

Programmierung

In diesem Fachgebiet erhalten Sie eine ausführliche und umfangreiche Einführung in die Programmierung des NDR-Computers als SPS. Wir beginnen mit der **Programmierung von logischen Verknüpfungen**, es folgt dann die **Programmierung von Schaltnetzen, Speicherfunktionen und Schaltwerken**. Die Programme von Verknüpfungs- und Ablaufsteuerungen bilden den Einstieg in die Steuerungstechnik.

Programmierung der logischen Grundverknüpfungen

Im Fachgebiet Digitaltechnik haben Sie ja die logischen Grundfunktionen bereits kennengelernt. Am Beispiel der UND-Funktion zeigen wir Ihnen ausführlich die Programmierung und den Test von Verknüpfungsfunktionen. Das Gelernte sollen Sie anschließend sinngemäß auf die Programmierung der Funktionen ODER und NICHT anwenden. Zunächst kommen wir mit den bereits verwendeten Steuerbefehlen (Kommandos) aus (siehe Seite D 10). Wir werden sie im Folgenden nicht wieder erläutern, sondern nur noch anwenden. Sie finden die Steuerbefehle und Steuerungsanweisungen auf den Tafeln 2 und 3 zusammengestellt und kommentiert. Tafel 5 auf Seite G 6 enthält die Programmsammlung logischer Grundverknüpfungen.

Programmierung der UND-Verknüpfung

Wir untersuchen eine UND-Verknüpfung mit drei Eingangsvariablen. Dafür gilt nach den Gesetzen der Digitaltechnik die Schaltfunktion:

$$A0 = E0 \wedge E1 \wedge E2$$

Unsere SPS löst diese Verknüpfung in einzelnen Schritten. Sie fragt **nacheinander** alle Signalzustände an den Eingängen E0 bis E2 ab. Die Eingänge und die zugehörigen Signale sind gleich bezeichnet; z. B. wird das Signal am Eingang E1 auch mit E1 bezeichnet. Das Gleiche gilt für die Ausgänge: Ausgang A0 führt das Ausgangssignal A0. Die Ausführung dieser logischen Signalverknüpfungen erfolgt **sequentiell**. Danach wird das Ergebnis der logischen Verknüpfung dem Ausgang A0 zugewiesen. In dieser Reihenfolge ist auch das Programm als Anweisungsliste aufgestellt.

Die **Anweisungsliste** besteht aus einer Folge von **Verknüpfungsanweisungen**. Jede Zeile enthält nur eine Anweisung. Die Anweisungen sind in der Reihenfolge aufgelistet, wie sie von der SPS verarbeitet werden.

Zur Übung schreiben wir die Anweisungsliste auf. Bei der Anwendung der mathematischen Schreibweise (siehe Tafel 4, Seite G 4) lautet die Liste:

```
! E 0
& E 1
& E 2
= A 0
! PE
```

Bei industriellen SPS wird die Anweisungsliste meist in einer **mnemotechnischen, problemorientierten** Sprache formuliert. Diese Programmiersprache verwendet einen leicht zu merkenden Buchstaben-code. In Tafel 4 auf Seite G 4 haben wir die mathematischen Zeichen den zugehörigen mnemotechnischen Abkürzungen gegenübergestellt. Mit mnemonischen Abkürzungen lautet die Anweisungsliste:

```
U E 0      (UND Eingang 0)
U E 1      (UND Eingang 1)
U E 2      (UND Eingang 2)
= A 0      ( = Ausgang 0)
= BE      ( = Baustein Ende)
```

Wird eine Anweisung programmiert, dann belegt sie einen Speicherplatz im Programmspeicher. Jedem Speicherplatz ist eine Adresse zugeordnet. Eine vollständig dokumentierte Anweisung setzt sich somit aus Adresse und Anweisungsteil zusammen.

Beim NDR-Computer als SPS erfolgt die Programmeingabe **ohne Angabe** einer Speicheradresse. Die Anweisungen werden mit mathematischen Zeichen eingegeben.

Über Programmiersprachen werden wir Sie in einem der folgenden Kapitel noch ausführlich informieren. Wir dürfen nun annehmen, daß Ihr Computer betriebsbereit ist und der Monitor das Anfangsmenü zeigt.

Beispiel E2.1

Speichern Sie die Schaltfunktion $A0 = E0 \wedge E1 \wedge E2$ als Anweisungsliste und geben Sie nach dem Einspeichern die Funktion wieder auf dem Bildschirm des Monitors aus.

Durch Betätigung von Taste 1 geht die SPS in den EDIT-Mode. Wir geben diese Anweisungsliste mit mathematischen Zeichen gemäß der bereits vorgenommenen Auflistung ein. Die Eingabe ist mit dem Ladebefehl „!“ zu beginnen. Jede Einzelanweisung muß in einer Zeile stehen. Das Weiterschalten von Zeile zu Zeile erfolgt durch die Betätigung der Taste „CR“. Mit „!PE“ ist das Programm abzuschließen. Die Übergabe in den Arbeitsspeicher erfolgt mit den Kommandos „CTRL K“ und „X“. Gleichzeitig erscheint auf dem Bildschirm wieder das Anfangsmenü.

Vollziehen Sie bitte alle Schritte mit Ihrer SPS nach. Wir wollen mit der ausführlichen Erläuterung der Beispiele erreichen, daß Sie Ihr System kennen und selbständig anwenden lernen.

Wie bereits angeführt, befindet sich der Computer wieder im Anfangsmenü. Durch die Betätigung von Taste 1 erscheint erneut das Menü 1. Auf dem Bildschirm des Monitors zeigt sich das eingespeicherte Programm. Es wird jedoch nicht mehr als Anweisungsliste ausgegeben, sondern auf Ihrem Monitor steht das Programm in folgender Form:

```
!E0&E1&E2=A0  
!PE
```

Die Struktur des Programms entspricht der Schaltfunktion der Digitaltechnik (Tabelle B3.1, Spalte 5). Im Unterschied zu dieser sind die Operationen in der Reihenfolge aufgeführt, in der sie von der SPS ausgeführt werden. Erst **nach der Ausführung aller Verknüpfungen** kann das Ergebnis der Ausgangsvariablen A0 zugewiesen werden.

Die vorstehende Schaltfunktion ist in der Sprache formuliert, die die SPS versteht und setzt sich aus einer Folge von Steuerungsanweisungen zusammen. Da der NDR-Computer das Programm in Form von Anweisungen unmittelbar versteht, wird diese Programmierungsart bevorzugt angewendet. Lassen Sie bitte das Programm der UND-Verknüpfung im Arbeitsspeicher und bringen Sie die SPS wieder in das Anfangsmenü.

Beispiel E3.1

Die programmierte UND-Verknüpfung soll nun auf dem Bildschirm des Monitors als Kontaktplan ausgegeben und durch Aufnahme der Funktionstabelle getestet werden.

Lösung:

- a) Im Beispiel D17.1 haben wir Ihnen schon gezeigt, wie der Kontaktplan einer Signalverknüpfung ausgegeben wird. Wir wiederholen die erforderlichen Kommandos. Die Kontaktplanausgabe

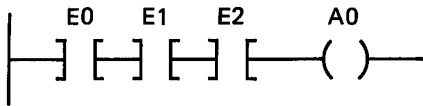


Bild E4.1
Kontaktplan zum Beispiel E3.1

erfolgt im Menü 3 (Taste mit Ziffer 3 betätigen). Im Menü 3 ist die Bildschirmausgabe mit Ziffer 0 zu wählen. Auf dem Bildschirm muß ein Kontaktplan nach Bild E4.1 erscheinen. Mit **CTRL „A“** gelangen Sie aus Menü 3 wieder in das Anfangsmenü.

- b) Zum Test einer logischen Verknüpfung muß der Computer als SPS arbeiten. Das ist im Menü 2 der Fall. Durch die Betätigung der Eingabeschalter E0 bis E2 auf der EIN-/AUSGABE-Baugruppe werden alle acht möglichen Kombinationen der Eingangsvariablen gebildet und der SPS zugeführt. Nach deren Verknüpfung muß am Ausgang A0 die in Tabelle E4.1 gezeigte Signalfolge entstehen. Zur Erinnerung: bei leuchtender LED führt der Ausgang den Signalzustand 1, bei dunkler LED ist $A = 0$.

Überprüfen Sie nun die programmierte UND-Verknüpfung. Die erforderlichen Schritte haben wir Ihnen bereits ausführlich erläutert.

E

4

Tabelle E4.1: Funktionstabelle zu Beispiel E3.1

E2	E1	E0	A0
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Zur Durchführung von Programmänderungen wird es verschiedentlich erforderlich sein, von einem Programm eine (oder auch mehrere) Zeilen zu löschen. Wie man eine Zeile löscht (die allerdings noch nicht im Programmspeicher war) haben Sie schon gelernt. Sie müssen den Cursor in die betreffende Zeile bringen und den Befehl „**CTRL Y**“ über die Tastatur eingeben. Nach Betätigung der Tasten „**CTRL K** und **X**“ erfolgt auch die Korrektur im Arbeitsspeicher.

Probieren Sie doch bitte diese Korrekturmöglichkeit aus: Löschen Sie das im Arbeitsspeicher stehende Programm der UND-Verknüpfung.

Am Beispiel einer UND-Verknüpfung wollen wir Ihnen noch eine weitere Möglichkeit der Signalverarbeitung zeigen.

