0
0	1	1	0	0
1	1	1	0	1
0	0	0	1	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	1

2.8 ODER- vor UND-Funktion

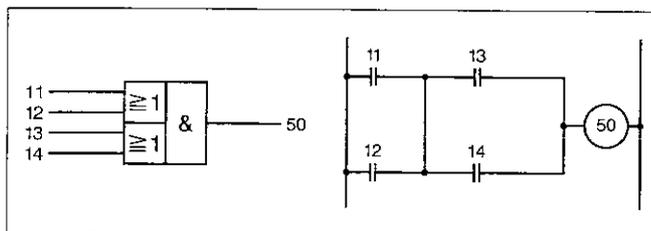


Bild 2.9

!	11	Satzanfang	Eingang 11 = „1“?
/	12	ODER	Eingang 12 = „1“?
STR	13	STR	Eingang 13 = „1“?
/	14	ODER	Eingang 14 = „1“?
&	STR	& STR	Verknüpfung nach UND
=	50	Zuweisung	Ausgang 50 = „1“

Die ODER- vor UND-Funktion (konjunktive Verknüpfung) wird mit dem STR-Befehl eingeleitet (zwischen gespeichert). Vor der Ausgangszuweisung wird mit & STR nach UND verknüpft.

11	12	13	14	Ausgang 50
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	0
0	0	1	0	0
1	0	1	0	1
0	1	1	0	1
1	1	1	0	1
0	0	0	1	0
1	0	0	1	1
0	1	0	1	1
1	1	0	1	1
0	0	1	1	0
1	0	1	1	1
0	1	1	1	1
1	1	1	1	1

2.9 Exklusiv-ODER-Glied

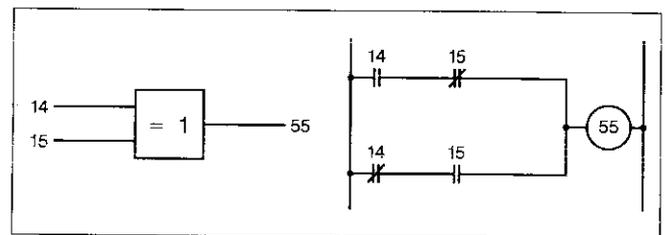


Bild 2.10

!	14	Satzanfang	Eingang 14 = „1“?
&N	15	UND	Eingang 15 = „0“?
STRN	14	ODER	Eingang 14 = „0“?
&	15	UND	Eingang 15 = „1“?
/	STR	/STR	Verknüpfung nach ODER
=	55	Zuweisung	Ausgang 55 = „1“

Der Ausgang 55 wird dann auf „1“-Signal gesetzt, wenn entweder Eingang 14 ein „1“-Signal und Eingang 15 ein „0“-Signal oder Eingang 14 ein „0“-Signal und Eingang 15 ein „1“-Signal führt.

14	15	Ausgang 55
0	0	0
1	0	1
0	1	1
1	1	0

2.10 Äquivalenz-Glied

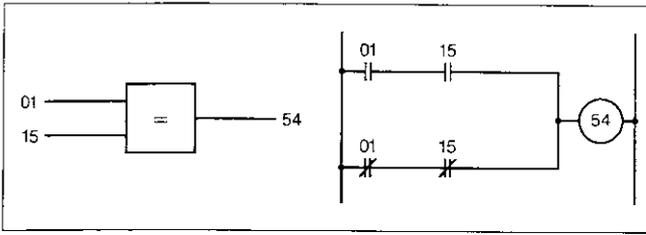


Bild 2.11

!	01	Satzanfang	Eingang 01 = „1“?
&	15	UND	Eingang 15 = „1“?
STRN	01	ODER	Eingang 01 = „0“?
&N	15	UND	Eingang 15 = „0“?
/	STR	/STR	Verknüpfung nach ODER
=	54	Zuweisung	Ausgang 54 = „1“

Der Ausgang 54 führt dann ein „1“-Signal, wenn am Eingang 01 ein „1“-Signal und am Eingang 15 ein „1“-Signal oder am Eingang 01 ein „0“-Signal und am Eingang 15 ein „0“-Signal ansteht.

01	15	Ausgang 54
0	0	1
1	0	0
0	1	0
1	1	1

2.11 Ansteuerung mehrerer Ausgänge (Ausgangsduplizierung)

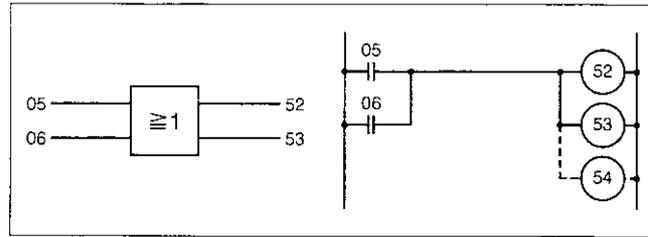


Bild 2.12

!	05	Satzanfang	Eingang 05 = „1“?
/	06	ODER	Eingang 06 = „1“?
=	52	Zuweisung 1	Eingang 52 = „1“?
=	53	Zuweisung 2	Ausgang 53 = „1“

Es können beliebig viele Zuweisungen (max. Anzahl der Zuweisungen entspricht der Anzahl der Ausgänge) negiert oder wahr geschrieben werden. Die Zuweisungen beziehen sich immer auf das Ergebnis der Signalabfrage vor dem ersten Zuweisungszeichen.

05	06	Ausgänge	
		52	53
0	0	0	0
1	0	1	1
0	1	1	1
1	1	1	1

3 Speicherfunktionen

3.1 Speicher für Ausgänge oder Merker – dominierend löschend

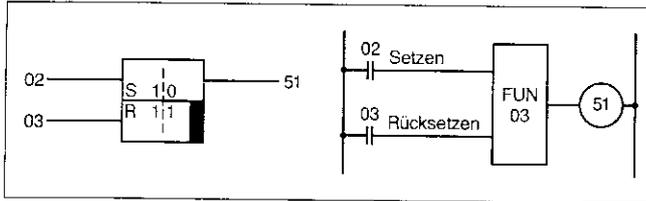


Bild 3.1

Programm:

I	02
STR	03
FUN	03
=	51

Wenn der Eingang 02 „1“-Signal und der Eingang 03 „0“-Signal hat, wird der Ausgang 51 durch FUN 03 gesetzt. Der Ausgang 51 bleibt auch dann auf „1“ gesetzt, wenn der Eingang 02 wieder „0“-Signal führt.

Der Ausgangsspeicher wird erst dann gelöscht, wenn Eingang 03 „1“-Signal hat.

Haben beide Eingänge (02 und 03) „1“-Signal, wird der Speicher gelöscht (dominierend löschend).

02	03	Ausgang 51
0	0	Vorzustand
0	1	0
1	0	1
1	1	0

Werden Merker mit Adreß-Nummern von 400 bis 767 benutzt, so haben die Speicher Haftverhalten, d. h. die Information der Merker bleibt auch nach Speisespannungsausfall oder Abschalten der Speisespannung erhalten.

3.2 Speicher für Ausgänge oder Merker – dominierend setzend

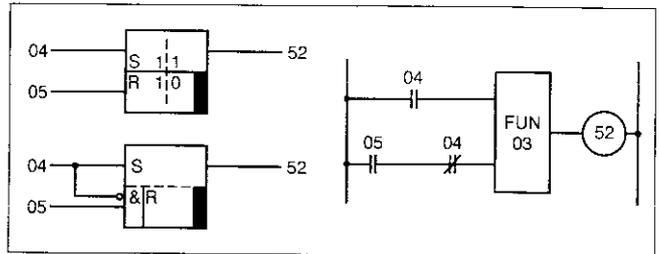


Bild 3.2

Programm:

I	04
STR	05
&N	04
FUN	03
=	52

Wenn der Eingang 04 „1“-Signal hat, wird der Ausgang 52 durch FUN 03 speichernd gesetzt. Hat der Eingang 05 „0“-Signal, so bleibt Ausgang 52 auf „1“ gesetzt, auch wenn Eingang 04 wieder auf „0“-Signal gesetzt wird.

Wenn der Eingang 05 „1“-Signal und der Eingang 04 „0“-Signal hat, wird der Ausgang 52 gelöscht.

Haben beide Eingänge (04 und 05) „1“-Signal, wird der Speicher gesetzt (dominierend setzend).

05	04	Ausgang 52
0	0	Vorzustand
0	1	1
1	0	0
1	1	1

Werden Merker mit Adreß-Nummern von 400 bis 767 benutzt, so haben die Speicher Haftverhalten.

3.3 FUN 45 (Speicher mit dynamischem Setzeingang)

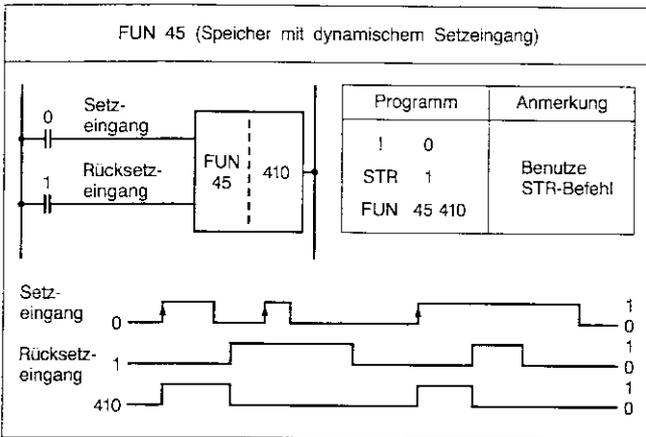


Bild 3.3

- a) FUN 45 ist eine Funktion zum Aufbau von Speichern mit dynamischem Setzeingang (mit den Merkern 200 bis 377 Speicher, mit Merkern 400 bis 767 Haftspeicher, jeweils mit dynamischen Eingängen). Die Funktion FUN 45 kann auch direkt an die Ausgänge gegeben werden (kein Haftverhalten), z. B. FUN 45 50.
- b) Der Unterschied zu einer normalen Selbsthalteschaltung soll an nachfolgendem Bild verdeutlicht werden.

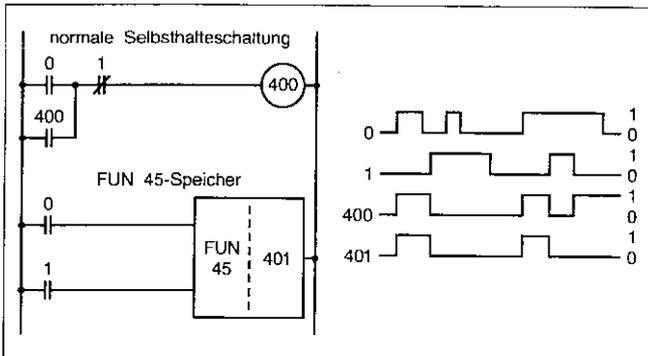


Bild 3.4

4 Flankenbewertung

4.1 Wischimpuls durch 0-1-Flanke am Eingang 02 ausgelöst

Programm:

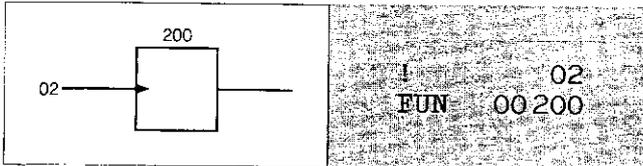


Bild 4.1

Nach erkennen der „0-1“-Flanke am Eingang 02 wird der Merker 200 für die Dauer eines Programmzykluses gesetzt.

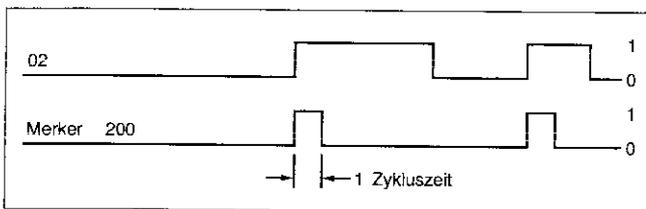


Bild 4.2

4.2 Wischimpuls durch 1-0-Flanke am Eingang 01 ausgelöst

Programm:

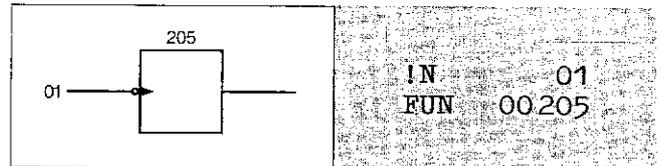


Bild 4.3

Nach erkennen der „1-0“-Flanke am Eingang 01 wird der Merker 205 für die Dauer eines Programmzykluses gesetzt.

4.3 Stromstoßrelais

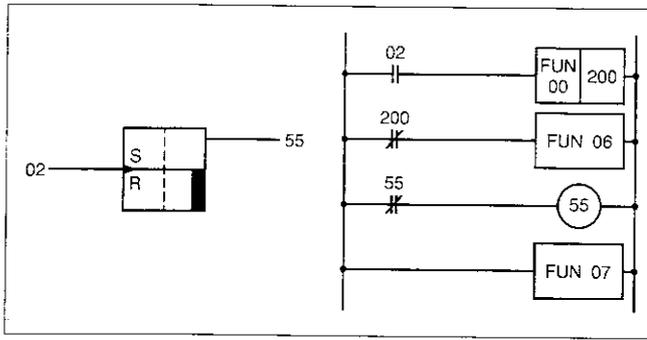


Bild 4.4

Programm:

```

I      02
FUN 00 200
IN     200
FUN    06
!N     55
-      55
FUN    07
    
```

Wenn am Eingang 02 das Signal von „0“ nach „1“ wechselt, wird durch die Funktion 00 aus diesem Dauersignal ein Impuls mit einer Breite der Zykluszeit gebildet. Ausgang 55 wird gesetzt. Beim nächsten Impuls wird er wieder gelöscht, usw. Dabei ist immer der „0-1“ Übergang des Eingangssignals für die Änderung verantwortlich.

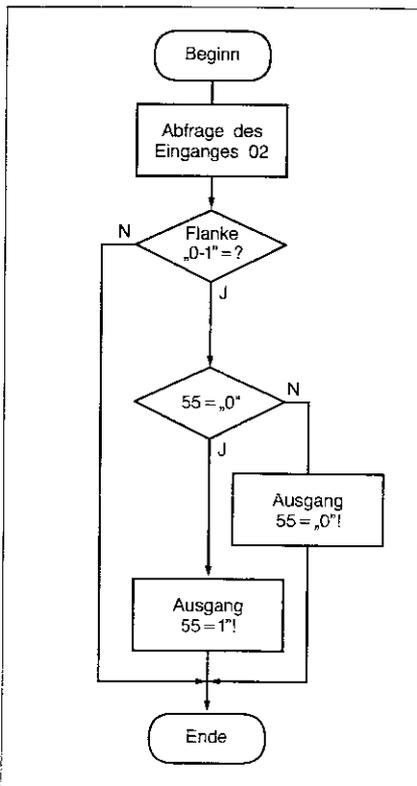


Bild 4.5

5 Zeitfunktionen

5.1 Einschaltverzögerer

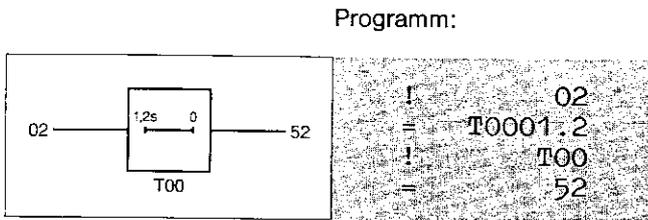


Bild 5.1

Bei der Einschaltverzögerung läuft die Zeit dann an, wenn der Eingang 02 „1“-Signal führt. Der Ausgang 52 wird gesetzt, wenn die Zeit abgelaufen ist und das Eingangssignal noch ansteht. Dies geschieht über die Abfrage der abgelaufenen Zeit.

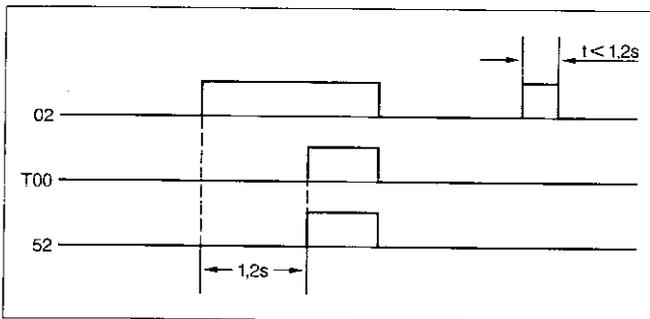


Bild 5.2

Steht das Eingangssignal kleiner der programmierten Verzögerungszeit an, so bleibt Ausgang 52 gelöscht.

5.2 Ausschaltverzögerer

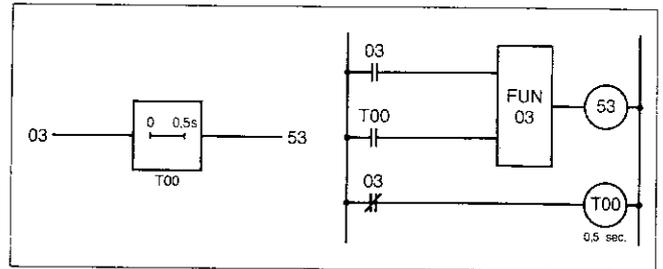
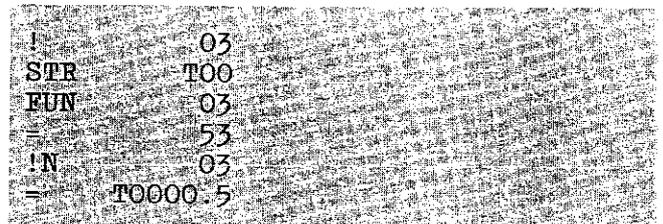


Bild 5.3

Programm:



Bei der Ausschaltverzögerung läuft die Zeit dann an, wenn der Eingang 03 „0“-Signal führt. Der Ausgang 53 wird sofort bei „1“-Signal an Eingang 03 gesetzt. Nach Ablauf der Zeit wird der Ausgang 53 zurückgesetzt.

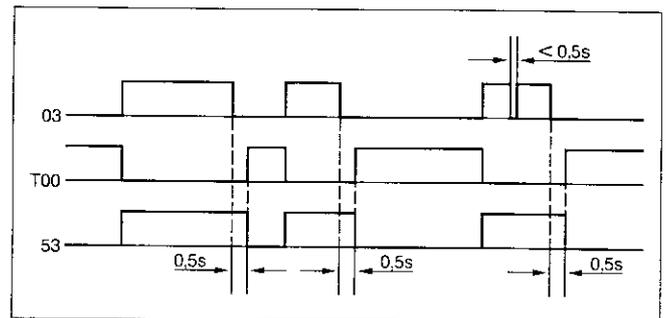


Bild 5.4

5.3 Ein-, Ausschaltverzögerer mit einem Zeitglied

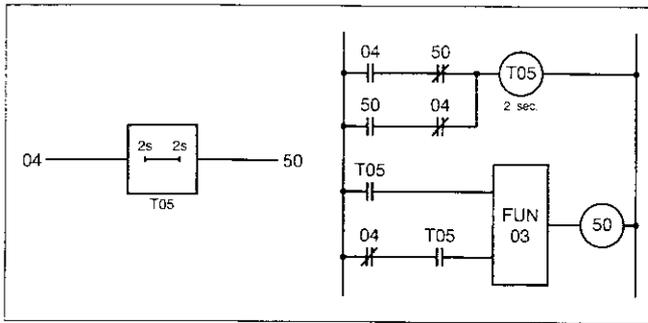


Bild 5.5

Programm:

```

!      04
&N    50
STR    50
&N    04
/      STR
=      T05 002
!      T05
STRN   04
&      T05
FUN    03
=      50
    
```

Mit einem „1“-Signal am Eingang 04 läuft die Einschaltverzögerungszeit T05 an. Nach Ablauf dieser Zeit wird der Ausgang 50 gesetzt.

Wird aus dem „1“-Signal am Eingang 04 ein „0“-Signal, läuft die gleiche programmierte Zeit nochmals ab (Ausschaltverzögerungszeit) und der Ausgang 50 wird zurückgesetzt.

Steht das Eingangssignal kleiner (<) der programmierten Zeit an, so bleibt der Ausgang im vorherigen Zustand.

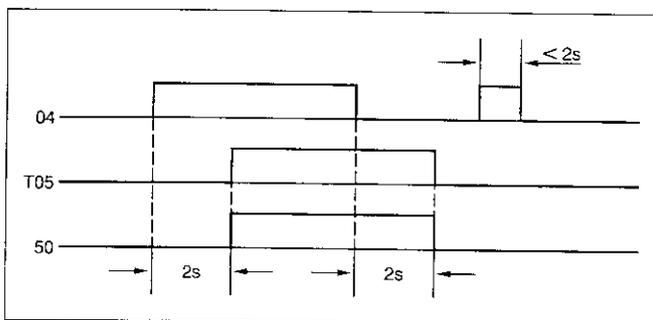


Bild 5.6

5.4 Ein-, Ausschaltverzögerer mit zwei Zeitgliedern

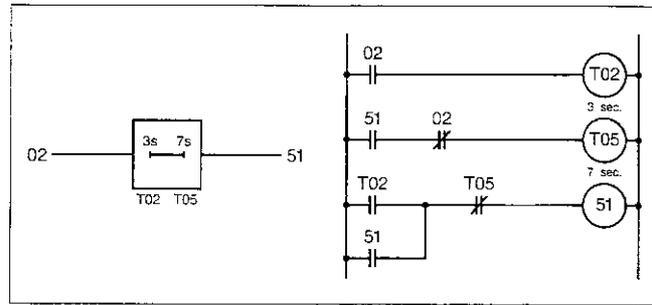


Bild 5.7

Programm:

```

!      02
=      T02 003
!      51
&N    02
=      T05 007
!      T02
/      51
&N    T05
=      51
    
```

Mit einem „1“-Signal am Eingang 02 läuft die Einschaltverzögerungszeit T02 an. Nach Ablauf dieser Zeit wird der Ausgang 51 gesetzt. Wird aus dem „1“-Signal am Eingang 02 ein „0“-Signal, dann läuft die Ausschaltverzögerungszeit T05 ab. Nach Ablauf der Ausschaltverzögerungszeit T05 wird der Ausgang 51 zurückgesetzt.