Bisher wurden die Eingangsvariablen immer auf Signalzustand 1 abgefragt. In der Verknüpfung nach Bild E5.1 soll nun die Eingangsvariable E2 auf Zustand 0 abgefragt werden. Damit trotzdem die UND-Bedingung erfüllt wird, muß das Signal E2 negiert werden. Für diese Verknüpfung lautet die Schaltfunktion:

$$A1 = E1 \wedge \overline{E2} \wedge E3$$

Beispiel E5.1

Die Verknüpfung $A1 = E1 \wedge \overline{E2} \wedge E3$ ist zu programmieren (Bild E 5.1). Zusätzlich soll am Ausgang A2 das Verknüpfungsergebnis negiert zur Verfügung stehen: $A2 = \overline{A1}$

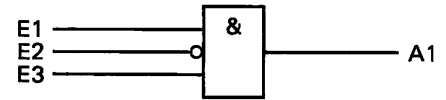


Bild E 5.1
Schaltung mit negiertem Eingang.

- Das Programm ist in Form von Anweisungen zu schreiben.
- Das Programm ist in den Arbeitsspeicher zu bringen und anschließend zu testen. Überprüfen Sie, ob das Programm die vorgegebene Funktionstabelle in Tabelle E 5.1 erfüllt. Die Funktionstabelle wurde nach der Schaltfunktion aufgestellt.
- Der Arbeitsspeicher ist zeilenweise zu löschen.

Lösung:

- Im Programm wird die Negation einer Eingangsvariablen E durch die Formulierung NE ausgedrückt. Das Programm für die Ausgangsvariable A1 lautet somit:

$!E1 \& NE2 \& E3 = A1$

Signal A2 entspricht der Negation von Signal A1. Im Programm kann man diese Umkehrung sehr elegant lösen. Wir brauchen nur das Verknüpfungsergebnis dem Ausgang A2 negiert zuzuweisen. Damit lautet das gesamte Programm:

$!E1 \& NE2 \& E3 = A1 = NA2$

!PE

- Vorstehendes Programm ist in den Arbeitsspeicher einzugeben. Die hierfür erforderlichen Schritte kennen Sie, wir wiederholen sie nicht mehr. Gehen Sie danach in das Menü 2. Durch Betätigung der Eingabeschalter ist das Programm zu testen. Vergleichen Sie die Verknüpfungsergebnisse mit den Vorgaben in der Funktionstabelle E 5.1.

Tabelle E5.1: Funktionstabelle zu Beispiel E5.1

E3	E2	E1	A1	A2
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

Die nachstehende Zusammenfassung enthält wichtige Hinweise zum Programmieren. Sie sind aus dem bisherigen Lehrstoff zusammengestellt.

Zusammenfassung

Das Programm kann als **Anweisungsliste** oder als Folge von Anweisungen in Form einer **Funktionsgleichung** aufgestellt werden. Im Programm sind die Operationen in der Reihenfolge anzuordnen, wie sie die SPS abarbeiten soll. Bei der Programmierung einer Verknüpfung stehen im Programm zuerst die Verknüpfungsoperationen. Das Verknüpfungsergebnis wird anschließend einer Ausgangsvariablen (bzw. einem Ausgang) zugewiesen.

Jede Anweisung oder Anweisungsfolge ist mit dem Zeichen „!“ = „lade“ zu beginnen. Mit Betätigung der Taste „CR“ erfolgt ein Zeilenwechsel, der Cursor springt an den Anfang der folgenden Zeile. Vom Cursor wird angezeigt, an welche Stelle auf dem Bildschirm das folgende Zeichen geschrieben wird. Die Eingabe „!PE“ markiert das Programmende. Korrekturen fehlerhafter Eingaben sind im Edit-Mode möglich. Mit Steuerbefehlen ist der Cursor auf das falsche Zeichen zu setzen. Mit der Tastatur braucht man nur noch das richtige Zeichen einzugeben. Das alte (falsche) Zeichen wird automatisch gelöscht.

Zum Löschen einer kompletten Zeile ist der Cursor an eine beliebige Stelle in der betreffenden Zeile zu bringen. Mit „CTRL Y“ wird die Bildschirmzeile gelöscht und mit CTRL Kund „X“ wird die Korrektur auch im Arbeitsspeicher ausgeführt.

Unsere Programme mit logischen Verknüpfungen testen wir durch Aufnahme der **Funktionstabelle**. Die Signaleingabe erfolgt mit Schaltern, LEDs zeigen den Zustand der Ausgangssignale an.

Aufgaben E6.1

1. Mit welchen Kommandos können Sie den Cursor um ein Zeichen nach „rechts“ bewegen?
2. Im Menü 1 ist auf dem Bildschirm folgende fehlerhafte Anweisung zu lesen:

!E1&EE5&E8=A3

Welche Maßnahme ist für folgende Korrektur notwendig:

!E1&E5&E6=A3

3. Ein Programm ist zu testen. Der Computer befindet sich im Anfangsmenü. Nach dem Test ist der Programmablauf abzubrechen; die SPS soll in das Anfangsmenü zurückkehren. Geben Sie die zum Programmtest erforderlichen Tastenbetätigungen bzw. Befehle in der richtigen Reihenfolge an.
4. Bild E6.1 zeigt einen Kontaktplan.

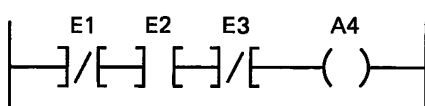


Bild E6.1
Kontaktplan zur Aufgabe E6.1, 4.

- a) Für die aus dem Plan zu ersiehende Signalverknüpfung sind Schaltfunktion und Programm (als Anweisungsfolge) aufzustellen.
- b) Bei welchen Zuständen der Eingangsvariablen nimmt die Ausgangsvariable den Wert 1 an?

- c) Geben Sie das Programm nach a) in den Arbeitsspeicher der SPS. Die SPS befindet sich im Anfangsmenü. Testen Sie Ihre Antwort zu Punkt b).

Die Lösungen finden Sie auf Seite F5.

Programmierung der ODER-Verknüpfung

Für die ODER-Verknüpfung mit drei Eingangsvariablen (Bild E7.1) lautet die Schaltfunktion:

$$A0 = E0 \vee E1 \vee E2$$

Die Schaltfunktion ist zu programmieren. Erinnern Sie sich bitte wieder an die Arbeitsweise der SPS: Die Eingangsvariablen E werden nacheinander abgefragt und sofort nach Abfrage **disjunktiv**, d.h. **ODER verknüpft**. Nach Ausführung aller Verknüpfungen wird das Verknüpfungsergebnis z. B. dem Ausgang A0 zugewiesen. In dieser Reihenfolge ist auch das Programm als Anweisungsliste zu schreiben.

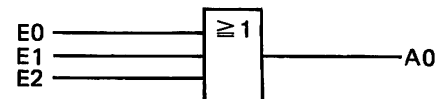


Bild E7.1
ODER-Verknüpfung.

Anweisungsliste: Mnemo-Code

NDR-
Computer

○ E 0 (ODER Eingang 0)	! E 0
○ E 1 (ODER Eingang 1)	/ E 1
○ E 2 (ODER Eingang 2)	/ E 2
= A 0 (= Ausgang 0)	= A 0
= BE (= Baustein Ende)	! PE

Als **Anweisungsfolge** lautet das Programm:

!E0/E1/E2 = A0
!PE

Mit folgendem Beispiel wollen wir die Programmierkenntnisse wieder etwas erweitern. Man kann nicht nur Eingangsvariablen abfragen und logisch verknüpfen. Nach dem Schema in Bild E7.2 können auch Ausgangsvariablen abgefragt werden, die im Verknüpfungsprogramm vorhanden sind. Das gilt allgemein, nicht nur für die ODER-Verknüpfung.

Beispiel E7.1

Nach dem Funktionsplan in Bild E7.2 ist dem Ausgang A1 der Signalzustand von E1 und dem Ausgang A2 der Signalzustand von E2 zuzuweisen. Danach sind die Variablen A1, A2 und E0 mit ODER zu verknüpfen. Das Verknüpfungsergebnis soll am Ausgang A0 zur Verfügung stehen.

- Die Anweisungslisten sind aufzustellen. Damit sind die Anweisungsliste im Mnemo-Code und die Anweisungsliste des NDR-Computers gemeint.
- Das Programm ist als Folge von Anweisungen aufzustellen und in den Arbeitsspeicher einzugeben.

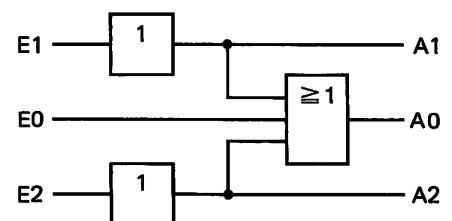


Bild E7.2
Schaltung zum Beispiel E7.1. Das Schaltzeichen für eine Zuweisung zeigt die Tafel 5 auf Seite G6.

- c) Entwerfen Sie den Kontaktplan und vergleichen Sie ihn mit dem Kontaktplan des ausgegebenen Programms.
- d) Die vorgegebene Funktionstabelle (Tabelle E 9.1) ist experimentell zu ergänzen.

Lösung:

a) Anweisungslisten:

	Mnemo-Code	NDR-Computer
Ausgang A1 übernimmt Zustand E1	U E 1 = A 1	! E 1 = A 1
Ausgang A2 übernimmt Zustand E2	U E 2 = A 2	! E 2 = A 2
Ergebnis der ODER-Verknüpfung Programmende	O A 1 O A 2 O E 0 = A 0 = BE	! A 1 / A 2 / E 0 = A 0 ! PE

b) Programm als Folge von Anweisungen:

Das Programm enthält die Anweisungsliste des NDR-Computers.

```
!E1=A1
!E2=A2
!A1/A2/E0=A0
!PE
```

Die Eingabe in den Arbeitsspeicher erläutern wir nicht mehr. Gehen Sie bei Ihren Überlegungen davon aus, daß sich der Computer im Anfangsmenü befindet.

- c) In Bild E 8.1 ist die Aufgabe als Kontaktplan wiedergegeben. Im Strompfad 1 wird E1 dem Ausgang A1 zugewiesen, im Strompfad 2 erfolgt die Zuweisung von E2 auf Ausgang A2. Im Strompfad 3 erfolgt die logische Verknüpfung der Signale. Die Ausgänge sind im Verknüpfungszweig durch Kontakte nachgebildet. Diese Kontakte haben den gleichen Signalzustand wie die zugehörigen Ausgänge. Nach der Eingabe in den Arbeitsspeicher ist Ihr Computer wieder in das Anfangsmenü zurückgekehrt. Über Menü 3 kom-

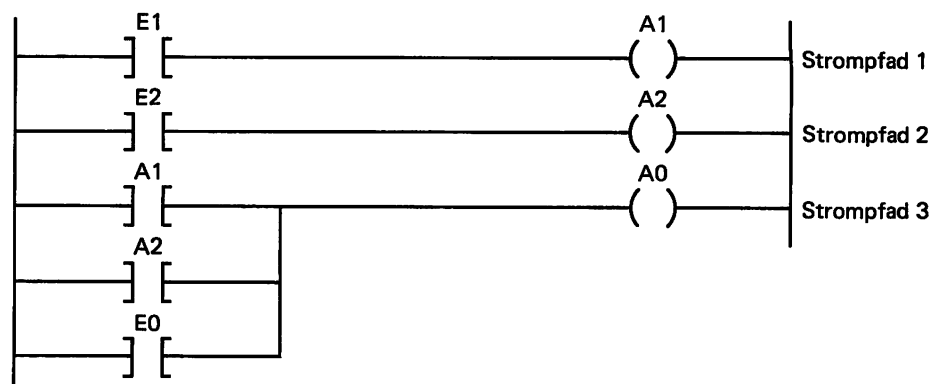


Bild E 8.1
Kontaktplan zum Beispiel E 7.1.

men Sie zur Ausgabe des Kontaktplans auf dem Bildschirm. Vergleichen Sie Bild E 8.1 mit der Bildschirmdarstellung. Mit „CTRL A“ kommen Sie wieder zurück in das Anfangsmenü.

- d) In Tabelle E 9.1 haben Sie die Signalzustände der Ausgangsvariablen für einige Signalkombinationen eingetragen. Beim Testen der eingetragenen Signalkombinationen müssen die Ergebnisse aus der Praxis und der Theorie miteinander übereinstimmen. Haben eine oder mehrere Eingangsvariablen den Wert 1, nimmt bei der disjunktiven Verknüpfung (ODER) die Ausgangsvariable ebenfalls Zustand 1 an.

Tabelle E 9.1: Funktionstabelle zum Beispiel E 7.1

E2	E1	E0	A2	A1	A0
0	0	0			
0	1	0			
1	0	0			
1	1	0			
1	1	1			

Aufgabe E 9.1

Bild E 9.1 zeigt den Kontaktplan einer ODER-Verknüpfung. Entwickeln Sie danach:

- Schaltzeichen und Schaltfunktion.
- Das Programm als Folge von Anweisungen. Das Programm ist in die NDR-SPS einzugeben (Menü 1).
- Vergleichen Sie die Bildschirmausgabe im Menü 3 mit dem vorgegebenen Kontaktplan (Bild E 9.1).
- Die Funktionstabelle E 9.2 ist durch Betätigung der Schalter im Menü 2 aufzunehmen. Tragen Sie die Signalzustände für A1 ein.

Tabelle E 9.2: Funktionstabelle für Aufgabe E 9.1, Frage d

E2	E1	E0	A1
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

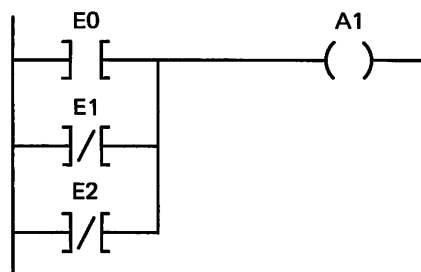


Bild E 9.1
Kontaktplan zur Aufgabe E 9.1.

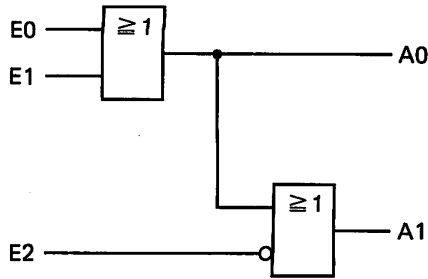


Bild E 10.1
Schaltung zur Aufgabe E 10.1.

Aufgabe E 10.1

Führen Sie bitte für die Schaltung in Bild E 10.1 folgende Schritte durch:

- Einen Kontaktplan aufzeichnen.
- Das Programm als Folge von Anweisungen aufstellen und in die NDR-SPS eingeben.
- Die Bildschirmausgabe des Kontaktplans mit dem Kontaktplan nach a) vergleichen.
- Die Funktionstabelle E 10.1 aufnehmen, die Signalzustände für A1 eintragen und die Ergebnisse nun mit der Theorie vergleichen.

Tabelle E 10.1: Funktionstabelle zur Aufgabe E 10.1

E0	E1	A0	E2	A1
0	0		0	
0	1		0	
1	0		0	
1	1		0	
0	0		1	
0	1		1	
1	0		1	
1	1		1	

Die Lösungen dieser Aufgaben finden Sie auf Seite F 6 und F 7.

Programmierung der Signalumkehr (NICHT-Funktion)

Die Signalumkehr oder auch **NICHT-Funktion** wurde in den bisherigen Beispielen schon mehrfach angewendet. Mit der NICHT-Funktion kann man beispielsweise einer Ausgangsvariablen den **negierten Signalzustand** der Eingangsvariablen zuweisen. Das Programm hierfür lautet als

Anweisungsliste:

Mnemo-Code

NDR-Computer

U E 1

! E 1

=NA 1

=NA 1

= BE

! PE

Programm als Folge von Anweisungen:

!E1=NA1

!PE

Der Ausgang A1 führt das negierte Signal des Eingangs E1.

Bei zweifacher Negation entsteht wieder der ursprüngliche Signalzustand. Es gilt:

$$\begin{aligned}\overline{\overline{1}} &= \overline{0} && \text{erste Negation} \\ \overline{\overline{0}} &= \overline{1} && \text{zweite Negation}\end{aligned}$$

Aufgabe E 11.1

Der Signalzustand von Eingang E1 soll negiert dem Ausgang A1 und der Signalzustand von A1 soll wiederum negiert dem Ausgang A2 zugewiesen werden. Führen Sie dazu bitte folgende Schritte aus:

- Zeichnen Sie bitte den Kontaktplan.
- Das Programm ist als Anweisungsliste und in Form von Anweisungen aufzustellen; die Anweisungen sind in die NDR-SPS einzugeben.
- Durch experimentelle Aufnahme der Funktionstabelle E 11.1 ist das Programm zu überprüfen.

Tabelle E 11.1: Funktionstabelle zur Aufgabe E 11.1

E1	A1	A2
0		
1		

Die Lösung dieser Aufgabe finden Sie auf Seite F7.

Programmierung der Standardverknüpfungen NAND und NOR

In den bisherigen Beispielen konnte das Verknüpfungsergebnis unmittelbar einer Ausgangsvariablen zugewiesen werden. In manchen Programmen ist es jedoch erforderlich, daß sich die SPS ein **Zwischenergebnis** oder ein Ergebnis merken muß. Das ist beispielsweise der Fall, wenn ein Verknüpfungsergebnis in unterschiedlichen Programnteilen benötigt wird. Zur vorübergehenden Aufbewahrung von Ergebnissen stehen uns **programmierbare Speicherelemente** zur Verfügung. Sie werden als **Merker** bezeichnet und haben das Operandenkennzeichen M.

Merker

Der Merker ist ein **1-Bit-Speicherelement** eines Schreib-Lese-Speichers (RAM). Er hat die Aufgabe, sich den Signalzustand 1 oder 0 für eine bestimmte Zeit zu merken. Der Merker behält solange das zuletzt eingegebene Signal, bis der Speicherzustand durch eine neue

Setzbedingung geändert wird. Unsere SPS hat 16 Merker mit den Adressen 0 bis 15. Das im Merker gespeicherte Signal steht uns nicht unmittelbar an einem Ausgang zur Verfügung, es ist intern im Computer vorhanden. Falls erforderlich, muß der Merkerinhalt einem Ausgang zugewiesen werden.

In einem eigenen Abschnitt werden Sie noch über Halbleiterspeicher ausführlich unterrichtet. Hier benutzen wir dann den Merker als Programmelement zur Lösung bestimmter Aufgaben. In den folgenden Beispielen sollen Sie zunächst allgemeine Anwendungen und Eigenschaften des Merkers kennenlernen.