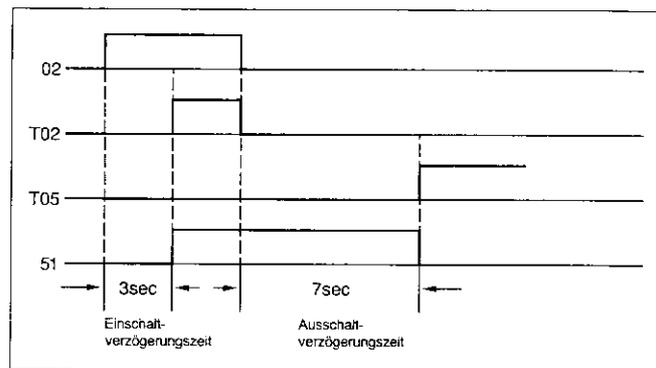


Bild 5.8

5.5 Blocker

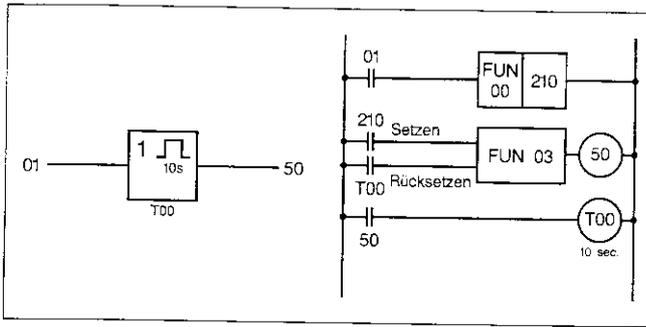


Bild 5.9

Programm:

```

!      01
FUN    00 210
!      210
STR    T00
FUN    03
=      50
!      50
=      T00 010
    
```

Hat der Eingang 01 „1“-Signal, so wird der Ausgang 50 (oder Merker) mit einem Impuls gesetzt. Ausgang 50 startet das Zeitglied T00. Ist die Zeit abgelaufen, so wird Ausgang 50 gelöscht. Bei einem kurzem Eingangssignal hält sich der Blocker bis zur eingestellten Zeit über FUN 03.

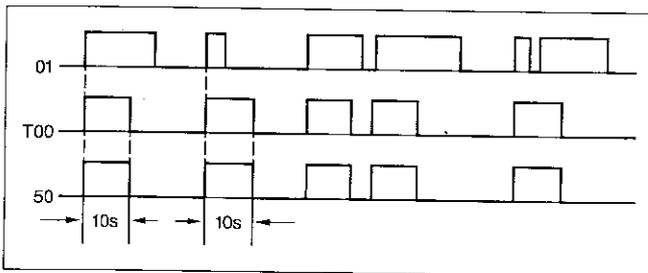


Bild 5.10

5.6 Blocker mit Signalabbruch

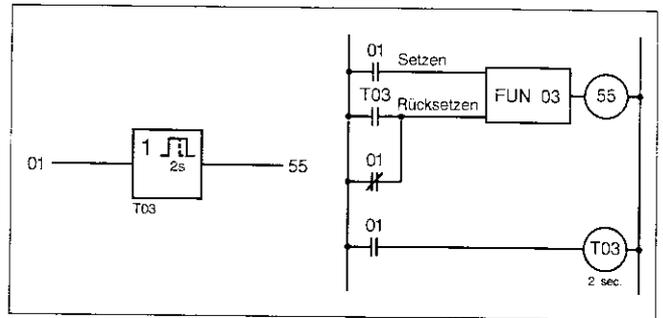


Bild 5.11

Programm:

```

!      01
STR    T03
/N     01
FUN    03
=      55
!      01
=      T03 002
    
```

Hat der Eingang 01 „1“-Signal, wird dem Ausgang 55 ein „1“-Signal zugewiesen. Gleichzeitig wird das Zeitglied T03 gestartet. Nach der eingestellten Zeit setzt das Zeitglied den Ausgang auf „0“-Signal. Hat das Eingangssignal während der Verzögerungszeit ein „0“-Signal, wird dem Ausgang sofort ein „0“-Signal zugewiesen.

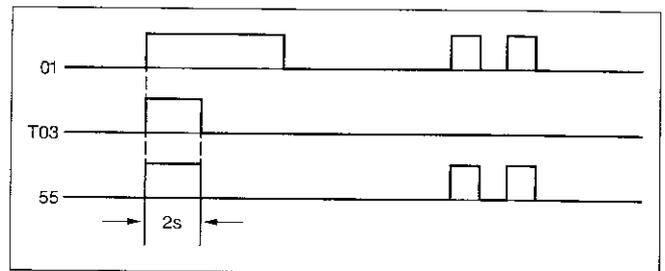


Bild 5.12

5.7 Oszillator mit einem Zeitglied

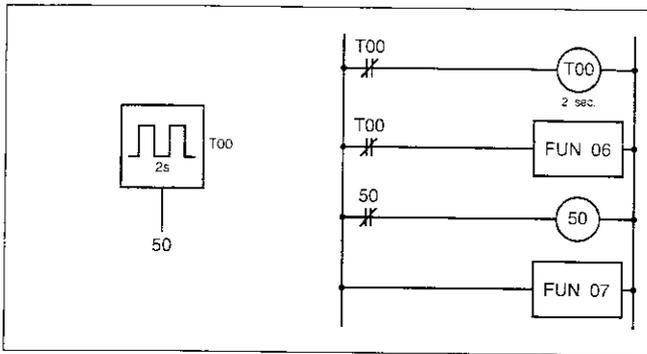


Bild 5.13

Programm:

```

!N      T00
=       T00 002
!N      T00
FUN     06
!N      50
=       50
FUN     07
    
```

Nach Starten des Programms läuft die Zeit T00 ab. Durch die Anweisung !NT00 wird mit FUN 06 der Programmteil bis FUN 07 übersprungen bis die Zeit T00 abgelaufen ist. Nach Ablauf der Zeit wird für die Dauer eines Zyklus der Programmteil zwischen FUN 06 und FUN 07 (!N50 = 50) abgearbeitet, und der Ausgang 50 bekommt „1“-Signal. Gleichzeitig wird die Zeit T00 erneut gestartet, wodurch der Ausgang 50 nach Ablauf der Zeit „0“-Signal hat, usw.

5.8 Oszillator mit zwei Zeitgliedern

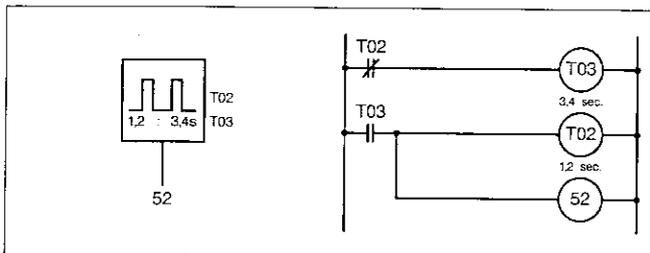


Bild 5.14

Programm:

```

!N      T02
=       T03 03.4
!N      T03
=       T02 01.2
=       52
    
```

Nach dem Starten des Systems läuft das Zeitglied T03 mit der eingestellten Zeit ab. Eingang 52 wird gesetzt. Gleichzeitig wird die Zeit T02 gestartet. Nach Ablauf der Zeit T02 wird Ausgang 52 zurückgesetzt, usw.

T02: impulszeit
T03: Pausenzeit

5.9 Hinweis zur Verwendung von Zeitgliedern

Überschreitet die Programmzykluszeit 16 ms, darf die Zeitbasis 10 ms nicht benutzt werden.

Bei Verwendung einer Grundausstufe ohne Speichererweiterung (950 Worte) ist die Programmzykluszeit nicht größer als 16 ms, d. h. alle Zeitbasen sind uneingeschränkt verwendbar.

Bei längeren Programmen (möglich bei Grundausstufen mit Speichererweiterung) kann die Programmzykluszeit 16 ms überschreiten. In diesem Fall darf die Zeitbasis 10 ms nicht benutzt werden. Die Zeitbasen 100 ms und 1 s sind immer uneingeschränkt verfügbar.

6 Zähler

6.1 Zähler

Softwarezähler in zyklisch arbeitenden Steuerungen haben eine relativ niedrige Zählerfrequenz. Die Zählerfrequenz ist von der Zykluszeit und der Eingangsverzögerung der Signale abhängig.

Bei einer Programmlänge von 1 K-Worten beträgt bei PROCONTIC K200 die Zykluszeit typ. 5 ms, mit den FUN-Befehlen entsprechend länger.

Die maximale Zählfrequenz ist unter Berücksichtigung der Eingangsverzögerung (typ. 4 ms)

für 1 KWorte-Programm max. 50 Hz

für 2 KWorte-Programm max. 40 Hz.

Für schnelle Zählvorgänge empfiehlt es sich, den schnellen Zähler zu nutzen (4 Dekaden BCD, 10 kHz).

Verwendung von Zählern:

Dezimalzähler:

Für interne Zählvorgänge mit vielen Sollwerten bzw. Abfragen.

BCD-Zähler:

Bei externer Sollwertvorgabe bzw. externer Istwertanzeige.

Alle Ausgänge bei den aufgeführten Zählern können genauso, als Merker zwischengespeichert werden, mit den Merkern 400 bis 767 als Zähler mit Haftverhalten.

6.2 Dezimalzähler, vorwärts

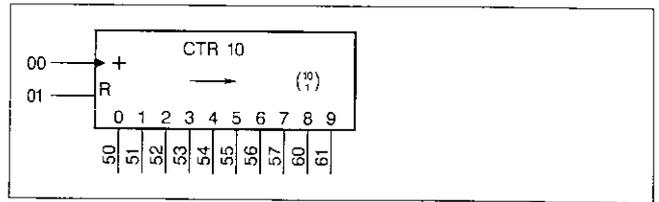
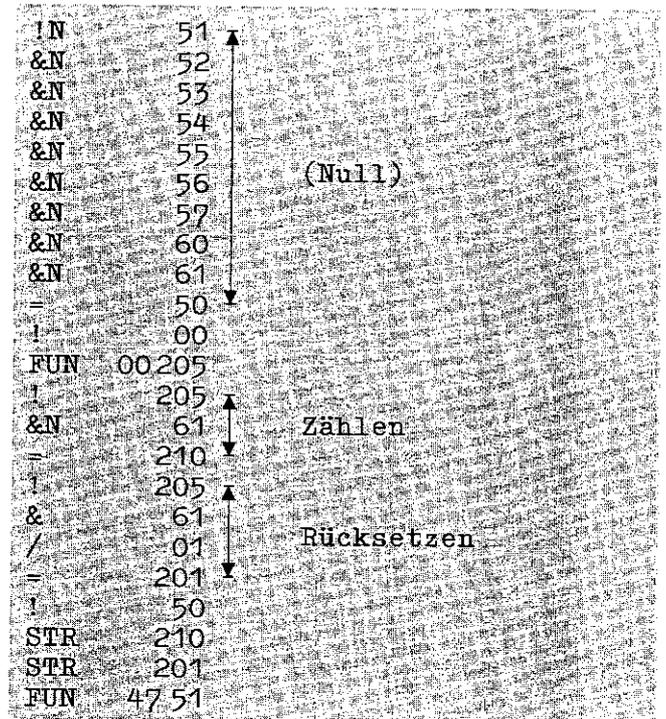


Bild 6.1

Programm mit FUN47



Mit jeder „0-1“-Flanke am Zähl Eingang 00 wird der Zähler um 1 weitergezählt.

Ein „1“-Signal an Eingang 01 setzt den Zähler auf Null zurück.

6.3 Dezimalzähler, rückwärts

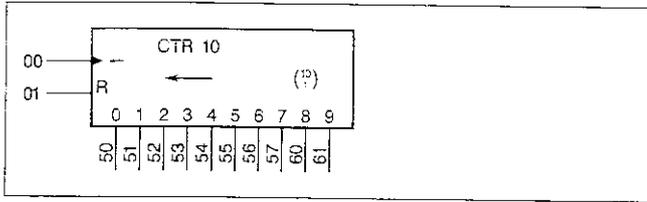


Bild 6.2

Programm mit FUN 47

```

!N      51
&N      52
&N      53
&N      54
&N      55
&N      56
&N      57
&N      60
&N      61
=       50
!       00
FUN 00 205
!       205
&N      51
=       210
!       205
&      51
/       01
=       201
!       50
STR     210
STR     201
FUN 47 301
!       301
=       61
!       302
=       60
!       303
=       57
!       304
=       56
!       305
=       55
!       306
=       54
!       307
=       53
!       310
=       52
!       311
=       51
    
```

(Null)

Zählen

Rücksetzen

Mit jeder „0-1“-Flanke am Zähleingang wird der Zähler um 1 rückwärtsgezählt.

Ein „1“-Signal am Eingang 01 setzt den Zähler auf 0 zurück.

6.4 Dezimalzähler, vorwärts/rückwärts 2 Dekaden

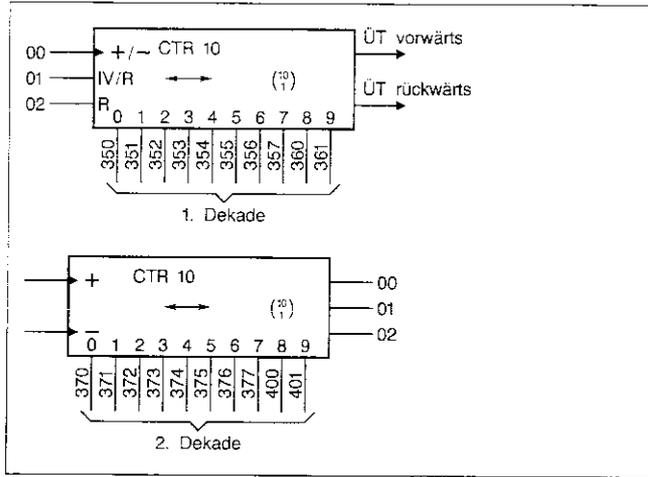


Bild 6.3

Programm:

```

!       01
FUN 00 205 - (Zählimpulsbildung)
!N     300
&      301
&      302
&      303
&      304
&      305
&      306
&      307
&N     00
=       203
!       300
&      303
&      304
&      307
&      00
=       202
!       205
STRN   202
/N     203
&      STR
=       210
!       205
STR    202
/      203
&      STR
=       201
!       00 - (V-R-Eingang)
STR    210 - (Zählen)
STR    201 - (Rücksetzen)
/      02
FUN 40 300
!N     300
&N     301
&N     302
&N     303
=       350
    
```

(Vorwärts)

(Rückwärts)

(Nullbildung 1. Dekade)

!N	304		!	304	
&N	305		&	305	
&N	306	(Nullbildung	&N	306	Umsetzung 2. Dekade
&N	307	2. Dekade)	&N	307	BCD in 1 aus 10
=	370		=	373	
!	300		!N	304	
&N	301		&N	305	
&N	302		&	306	
&N	303		&N	307	
=	351		=	374	
!N	300		!	304	
&	301		&N	305	
&N	302		&	306	
&N	303		&N	307	
=	352		=	375	
!	300		!N	304	
&	301		&	305	
&N	302		&	306	
&N	303		&N	307	
=	353		=	376	
!N	300		!	304	
&N	301	Umsetzung 1. Dekade	&	305	
&	302	BCD in 1 aus 10	&	306	
&N	303		&N	307	
=	354		=	377	
!	300		!N	304	
&N	301		&N	305	
&	302		&N	306	
&N	303		&	307	
=	355		=	400	
!N	300		!	304	
&	301		&N	305	
&	302		&N	306	
&N	303		&	307	
=	356		=	401	
!	300				
&	301				
&	302				
&N	303				
=	357				
!N	300				
&N	301				
&N	302				
&	303				
=	360				
!	300				
&N	301				
&N	302				
&	303				
=	361				
!	304				
&N	305				
&N	306				
&N	307				
=	371				
!N	304				
&	305				
&N	306				
&N	307				
=	372				

Werden mehr als 2 Dekaden benötigt, so kann der BCD-Zähler auch für 3 oder 4 Dekaden umgesetzt werden. Es ist dann jeweils nur noch eine Decodierung von BCD nach ⁽¹⁰⁾ je Dekade zu programmieren. Eingang 00 ist bei oben aufgeführtem Beispiel V-R-Umschalteingang. Eingang 01 ist der Zähleingang. Eingang 02 ist der Rücksetzeingang.

6.5 BCD-Zähler vorwärts

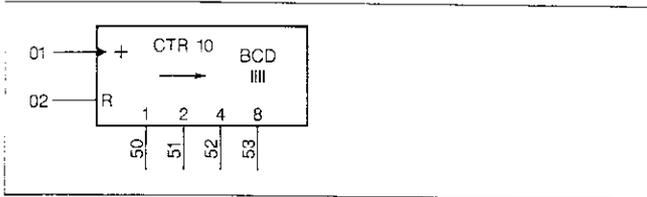


Bild 6.4

Programm:

```

!      01
FUN 00205 -   Zählimpulsbildung
!      50
&      53   ↑   1*
=      202   ↓
!      205
&N     202
=      210
!      205
&      202
=      201
!      00   (V-R-Eingang mit 1
STR     210   Zähleringang
STR     201   beschalten)
/      02   ↓   Rücksetzeingang
FUN 4050
    
```

Wird anstelle des mit 1* gekennzeichneten Satzes, der Satz

```

!      50
&      53
&      54
&      57
=      202
    
```

geschrieben, so arbeitet der V-Zähler mit 2 Dekaden.

Die Ausgänge gehen von 50–57, wobei 50 das niederwertigste und 57 das höchstwertigste Bit darstellt.

Wird anstelle des mit 1* gekennzeichneten Satzes, der Satz

```

!      50
&      53
&      54
&      57
&      60
&      63
=      202
    
```

geschrieben, so arbeitet der V-Zähler mit 3 Dekaden.

Die Ausgänge gehen von 50–63, wobei 50 das niederwertigste und 63 das höchstwertigste Bit darstellt.

Mit jeder „0-1“ Flanke am Zähleringang 01 wird der V-Zähler um 1 vorwärtsgezählt.

Ein „1“-Signal am Rücksetzeingang 02 setzt den Zähler auf Null zurück.

6.6 BCD-Zähler rückwärts

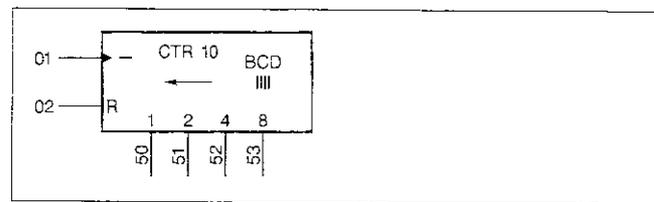


Bild 6.5

Programm:

```

!      01
FUN 00205
!N     300
&      301
&      302
&      303
=      202
①
!      205
&N     202
=      210
!      205
&      202
=      201
!      00 -   (V-R-Eingang
STR     210 -   Zähleringang
STR     201   nicht beschalten)
/      02   ↓   Rücksetzeingang
FUN 40300
!      300
=      50
!      301
=      51
!      302
=      52
!      303
=      53
    
```

① wird hier zusätzlich vor der Ausgangszuweisung (=202) &304 &305 &306 &307 eingefügt und nach dem letzten programmierter Satz (!303=53) folgendes Programmteil angehängt,

```

!      304
=      54
!      305
=      55
!      306
=      56
!      307
=      57
    
```

so arbeitet der R-Zähler zweidekadisch. Die Ausgänge gehen von 50–57, wobei 50 das niederwertigste und 57 das höchstwertigste Bit darstellt.

①
 Wird bei der Anmerkung dagegen folgendes Programmteill eingefügt (vor = 202) & 304 & 305 & 306 & 307 & 310 & 311 & 312 & 313 sowie nach = 53 folgender Programmteil angehängt,

```
! 304
= 54
! 305
= 55
! 306
= 56
! 307
= 57
! 310
= 60
! 311
= 61
! 312
= 62
! 313
= 63
```

so arbeitet der Rückwärtszähler dreidekadisch. Die Ausgänge gehen von 50 – 63, wobei 50 das niederwertigste, 63 das höchstwertigste Bit darstellt.

Mit jeder „0-1“-Flanke am Zähleingang 01 wird der R-Zähler um 1 zurückgezählt. Ein „1“-Signal am Rücksetzeingang 02 setzt den R-Zähler auf 0.

6.7 BCD-Zähler vorwärts/rückwärts (FUN 40)

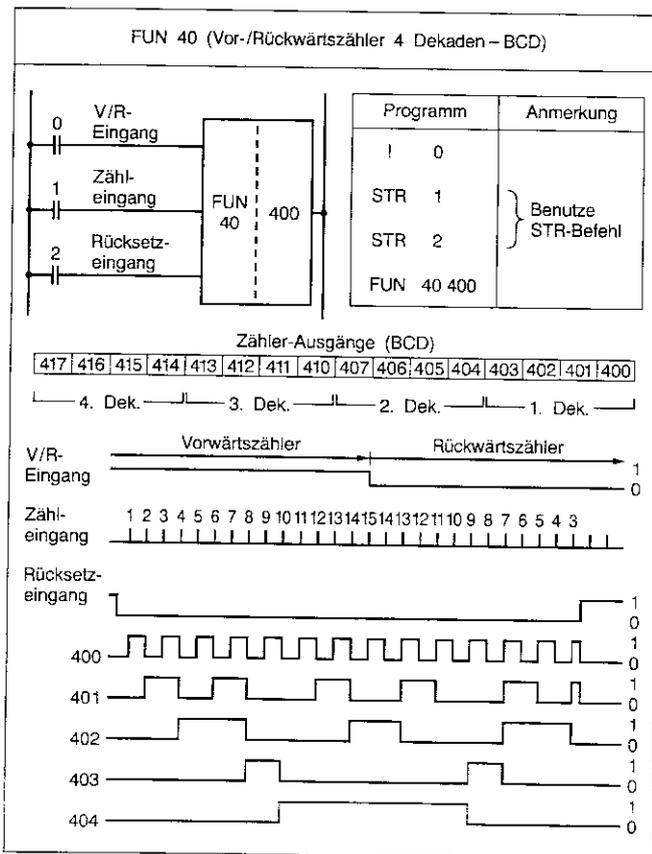


Bild 6.6

- a) Vor/Rückwärtszähler (4 Dekaden) werden programmiert mit FUN 40 in Kombination mit Merkern oder direkt mit den Ausgängen.
 Wird der Zählerstand auf Merker 400 bis 767 abgelegt und dann auf die Ausgänge durchgeschaltet, so bleibt der Zählerstand bei Spannungsunterbrechung gespeichert. Wird der Zählerstand direkt an die Ausgänge geschaltet, so wird der Zähler bei Spannungsausfall auf 0 zurückgesetzt.
- b) Wird der Zähler als Vorwärtszähler benutzt, so wird bei Erreichen von 9999 mit dem nächsten Impuls wieder mit 0 begonnen.
- c) Wird der Zähler als Rückwärtszähler benutzt, so wird bei Erreichen von 0 mit dem nächsten Impuls wieder mit 9999 begonnen.
- d) Der Zähler wird mit einem „1“-Signal am Rücksetz-Eingang auf 0 gesetzt.