Beispiel E 12.1

Wir wollen den Merker M0 mit dem Zustand des Signals E0 laden. Anschließend soll der Merkerinhalt auf Ausgang A0 gegeben werden. Zur Kontrolle ist Signal E0 unmittelbar über Ausgang A1 auszugeben. Im einzelnen wird folgendermaßen vorgegangen:

- Zeichnen Sie bitte den Kontaktplan.
- Die Anweisungslisten sind anzugeben.
- Stellen Sie bitte das Programm auf.
- Das aufgestellte Programm ist in die SPS einzugeben (Menü 1), als Kontaktplan auf dem Bildschirm des Monitors auszugeben (Menü 3) und mit a) zu vergleichen. Anschließend soll das Programm im Menü 2 auf der Experimentier-Ein-Ausgabe durch Betätigung der Eingabeschalter getestet werden. Folgende Werte sind aufzunehmen; d.h. die Signalzustände von A0 und A1 sind zu bestimmen und einzutragen.

E0	A0	A1
0		
1		

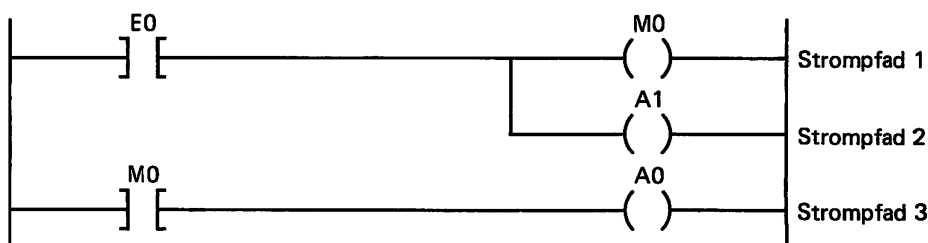


Bild E 12.1
Kontaktplan zum Beispiel E12.1.

Lösung:

- Den Kontaktplan zeigt Bild E12.1. Im Strompfad 1 wird der Zustand von E0 dem Merker M0 zugewiesen. In diesem Fall ist der Merker als Ausgabe gezeichnet. Im Strompfad 3 wird der Merker abgefragt. Jetzt ist der Merker durch ein Eingabesymbol darzustellen. Im Strompfad 2 erfolgt die Zuweisung von E0 auf Ausgang A1.

b) Anweisungsliste:

Mnemo-Code	NDR-Computer
U E 0	! E 0
= M 0	= M 0
= A 1	= A 1
U M 0	! M 0
= A 0	= A 0
= BE	! PE

c) Anweisungen des Programms:

!E0=M0=A1
 !M0=A0
 !PE

- d) Wählen Sie aus dem Menü die Funktion „Edit“. Nach Eingabe des Programms und anschließender Rückkehr in das Anfangsmenü wird der Kontaktplan (Menü 3) aufgerufen. Nun können Sie die Lösung unter a) mit der Bildschirmdarstellung vergleichen. Nach Rückkehr ins Anfangsmenü erfolgt mit der Funktion „Start“ der Programmtest. Durch Betätigung des Eingabeschalters E0 ermitteln Sie bitte die Tabellenwerte. Bei ordnungsgemäßer Funktion müssen die Signalzustände der Variablen E0, A0 und A1 immer gleich sein. Damit haben Sie den Beweis, daß der Merker M0 den Signalzustand E0 gespeichert hat, denn A0 ist ja identisch mit E0.

Die Kommandos zur Rückkehr in das Menü wurden mit Absicht nicht aufgeführt. Wichtige Kommandos sollten Sie sich einprägen. Für alle Fälle haben wir für Sie einen „persönlichen Hilfsmerker“ beigelegt. Auf Tafel 3, Seite G3 finden Sie die Kommandos der SPS mit einem Kurzkomentar versehen zusammengestellt. In den einführenden Programmierbeispielen hat nicht jedes Beispiel eine anwendungsorientierte Bedeutung. Verschiedentlich sollen Sie nur den „Umgang“ mit bestimmten Anweisungen lernen.

Aufgabe E 13.1

Setzen Sie den Signalzustand der Eingangsvariablen E1 in Merker M1, den von E2 in Merker M2. Die Merkerinhalte von M1 und M2 sind konjunktiv, d. h. mit UND zu verknüpfen. Das Ergebnis soll als Ausgangsvariable A0 ausgegeben werden. Führen Sie dazu folgende Schritte aus:

- Zeichnen Sie bitte den Kontaktplan.
- Die Aufgabe ist zu programmieren.
- Geben Sie das Programm b) in den Arbeitsspeicher der SPS ein und testen Sie das Programm anhand der Funktionstabelle E14.1.

Tabelle E 14.1: Funktionstabelle zur Aufgabe E 13.1

E2	E1	A0
0	0	
0	1	
1	0	
1	1	

d) Welche logische Verknüpfung erfüllt das Programm?

Die Lösung dieser Aufgabe finden Sie auf Seite F7.

E

14

Programmierung der NAND-Verknüpfung

Im Fachgebiet Digitaltechnik wurde die Wirkung dieser Funktion bereits besprochen. Für eine NAND-Verknüpfung mit zwei Eingangsvariablen lautet die Schaltfunktion: $A0 = \overline{E1 \wedge E2}$. Daraus folgt allgemein: Die Eingangsvariablen sind erst mit UND zu verknüpfen. Anschließend ist das Ergebnis der UND-Verknüpfung zu negieren. In dieser Reihenfolge muß auch die programmierte NAND-Verknüpfung die Signale verarbeiten. Wir zeigen Ihnen anschließend eine Lösungsmöglichkeit. Weitere folgen in einem späteren Abschnitt.

Programm mit Merker

Die drei Eingangsvariablen E0, E1 und E2 wollen wir mit NAND verknüpfen, A0 ist die Ausgangsvariable. Zur Lösung wird folgender Weg gewählt:

- Die Verknüpfung der Variablen E0, E1 und E2 erfolgt mit UND.
- Das Ergebnis der UND-Verknüpfung wird in einen Merker (M0) gesetzt.
- Die Ausgangsvariable A0 erhält den Merkerinhalt negiert zugewiesen.

Die Schaltfunktionen der Lösungen lauten:

$$M0 = E0 \wedge E1 \wedge E2$$

$$A0 = \overline{M0}$$

Mit unseren Kenntnissen können wir bereits die vorstehenden Schaltfunktionen programmieren. Zunächst stellen wir das Programm als **Anweisungsliste im Mnemo-Code** und mit mathematischen Zeichen für den NDR-Computer auf.

Anweisungsliste:	Mnemo-Code	NDR-Computer
	U E 0	! E 0
	U E 1	& E 1
	U E 2	& E 2
	= M 0	= M 0
	UNM 0	!NM 0
	= A 0	= A 0
	= B E	! P E

Als **Anweisungsfolge** lautet das Programm:

```
!E0&E1&E2=M0
!NM0=A0
!PE
```

Wir wollen im folgenden Beispiel untersuchen, ob das Programm zum richtigen Ergebnis führt.

Beispiel E 15.1

Wir geben das gerade entwickelte Programm der Schaltfunktion $A0 = \overline{E0 \wedge E1 \wedge E2}$ in den Arbeitsspeicher der SPS ein. Anschließend wird die SPS gestartet. Mit den Eingabeschaltern sind nun die acht möglichen Signalkombinationen einzustellen. Überprüfen Sie, ob die experimentellen Verknüpfungsergebnisse ($A0,2$) mit den in der Tabelle E 15.1 enthaltenen, theoretisch ermittelten Werten ($A0,1$) übereinstimmen.

Tabelle E 15.1: Funktionstabelle einer NAND-Verknüpfung mit drei Eingangsvariablen.

E2	E1	E0	A0,1	A0,2
0	0	0	1	
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	0	

A0,1 theoretisches Verknüpfungsergebnis
A0,2 experimentelles Verknüpfungsergebnis

Lösung:

Programmeingabe und Test haben Sie nun schon mehrfach ausgeführt. Das Vorgehen wird deshalb nur noch erläutert, falls besondere Punkte zu beachten sind. In unserem Beispiel müssen die theoretischen und experimentellen Ergebnisse übereinstimmen.

Programmierung der NOR-Verknüpfung

Die Lösung dieser Aufgabe dürfte Ihnen nunmehr keine Schwierigkeiten mehr bereiten. Analog zur NAND-Verknüpfung enthält die Schaltfunktion $A0 = \overline{E1 \vee E2}$ folgenden Lösungsweg:

Die Eingangsvariablen sind zuerst mit ODER zu verknüpfen. Anschließend ist das Ergebnis der ODER-Verknüpfung zu negieren.

Daraus ergeben sich die gleichen Programmschritte, die Sie bei der NAND-Programmierung gelernt haben. Das Ergebnis der **disjunktiven Verknüpfung (ODER)** ist in einen Merker zu setzen. Das negierte Merkersignal wird dann der Ausgangsvariablen A0 zugewiesen.

E

16

Aufgabe E 16.1

Programmieren Sie bitte die Schaltfunktion $A0 = \overline{E1 \vee E2 \vee E3}$.

- Stellen Sie die Anweisungsliste (im Mnemo-Code und mit mathematischen Zeichen) und das Programm auf.
- Zeichnen Sie das Programm als Kontaktplan auf.
- Testen Sie das Programm mit der SPS durch Aufnahme der Funktionstabelle E 16.1. Diese Tabelle enthält bereits die theoretisch ermittelten Verknüpfungsergebnisse. Vergleichen Sie die experimentell gewonnenen Ergebnisse mit den vorgegebenen Werten.

Tabelle E 16.1: Funktionstabelle zur Aufgabe E 16.1

E3	E2	E1	A0,1	A0,2
0	0	0	1	
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	

Die Lösung dieser Aufgabe finden Sie auf Seite F 8.

Zusammenfassung

Zur Programmierung bestimmter Verknüpfungen müssen oft Zwischenergebnisse gespeichert werden. Hierfür gibt es **Merker M**. Merker sind Schreib-Lesespeicher. Ein Merker hat die Speicherkapazität 1 Bit.

Mit Merkern lassen sich einfache Programme für die Standardfunktionen NAND und NOR aufstellen. Das Ergebnis der **konjunktiven (UND-)** oder **disjunktiven (ODER-)** Verknüpfung wird in einem Merker zwischengespeichert und danach negiert der Ausgangsvariablen zugewiesen.

Programmierung einfacher Schaltnetze

Schaltnetze enthalten im wesentlichen die Grund- und Standardverknüpfungen. Die folgenden Anwendungen und Beispiele benötigen noch keine Speicherfunktionen. Zur Lösung einer Aufgabe gibt es auch hier verschiedene Wege. In unserem Fall muß nicht immer das kürzeste Programm die beste Lösung sein. Der Lösungsweg wird oft sowohl von der Aufgabenstellung als auch von der Auswertung der Verknüpfungsergebnisse beeinflusst.

Zur **Aufstellung der Steuerungsanweisungen** kommen wir im Moment noch mit den bereits bekannten Operanden aus. Die Operationskennzeichen sind durch „Klammer auf“ und „Klammer zu“ erweitert. Die „Klammern“ gehören zur Operationsgruppe „Programmorganisation“ (siehe auch Tafel 2, Lehrbrief 1).

Eine Aufgabenstellung oder eine Lösung enthält u. a. meistens

- eine Schaltfunktion,
- einen Kontaktplan,
- einen Funktionsplan,
- eine Anweisungsliste oder
- Anweisungen.

Bei der Aufstellung der Anweisungsliste schließen wir einen Kompromiß. Nach DIN 19 239 enthält die Anweisungsliste mnemotechnische Kurzbezeichnungen für die Operationen. Unser System versteht nur mathematische Zeichen. Eine Gegenüberstellung der mathematischen und mnemotechnischen Zeichen enthält Tabelle E 17.1. Wir stellen deshalb zwei Anweisungslisten auf; eine Liste im Mnemonik-Code, die andere mit mathematischen Zeichen. Nahezu alle Programmiergeräte erlauben die Programmeingabe als Anweisungsliste. Sie sollen deshalb auch das Aufstellen der Anweisungsliste üben.

Tabelle E 17.1: Operationen zur Signalverarbeitung nach DIN 19 239

Benennung	Zeichen	
	mnemotechnische Abkürzung	mathematische Schreibweise
UND	U	&
ODER	O	/
NICHT	N	
Zuweisung	=	=

Für Operationen, die kein eigenes, mathematisches Zeichen besitzen, werden die Mnemo-Kurzzeichen verwendet.

In Schaltnetzen entsprechen die Funktions-Symbole nahezu den Schaltzeichen der Digitaltechnik. Der Unterschied zwischen einer Schaltung mit digitalen Schaltzeichen und dem Funktionsplan wird noch ausführlich besprochen.

UND- vor ODER-Verknüpfung

Wir wollen die Verknüpfung der Schaltung in Bild E 19.1 programmieren. Gleichzeitig sollen Sie verschiedene Lösungsmöglichkeiten kennenlernen. Laut Schaltung sind zunächst zwei UND-Verknüpfungen zu bilden. Deren Ausgangsvariablen und die Variable E4 sind die Eingangsvariablen der nachfolgenden ODER-Verknüpfung.

a) Programm mit Merkern:

Dieses Programm setzt die Ergebnisse der UND-Verknüpfungen in Merker. Anschließend werden die Merkerinhalte und die Variable E4 mit ODER verknüpft.

Das Feld a in Bild E 19.1 enthält die Schaltung, den Kontaktplan, den Funktionsplan, die Anweisungslisten und die Anweisungen. Die Ausgänge der UND-Verknüpfungen im Kontaktplan sind als Merker dargestellt. Zur ODER-Verknüpfung werden die Merkerinhalte abgefragt. Die Merker sind in diesem Fall als Eingangssymbole zu zeichnen.

Im Funktionsplan muß der Ausgang der UND-Funktion die Bezeichnungen des Merkers erhalten, in den das Verknüpfungsergebnis gesetzt wird. Der Funktionsplan kann ähnlich der Digital-schaltung in geschlossener Form gezeichnet werden. Zusätzlich ist noch der aufgelöste Funktionsplan dargestellt. Dieser läßt sich unmittelbar programmieren. Vergleichen Sie bitte Anweisungen und Anweisungsliste mit Kontaktplan und Funktionsplan. Bezüglich der Informationsverarbeitung muß Übereinstimmung bestehen.

b) Programm mit Klammern:

Im Fachgebiet „Digitaltechnik“ haben Sie gelernt, daß man zusammengehörende Verknüpfungen in Klammer setzen soll. Für die Schaltung in Bild E 19.1 ergibt die Analyse folgende Schaltfunktion:

$$A1=(E0\wedge E1)\vee(E2\wedge E3)\vee E4$$

Für diese Schaltfunktion zeigt das Feld b in Bild E 19.1 den Kontaktplan und die Anweisungen. Der Funktionsplan hat sich im Prinzip nicht geändert. An die Ausgänge der UND-Funktionen müssen anstelle der Merker die Klammerausdrücke geschrieben werden.

c) Programm nach der Regel „UND vor ODER“:

Eine Regel der Schaltalgebra lautet:

UND-Verknüpfung hat Vorrang vor der ODER-Verknüpfung.

Das Verknüpfungsprogramm unserer SPS enthält diese Regel. Danach führt das Programm zuerst die UND-Verknüpfungen aus. Anschließend werden deren Verknüpfungsergebnisse mit ODER zusammengefaßt. Nach dieser Regel können Sie die Schaltfunktion auch folgendermaßen schreiben:

$$A1=E0\wedge E1\vee E2\wedge E3\vee E4$$

Die Klammern sind also nicht mehr erforderlich. Allerdings ergibt diese Schreibweise eine recht unübersichtliche Formulierung. Das Feld c in Bild E 19.1 enthält das Programm mit Anweisungen ohne Klammern. Kontakt- und Funktionsplan haben sich nicht geändert.

Aufgabe E 19.1

Die Programme der Schaltung von Bild E 19.1 sind zu testen. Geben Sie bitte erst das Programm nach Bild E 19.1, Feld a ein und testen Sie es anschließend. Nach der Aufnahme der Funktionstabelle können Sie das Programm nach den Anweisungen in Feld b und c ebenfalls eingeben und testen. Die Funktionstabelle E 19.1 mit einer Auswahl möglicher Schaltkombinationen ist durch die Betätigung der Eingabeschalter aufzunehmen.

Programm als					
	Schaltung	Kontaktplan	Funktionsplan	Anweisungsliste	Anweisungen
a				U E0 ! E0 U E1 & E1 = M1 = M1 U E2 ! E2 U E3 & E3 = M2 = M2 O M1 ! M1 O M2 / M2 O E4 / E4 = A1 = A1 PE ! PE (α) (β)	!E0&E1=M1 !E2&E3=M2 !M1/M2/E4=A1 !PE
b					Anweisungen !((E0&E1)/(E2&E3)/E4=A1 IPE
c					!E0&E1/E2&E3/E4=A1 IPE

Tabelle E 19.1: Funktionstabelle zur Aufgabe E 19.1

E4	E3	E2	E1	E0	A1a	A1b	A1c
0	0	0	0	0			
0	0	0	0	1			
0	0	0	1	1			
0	1	0	0	0			
0	1	1	0	0			
1	0	0	0	0			
1	1	1	1	1			

Die Lösung dieser Aufgabe finden Sie auf Seite F 12.

Bild E 19.1

UND- vor ODER-Verknüpfung
 α = Anweisungsliste im Mnemo-Code,

β = Anweisungsliste mit mathematischen Zeichen, (NDR-Computer als SPS).

ODER- vor UND-Verknüpfung

In der praktischen Schaltungstechnik treten derartige Verknüpfungen ebenfalls häufig auf. Nach Bild E 20.1 erfolgt erst die disjunktive Verknüpfung (ODER) der Eingangsvariablen. Deren Verknüpfungsergebnisse werden anschließend konjunktiv (UND) verarbeitet.

Zur Programmierung der Schaltung sind in Bild E 20.1 zwei Lösungen angegeben. Im Feld a enthält das Programm Merker und im Feld b werden Klammern verwendet. Das Programm nach a verknüpft die Merkerinhalte, das Programm nach b verknüpft die Klammerinhalte mit UND. Beide Programme müssen zum gleichen Ergebnis führen.

Aufgabe E 20.1

Die Schaltung von Bild E 20.1 ist zu programmieren und zu testen. Geben Sie bitte erst das Programm nach Feld a und dann nach Feld b ein (siehe Aufgabe E 19.1). Durch Betätigung der Eingabeschalter ist die Funktionstabelle E 20.1 mit einer Auswahl möglicher Schaltkombinationen aufzunehmen.