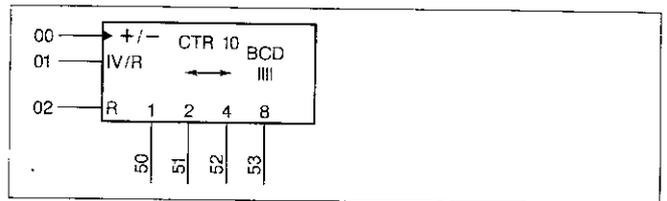


Bild 6.7

Programm

```
! 01
FUN 00 205
!N 300
& 301
& 302
& 303
&N 00
= 203
! 300
& 303
& 00
= 202
! 205
STRN 202
/N 203
& STR
= 210
! 205
STR 202
/ 203
& STR
= 201
! 00 - (V-R-Eingang)
STR 210 - (Zähleingang)
STR 201
/ 02 - (Rücksetzeingang)
FUN 40 300
! 300
= 50
! 301
= 51
! 302
= 52
! 303
= 53
```

Soll der V-R-Zähler zweidekadisch arbeiten, so sind folgende Programmteile hinzuzufügen:

■ Satz Vorwärts vor der Ausgangszuweisung = 203:
& 304 & 305 & 306 & 307

■ Satz Rückwärts vor der Ausgangszuweisung = 202:
& 304 & 307

Außerdem ist nach dem letzten Programmwort = 53 folgender Programmteil anzuhängen:

```
!      304
=      54
!      305
=      55
!      306
=      56
!      307
=      57
```

Diegänge gehen von 50 bis 57, wobei 50 das niederwertigste, 57 das höchstwertigste Bit darstellt.

Soll der V-R-Zähler dreidekadisch arbeiten, so sind dagegen folgende Programmteile einzufügen:

Satz Vorwärts vor = 203
& 304 & 305 & 306 & 307 & 310 & 311 & 312 & 313

Satz Rückwärts vor = 202
& 304 & 307 & 310 & 313

Außerdem ist nach dem letzten Programmwort = 53 folgender Programmteil anzuhängen:

```
!      304
=      54
!      305
=      55
!      306
=      56
!      307
=      57
!      310
=      60
!      311
=      61
!      312
=      62
!      313
=      63
```

Diegänge gehen hierbei von 50 bis 63 wobei 50 das niederwertigste, 63 das höchstwertigste Bit darstellt.

6.8 BCD-Zähler – setzen

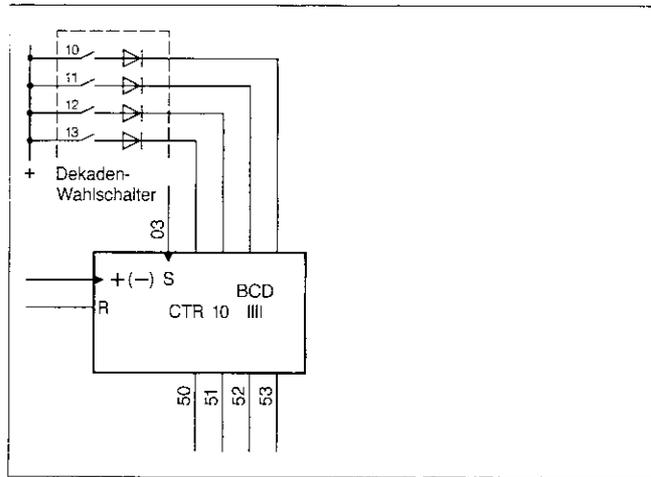


Bild 6.8

Programm für die Zähler
– siehe BCD-Zähler, vorwärts oder rückwärts

Programm:

```
!      03
FUN    00230
! N    230
FUN    06
!      10
FUN    02
=      50
!      11
FUN    02
=      51
!      12
FUN    02
=      52
!      13
FUN    02
=      53
FUN    07
```

Eventuell muß bei dem Setzeingang des entsprechenden Zählers eine &-Verknüpfung eingefügt werden, damit während des Setzens keine Zählimpulse verarbeitet werden.

7 Register

7.1 Schieberegister 16 bit (FUN 47)

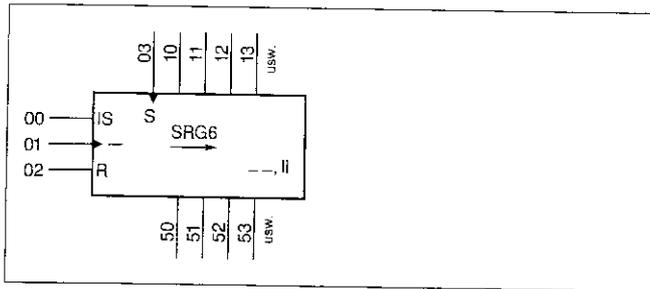
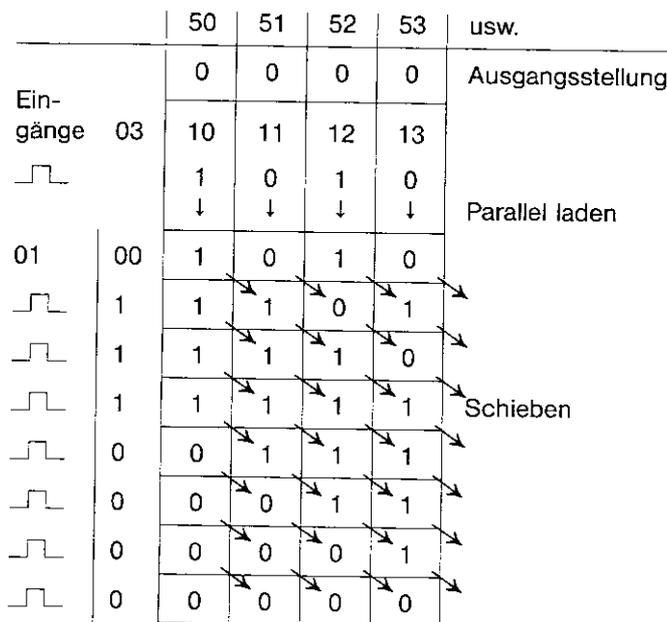


Bild 7.1

Allgemeines:

In einem Schieberegister wird eine Information (1 Bit) über die Anzahl der Schiebepätze (16) mit einem Taktsignal um jeweils einen Platz seriell weitergeschoben. Dabei wird in den ersten Platz eine Information von außen (00) eingelesen, die letzte Information entfällt. Zum Laden und zur Korrektur kann das Schieberegister auch durch Eingänge (10 bis 27) parallel geladen werden, wenn am Eingang 03 ein Impuls ansteht.



Das nachfolgende Programm stellt ein Schieberegister mit 16 Bit dar. Aus Gründen der besseren Übersicht wurde für paralleles Setzen nur die erste Dekade berücksichtigt. Durch entsprechende Eingänge können alle anderen Dekaden ebenso gesetzt werden.

Programm:

```

!      01
FUN   00200
!      00      (serieller Eingang)
STR   200      (Taktingang)
STR   02      (Rücksetzeingang)
FUN   47 50
!      03
FUN   00201
!N    201
FUN   06
!      10
FUN   02
=     50
!      11
FUN   02
=     51
!      12
FUN   02
=     52
!      13
FUN   02
=     53
FUN   07
    
```

Die Anzahl der Dekaden der Schieberegister kann beliebig erhöht werden.

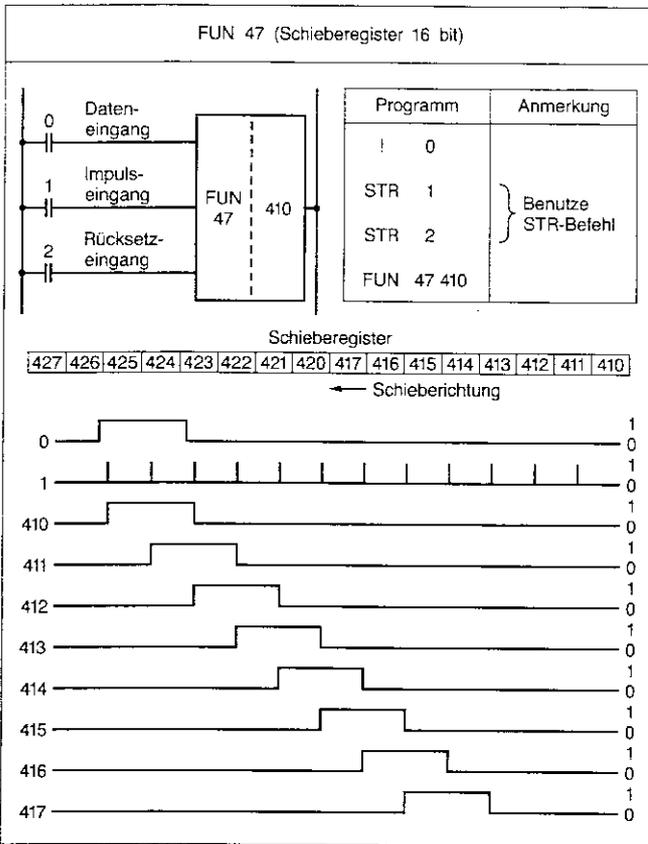


Bild 7.2

- a) FUN47 ist eine Funktion zum Aufbau eines seriellen 16 Bit-Schieberegisters und wird zusammen mit den Merkern benutzt, bei Benutzung der Merker 400 bis 767 mit Haftverhalten.
- b) Die Daten an dem Dateneingang werden vom niederwertigsten Bit (im obigen Beispiel) Merker 410 zum höchstwertigem Punkt geschoben (im obigen Beispiel Merker 427). Statt Merker können auch direkte Ausgänge programmiert werden.
- c) Es besteht die Möglichkeit, mehrere Schieberegister mit verschiedenen Merkern bzw. Ausgängen anzusprechen.

7.2 Siloregister

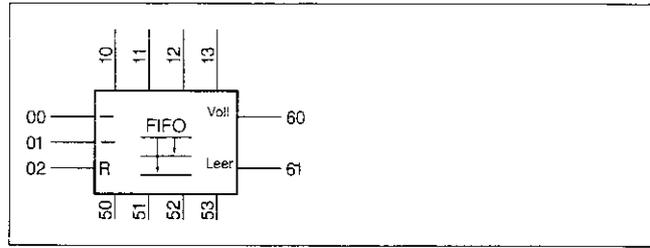


Bild 7.3

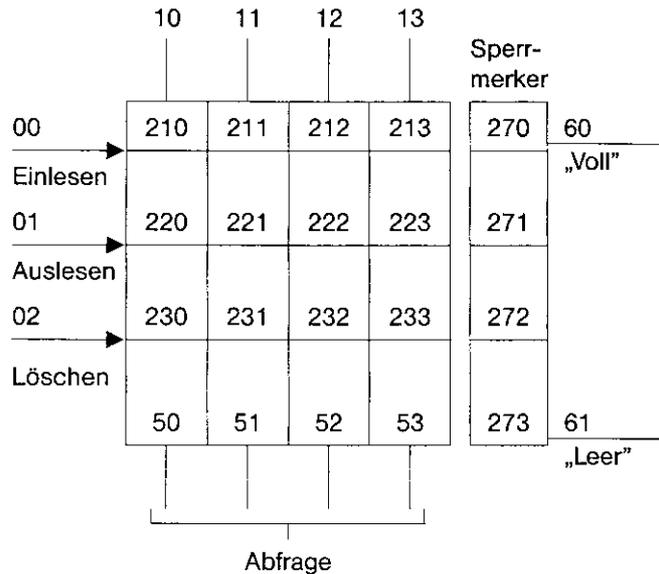
Allgemeines:

Bei einem Siloregister (FIFO = first in – first out) wird eine Parallelinformation (10 – 13) zeilenweise mit einem Einlese-signal am Eingang 00 in einen Speicher eingelesen. Die Parallelinformation fällt bei leerem Register bis zur letzten Zeile durch und kann an den Ausgängen abgelesen werden. Sind bereits Zeilen belegt, fällt die Information bis zur letzten freien Zeile.

Mit einem Auslesesignal an Eingang 01 wird die letzte Zeile gelöscht, und alle anderen Zeilen rücken nach. Mit dem Löschsinal an Eingang 03 wird der gesamte Speicher gelöscht.

Ist der Speicher voll, d.h. sind alle Zeilen belegt, wird keine Information angenommen.

Die Zustände „Voll“ und „Leer“ des Siloregisters werden angezeigt.



Programm:

```

!      00  ↑
&N    270  ↓ (Einlesetakt)
FUN 00200
!N    200
FUN   06
!     10
=     210
!     11
=     211
!     12
=     212
!     13
=     213
!N    270
=     270
FUN   07
!     271
FUN   06
!     210
=     220
!     211
=     221
!     212
=     222
!     213
=     223
!     270
=     271
FUN   02
=N    270
=N    210
=N    211
=N    212
=N    213
FUN   07
!     272
FUN   06
!     220
=     230
!     221
=     231
!     222
=     232
!     223
=     233
!     271
=     272
FUN   02
=N    271
FUN   07
!     273
FUN   06
!     230
=     50
!     231
=     51
!     232
=     52
!     233
=     53

```

```

!      272
=      273
FUN    02
=N     272
FUN    07
!      01  ↓ (Auslesetakt)
FUN 00274
!      274
FUN    02
=N     273
!      02
FUN    02  ↑
=N     271  ↑ (Löschen)
=N     272  ↓
=N     273  ↓
=N     274  ↓
!N    273  ↑
&N    272  ↑ (Meldung „Leer“)
&N    274  ↓
=      61
!      270
=      60  ↓ (Meldung „Voll“)

```


8 Code-Umsetzer

8.1 Code-Umsetzer, BCD in 1 aus 10

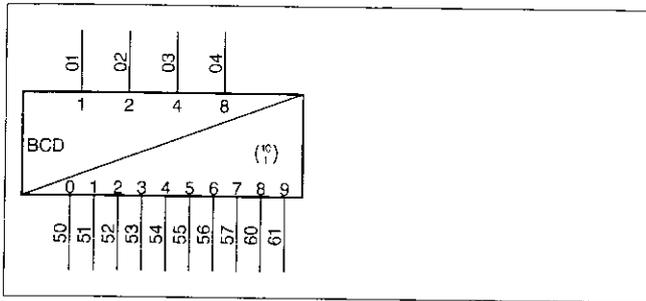


Bild 8.1

Programm:

```

!N 01
&N 02
&N 03
&N 04
= 50
! 01
&N 02
&N 03
&N 04
= 51
!N 01
& 02
&N 03
&N 04
= 52
! 01
& 02
&N 03
&N 04
= 53
!N 01
&N 02
& 03
&N 04
= 54
! 01
&N 02
& 03
&N 04
= 55
!N 01
& 02
& 03
&N 04
= 56
! 01
& 02
& 03
&N 04
= 57
    
```

```

!N 01
&N 02
&N 03
& 04
= 60
! 01
&N 02
&N 03
& 04
= 61
    
```

8.2 Code-Umsetzer, 1 aus 10 in BCD

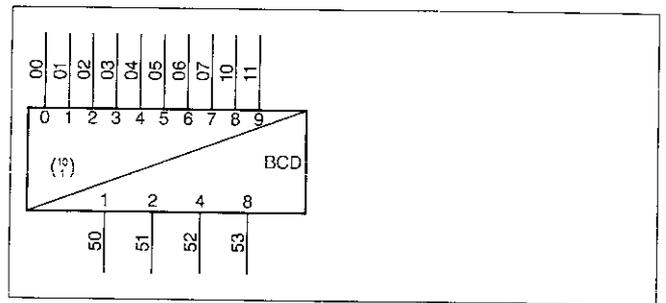


Bild 8.2

Programm:

```

! 01
/ 03
/ 05
/ 07
= 50
! 02
/ 03
/ 06
= 51
! 04
/ 05
/ 06
= 52
! 10
/ 11
= 53
    
```

8.3 Code-Umsetzer, BCD in 1 aus 8

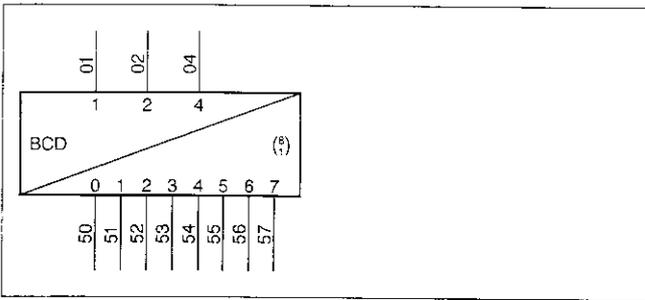
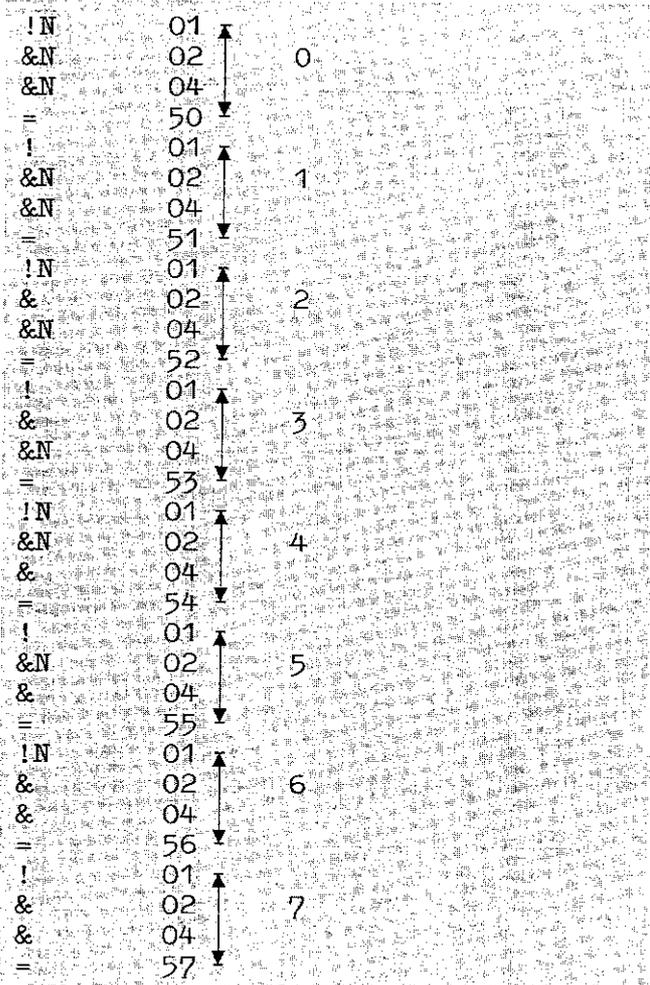


Bild 8.3

Programm:



8.4 Code-Umsetzer, 1 aus 8 in BCD

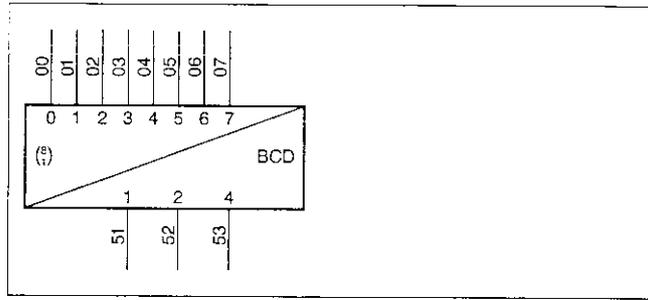
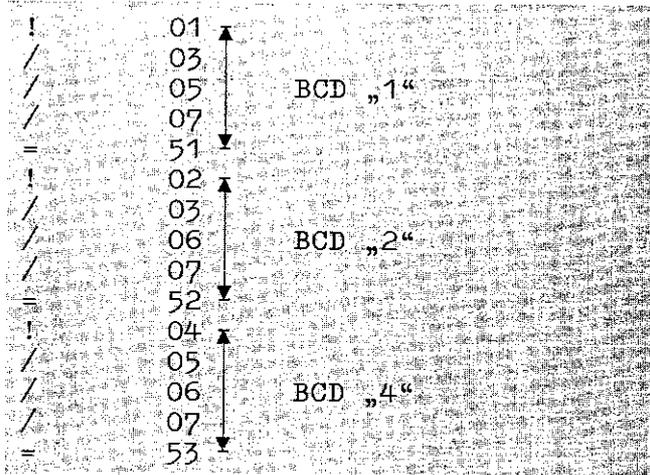


Bild 8.4

Programm:



8.5 Code-Umsetzer, DUAL in 1 aus 16

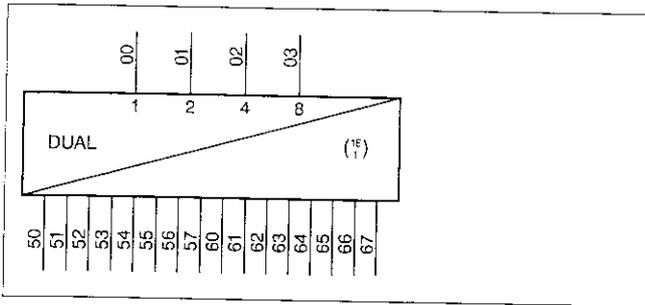


Bild 8.5

Programm:

```

!N      00
&N     01
&N     02
&N     03
=      50
!      00
&N     01
&N     02
&N     03
=      51
!N     00
&      01
&N     02
&N     03
=      52
!      00
&      01
&N     02
&N     03
=      53
!N     00
&N     01
&      02
&N     03
=      54
!      00
&N     01
&      02
&N     03
=      55
!N     00
&      01
&      02
&N     03
=      56
!      00
&      01
&      02
&N     03
=      57
!N     00
&N     01
&N     02
&      03
=      60
    
```

```

!      00
&N     01
&N     02
&      03
=      61
!N     00
&      01
&N     02
&      03
=      62
!      00
&      01
&N     02
&      03
=      63
!N     00
&N     01
&      02
&      03
=      64
!      00
&N     01
&      02
&      03
=      65
!N     00
&      01
&      02
&      03
=      66
!      00
&      01
&      02
&      03
=      67
    
```

8.6 FUN 24 – FUN 25, Umwandlung Binär (Hex) in BCD und umgekehrt

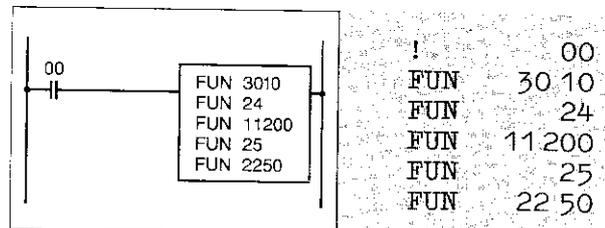


Bild 8.6

Sobald der Eingang 00 „1“-Signal führt (Startbedingung erfüllt), wird der Zustand der Eingänge 10 bis 27 mit FUN 30 in das Arithmetik-Register übernommen.