Tabelle E 20.1: Funktionstabelle zur Aufgabe E 20.1

E4	E3	E2	E1	E0	A2a	A2b
0	0	0	0	0		
0	1	0	1	1		
0	0	1	1	0		
0	1	1	1	0		
1	0	0	0	0		
0	1	0	0	0		

Bild E 20.1
ODER- vor UND-Verknüpfung,
Anweisungslisten wie im Bild
E 19.1.

Programm als																																																									
Schaltung		Kontaktplan	Funktionsplan	Anweisungsliste	Anweisungen																																																				
a				<table><tr><td>O</td><td>E0</td><td>!</td><td>E0</td></tr><tr><td>ON</td><td>E1</td><td>/</td><td>NE1</td></tr><tr><td>=</td><td>M1</td><td>=</td><td>M1</td></tr><tr><td>O</td><td>E3</td><td>!</td><td>E3</td></tr><tr><td>O</td><td>E4</td><td>/</td><td>E4</td></tr><tr><td>=</td><td>M2</td><td>=</td><td>M2</td></tr><tr><td>U</td><td>M1</td><td>!</td><td>M1</td></tr><tr><td>U</td><td>M2</td><td>&</td><td>M2</td></tr><tr><td>UN</td><td>E2</td><td>&</td><td>NE2</td></tr><tr><td>=</td><td>A2</td><td>=</td><td>A2</td></tr><tr><td>PE</td><td></td><td>!</td><td>PE</td></tr></table>	O	E0	!	E0	ON	E1	/	NE1	=	M1	=	M1	O	E3	!	E3	O	E4	/	E4	=	M2	=	M2	U	M1	!	M1	U	M2	&	M2	UN	E2	&	NE2	=	A2	=	A2	PE		!	PE	<table><tr><td>!</td><td>E0/NE1=M1</td></tr><tr><td>!</td><td>E3/E4=M2</td></tr><tr><td>!</td><td>M1&M2&NE2=A2</td></tr><tr><td>!</td><td>PE</td></tr></table>	!	E0/NE1=M1	!	E3/E4=M2	!	M1&M2&NE2=A2	!	PE
	O	E0	!	E0																																																					
ON	E1	/	NE1																																																						
=	M1	=	M1																																																						
O	E3	!	E3																																																						
O	E4	/	E4																																																						
=	M2	=	M2																																																						
U	M1	!	M1																																																						
U	M2	&	M2																																																						
UN	E2	&	NE2																																																						
=	A2	=	A2																																																						
PE		!	PE																																																						
!	E0/NE1=M1																																																								
!	E3/E4=M2																																																								
!	M1&M2&NE2=A2																																																								
!	PE																																																								
b				<table><tr><td>!</td><td>(E0/NE1)&(E3/E4)&NE2=A2</td></tr><tr><td>!</td><td>PE</td></tr></table>	!	(E0/NE1)&(E3/E4)&NE2=A2	!	PE																																																	
!	(E0/NE1)&(E3/E4)&NE2=A2																																																								
!	PE																																																								

Über das Menü 3 ist das gespeicherte Programm als Kontaktplan auszugeben und mit dem Kontaktplan von Bild E 20.1 zu vergleichen.

Die Lösung dieser Aufgabe finden Sie auf Seite F13.

Aufgabe E 21.1

Das Bild E 21.1 enthält zwei einfache Verknüpfungsschaltungen.

- Für die Schaltung in Bild E 21.1a ist der Kontaktplan aufzuzeichnen. Das Ergebnis der ODER-Verknüpfung ist in Merker M0 zu setzen.
- Stellen Sie bitte für den Kontaktplan von Punkt a) das Programm in Form von Anweisungen auf. Nach Eingabe des Programms in die SPS sind für die Funktionstabelle E 21.1 experimentell die Zustände von A1 zu bestimmen.

Tabelle E 21.1: Funktionstabelle zur Aufgabe E 21.1

E3	E2	E1	E0	A1	A2,1	A2,2	A2,3
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	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				

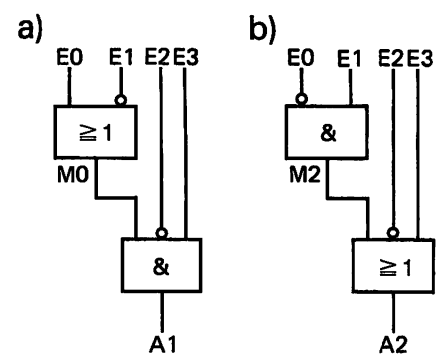


Bild E 21.1
Schaltungen zur Aufgabe E 21.1.

- Für die Schaltung in Bild E 21.1b ist das Programm in Anweisungsform in folgenden Varianten aufzustellen:

Variante 1: Das Ergebnis der UND-Verknüpfung ist in Merker M2 zu setzen,

Variante 2: die UND-Verknüpfung ist in Klammer zu setzen und

Variante 3: die Regel „UND vor ODER“ ist anzuwenden.

Diese 3 Programme sind nacheinander in die SPS einzugeben und zu testen. Tragen Sie die Testergebnisse in die vorbereitete Tabelle E 21.1 ein.

Die Lösung dieser Aufgabe finden Sie auf Seite F13.

Überwachungsschaltung

Das Bild E 22.1 zeigt den Stromlaufplan einer Überwachungsschaltung. Lampe H1 soll leuchten, wenn mindestens einer der Signalgeber E0 bzw. E1 betätigt ist. Ist die Meldeleuchte H1 eingeschaltet ($A0=1$) und sind die Schalter E2 und E3 geschlossen, soll Hupe H2 ertönen ($A1=1$).

Beispiel E 22.1

Die Überwachungsschaltung von Bild E 22.1 ist zu programmieren. Kontaktplan, Funktionsplan, Anweisungsliste und Programmanweisungen sind anzufertigen bzw. aufzustellen.

E

22

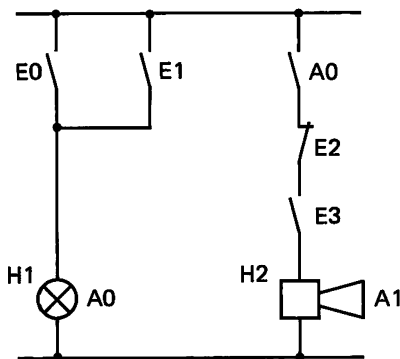


Bild E 22.1
Überwachungsschaltung, Beispiel E 22.1.

Lösung:

Die beiden Eingänge E0 und E1 werden mit der Operation **ODER** auf Signalzustand 1 abgefragt. Sobald einer der beiden Signalgeber Zustand 1 hat, übernimmt Ausgang A0 ebenfalls Zustand 1. Dieses Signal wird im Ausgangsmerker zwischengespeichert. Im Hupenzweig erfolgt die Abfrage des Signalzustands von A0, die Abfrage von E2 (Offner) auf Zustand 0 und die Abfrage von E3 (Schließer) auf Zustand 1. Bei Erfüllung der **UND-Bedingung** nimmt Ausgang A1 den Wert 1 an.

Das Bild E 22.2 zeigt den nach der verbalen Beschreibung und dem Stromlaufplan entwickelten Funktionsplan und den Kontaktplan. Die Pläne ergeben unmittelbar die Anweisungslisten.

Anweisungslisten:

Mnemo-Code	NDR-Computer
O E 0	!E0
O E 1	/E1
= A 0	=A0
U A 0	!A0
UN E 2	&NE2
U E 3	&E3
= A 1	=A1
BE	!PE

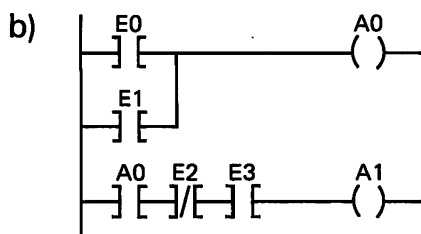
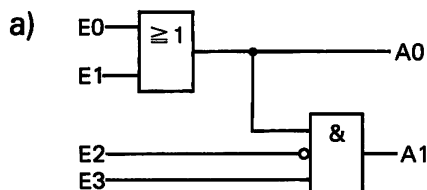


Bild E 22.2
Funktionsplan (a) und Kontaktplan (b) der Überwachungsschaltung von Bild E 22.1.

Anweisungen des Programms:

```
!E0/E1=A0
!A0&NE2&E3=A1
!PE
```

Geben Sie bitte vorstehendes Programm in die SPS ein. Prüfen Sie mit Menü 3, ob der ausgegebene Kontaktplan mit dem in Bild E 22.2b übereinstimmt. Testen Sie im Menü 2 durch Betätigung der Eingabeschalter die Funktionen der Schaltung.

Aufgabe E 23.1

Für den Kontaktplan in Bild E 23.1

- a) sind die Anweisungen für das Programm aufzustellen und
b) ist der Funktionsplan zu zeichnen.

Verwenden Sie im Programm die Klammern (keine Merker).

Die Lösung dieser Aufgabe finden Sie auf Seite F14.

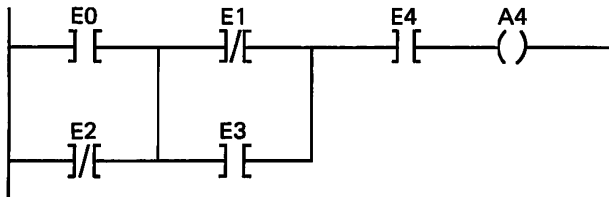


Bild E 23.1
Kontaktplan zur Aufgabe E 23.1.