FUN 24 wandelt den Binärwert (Hex) in einen BCD-Wert um.

Zu diesem BCD-Wert wird mit FUN 11 der Wert der Merker 200 bis 217 hinzuaddiert, und das Ergebnis wird im Arithmetik-Register abgespeichert.

Mit FUN 25 wird nun der Wert des Arithmetik-Registers in einen Binärwert (Hex) umgewandelt und mit FUN 22 an die Ausgänge 50 bis 67 weitergegeben (s. Kapitel 10.5).

9 Vergleicher/Arithmetik-Befehle

9.1 Vergleicher auf Gleichheit

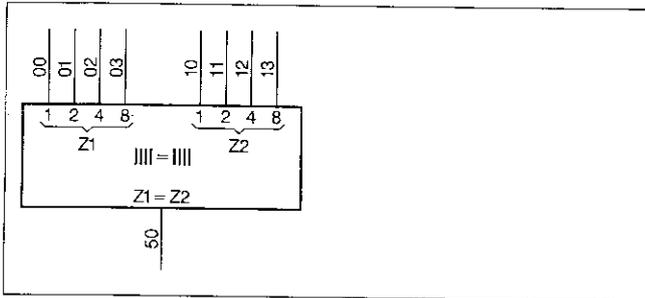


Bild 9.1

Realisierung im Programm

Es wird jede einzelne Bitstelle auf größer oder kleiner geprüft. Ist dies nicht der Fall, sind die beiden Werte gleich.

Programm: mit STR Befehl

```

!N      00
&       10
STR     00
&N     10
STRN    01
&       11
STR     01
&N     11
STRN    02
&       12
STR     02
&N     12
STRN    03
&       13
STR     03
&N     13
/       STR
=N      50
    
```

Programm: mit Merkern

```

!N      00
&       10
=       200
!       00
&N     10
=       201
!N     01
&       11
=       202
!       01
&N     11
=       203
!N     02
&       12
=       204
!       02
&N     12
=       205
!N     03
&       13
=       206
!       03
&N     13
=       207
/       200
/       201
/       202
/       203
/       204
/       205
/       206
/       207
=N      50
    
```

Wenn beide Werte gleich sind, wird Ausgang 50 geschaltet (1-Signal).

9.2 Größer-Kleiner-Gleich-Vergleicher (4 bit)

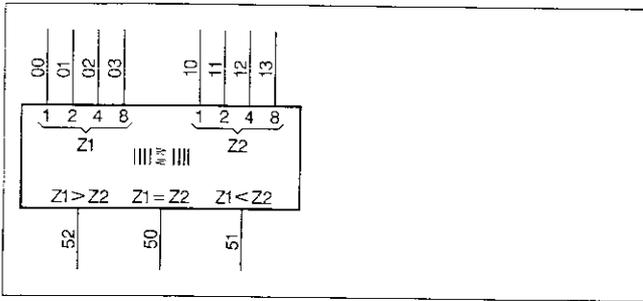


Bild 9.2

Realisierung im Programm

Ausgehend von der höchsten Bit-Wertigkeit wird nacheinander jeder einzelne Bitwert auf > oder < geprüft, und das Ergebnis auf den entsprechenden Ausgang geschaltet.

Programm:

```

!      03
&N    13
=      200
!N    03
&     13
=      201
!N    03
&N    13
STR    3
&     13
/      STR
=      202
!     02
&N    12
&     202
=      203
!N    02
&     12
&     202
=      204
!N    02
&N    12
STR    2
&     12
/      STR
=      205
!     01
&N    11
&     202
&     205
=      206
!N    01
&     11
&     202
&     205
=      207
!N    01
&N    11
STR    01
&     11
/      STR
=      210
    
```

```

!      00
&N    10
&     202
&     205
&     210
=      211
!N    00
&     10
&     202
&     205
&     210
=      212
!     200
/      203
/      206
/      211
=      52
!     201
/      204
/      207
/      212
=      51
!N    52
&N    51
=      50
    
```

9.3 Größer-Kleiner-Gleich Vergleich (2 bit)

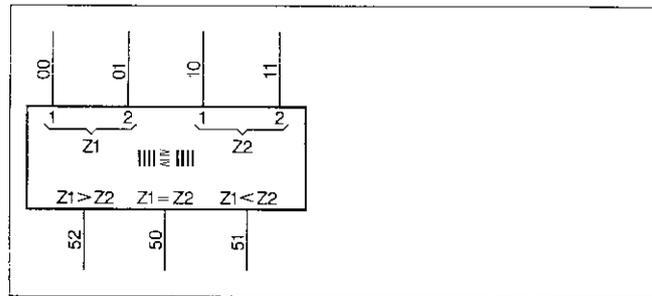


Bild 9.3

Programm:

```

!      01
&N    11
=      200
!N    01
&     11
=      201
!N    01
&N    11
STR    01
&     11
/      STR
=      202
!     00
&N    10
&     202
=      203
!N    00
&     10
&     202
=      204
    
```

```

!      200
/      203
=      52
!      201
/      204
=      51
!N     52
&N    51
=      50

```

9.4 Allgemeines über Arithmetik-Befehle

Mit der PROCONTIC K200 können arithmetische Befehle als Wort (1 Wort = 16 Bit) verarbeitet werden.

Arithmetische Befehle werden dargestellt, wie in Bild 9.4 gezeigt. Sie müssen immer mit einer Startbedingung programmiert werden.

Die arithmetische Operation (Funktion) wird dann ausgeführt, wenn die Startbedingung erfüllt ist.

Ist die Startbedingung nicht mehr erfüllt, so bleiben die Ausgänge oder Merker in dem Status, den sie gerade angenommen hatten.

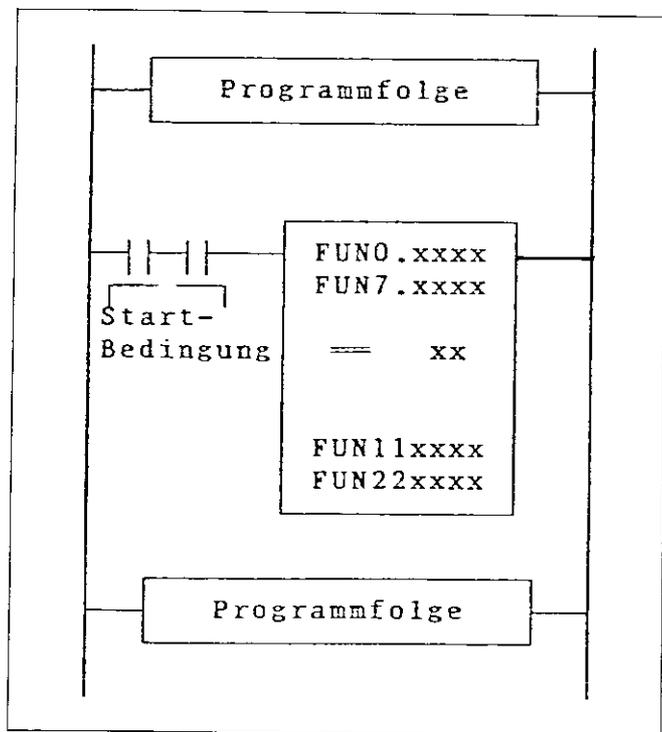


Bild 9.4

Ergebnisse von arithmetischen Operationen wie Addition, Subtraktion, Multiplikation, Division und BNR-Daten werden nur im BCD-Code ausgegeben. Deshalb müssen Arithmetik-Operanden in BCD-Code umgewandelt und eingegeben werden (s. Kapitel 10.5 in dieser Beschreibung).

9.5 FUN 0. - Ladekonstante

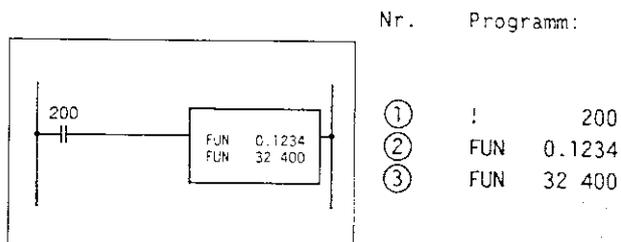


Bild 9.5

Anmerkungen zu Nr. ① - ③:

- ①: ! 200 --> Startbedingung
- ②: "1234" --> AR
- ③: AR --> 400 - 417

Der Befehl FUN 0. wird benutzt, um eine Konstante im BCD-Format in das Arithmetikregister zu laden.

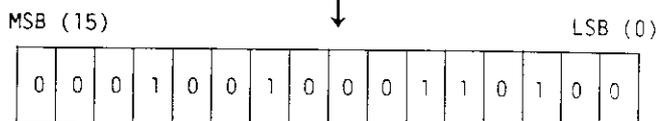
Vor dem Wert der Konstanten muß ein Dezimalpunkt programmiert werden.

Beispiel:

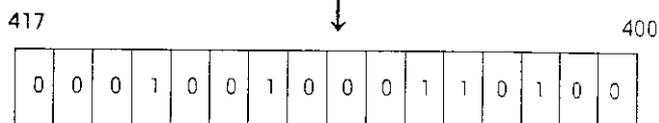
Lade-Konstante (BCD)



FUN 0.1234



FUN 32 400



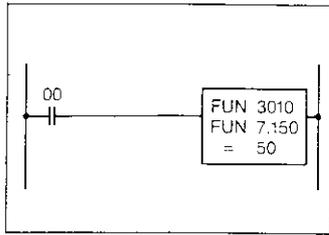
Interner Ausgang (Merker)

Bild 9.6

Die vier Stellen BCD-Daten "1234" werden in das Arithmetik-Register (AR) geladen, und der Inhalt des AR wird in diesem Beispiel auf die gepufferten Merker 400 - 417 gegeben, sobald die Startbedingung (Merker 200 hat "1"-Signal erfüllt ist).

Wird die Startbedingung (Merker 200 hat "1"-Signal nicht mehr erfüllt (Merker 200 hat "0"-Signal), so bleiben die Merker auf dem Status gesetzt, den sie gerade hatten.

9.6 FUN 7. – Vergleich der Konstanten auf \geq mit dem Inhalt des Arithmetik-Registers



Programm

```
!      00
FUN   30 10
FUN   7.150
=     50
```

Bild 9.7

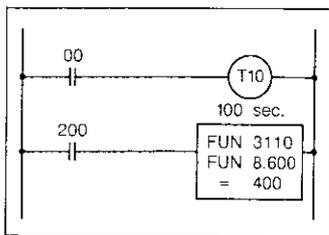
Sobald Eingang 00 "1"-Signal führt (Startbedingung erfüllt), wird der Zustand der Eingänge 10 bis 27 mit FUN 30 ins Arithmetik-Register übernommen. Der Inhalt des Arithmetik-Registers wird mit der Konstanten (in oben aufgeführtem Beispiel 150) von FUN 7. verglichen.

Ist der Inhalt des Arithmetik-Registers größer oder gleich (\geq) der Konstanten von FUN 7., dann bekommt der Ausgang 50 "1"-Signal.

Ist der Inhalt des Arithmetik-Registers kleiner ($<$) der Konstanten von FUN 7., dann bleibt der Ausgang 50 in Ruhestellung ("0"-Signal).

Anmerkung: Bei Vergleich von Zeiten mit FUN 7. wird die Konstante in Schritten von 0,1 s eingegeben (z. B. Eingabe 600 ist gleich 60 s).

9.7 FUN 8. – Vergleich der Konstanten auf = mit dem Inhalt des Arithmetik-Registers



Programm

```
!      00
=T     10 100
!      200
FUN   31 10
FUN   8.600
=     400
```

Bild 9.8

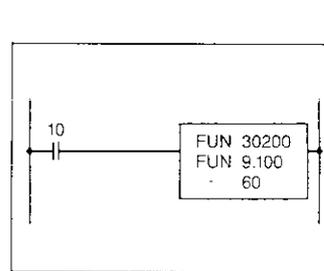
Sobald die Startbedingung (Merker 200 hat "1"-Signal) erfüllt ist, wird der augenblickliche Wert (Istwert) des Zeitgliedes T10 mit FUN 31 in das Arithmetik-Register übernommen. Der Inhalt des Arithmetik-Registers wird mit der Konstanten von FUN 8. (in diesem Beispiel 600) verglichen.

Ist der Inhalt des Arithmetik-Registers gleich (=) der Konstanten von FUN 8., dann bekommt der Merker 400 "1"-Signal.

Ist der Inhalt des Arithmetik-Registers ungleich der Konstanten von FUN 8., dann bleibt der Merker 400 in Ruhestellung ("0"-Signal).

Anmerkung: Bei Vergleich von Zeiten mit FUN 8. wird die Konstante (i. o. aufgeführtem Beispiel 600) in Schritten von 0,1 s eingegeben (600 x 0,1 s = 60 s).

9.8 FUN 9. – Vergleich der Konstanten auf $<$ mit dem Inhalt des Arithmetik-Registers



Programm:

```
!      10
FUN   30 200
FUN   9.100
=     60
```

Bild 9.9

Sobald Eingang 10 "1"-Signal führt (Startbedingung erfüllt), wird der Zustand der Merker 200 bis 217 mit FUN 30 in das Arithmetik-Register übernommen.

Der Inhalt des Arithmetik-Registers wird mit der Konstanten von FUN 9. (in diesem Beispiel 100) verglichen.

Ist der Inhalt des Arithmetik-Registers kleiner ($<$) der Konstanten von FUN 9., dann bekommt der Ausgang 60 "1"-Signal.

Ist der Inhalt des Arithmetik-Registers größer oder gleich (\geq) der Konstanten, so bleibt der Ausgang 60 in Ruhestellung ("0"-Signal).

Anmerkung: Bei Vergleich von Zeiten mit FUN 9. wird die Konstante (i. o. aufgeführtem Beispiel 100) in Schritten von 0,1 s eingegeben (100 x 0,1 s = 10 s).

9.9 Zusammenfassendes Beispiel für die Funktionen FUN 7., FUN 8. und FUN 9.

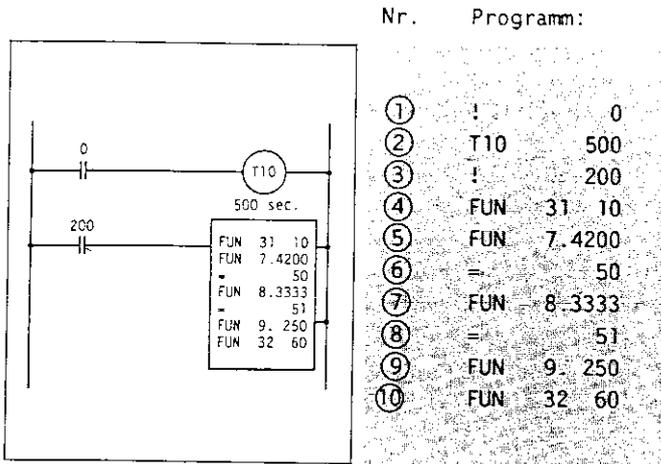


Bild 9.10

Anmerkungen zu Nr. ④ - ⑩ :

- ④ : Aktueller Wert von T10 --> AR
- ⑤ : $AR \geq "4200" \dots 1 \rightarrow C$
- ⑥ : Wenn C = 1, dann 50 ein
- ⑦ : $AR = "3333" \dots 1 \rightarrow C$
- ⑧ : Wenn C = 1, dann 51 ein
- ⑨ : $AR < "250" \dots 1 \rightarrow C$
- ⑩ : Wenn C = 1, AR --> (60-77)
Wenn C = 0, 0 --> (60-77)

Wenn die Startbedingung erfüllt ist (Merker 200 hat "1"-Signal), dann wird der aktuelle Wert des Zeitgliedes T10 mit FUN 31 in das Arithmetikregister (AR) übergeben und mit den Konstanten von FUN7., FUN8. und FUN9. verglichen.

Der aktuelle Wert von T10 soll bei folgenden Erklärungen 4035 (403,5 s) betragen.

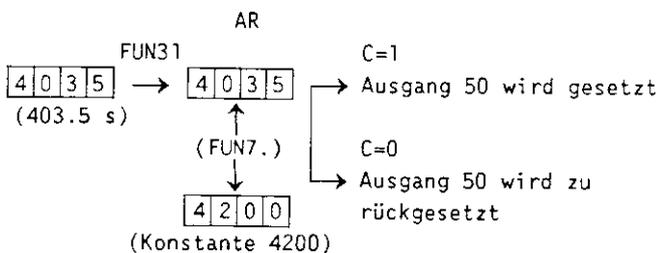


Bild 9.11

Mit FUN31 wird der aktuelle Wert von T10 (4035) ins Arithmetikregister (AR) geladen. Der Vergleich mit der Konstanten von FUN7. (4200) ergibt, daß der Wert im AR kleiner als die Konstante von FUN7. ist ($4035 < 4200$). Somit wird der Übertragungsmerker (carry flag C) auf 0 gesetzt, was zur Folge hat, daß der Ausgang 50 zurückgesetzt wird.

Das Ergebnis des Vergleichs kann im Bit-Format ausgegeben werden. Der Ein-/Ausstatus des C (carry flag) wird zusammen mit dem Ergebnis ausgegeben.

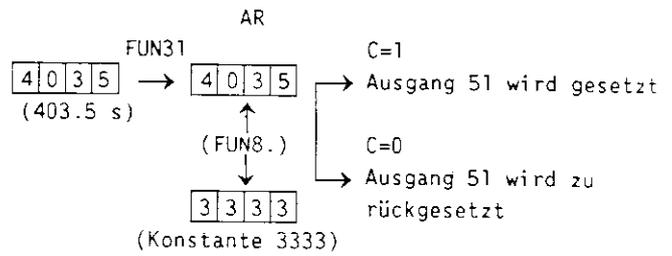


Bild 9.12

Mit FUN31 wird der aktuelle Wert von T10 (4035) ins Arithmetikregister (AR) geladen. Der Vergleich mit der Konstanten von FUN8. (3333) ergibt, daß der Wert im AR ungleich der Konstanten von FUN8. ist. Somit wird der Übertragungsmerker (carry flag C) auf 1 gesetzt, was zur Folge hat, daß der Ausgang 51 zurückgesetzt wird.

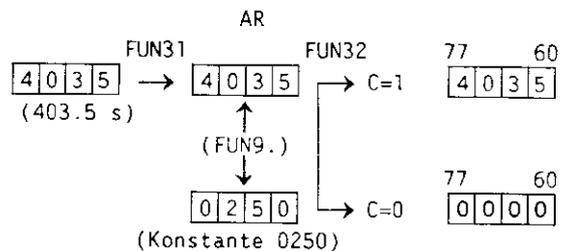


Bild 9.13

Mit FUN31 wird der aktuelle Wert von T10 (4035) ins Arithmetikregister (AR) geladen. Der Vergleich mit der Konstanten von FUN9. (0250) ergibt, daß der Wert im AR größer als die Konstante von FUN9. ist. Somit wird der Übertragungsmerker (carry flag C) auf 1 gesetzt, was zur Folge hat, daß durch FUN 32 das Wort 0000 an die Ausgänge 60 bis 77 gegeben wird.

9.10 Status des Übertragungsmerkers (Carry Flag) bei den Vergleichsbefehlen

Folgende Tabelle zeigt den Wert des carry flag C in Abhängigkeit davon, ob die jeweilige Konstante der Vergleichsbefehle FUN 7., FUN 8. und FUN 9. grösser, gleich oder kleiner dem aktuellen Wert des Arithmetikregisters ist.

Befehl	AR > Konst.	AR = Konst.	AR < Konst.
FUN7. \geq AR	C = 1	C = 1	C = 0
FUN8. = AR	C = 0	C = 1	C = 0
FUN9. < AR	C = 0	C = 0	C = 1

Die mittlere Befehlsausführungszeit der Rechenbefehle entnehmen Sie bitte der Tabelle auf Seite 1-5 in dieser ABB Procontic K200-Software-Beschreibung.

10.1 FUN 11 - Addition

Beispiel 1:

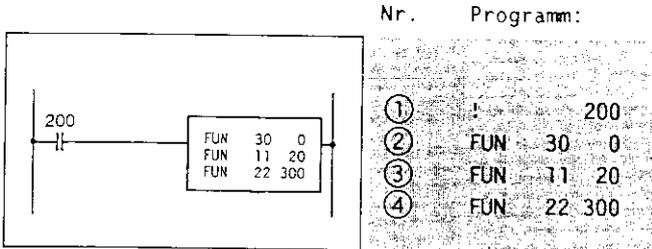


Bild 10.1

Anmerkung zu Nr. ① - ④:

- ① : ! 200 --> Startbedingung
- ② : 0 - 17 --> AR
- ③ : AR + (20-37) --> AR
- ④ : AR --> 300-317

Der Befehl FUN 11 wird eingesetzt, wenn zum Inhalt des AR etwas addiert werden soll.

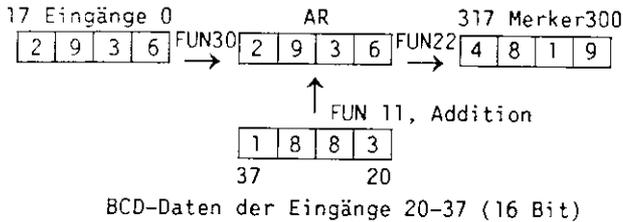


Bild 10.2

In obigem Beispiel werden die BCD-Daten der Eingänge 0-17 mit FUN 30 ins Arithmetikregister (AR) übernommen, wenn die Startbedingung (Merker 200 hat "1"-Signal) erfüllt ist.

Mit FUN 11 wird der BCD-Wert, der an den Eingängen 20-37 anliegt, zum Inhalt des AR addiert, und der neue Wert wird im AR abgelegt. Das Ergebnis wird mit FUN 22 an die Merker 300-317 gegeben.

Einschränkung: Bei einem Ergebnis größer als 9999 wird ein Übertragungsmerker (carry flag C) gesetzt. Mit dem Wert, zu dem etwas addiert werden sollte, wird weitergearbeitet (bei diesem Beispiel mit dem Wert der an den Eingängen 10-17 anliegt), d. h. der Wert wird unbearbeitet auf die Merker weitergegeben.

Mit FUN 22 wird der Inhalt des AR auf die Ausgänge oder Merker gegeben (Ausgabe Wort).

Anmerkung: Sobald das AR ausgelesen ist, muß bei erneutem Zugriff auf das AR wieder eine Startbedingung definiert werden.

Der Unterschied zwischen FUN 32 und FUN 22 ist in folgender Tabelle dargestellt:

Befehl	Carry-Flag Status	Funktion
FUN 32	C = 0	keine Ausgabe auf Ausgänge/ Merker
	C = 1	Inhalt AR auf Ausgänge/ Merker
FUN 22	C = 0 oder C = 1	Inhalt AR auf Ausgänge/ Merker

Beispiel 2:

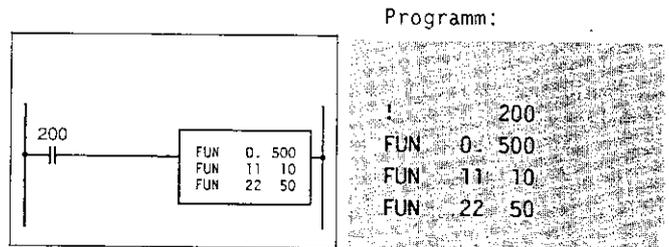


Bild 10.3

Wird die Startbedingung (Merker 200 hat "1"-Signal) erfüllt, so wird der Wert der Konstanten (in diesem Beispiel 500) mit FUN 0. in das Arithmetikregister übernommen. Zu diesem Wert wird mit FUN 11 der Zustand der Eingänge 10 bis 27 addiert und im Arithmetik-Register abgelegt.

Mit FUN 22 wird der Wert des Arithmetik-Registers an die Ausgänge 50 bis 67 weitergegeben.

Einschränkung: Ist die Addition von zwei Zahlen (z. B. 3350 + 7000) >9999, so wird die Zahl im Programm weiter berücksichtigt, zu der etwas addiert werden soll.

10.2 FUN 12 - Subtraktion

Beispiel 1:

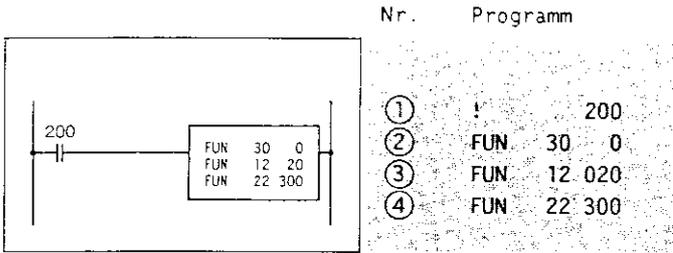


Bild 10.4

Anmerkung zu Nr. ① - ④:

- ①: ! 200 --> Startbedingung
- ②: 0 - 17 --> AR
- ③: AR minus (20-37) --> AR
- ④: AR --> 300 - 317

Der Befehl FUN 12 wird eingesetzt, wenn vom Inhalt des Arithmetikregisters (AR) etwas subtrahiert werden soll.

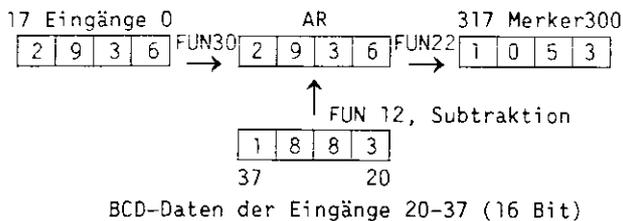


Bild 10.5

In obigem Beispiel werden die BCD-Daten der Eingänge 0-17 mit FUN 30 ins AR übernommen, wenn die Startbedingung (Merker 200 hat "1"-Signal) erfüllt ist.

Mit FUN 12 wird der BCD-Wert, der an den Eingängen 20-37 anliegt, vom Inhalt des AR abgezogen, und der neue Wert wird im AR abgelegt. Das Ergebnis wird mit FUN 22 an die Merker 300-317 gegeben.

Einschränkung: Bei negativem Ergebnis wird ein Übertragungsmerker (carry flag C) gesetzt. Mit dem Wert, von dem etwas abgezogen werden sollte, wird weitergearbeitet (bei diesem Beispiel mit dem Wert der an den Eingängen 10-17 anliegt), d. h. der Wert wird unbearbeitet durch FUN 22 auf die Merker 300-317 weitergegeben.

Beispiel 2:

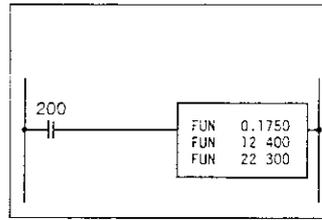


Bild 10.6

Programm:

!	200
FUN	0.1750
FUN	12 400
FUN	22 300

Sobald die Startbedingung (Merker 200 hat "1"-Signal) erfüllt ist, wird der Wert der Konstanten (in diesem Beispiel 1750) mit FUN 0. in das Arithmetik-Register übernommen.

Von diesem Wert wird mit FUN 12 der Zustand der Merker 400 bis 417 subtrahiert und im Arithmetik-Register abgelegt.

Mit FUN 22 wird der Wert des Arithmetik-Registers an die Merker 300 bis 317 weitergegeben.

Einschränkung: Die Darstellung eines negativen Resultats ist nicht möglich (z. B. 200 - 300). Bei negativem Resultat wird die Zahl, von der etwas subtrahiert werden soll, im Programm weiter berücksichtigt.

10.3 FUN 13 - Multiplikation

Beispiel 1:

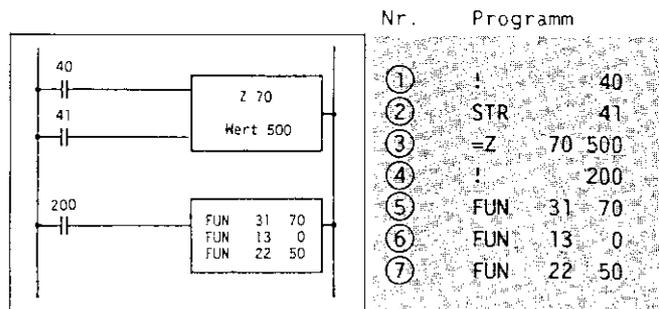


Bild 10.7

Anmerkung zu Nr. ④ - ⑦:

- ④: ! 200 --> Startbedingung
- ⑤: Z70 --> AR
- ⑥: AR x (0-17) --> AR
- ⑦: AR --> 50-67

Der Befehl FUN 13 wird eingesetzt, wenn der Wert des Arithmetikregisters (AR) mit einem anderen BCD-Wert multipliziert werden soll.

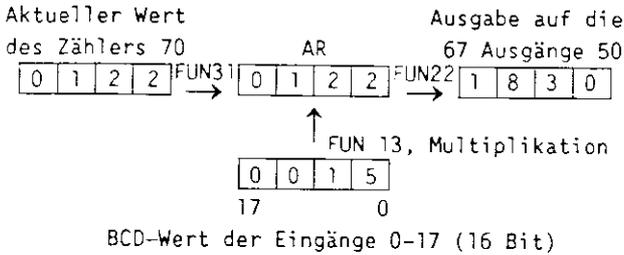


Bild 10.8

In obigem Beispiel wird der aktuelle Zählerstand mit FUN 31 in das Arithmetikregister (AR) übernommen, wenn die Startbedingung (Merker 200 hat "1"-Signal) erfüllt ist.

Mit FUN 13 wird der BCD-Wert des AR mit dem aktuellen Wert, der an den Eingängen 0-17 anliegt, multipliziert und im AR abgelegt. Mit FUN 22 wird der Wert des AR an die Ausgänge 50-67 gegeben.

Einschränkung: Bei einem Ergebnis größer als 9999 wird ein Übertragungsmerker (carry flag C) gesetzt. Mit dem Wert, der mit einer Zahl multipliziert werden sollte, wird weitergearbeitet, d. h. der Wert wird unbearbeitet durch FUN 22 auf die Ausgänge 50-67 weitergegeben.

Beispiel 2:

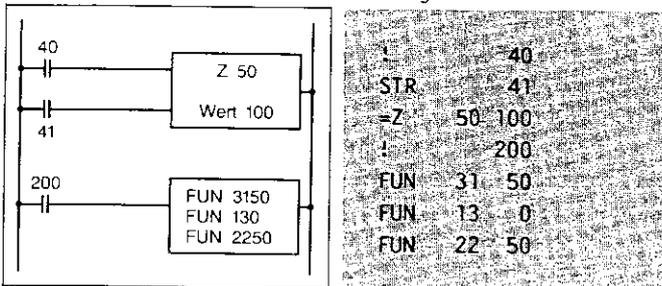


Bild 10.9

Sobald die Startbedingung (Merker 200 hat "1"-Signal) erfüllt ist, wird der Istwert des Zählers 50 mit FUN 31 in das Arithmetik-Register übernommen. Dieser Wert wird mit dem Zustand der Eingänge 0 bis 17 multipliziert (FUN 13) und im Arithmetik-Register abgelegt. Mit FUN 22 wird der Wert des Arithmetik-Registers an die Ausgänge 50 bis 67 weitergegeben.

Einschränkung: Ist die Multiplikation zweier Zahlen (z. B. 900 x 700) > 9999, so wird die Zahl im weiteren Programm berücksichtigt, mit der eine Zahl multipliziert werden soll.

10.4 FUN 14 - Division

Beispiel 1:

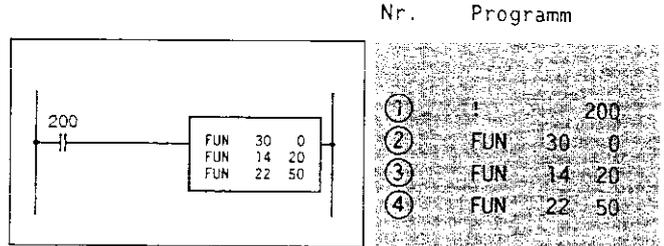


Bild 10.10

Anmerkung zu Nr. ① - ④:

- ①: ! 200 --> Startbedingung
- ②: 0-17 --> AR
- ③: AR : (20-37) --> AR
- ④: AR --> 50-67

Der Befehl FUN 14 wird eingesetzt, wenn der Wert des Arithmetikregisters (AR) durch einen anderen BCD-Wert dividiert werden soll.

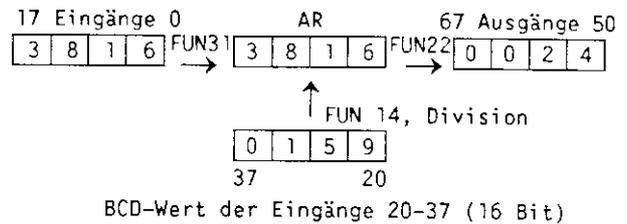


Bild 10.11

In obigem Beispiel werden die BCD-Daten der Eingänge 0-17 mit FUN 30 in das Arithmetikregister (AR) übernommen, wenn die Startbedingung (Merker 200 hat "1"-Signal) erfüllt ist.

Mit FUN 14 wird der BCD-Wert des AR durch den aktuellen Wert, der an den Eingängen 20-37 anliegt, dividiert und im AR abgelegt. Mit FUN 22 wird der Wert des AR an die Ausgänge 50-67 gegeben.

Beispiel 2:

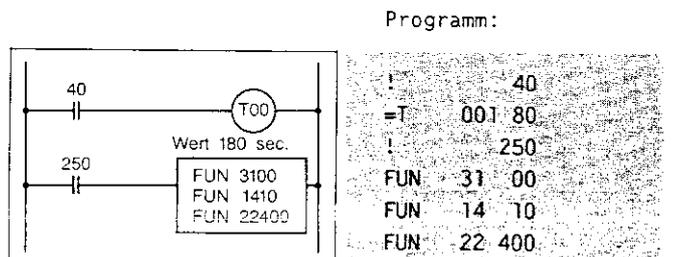


Bild 10.12

Sobald die Startbedingung (Merker 250 hat "1"-Signal) erfüllt ist, wird der Istwert des Zeitgliedes T00 mit FUN 31 in das Arithmetik-Register übernommen.

Dieser Wert wird durch den Zustand der Eingänge 10 bis 27 mit FUN 14 dividiert und im Arithmetik-Register abgelegt.

Mit FUN 22 wird der Wert des Arithmetik-Registers an die Merker 400 bis 417 weitergegeben.

Einschränkungen: a) Bei einer Division durch 0 wird ein Übertragungsmerker (carry flag C) gesetzt. Mit dem Wert, der durch eine Zahl dividiert werden sollte, wird weitergearbeitet, d. h. der Wert wird unbearbeitet durch FUN 22 an die Ausgänge 50-67 weitergegeben.

b) Liefert die Division zweier Zahlen ein Ergebnis, das keine ganze Zahl ist (z. B. $19 : 8 = 2,275$), so wird das Ergebnis zur nächst kleineren ganzen Zahl abgerundet (hier 2).

10.5 FUN 24 – FUN 25, Umwandlung BNR-BCD und BCD-BNR

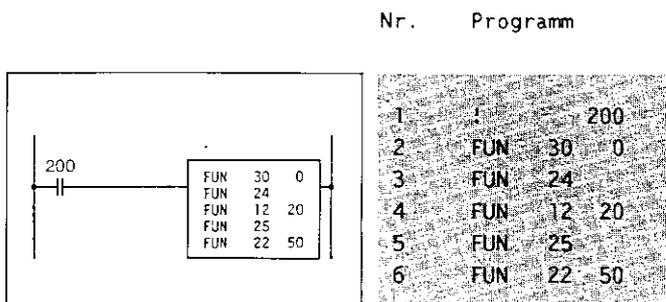


Bild 10.13

Anmerkung zu Nr. ① – ⑥:

- ①: ! 200 --> Startbedingung
- ②: 0-17 --> AR
- ③: AR (BNR) --> AR (BCD)
- ④: AR (20-37) --> AR
- ⑤: AR (BCD) --> AR (BNR)
- ⑥: AR -- (50-67)

Der Befehl FUN 24 wird eingesetzt, um binäre Daten in BCD-Daten umzuwandeln.

Der Befehl FUN 25 wird eingesetzt, um BCD-Daten in binäre Daten umzuwandeln.

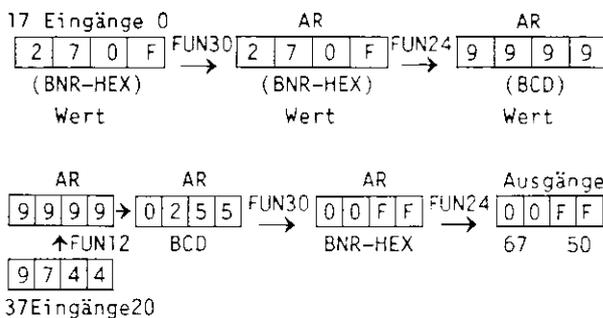


Bild 10.14

Wenn der BCD-Wert bei Umwandlung des Binärwertes größer als 9999 wird, findet keine Umwandlung statt. Das Carry-Flag C wird gesetzt.

10.6 Status des Übertragungsmerkers (Carry Flag) bei den Rechenbefehlen

Folgende Tabelle zeigt den Wert des carry flag C bei den Rechenbefehlen.

Befehl	AR oder Eingabewert		Ergebnis		
	BNR	BCD	<0	0-9999	>9999 bzw. :0
FUN 11 ADD	C = 0	C = 0	-	C = 0	C = 1
FUN 12 SUB	C = 0	C = 0	C = 1	C = 0	-
FUN 13 MUL	C = 1	C = 0	-	C = 0	C = 1
FUN 14 DIV	C = 1	C = 0	-	C = 0	C = 1
FUN 24 BNR-BCD	C = 0	-	-	C = 0	C = 1
FUN 25 BCD-BNR	C = 1	C = 0	-	-	-

Die mittlere Befehlsausführungszeit der Rechenbefehle entnehmen Sie bitte der Tabelle auf Seite 1-5 in dieser ABB Procontic K200-Software-Beschreibung.

10.7 Programmbeispiel für eine arithmetische Gleichung

Zuordnung der Analog-Eingabe-/Ausgabe-Module:

Das folgende Bild zeigt die Konfiguration einer ABB Procontic K200 Grundausbaustufe, die mit zwei Analogmodulen erweitert wurde.

Grundausbaustufen	07 EA 200	07 AA 200
07 KR 220, 07 KR 228	Analog-Eingabe-Modul	Analog-Ausgabe-Modul
07 KT 228, 07 KR 240		
07 KR 264		

Bild 10.15

Eingabe Kanal 0
 100-107 110-117
 Analog- unbenutzt
 Wert

Ausgabe Kanal 1
 160-167 170-177
 Analog- unbenutzt
 Wert

Arithmetische Gleichung: $Y = X * A : B$

X: Analog-Eingabe, Kanal 0 zugeordnet

Y: Analog-Ausgabe, Kanal 1 zugeordnet

A: Konstante, Wert: 5

B: Konstante, Wert: 25

FUN 0. 5	Wert 5 ins AR (8-bit BCD)
FUN 22 200	AR --> 200-217
FUN 0. 25	Wert 25 in AR (8-bit BCD)
FUN 22 220	AR --> 220-237
FUN 30 100	Analogwert-Eingabe (Kanal 0) ins AR (8-bit BCD)
FUN 24	Umwandlung von BNR in BCD des Analogwerts
FUN 13 200	AR*5 (BCD Merker 200-217) --> AR (8-bit BCD)
FUN 14 220	AR:25 (BCD Merker 220-237)
FUN 25	Umwandlung des BCD-Werts des AR-Inhalt in einen BNR-Wert (8-bit BNR)
FUN 22 160	Ausgabe Wert des AR mit FUN 22 auf Analog-Ausgang 160-167 (Kanal 1)

Bild 10.16

10.8 3-Punkt-Regler mit Hysterese

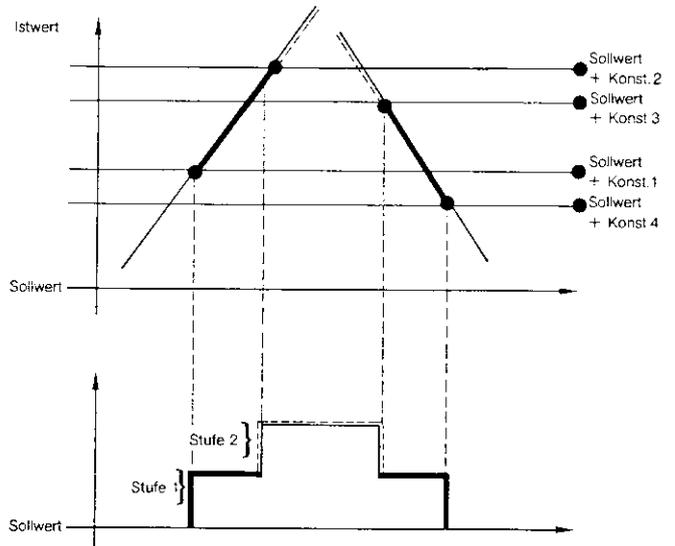


Bild 10.17

Merkerbereich 1	Istwert	Merker 300-317
Merkerbereich 2	Sollwert	Merker 320-337
Merkerbereich 3	Sollwert + Konstante 1	Merker 340-357
Merkerbereich 4	Sollwert + Konstante 2	Merker 360-377
Merkerbereich 5	Sollwert + Konstante 3	Merker 240-257
Merkerbereich 6	Sollwert + Konstante 4	Merker 260-277

Freigabe = Startbedingung = Eingang 2

Programm:

!	2	Startwert
FUN	30 100	Istwert analog einlesen in AR
FUN	24	Umwandlung BNR-BCD
FUN	22 300	Istwert BCD
!	2	Startbedingung
FUN	30 120	Sollwert analog einlesen in AR
FUN	24	Umwandlung BNR-BCD
FUN	22 320	Sollwert BCD
!	2	Startbedingung
FUN	0. 40	Konstante 1
FUN	11 320	Addition Sollwert + Konstante 1
FUN	22 340	Sollwert + Konstante 1

```

!           2      Startbedingung
FUN  0.  50      Konstante 2
FUN  11 320      Addition Sollwert + Konstante 2
FUN  22 360      Sollwert + Konstante 2

```

```

!           2      Startbedingung
FUN  0.  30      Konstante 3
FUN  11 320      Addition Sollwert + Konstante 3
FUN  22 240      Sollwert + Konstante 3

```

```

!           2      Startbedingung
FUN  0.  45      Konstante 4
FUN  11 320      Addition Sollwert + Konstante 4
FUN  22 260      Sollwert + Konstante 4

```

```

!           2      Startbedingung
FUN  30 340      Ergebnis Sollwert + Konstante 1
FUN  34 300      Vergleich mit Istwert
=           400      Merker 400 "1"-Signal, wenn
                    Istwert ≥ Sollwert + Konstante 1
!           2      Startbedingung
FUN  30 360      Ergebnis Sollwert + Konstante 2
FUN  34 300      Vergleich mit Istwert
=           401      Merker 401 "1"-Signal, wenn
                    Istwert ≥ Sollwert + Konstante 2

```

```

!           2      Startbedingung
FUN  30 300      Istwert (BCD)
FUN  34 240      Vergleich mit Sollwert + Konst. 3
=           402      Merker 402 "1"-Signal, wenn
                    Istwert ≤ Sollwert + Konstante 3

```

```

!           2      Startbedingung
FUN  30 300      Istwert (BCD)
FUN  34 260      Vergleich mit Sollwert + Konst. 4
=           403      Merker 403 "1"-Signal, wenn
                    Istwert ≤ Sollwert + Konstante 4

```

Stufe 1 wird über RS-Flip-Flop (FUN 03) durch Merker 400 gesetzt (eingeschaltet) und durch Merker 403 wieder zurückgesetzt.

```

!           400      setzen (einschalten)
STR          403      rücksetzen (ausschalten)
FUN           03      R/S-Speicher
=            50

```

Stufe 2 wird über RS-Flip-Flop (FUN 03) durch Merker 401 gesetzt und durch Merker 402 wieder zurückgesetzt.

```

!           401      setzen (einschalten)
STR          402      rücksetzen (ausschalten)
FUN           03      R/S-Speicher
=            51

```

Merker 400 wird "1"-Signal, sobald der Istwert die Schwelle (Sollwert + Konstante 1) erreicht bzw. überschreitet.