Beispiel E 23.1

In Beispiel B 15.1 und Aufgabe B 15.1 wurde eine **Vergleicherschaltung** untersucht. Nach den Regeln der Digitaltechnik ergab sich als Schaltfunktion für die Ausgangsvariable:

$$A1 = \underbrace{[(E1 \wedge E2) \vee (\overline{E1} \wedge \overline{E2})]}_{M1} \wedge \underbrace{[(E3 \wedge E4) \vee (\overline{E3} \wedge \overline{E4})]}_{M2}$$

Die Schaltung (Schaltfunktion) ist zu programmieren. Wie angegeben, sind die Verknüpfungsergebnisse der rechteckigen Klammern in Merker zu setzen. Kontaktplan und Programm sind aufzustellen. Für einige vorgegebene Kombinationen der Eingangsvariablen ist das Programm zu testen.

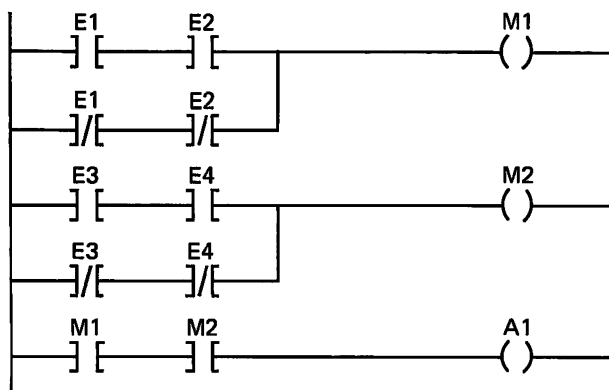


Bild E 23.2
Kontaktplan der Vergleicherschaltung, Beispiel E 23.1.

Lösung:

Bei Anwendung der Regel „UND vor ODER“ erhält man als Programm die Anweisungen:

$\neg E1 \wedge E2 / \neg E1 \wedge \neg E2 = M1$
 $\neg E3 \wedge E4 / \neg E3 \wedge \neg E4 = M2$
 $M1 \wedge M2 = A1$
 $\neg PE$

Den Kontaktplan, der dem vorstehenden Programm entspricht, zeigt das Bild E 23.2. Geben Sie bitte das Programm in unsere SPS ein. Vergleichen Sie in Menü 3 den Kontaktplan des Programms mit dem von Bild E 23.2.

Geben Sie nun die Signalkombinationen der Tabelle E 24.1 (Menü 2) durch Betätigung der Schalter in die Steuerung ein. Prüfen Sie, ob die theoretischen Ergebnisse (A1,1) mit den experimentell zu ermittelnden Ergebnissen (A1,2) übereinstimmen.

Tabelle E 24.1: Funktionstabelle zu Beispiel E 23.1

E4	E3	E2	E1	A1,1	A1,2
0	0	0	0	1	
0	0	0	1	0	
0	0	1	1	1	
1	0	1	1	0	
1	0	1	0	0	
1	1	0	0	1	
1	1	1	0	0	
1	1	1	1	1	

Programmierung von Schaltnetzen

Den bisher behandelten Lehrstoff wollen wir nun zur Lösung praktischer Beispiele anwenden. Außerdem sollen Sie einige Programme für Schaltnetze selbst entwickeln.

Beispiel E 25.1

Prioritätsschaltung

Wir wollen das Programm für eine Prioritätsschaltung mit drei Eingangssignalen entwickeln. Priorität bedeutet, daß bestimmte Variablen gegenüber anderen Vorrang haben. Als Eingangsvariablen wirken die Signale E0, E1 und E2. Signal E2 soll höherwertig gegenüber E1, und Signal E1 soll höherwertig gegenüber E0 sein. Das höherwertige Signal bestimmt immer den Zustand der Ausgangsvariablen. Es gilt die Zuordnung:

Variable E0 ist Ausgang A0 zugeordnet,
Variable E1 ist Ausgang A1 zugeordnet,
Variable E2 ist Ausgang A2 zugeordnet.

Es darf immer nur **ein Ausgang** den Zustand 1 haben. Eine Eingangsvariable ist nur wirksam, wenn sie den Wert 1 hat. Die Variable mit höchster Priorität bestimmt, welcher Ausgang den Zustand 1 hat.

- Stellen Sie die Funktionstabelle auf.
- Nach der Funktionstabelle sind die Schaltfunktionen zu entwickeln und, falls möglich, mit den Regeln der Schaltalgebra zu vereinfachen.
- Die Schaltung bzw. die Schaltfunktionen nach b) sind zu programmieren. Überprüfen Sie das Programm durch Aufnahme der Funktionstabelle. Sie muß mit der unter a) aufgestellten Tabelle übereinstimmen.

Lösung:

Tabelle E25.1: Funktionstabelle zu Beispiel E25.1

E2	E1	E0	A0	A1	A2
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	1

- a) In der Funktionstabelle führen wir alle möglichen Signalkombinationen der Eingangsvariablen auf. Die Eingangsvariable höchster Priorität bestimmt den jeweiligen Wert der Ausgangsvariablen. Sehen Sie sich bitte die Tabelle E25.1 an.

- b) Diejenigen Signalkombinationen, die zum Zustand 1 einer Ausgangsvariablen führen, ergeben die Schaltfunktion:

$$A_0 = E_0 \overline{E_1} \overline{E_2}$$

Hier ist keine Vereinfachung möglich.

$$A_1 = \overline{E_0} E_1 \overline{E_2} \vee E_0 E_1 \overline{E_2}$$

Die Verknüpfung $E_1 \overline{E_2}$ kann ausgeklammert werden, sie ist in beiden Termen enthalten.

$$A_1 = E_1 \overline{E_2} (\underbrace{\overline{E_0} \vee E_0}_1) = E_1 \overline{E_2}$$

Die Regeln der Theoreme und Postulate, nach denen Schaltungsvereinfachungen vorgenommen werden, sind jetzt nicht mehr angeführt. Ermitteln Sie die benutzten Regeln an Hand der Tabelle B20.1 bitte selbst.

$$A_2 = \overline{E_0} \overline{E_1} E_2 \vee E_0 \overline{E_1} E_2 \vee \overline{E_0} E_1 E_2 \vee E_0 E_1 E_2$$

Die farbig markierten Kombinationen lassen sich ausklammern. Wir erhalten:

$$A_2 = \overline{E_1} E_2 (\overline{E_0} \vee E_0) \vee E_1 E_2 (\overline{E_0} \vee E_0)$$

$$A_2 = \overline{E_1} E_2 \vee E_1 E_2$$

In diesem Ausdruck läßt sich noch E_2 ausklammern.

$$A_2 = E_2 (\underbrace{\overline{E_1} \vee E_1}_1) = E_2$$

Die vereinfachten Schaltfunktionen lauten also:

$$A_0 = E_0 \overline{E_1} \overline{E_2}$$

$$A_1 = E_1 \overline{E_2}$$

$$A_2 = E_2$$

- c) Sie ergeben den Kontaktplan nach Bild E27.1a und den Funktionsplan nach Bild E27.1b.

Die Anweisungsliste im Mnemo-Code lautet:

```

U  E 0
UN E 1
UN E 2
=  A 0
U  E 1
UN E 2
=  A 1
U  E 2
=  A 2
=  BE

```

Wir erhalten folgendes Programm:

```

!E0&NE1&NE2=A0
!E1&NE2=A1
!E2=A2
!PE

```

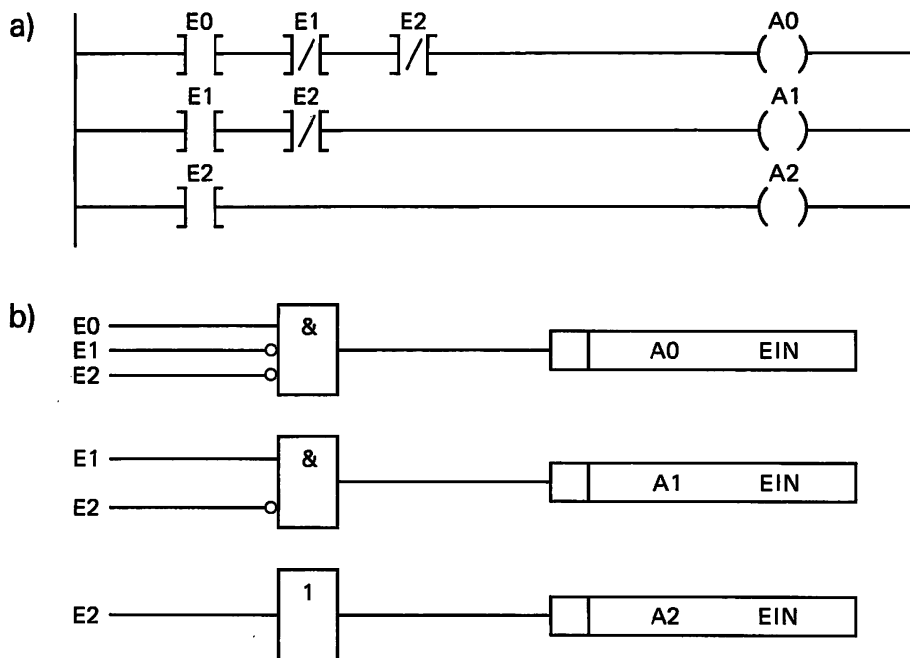


Bild E 27.1
Prioritätsschaltung, Beispiel E 25.1.
a) Kontaktplan,
b) Funktionsplan.