Merker 401 wird "1"-Signal, sobald der Istwert die Schwelle (Sollwert + Konstante 2) erreicht bzw. überschreitet.

Merker 402 wird "1"-Signal, sobald der Istwert die Schwelle (Sollwert + Konstante 3) erreicht bzw. unterschreitet.

Merker 403 wird "1"-Signal, sobald der Istwert die Schwelle (Sollwert + Konstante 4) erreicht bzw. unterschreitet.

10.9 Ein- und Auslesen von Analogwerten

Beispiel 1: Einlesen eines Analogwertes vom Analogeingang (Kanal 0) und Ausgabe des Analogwertes auf Analogausgang Kanal 0.

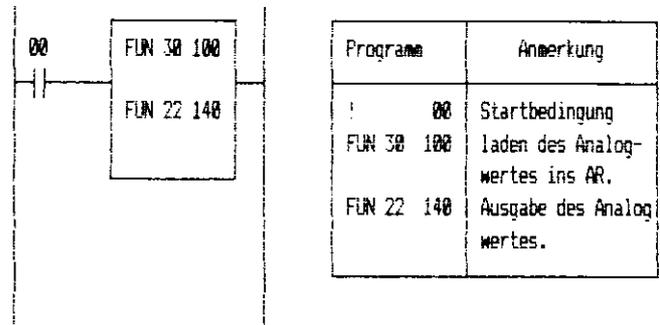


Bild 10.18

Beispiel 2: Einlesen eines Analogwertes vom Analogeingang (Kanal 1) und Ausgabe des Analogwertes auf Analogausgang Kanal 1.

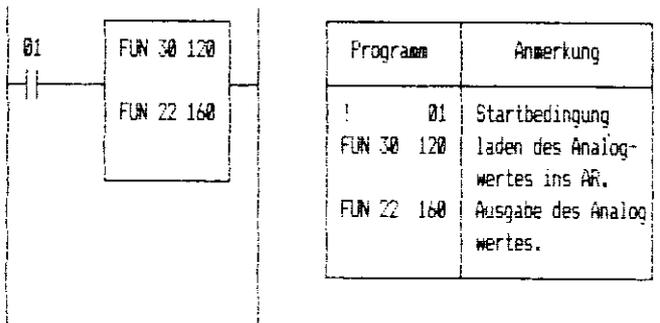


Bild 10.19

10.10 Addition Konstante (BCD) zu Analogwert – Ausgabe des neuen Wertes als Analogwert

Im folgenden Beispiel wird zu einem Analogwert eine Konstante im BCD-Format addiert. Der daraus resultierende Wert wird als Analogwert auf einen Analogausgang ausgegeben.

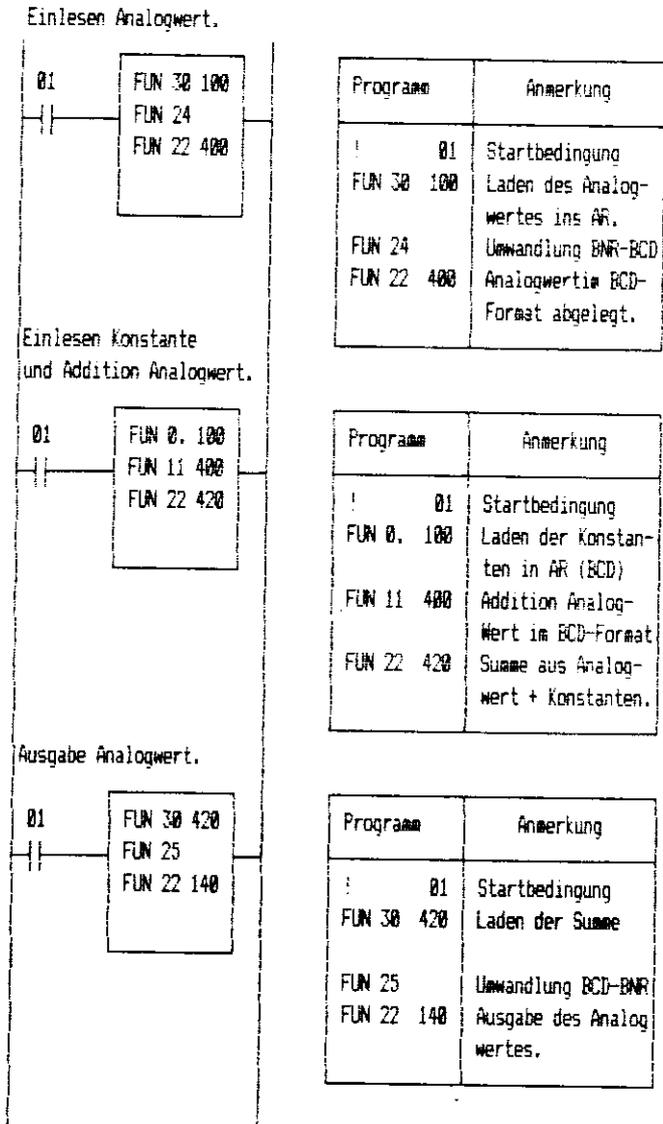


Bild 10.20

10.11 Addition BCD-Wert zu Analogwert – Ausgabe des Wertes auf Analogausgang bzw. Ausgabe im BCD-Format

Im folgenden Beispiel wird ein BCD-Wert zu einem Analogwert addiert. Das Ergebnis wird auf einen Analogausgang gegeben bzw. im BCD-Format ausgegeben.

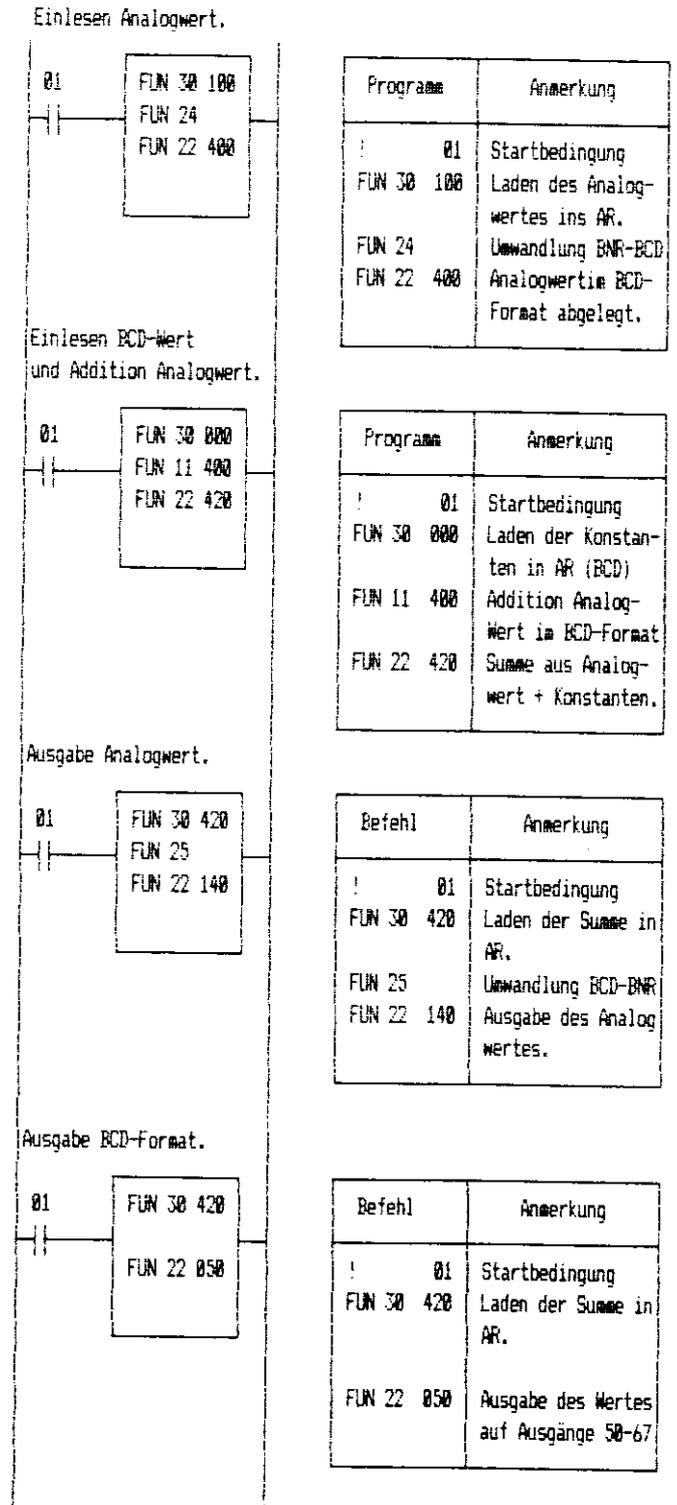


Bild 10.21

10.12 Begrenzung eines Analogwertes auf eine obere Grenze (z. B. 8 V)

a) Nach erfolgter Startbedingung (Eingang 01 hat "1"-Signal) wird mit FUN 30 der Analogwert in das AR eingelesen. Mit FUN 24 wird der Wert in das BCD-Format umgewandelt. Durch FUN 22 wird der gewandelte Analogwert im BCD-Format ausgegeben (in diesem Beispiel auf die Merker 300-317).

```
!      01      Startbedingung
FUN  30 100    Analogwert einlesen
FUN   24      Umwandlung von BNR in BCD
FUN   22 300    Ausgabe auf Merker 300-317
```

b) Der gewandelte Analogwert wird wieder in das AR mit FUN 30 eingelesen. Es erfolgt ein Vergleich mit der Konstanten 205, die 8 Volt entspricht (255=10 Volt). Ist der Wert \geq 8 Volt, so erfolgt keine Analogausgabe. Ausgang 50 wird gesetzt.

```
!      01      Startbedingung
FUN  30 300    gewandelten Analogwert einlesen
FUN   7. 205    Vergleich auf  $\geq$  mit Konstante 205 (entspricht 8 Volt)
=      50      Ausgang 50
```

c) Unterprogramm für Analogwert kleiner 8 Volt: Sobald der Analogwert kleiner 8 Volt ist, wird der eingelesene Analog-Istwert direkt ausgegeben.

```
!      50      Wert ist  $\leq$  8 Volt
FUN   06      Sprung-Anfang
!      01      Startbedingung
FUN  30 300    gewandelten Analogwert einlesen
FUN   25      Umwandlung BCD-BNR
FUN   22 140    Ausgabe Analogwert
FUN   07      Sprung-Ende
```

d) Unterprogramm für Analogwert größer 8 Volt: Sobald der Analogwert größer 8 Volt ist, wird der Wert der Konstanten 205 (entspricht 8 V) mit FUN 25 gewandelt (BCD-BNR) und ausgegeben.

```
!N     50      Wert ist  $\geq$  8 Volt
FUN   06      Sprung-Anfang
!      1       Startbedingung
FUN   0. 205    Konstante 205 (8 V) laden
FUN   25      Umwandlung BCD-BNR
FUN   22 140    Ausgabe Analogwert begrenzt auf 8 V
FUN   07      Sprung-Ende
```

10.13 Begrenzung eines Analogwertes auf eine obere und untere Grenze (Obergrenze 8 V, Untergrenze 2 V)

a) Nach erfolgter Startbedingung (Eingang 01 hat "1"-Signal) wird mit FUN 30 der Analogwert in das AR eingelesen. Mit FUN 24 wird der Wert in das BCD-Format umgewandelt. Durch FUN 22 wird der gewandelte Analogwert im BCD-Format ausgegeben (in diesem Beispiel auf die Merker 300-317).

```
!      01      Startbedingung
FUN  30 100    Analogwert einlesen
FUN   24      Umwandlung von BNR in BCD
FUN   22 300    Ausgabe auf Merker 300-317
```

b) Der gewandelte Analogwert wird in das AR mit FUN 30 eingelesen. Mit FUN 25 wird der Wert in das BNR-Format gewandelt. Mit FUN 22 erfolgt die Ausgabe auf den Analogausgang.

```
!      01      Startbedingung
FUN  30 300    gewandelten Analogwert einlesen
FUN   25      Umwandlung BCD-BNR
FUN   22 140    Analogausgabe
```

c) Es erfolgt nun ein Vergleich mit der Konstanten 205 (Obergrenze 8 Volt). Wenn der Wert \geq 8 Volt ist, wird der Ausgang 52 gesetzt.

```
!      01      Startbedingung
FUN  30 300    gewandelten Analogwert einlesen
FUN   7. 205    Vergleich Konstante 205 (8 V)
=      52      Ausgang 52 wird gesetzt, wenn der Analogwert  $\geq$  8 Volt ist
```

d) Es erfolgt nun ein Vergleich mit der Konstanten 052 (Untergrenze 2 Volt). Wenn der Wert \leq 2 Volt ist, wird der Ausgang 50 gesetzt.

```
!      01      Startbedingung
FUN  30 300    gewandelten Analogwert einlesen
FUN   9. 052    Vergleich Konstante 052 (2 V)
=      50      Ausgang 50 wird gesetzt, wenn der Analogwert  $\leq$  2 Volt ist
```

e) Abfrage des Analogwertes innerhalb des Bereichs 2-8 Volt.

```
!N     50      Analogwert ist kleiner 2 Volt
&N    52      Analogwert ist größer 8 Volt
=      51      Analogwert innerhalb des Bereiches 2-8 Volt
```

11 Meldesteuerungen

11.1 Meldung mit Dauerlicht

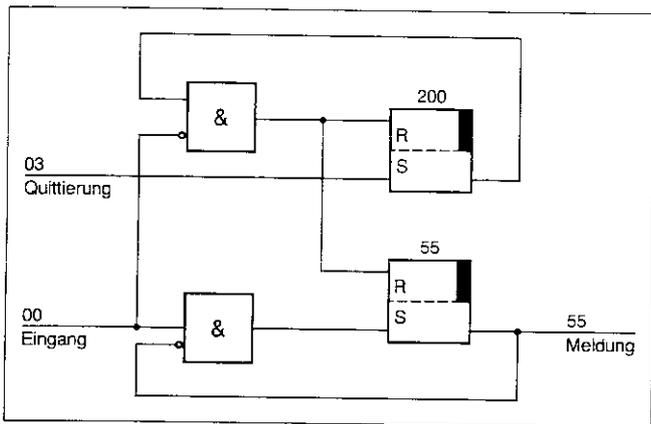


Bild 11.1

Der zu meldende Betriebszustand an Eingang 00 wird durch Dauerlicht an Ausgang 55 angezeigt. Steht beim Quittieren an Eingang 03 die Meldung noch an, bleibt das Dauerlicht erhalten, bis die Meldung verschwindet. Im anderen Fall geht die Lampe unmittelbar nach der Quittierung aus.

Funktionsdiagramm

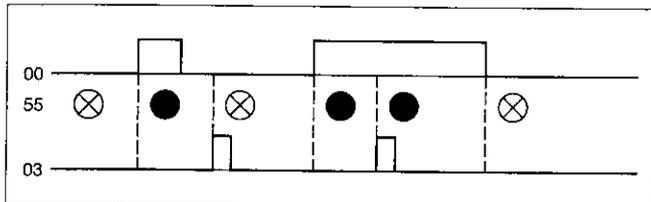


Bild 11.2

Programm:

```

!      00
&N    55
FUN    02
=      55
!      03
FUN    02
=      200
!      200
&N    00
FUN    02
=N     55
=N     200
    
```

11.2 Neuwertmeldung mit Einfachblinklicht

Jede neue Meldung am Signaleingang 00 wird am Ausgang 54 durch Blinken, (0,5 Hz) angezeigt (FUN 774). Im gleichen Moment erfolgt eine akustische Meldung am Ausgang 50. Wird die Quittiertaste 01 betätigt, so geht das Blinklicht in Dauerlicht über. Steht die Meldung nicht mehr an, erlischt der Sichtmelder am Ausgang 54.

Die akustische Meldung kann nur mit der eigenen Quittier-taste an Eingang 02 gelöscht werden.

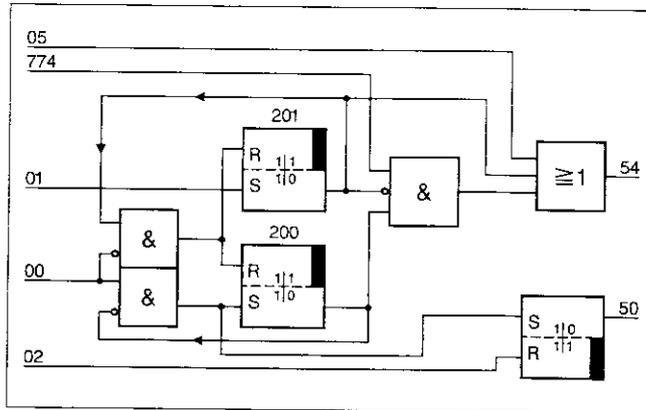


Bild 11.3

Funktionsdiagramm

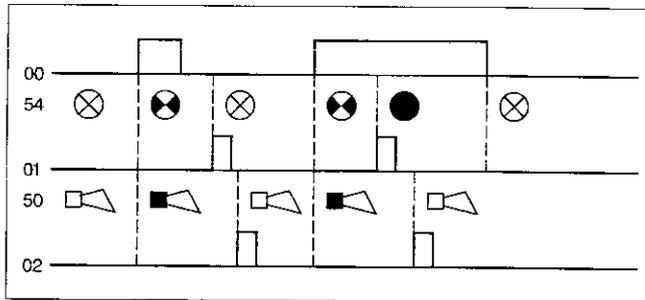


Bild 11.4

Programm:

```

!      00
&N    200
FUN    02
=      200
FUN    02
=      50
!      01
FUN    02
=      201
!N     00
&      201
FUN    02
=N     200
FUN    02
=N     201
!      200
&N    201
&      774
/      201
/      05
=      54
!      02
FUN    02
=N     50
    
```

Mit Eingang 05 kann eine Lampenprüfung vorgenommen werden, d. h. wird Eingang 05 auf „1“ geschaltet während 54 blinkt, so geht das Blinklicht in Dauerlicht über.

11.3 Neuwertmeldung mit Doppelblinklicht

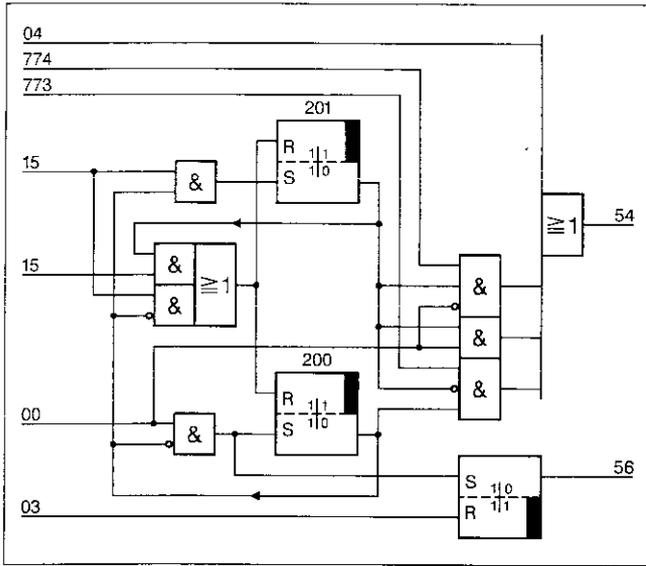


Bild 11.5

- 00 - Signaleingang, Meldung
- 15 - Quittierung Meldung
- 10 - Quittierung, Meldespeicher (Löschen)
- 773 - Blinkschiene, 5 Hz
- 774 - Blinkschiene, 0,5 Hz
- 04 - Lampenprüfen
- 03 - Quittierung Hörmelder
- 54 - Sichtmeldeausgang
- 56 - Hörmeldeausgang

Jede neue Meldung am Signaleingang 00 wird am Ausgang 54 durch Blinken (5 Hz) angezeigt. Im gleichen Moment erfolgt auch die akustische Meldung an Ausgang 56. Nach Betätigen der Quittiertaste 15 geht, wenn die Meldung nicht mehr ansteht, das Blinklicht in langsames Blinken (0,5 Hz) über. Steht die Meldung noch an, so geht das Blinklicht in Dauerlicht über. Das langsame Blinken kann mit der Quittiertaste 10 für den Meldespeicher gelöscht werden.

Die akustische Meldung an Ausgang 56 wird mit der eigenen Quittiertaste (03) gelöscht.

Funktionsdiagramm

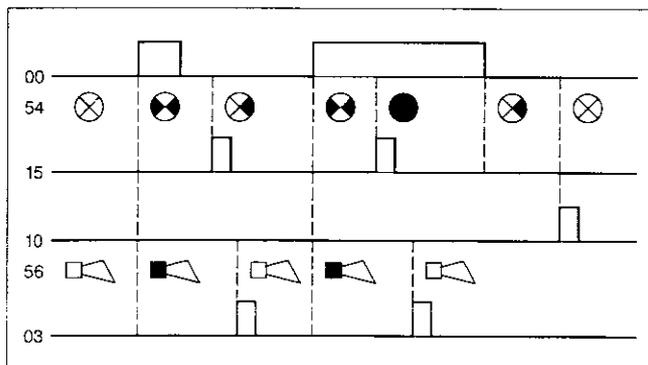


Bild 11.6

Programm:

```

!      00
&N    200
FUN   02
=      200
FUN   02
=      56
!      15
&     200
FUN   02
=      201
!      15
&N    200
STR   201
&     10
/      STR
FUN   02
=N     200
FUN   02
=N     201
!      200
&N    201
&     773
STR   201
&     00
STR   201
&N    00
&     774
/      STR
/      STR
/      04
=      54
!      03
FUN   02
=N     56
    
```

Wird ein Oszillator von 2 Hz benötigt, so ist statt & 773, & 211 zu programmieren. Zusätzlich muß folgender Programmteil hinzugefügt werden (siehe auch Kapitel Zeitfunktionen):

Programm:

```

IN     T00
=      T00.25
!      T00
FUN   00 210
IN     210
FUN   06
IN     211
=      211
FUN   07
    
```

11.4 Erstwertmeldung mit Einfach-Quittung nach DIN 19 235

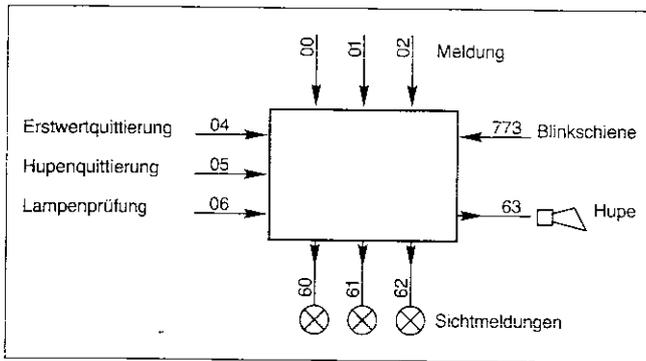


Bild 11.7

Die Erstwertmeldung hebt aus einer Gruppe zu meldender Betriebszustände an den Eingängen 00 – 02 den zuerst eingetretenen hervor. Nur dem zuerst gemeldeten Betriebszustand wird Blinken (5 Hz), allen nachfolgenden Dauerlicht zugeordnet. Die Quittierung erfolgt nur über eine Erstwert-Quittier-Taste 04. Steht nach der Erstwertquittierung die Meldung noch an, geht die Lampe in Dauerlicht über, bis die Meldung verschwunden ist.

Die weiteren Meldungen stehen solange als Dauersignal an, bis nach einer Quittierung der zu meldende Betriebszustand verschwunden ist. Der Hörmelder wird bei jeder Neuwertmeldung zugeschaltet.

Funktionsdiagramm

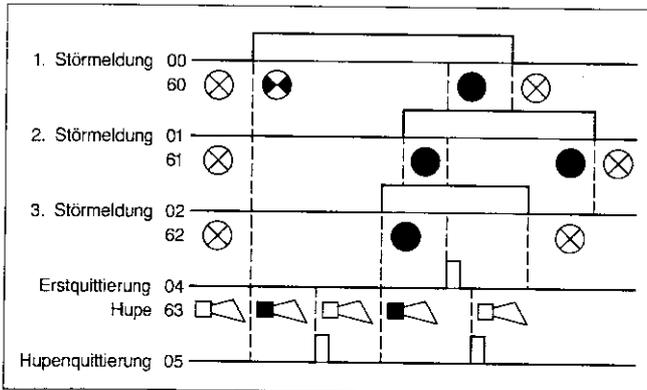


Bild 11.8

Programm:

```

!      00
FUN 00 200
!      01
FUN 00 201
!      02
FUN 00 202
!      200
/      201
/      202
FUN   02
=      63      Hupe EIN

!      04
FUN   02
=N     210
=N     211
=N     212      Quittierung
    
```

```

=N     220
=N     221
=N     222
=N     223
!      200
FUN   02
=      210
!      201
FUN   02
=      211
!      202
FUN   02
=      212

!      773
/      220
&     210
STR   00
&N   210
STR   06
/      STR
/      STR
=      60
!      773
/      221
&     211
STR   01
&N   211
STR   06
/      STR
/      STR
=      61
!      773
/      222
&     212
STR   02
&N   212
STR   06
/      STR
/      STR
=      62
!      05
FUN   02
=N     63
!      210
/      212
&N   223
FUN   02
=      221
!      210
/      211
&N   223
FUN   02
=      222
!      211
/      212
&N   223
FUN   02
=      220
!      210
/      211
/      212
FUN   02
=      223
    
```

Erstwert

Sichtmelder

Hupenquittierung

Auswertung Erstwert

11.5 Erst- und Neuwertmeldung mit Zweifach-Quittierung nach DIN 19 235

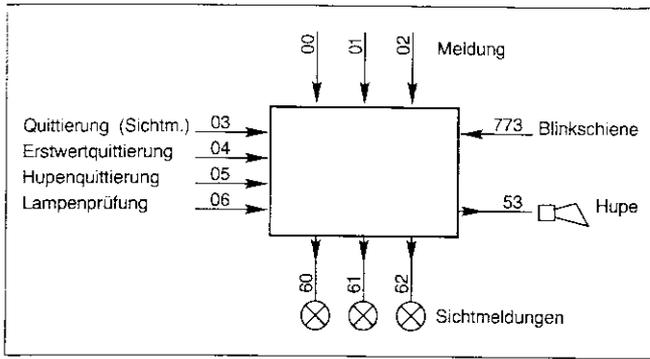


Bild 11.9

Bei der Erst- und Neuwertmeldung werden alle zu meldenden Betriebszustände mit Blinklicht (5 Hz) als Neuwert angezeigt. Nach Quittierung des Sichtmelders an Eingang 03 blinkt nur noch der Erstwertmelder. Die anderen Meldungen bleiben solange als Dauerlicht anstehen, wie der jeweilige zu meldende Betriebszustand ansteht. Der Erstwert kann nur über die Erstwertquittiertaste 04 quittiert werden und bleibt dann solange als Dauerlicht anstehen, wie die Meldung noch ansteht.

Funktionsdiagramm

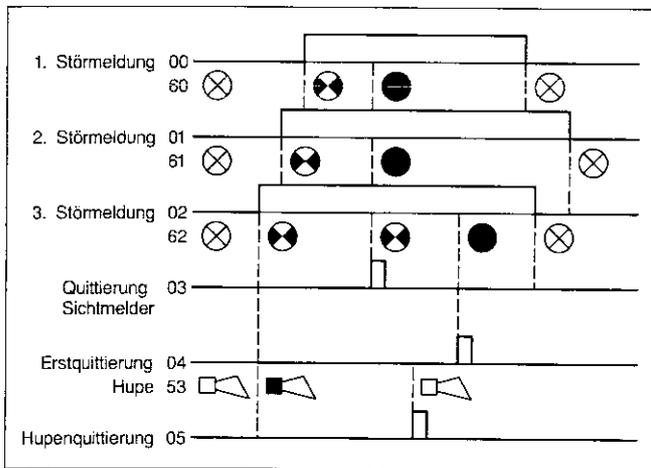


Bild 11.10

Programm:

```

!      00
FUN 00 200
!      01
FUN 00 201
!      02
FUN 00 202
!      200
/      201
/      202
FUN    02
=      53
    
```

```

!      03
&      220
FUN    02
=N     210
!      03
&      221
FUN    02
=N     211
!      03
&      222
FUN    02
=N     212
    
```

Quittierung
Sichtmelder

```

!      200
FUN    02
=      210
!      201
FUN    02
=      211
!      202
FUN    02
=      212
    
```

```

!      773
&      210
STR    00
&N    210
STR    06
/      STR
/      STR
=      60
!      773
&      211
STR    01
&N    211
STR    06
/      STR
/      STR
=      61
!      773
&      212
STR    02
&N    212
STR    06
/      STR
/      STR
=      62
    
```

Sichtmelder

```

!      05
FUN    02
=N     53
    
```

Hupenquittierung

!	210	↑
/	212	
&N	223	
FUN	02	
=	221	
!	210	
/	211	
&N	223	
FUN	02	
=	222	
!	211	Auswertung Erstwert
/	212	
&N	223	
FUN	02	
=	220	
!	210	
/	211	
/	212	
FUN	02	
=	223	↓
!	04	↑
FUN	02	
=N	220	
=N	221	
=N	222	
=N	223	
!	03	
FUN	02	Quittierung Erstwert
=	204	
!	04	
&	204	
FUN	02	
=N	210	
=N	211	
=N	212	
=N	204	↓

12 Wortbefehle 16 Bit

12.1 Funktion 30 (Lade Wort), Funktion 32 (Ausgabe Wort)

Programm:

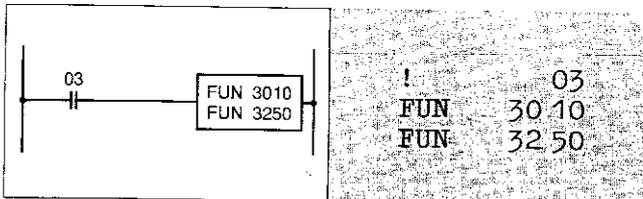


Bild 12.1

Mit einem „0-1“-Impuls an Eingang 03 wird der Status der Eingänge 10 bis 27 in das Arithmetik-Register übernommen (mit FUN 30) und mit FUN 32 an die Ausgänge gegeben, und zwar ab Ausgang 50 bis Ausgang 67. Nach Ausschalten („0“-Signal) am Eingang 03, bleiben die Ausgänge gesetzt, bis ein „1“-Signal am Eingang STOP den Programmablauf unterbricht oder eine Spannungsunterbrechung das System zurücksetzt. Oder aber es wird eine neue Information an den Eingängen 10 bis 27 mit Eingang 03 eingegeben. Bei einem Dauersignal an Eingang 03 wird der geänderte Wert immer sofort an die Ausgänge gegeben.

12.2 Funktion 31 (Lade Wort von T/Z), Funktion 32 (Ausgabe Wort)

Programm:

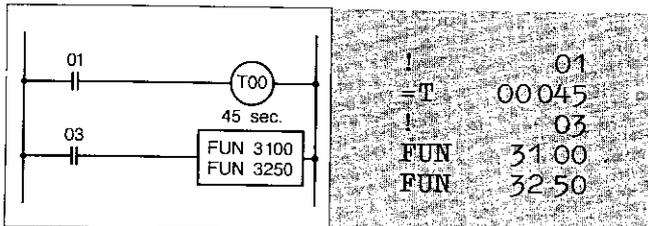


Bild 12.2

Mit einem „0-1“-Impuls am Eingang 03 wird der Wert des Zeitgliedes T00 in das Arithmetikregister übernommen (mit FUN 31) und mit FUN 32 an die Ausgänge 50 bis 67 gegeben (gespeichert).

Mit einem Dauersignal („1“-Signal) am Eingang 03 wird der jeweils aktuelle Zeitwert sofort an die Ausgänge gegeben (4 Dekaden BCD).

12.3 Funktion 30 (Lade Wort), Funktion 33 (Lade (T/Z))

Programm:

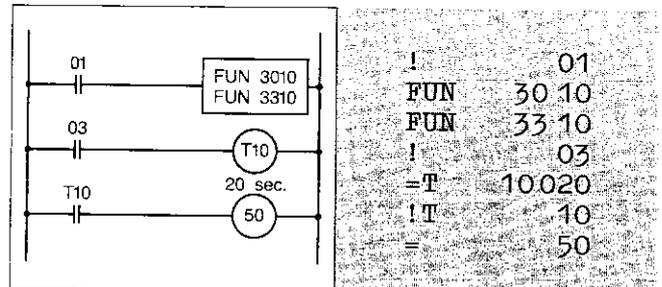


Bild 12.3

Mit einem „0-1“-Impuls am Eingang 01 wird der Status der Eingänge 10 bis 27 in das Arithmetik-Register übernommen (mit FUN 30).

Mit FUN 33 wird der Inhalt des Arithmetik-Registers als Sollwert dem Zeitglied (in oben aufgeführtem Beispiel Nr. 10) übergeben. Während des Impulses an 01 kann das Zeitglied T10 nicht gestartet werden.

Nach Ablauf der Zeit wird Ausgang 50 gesetzt („1“-Signal).

Bei Zyklus-Start ist der programmierte Wert des Zeitgliedes gültig (ohne Impuls an 01).

Achtung: Die Informationen an den Eingängen müssen BCD spezifisch sein.

12.4 Funktion 36 (Lade Inhalt schneller Zähler), Funktion 32 (Ausgabe Wort)

Programm:

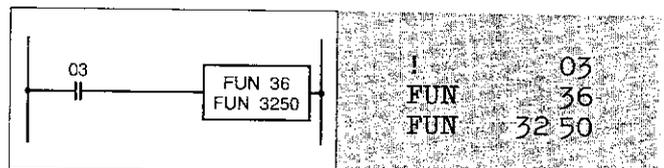


Bild 12.4

Mit einem „0-1“-Impuls am Eingang 03 wird der jeweilige Istwert des schnellen Zählers in das Arithmetik-Register übernommen (mit FUN 36). Mit FUN 32 wird dieser Wert an die Ausgänge 50 – 67 gegeben (gespeichert).

Mit einem Dauersignal an 03 ist der jeweils gültige Zählerstand dauernd an den Ausgängen sichtbar.

Der schnelle Zähler ist ein Vorwärtszähler (10 kHz Zählerfrequenz, 4 Dekaden) mit dem externen Zähleringang HZ und dem Löscheingang HR.

12.5 Funktion 31 (Lade von T/Z),
Funktion 34 (Vergleiche Wort auf \leq)

Programm:

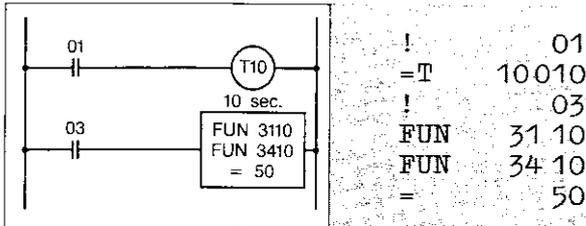


Bild 12.5

Mit einem „0-1“-Impuls an Eingang 03 wird der Wert des Zeitgliedes T10 in das Arithmetik-Register übernommen (mit FUN 31) und mit FUN 34 mit dem Status der Eingänge 10 bis 27 verglichen. Ist der Wert der Eingänge 10 bis 27 größer oder gleich dem Wert des Zeitgliedes so wird Ausgang 50 geschaltet („1“-Signal). Ist der Wert der Eingänge kleiner als der Wert des Zeitgliedes, so bleibt der Ausgang 50 in Ruhestellung („0“-Signal).

Mit einem Dauersignal („1“-Signal) an 03 und Ablauf des Zeitgliedes wird ein dauernder Vergleich zwischen Eingängen und den Zeitgliedern durchgeführt.

Beim Ausschalten von Eingang 3 („0“-Signal) wird der Zustand an Ausgang 50 gespeichert.

12.6 Funktion 31 (Lade von T/Z),
Funktion 35 (Vergleiche mit T/Z \leq)

Programm:

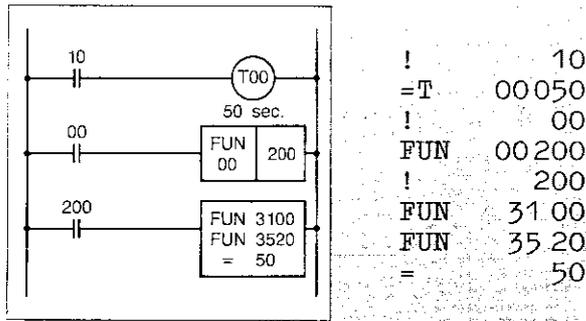


Bild 12.6

Hat der Eingang 10 ein „1“-Signal, dann läuft das Zeitglied T00 an.

Wird der Eingang 00 mit einem „1“-Signal beschaltet, wird mit FUN 00 am Merker 200 ein Impuls gebildet und zwar in der Länge der Zykluszeit.

Mit diesem Impuls wird dann der Istwert des Zeitgliedes T00 in das Arithmetik-Register übernommen. Mit FUN 35 wird der Istwert mit dem Sollwert des Zeitgliedes T20 verglichen.

Ist der Wert des Zeitgliedes T00 kleiner oder gleich T20, dann wird der Ausgang 50 gesetzt („1“-Signal). Wenn der Wert von T00 größer als T20 ist, bleibt der Ausgang 50 in Ruhestellung („0“-Signal).

13 Sonderfunktionen

13.1 Sonderfunktionen 770 bis 777

Die Sonderfunktionen sind softwaremäßig abgelegt unter den Adress-Nummern 770 bis 777. Sie werden auch mit diesen Nummern programmiert und aufgerufen.

Nr. 770 Alle Ausgänge Null setzen (Zuweisung, statisch)

Mit dieser Funktion besteht die Möglichkeit, Eingänge oder Ausgänge (Merker), die nicht zur gleichen Zeit geschaltet sein dürfen, zu überwachen.

Beispiel: Wenn Eingang 0 und Eingang 1 gleichzeitig 1-Signal haben, werden alle Ausgänge zurückgesetzt, und der arithmetische Ablauf wird unterbrochen. In der Anzeige erscheint die Meldung 770 E (s. Bild).

Nach der Korrektur dieses Fehlerzustandes wird nach Betätigen der STO-Taste und anschließendem Betätigen der STA-Taste bzw. nach Aus- und Einschalten der Spannung und Betätigen der STA-Taste das Anwenderprogramm fortgesetzt.

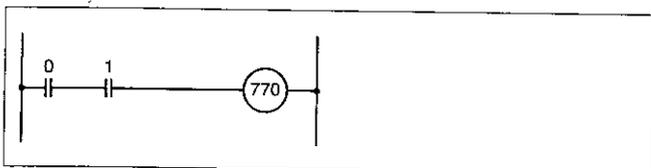


Bild 13.1

Nr. 771 Rücksetzen aller gepufferten Werte wie Merker, Zeiten, Zähler. Diese Funktion wird zu Beginn eines Programms eingesetzt, um alle gepufferten Merker, Zeiten und Zähler zurückzusetzen, z. B. nach Spannungsausfall oder Abschalten, wenn Zustände vor dem Ausfall bzw. Abschalten nicht berücksichtigt werden sollen. Da die Funktion 771 statisch ist, d. h. sie wird einmal pro Programmzyklus ausgeführt, ist es sinnvoll, Nr. 771 in Verbindung mit Nr. 777 (Impuls nach Programmstart) einzusetzen (s. Bild 13.2).

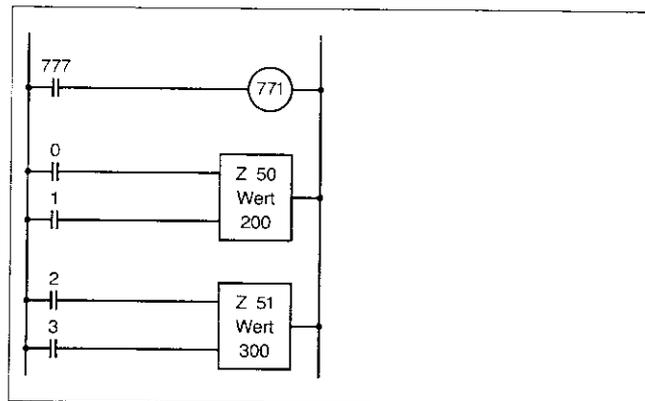


Bild 13.2

Nr. 772 Zyklus-Zeit-Oszillator (1 Zyklus Impuls, 1 Zyklus Pause)

Nr. 773 0,1 s Oszillator

Nr. 774 1 s Oszillator

Nr. 775 0,01 s Oszillator

Achtung: Diese Funktion darf nur eingesetzt werden, wenn die Zykluszeit kleiner als 0,01 s ist.

Nr. 776 1 min. Oszillator

Nr. 777 Ein Impuls nach Starten des Programmes (Zyklus-Zeit, s. Bild)).



Bild 13.3

Bei der Oszillation ist das Impuls-Pausenverhältnis 1:1 (s. Bild).

Beispiel: Nr. 774: 0,5 s Impuls; 0,5 s Pause
Nr. 772: 1 Zyklus Impuls, 1 Zyklus Pause

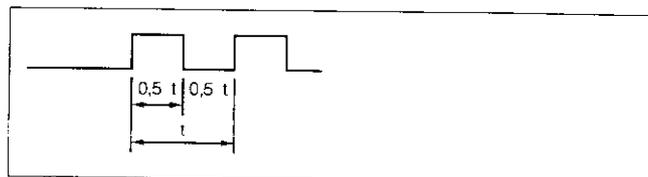


Bild 13.4

14 Programmierung

14.1 Konzept der Programmierung

Bitte beachten Sie nachfolgend aufgeführte Punkte:

a) Die Anzahl der Kontakte für Ein-/Ausgänge, Merker, Zeitglieder und Zähler ist nicht begrenzt, d. h. beliebig viele Kontakte können zugeordnet werden (s. Bild 14.1).

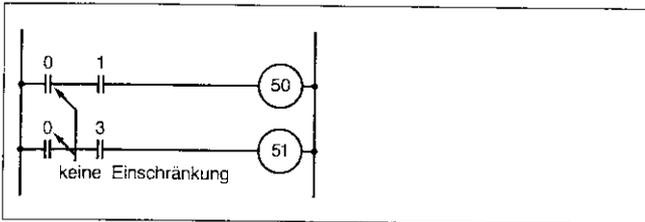


Bild 14.1

b) Die Anzahl der Kontakte, die in Reihe oder Parallel verknüpft werden sollen, ist nicht begrenzt (Bild 14.2).

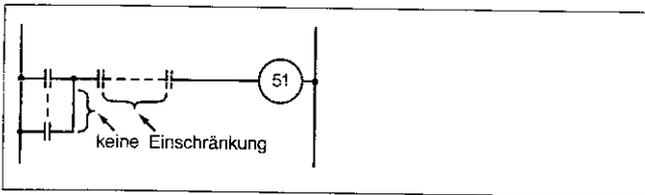


Bild 14.2

c) Ein Ausgang wie auch ein Zeitglied/Zähler kann nicht ohne Eingangsbedingung zugewiesen werden (Bild 14.3 oberer Teil: nicht erlaubte Zuweisung, Bild 14.3 unterer Teil: erlaubte Zuweisung).

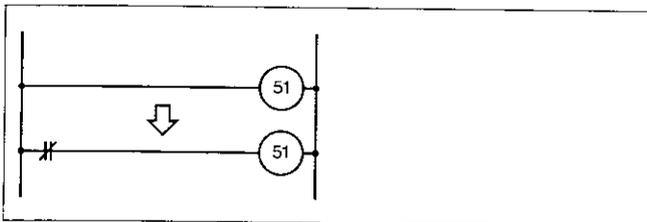


Bild 14.3

d) Nach einer Ausgangszuweisung darf keine weitere Verknüpfung im Programm angehängt werden (Bild 14.4 oberer Teil).

Das zu verknüpfende Element muß vor der Ausgangszuweisung programmiert werden (Bild 14.4 unterer Teil).

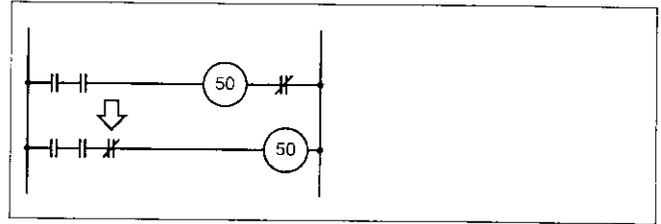


Bild 14.4

e) Ein Ausgang wie auch ein Zeitglied/Zähler darf im Programm nur einmal zugewiesen werden (s. oberer Teil des Bildes 14.5: nicht erlaubte Doppelzuweisung). Muß der Ausgang mehreren Schaltungskonstellationen zugewiesen werden, so ist der Schaltplan (und dementsprechend das Programm) wie im unteren Teil des Bildes 14.5 dargestellt zu ändern.

Anmerkung: Dies gilt jedoch nicht für „FUN 02“, „FUN 03“, „FUN 06“ und „FUN 07“.

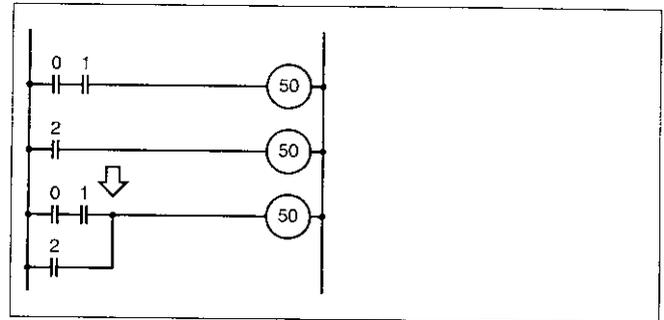


Bild 14.5

f) Die Programmierung muß von links nach rechts vorgenommen werden (siehe Bild 14.1 bis 14.5).

g) Bei der Programmierung von Zählern und Schieberegistern, welche zwei oder mehr Eingänge haben, muß mit spezifischen (festgelegten) Programmierbefehlen gearbeitet werden (s. z.B. Bild 14.6).

h) Werte für Zeitglieder oder Zähler werden bei der Programmierung sofort bei der T/Z-Zuweisung eingegeben (Bild 14.7).

i) Eine Schritt-Nr. (STEP-NR.) oder ein Ein-/Ausgang wird mit einer dreistelligen Zahl programmiert, führende Nullen müssen aber nicht eingegeben werden.

Beispiele: – Eingang 5 kann als 5, 05 oder 005 eingegeben werden. In jedem Fall der Eingabe wird der Eingang mit einer 5 angezeigt.

– Die Schritt-Nr. 050 kann als Schritt 50 oder 050 eingegeben werden.

j) Zeiten kleiner oder gleich 99 s können z. B. als 099 oder 99.0 oder 99 programmiert werden. In der Anzeige erscheint 99.0

Beispiel:	Eingabe	Display
	010	10.0
	15	15.0
	020	20.0

Programm:

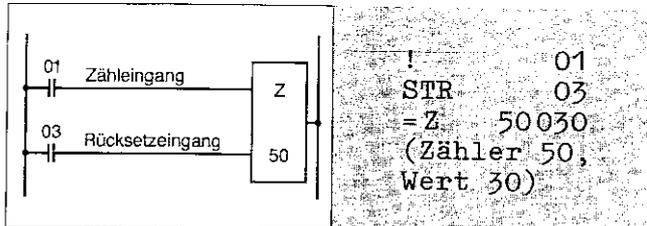


Bild 14.6

Programm:

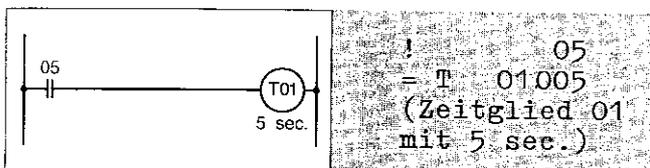


Bild 14.7

14.2 Vorgehensweise bei der Programmierung

- Nehmen Sie zur Erstellung des Programms den Kontaktplan zu Hilfe (Beispiel s. Bild 14.8).
- Unterteilen Sie den Kontaktplan in Blöcke, wobei Sie links oben mit den Startbedingungen beginnen. In Bild 14.9 sind die Blöcke (1) bis (11) angezeigt, wie sie in dem gegebenen Beispiel eingeteilt werden müssen.
- Verknüpfen Sie die einzelnen Blöcke in der Reihenfolge wie in Bild 14.10, 3. Spalte angegeben miteinander, bis Sie sich nach rechts zur Ausgangszuweisung vorgearbeitet haben.

Beispiel:

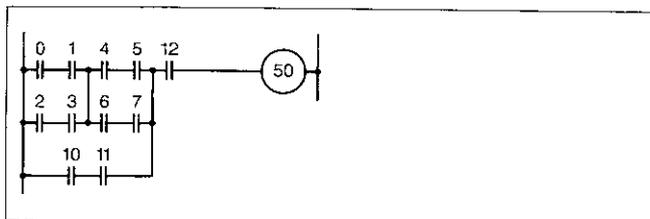


Bild 14.8

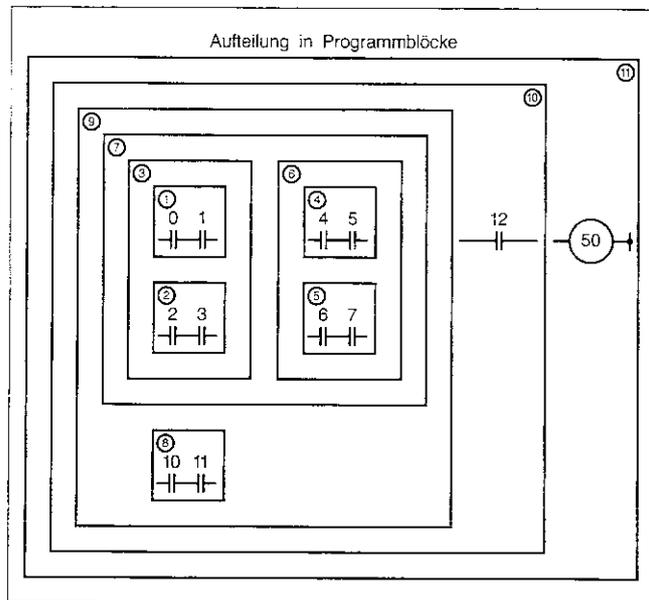


Bild 14.9

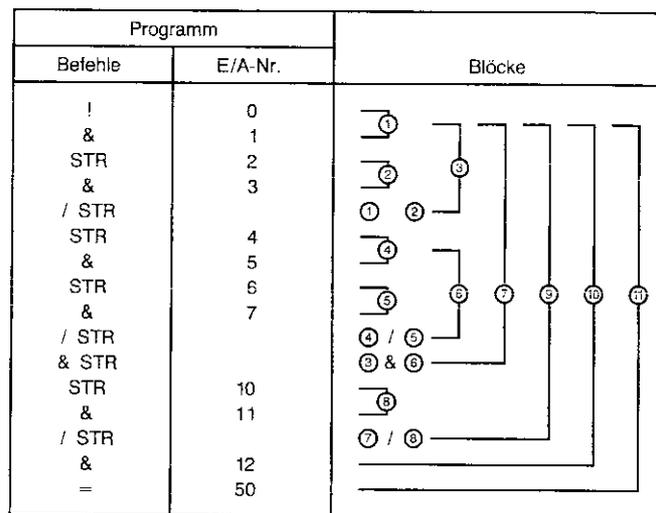


Bild 14.10

14.3 Anwenderprogrammspeicher

- Mit dem im System eingebauten Programmspeicher (EEPROM) können 950 Anweisungen programmiert werden. 1970 Anweisungen können mit den externen Speichermodulen EEPROM 07 PR 201 bzw. EPROM 07 PR 210 programmiert werden. Die Step-Nr. (Schritt-Nr.) beginnt mit dezimal 000.
- Bei Erstellen eines neuen Anwenderprogramms wird immer bei Step 000 begonnen.
- Programmteile, die hinzugefügt werden, beginnen immer an der ersten freien Adresse des Anwenderprogrammspeichers (s. Bild 14.11).
- Ein falscher Programmschritt kann nicht in ein Anwenderprogramm übertragen werden (Fehlermeldung).
- Werden im Programmspeicher Anweisungen eingefügt oder gelöscht, so werden die Step-Nrn. der Anweisungen des bereits bestehenden Anwenderprogramms automatisch erhöht bzw. erniedrigt (s. Bild 14.12 und Bild 14.13).

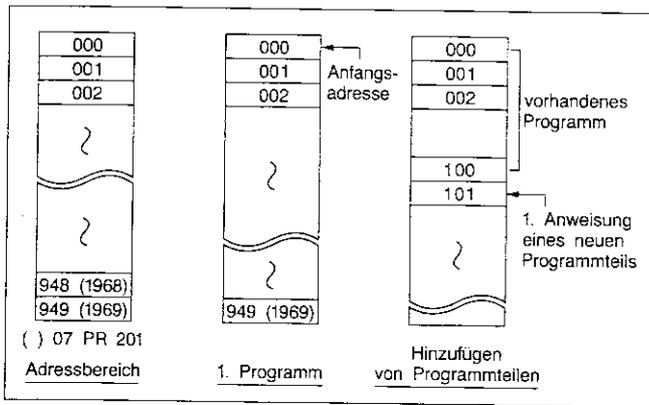


Bild 14.11

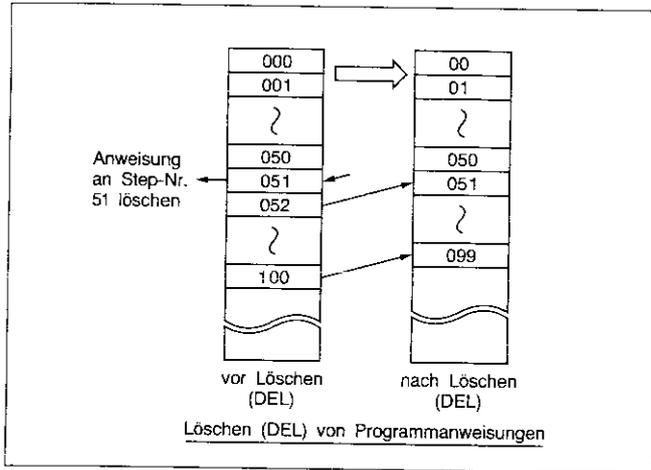


Bild 14.12

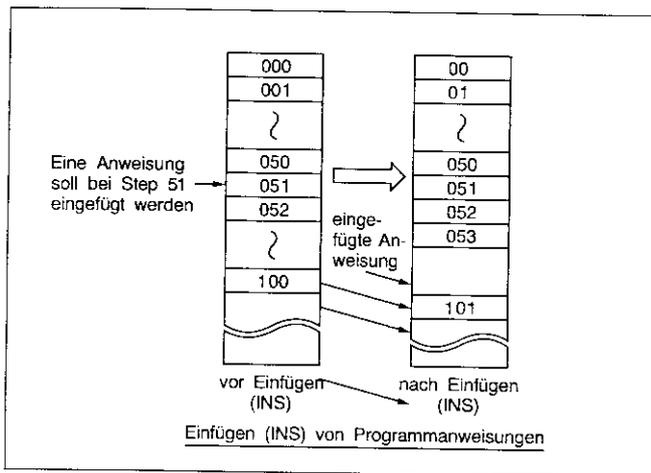


Bild 14.13

14.4 Applikationsbeispiele

	Ersatzschaltung	Programm		Anmerkung
		Befehl		
Parallel-/Reihen- schaltung		!	0	} a
		&	1	
		/	10	} b
		&	2	
		& N	3	
		=	50	
Reihen-/Parallelschaltung		!	0	} a
		& N	1	
		STR	2	} b
		&	3	
		/	10	
		/	4	
		& STR		zwei Blöcke a und b werden verknüpft mit & STR
		=	50	
Reihen-/Parallelschaltung		!	0	} a
		&	1	
		STR	2	} b1
		& N	3	
		STR N	4	} b2
		10	10	
		/ STR		b1 und b2 mit / STR verknüpfen zwei Blöcke a und b werden verknüpft mit & STR
		& STR		
		=	50	
zwei Parallelschaltungen in Reihe		!	0	} a1
		&	1	
		STR	2	} a2
		& N	3	
		/ STR		} b1
		STR N	4	
		&	5	} b2
		STR	6	
		&	7	
		/ STR		b1 + b2 mit / STR verknüpfen Blöcke a + b werden verknüpft mit & STR
		& STR		
		=	50	

Bild 14.14

	Ersatzschaltung	Programm		Anmerkung
		Befehl		
Verlängerung von Zeiten		!	0	
		= T ! T = T ! T =	00090 00 01090 01 50	
		! & N = T ! STR = Z ! T = ! Z =	0 200 200 1 60090 00 200 60 50	
Zeiten/Zähler		= T ! & N = T ! T / & NT =	00010 50 0 01005 00 50 01 50	
		! & NT = T ! T = T =	0 01 00001 00 01003 50	
Ein-/Ausschaltverzögerer				
Blinkgeber				

Bild 14.15

	Ersatzschaltung	Programm		Anmerkung
		Befehl		
Zähler auf einen Wert setzen Auswertung auf 0		! STR STR FUN 40 ! FUN 00 ! FUN 02 = = N = N ! N & N & N & N / & N =	0 1 2 400 200 200 200 400 401 402 403 400 401 402 403 50 200 50	
Sonderfunktion 773, 777		! / & N & N =	0 400 1 777 400	
		! STR = Z	773 0 50050	

Bild 14.16

	Ersatzschaltung	Programm		Anmerkung
		Befehl		
Blocker		!	200	
		& NT	00	
32-bit-Schieberegister		!	417	Zwei 16 bit Schieberegister sollen zu einem 32 bit Register zusammenschaltet werden. Es werden zuerst die oberen 16 bit programmiert.
		STR	1	
V/R-Zähler-Anwendung		!	0	
		STR	1	

Bild 14.17

	Ersatzschaltung	Programm		Anmerkung
		Befehl		
Komplexe Schaltung		!	1	Wenn eine Schaltung zu komplex aufgebaut ist, ist es möglich sie durch Umzeichnen einfacher zu machen.
		STR	2	
Äquivalentschaltungen		&	3	
		STR	4	
Brückenschaltung		STR	5	
		&	6	
		/ STR	7	
		=	50	
		!	1	
		& N	2	
		&	3	
		STR	4	
		&	5	
		STR	6	
		/ STR	1	
		STR	4	
		& N	7	
		/ STR	=	
		=	50	
		!	0	
		STR	2	
		&	4	
		/ STR	1	
		=	50	
		!	0	
		&	4	
		/	2	
		&	3	
		=	51	

Bild 14.18

15 Formulare zur Projektierung

Dieses Kapitel enthält folgende Formulare zur Projektierung der PROCONTIC K200:

Operandenliste – Eingänge

Operandenliste – Ausgänge

Variablenliste (verschiedene)

Variablenliste (Merker ungepuffert)

Variablenliste (Merker gepuffert)

Anweisungsliste

Zeitglieder T		Zeitglieder T		Zähler Z		Sonderfunktionen	
00		40		50		770	Alle Ausgänge Nullsetzen
01		41		51		771	Reset Daten nach Spannungsausf.
02		42		52		772	Zyklus-Zeit-Oszillator
03		43		53		773	0,1 sec. Oszillator
04		44		54		774	1 sec. Oszillator
05		45		55		775	0,01 sec. Oszillator
06		46		56		776	1 min. Oszillator
07		47		57		777	1 Impuls nach STA
10				60			
11				61			
12				62			
13				63			
14				64			
15				65			
16				66			
17				67			
20				70			
21				71			
22				72			
23				73			
24				74			
25				75			
26				76			
27				77			
30							
31							
32							
33							
34							
35							
36							
37							
PROCONTIC K200		Variablenliste versch.		R-Zähler Z 50 ... Z 77		Sonderfunktionen 770 ... 777	
Programmname:		Programmiert		T 00 ... T 47		Zeitglieder (0-1 Verzögern)	
Programmtitel:		Gepr. v. Ladung				Blatt	
Programm für:		Gepr. n. Ladung				insges.:	
		Inbetriebsetzung					
		Auftragsnummer:					



400	440	500	540
401	441	501	541
402	442	502	542
403	443	503	543
404	444	504	544
405	445	505	545
406	446	506	546
407	447	507	547
410	450	510	550
411	451	511	551
412	452	512	552
413	453	513	553
414	454	514	554
415	455	515	555
416	456	516	556
417	457	517	557
420	460	520	560
421	461	521	561
422	462	522	562
423	463	523	563
424	464	524	564
425	465	525	565
426	466	526	566
427	467	527	567
430	470	530	570
431	471	531	571
432	472	532	572
433	473	533	573
434	474	534	574
435	475	535	575
436	476	536	576
437	477	537	577
PROCONTIC K200		Variablenliste Merker gepuffert	
Programmname:		Programmiert	Blatt
		Gepr. v. Ladung	
Programmtitel:		Gepr. n. Ladung	
		Inbetriebsetzung	insges.:
Programm für:		Auftragsnummer:	

Merker gepuffert 400 ... 577

600	640	700	740
601	641	701	741
602	642	702	742
603	643	703	743
604	644	704	744
605	645	705	745
606	646	706	746
607	647	707	747
610	650	710	750
611	651	711	751
612	652	712	752
613	653	713	753
614	654	714	754
615	655	715	755
616	656	716	756
617	657	717	757
620	660	720	760
621	661	721	761
622	662	722	762
623	663	723	763
624	664	724	764
625	665	725	765
626	666	726	766
627	667	727	767
630	670	730	770
631	671	731	771
632	672	732	772
633	673	733	773
634	674	734	774
635	675	735	775
636	676	736	776
637	677	737	777
PROCONTIC K200		Variablenliste Merker gepuffert	
Programmname:		Programmiert	Blatt
		Gepr. v. Ladung	
Programmtitel:		Gepr. n. Ladung	insges.:
		Inbetriebsetzung	
Programm für:		Auftragsnummer:	

Merker gepuffert 600 ... 767

Step Nr. Wortadr.	Anweisung	Symbolischer Name	Befehl	Name
0			!	Wenn
1			!N	Wenn nicht
2			STR	Store
3			STR N	Store nicht
4			&	Und
5			& N	Und nicht
6			/	Oder
7			/ N	Oder nicht
8			& STR	Und store
9			/ STR	Oder store
0			=	Dann
1			= N	Dann nicht
2			FUN 0	Ladekonstante
3			FUN 00	Impulserzeugung
4			FUN 02	Set./Rücks. A/M
5			FUN 03	RS-Speicher
6			FUN 04	MC-setzen
7			FUN 05	MC-rücksetzen
8			FUN 06	Sprung auf Fun 07
9			FUN 07	Sprung Ende
0			FUN 7	Register ≥ Konstante
1			FUN 8	Register = Konstante
2			FUN 9	Register < Konstante
3			FUN 11	Addieren
4			FUN 12	Subtrahieren
5			FUN 13	Multiplizieren
6			FUN 14	Dividieren
7			FUN 22	Ausgabe Wort
8			FUN 24	Bin → BCD
9			FUN 25	BCD → Bin
0			FUN 30	Lade Wort E/A's
1			FUN 31	Lade Wort von T/Z
2			FUN 32	Lade Wort nach A/M
3			FUN 33	Lade Wort nach T/Z
4			FUN 34	Vergleiche E/A's mit Register
5			FUN 35	Vergleiche Register mit T/Z
6			FUN 36	Lade Inhalt schneller Zähler
7			FUN 40	V-R-Zähler
8			FUN 45	dyn. Speicher
9			FUN 47	Schieberegister
0			FUN 99	Ende
1			T/Z	Zeitglied
2				Zähler
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
0				
1				</



ABB Schalt- und Steuerungstechnik GmbH
D-6900 Heidelberg
Eppelheimer Straße 82

Telefon (06221) 777-0
Telefax (06221) 777-111