Geben Sie das Programm in Ihre SPS ein und testen Sie es. Der Test muß Übereinstimmung mit der Funktionstabelle E 25.1 von Punkt a) ergeben.

Im nachfolgenden Programm ist eine Schaltung zur Überwachung von Signalleitungen nachzubilden. Diese Aufgabe sollen Sie jetzt alleine lösen.

Aufgabe E 27.1

Überwachungsschaltung

Es ist eine Schaltung zu programmieren, mit der der Schaltzustand von drei Signalleitungen überwacht wird. Eingebaute Signalgeber

zeigen uns an, ob die jeweilige Leitung frei oder belegt ist. Die Zahl der belegten Signalleitungen soll gemeldet werden. Beachten Sie bitte folgende Vorgaben:

Jeder Signalleitung ist ein Geber zugeordnet. Die Geber liefern die Signale E1, E2 und E3.

„Leitung besetzt“ entspricht $E=1$

„Leitung nicht besetzt“ entspricht $E=0$

Die Leitungszustände sind wie folgt anzuzeigen:

	A1	A2	A3
eine Leitung besetzt	1	0	0
zwei Leitungen besetzt	0	1	0
drei Leitungen besetzt	0	0	1

- a) Stellen Sie bitte die vollständige Funktionstabelle auf, sie ist in Tabelle E28.1 vorbereitet.

Tabelle E28.1: Funktionstabelle zur Aufgabe E27.1

E3	E2	E1	A1	A2	A3
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

- b) Nach der Funktionstabelle sind die Schaltfunktionen für die Ausgangsvariablen aufzustellen.
- c) Zeichnen Sie nach den Schaltfunktionen von b) den Kontaktplan.
- d) Die Schaltfunktionen von b) sind zu programmieren. Anzugeben sind die Anweisungen.
- e) Geben Sie das Programm d) in die SPS ein. Die Funktion in Tabelle E28.1 und der Kontaktplan c) sind zu überprüfen.

Die Lösung dieser Aufgabe finden Sie auf Seite F14.

Aufgabe E28.1

Steuerung eines Lastaufzugs

Die Gesamtlösung erfordert außer den Verknüpfungsfunktionen auch noch Speicherfunktionen. Da Sie noch keine Speicherfunktionen programmieren können, umfaßt die Aufgabe nur das Programm für ein Schaltnetz zur Erzeugung der erforderlichen Steuersignale.

Im Lastaufzug sind drei lastabhängige Kontaktgeber eingebaut. Sie sind wie folgt eingestellt:

- Kontakt E0 geht in Zustand 1, wenn die Last ≥ 100 kg ist,
- Kontakt E1 geht in Zustand 1, wenn die Last > 600 kg ist,
- Kontakt E2 geht in Zustand 1, wenn die Last > 1200 kg ist.

Folgende Bedingungen sind von der Steuerung zu erfüllen:

Bei Lasten unter 100 kg fährt der Aufzug nicht. Liegt die Last zwischen 100 kg und 600 kg, soll er mit großer Geschwindigkeit H, bei Lasten > 600 kg bis 1200 kg soll er mit kleiner Geschwindigkeit L betrieben werden. Ab 1200 kg ist der Aufzug überlastet. Bei dieser Belastung soll ein Verriegelungssignal V entstehen, das den Aufzug blockiert.

Die Ursache einer falschen Kombination der Signale E0, E1 und E2 kann nur eine Störung in der Anlage sein. Sie wird mit einer Störungsmeldung S angezeigt.

Die verschiedenen Steuerungssignale für den Betrieb mit großer bzw. kleiner Geschwindigkeit, Verriegelung und Störung sind zu erzeugen und als Ausgangssignale A anzuzeigen. Wir legen fest:

A0=1 Fahrt mit großer Geschwindigkeit EIN (H=1)

A0=0 Fahrt mit großer Geschwindigkeit AUS (H=0)

A1=1 Fahrt mit kleiner Geschwindigkeit EIN (L=1)

A1=0 Fahrt mit kleiner Geschwindigkeit AUS (L=0)

A2=1 Verriegelung, keine Aufzugsbewegung (V=1)

A2=0 keine Verriegelung, Aufzug kann betätigt werden (V=0)

A3=1 Störung (S=1)

A3=0 Keine Störung (S=0)

a) Stellen Sie nach der verbalen Beschreibung die in Tabelle E 29.1 vorbereitete Funktionstabelle auf.

Tabelle E 29.1: Funktionstabelle zur Aufgabe E 28.1

E2	E1	E0	A0 H	A1 L	A2 V	A3 S
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

- b) Nach der Funktionstabelle sind die Schaltfunktionen für die Steuerungssignale zu entwickeln. Vereinfachen Sie die Schaltfunktionen nach den Regeln der Schaltalgebra.
- c) Stellen Sie die Programme für die Steuersignale mit den nicht-vereinfachten und vereinfachten Schaltfunktionen auf.
- d) Die Programme sind zu testen. Beide Programme müssen die gleichen Lösungen ergeben. Außerdem muß Übereinstimmung mit der Funktion in Tabelle E 29.1 bestehen.

Die Lösung dieser Aufgabe finden Sie auf Seite F 15.

Beispiel E 30.1

Betriebszustandsanzeige von drei elektrischen Geräten.

Jedes Gerät hat einen geeigneten elektrischen Schalter eingebaut, der dem Eingang der SPS den Betriebszustand des jeweiligen Geräts – entweder EIN oder AUS – mitteilt.

Festlegungen:

E=1 bedeutet: Gerät eingeschaltet

E=0 bedeutet: Gerät ausgeschaltet

Die Ausgänge der SPS sprechen Meldelampen an.

A=1 bedeutet: Meldelampe eingeschaltet

A=0 bedeutet: Meldelampe ausgeschaltet

Folgende Anzeigen werden gefordert:

1. Alle Geräte ausgeschaltet
2. Nur ein Gerät eingeschaltet
3. Nur zwei Geräte eingeschaltet
4. Alle Geräte eingeschaltet

A0	A1	A2	A3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

- a) Füllen Sie bitte die in Tabelle E 30.1 vorbereitete Funktionstabelle aus.

Tabelle E 30.1

E3	E2	E1	A0	A1	A2	A3
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

- b) Nach der Funktionstabelle sind die Schaltfunktionen für die Ausgangssignale A0, A1 und A2 aufzustellen. Das Ausgangssignal A3 ist aus A0, A1 und A2 zu erzeugen.
- c) Die Schaltfunktionen sind direkt (ohne Vereinfachung) zu programmieren. Alle Zwischenergebnisse sind in Merker zu setzen.
- d) Fertigen Sie den Kontaktplan und den Funktionsplan an.
- e) Stellen Sie die Anweisungsliste im Mnemo-Code auf.
- f) Das unter c) aufgestellte Programm ist zu testen. Überprüfen Sie experimentell die Funktionstabelle.

Lösung:

- a) Die Operanden für die Ausgangssignale enthält die Aufgabenstellung. Für die Eingangssignale gelten folgende Festlegungen:

	Operand
Schalter an Gerät 1	E1
Schalter an Gerät 2	E2
Schalter an Gerät 3	E3

In Tabelle E31.1 ist die Funktionstabelle nach der verbalen Aufgabenstellung zusammengestellt.

Tabelle E31.1: Funktionstabelle zum Beispiel E30.1

E3	E2	E1	A0	A1	A2	A3
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	1	0	0
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	0	1	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

- b) Aus der Funktionstabelle können Sie unmittelbar die folgenden Schaltfunktionen ablesen:

$$A0 = \overline{E1} \overline{E2} \overline{E3}$$

$$A1 = E1 \overline{E2} \overline{E3} \vee \overline{E1} E2 \overline{E3} \vee \overline{E1} \overline{E2} E3$$

$$A2 = E1 E2 \overline{E3} \vee E1 \overline{E2} E3 \vee \overline{E1} E2 E3$$

Aus der Tabelle E31.1 erkennen Sie: A3 hat nur dann den Wert 1, wenn die Variablen A0, A1 und A2 den Wert 0 haben. Das ist dann der Fall, wenn alle 3 Geräte eingeschaltet sind. Daraus folgt die Verknüpfung:

$$A3 = \overline{A0} \overline{A1} \overline{A2}$$

Die vorstehenden Schaltfunktionen kann man unmittelbar programmieren.

- c) Zwischenergebnisse sind in den Schaltfunktionen für A1 und A2 die UND-Verknüpfungen. Diese sind in Merker zu setzen. Wir bezeichnen den ersten Merker mit M0. Die weiteren Merker sind dann M1, M2, M3 usw.

Wir stellen nun die Programmanweisung auf:

Meldung: kein Gerät eingeschaltet. Ausgangssignal A0=1.

$!NE1 \& NE2 \& NE3 = A0$

Meldung: ein Gerät eingeschaltet. Ausgangssignal A1=1.

$!E1 \& NE2 \& NE3 = M0$

$!NE1 \& E2 \& NE3 = M1$

$!NE1 \& NE2 \& E3 = M2$

$!M0 / M1 / M2 = A1$

Meldung: zwei Geräte eingeschaltet. Ausgangssignal A2=1.

$!E1 \& E2 \& NE3 = M3$

$!E1 \& NE2 \& E3 = M4$

$!NE1 \& E2 \& E3 = M5$

$!M3 / M4 / M5 = A2$

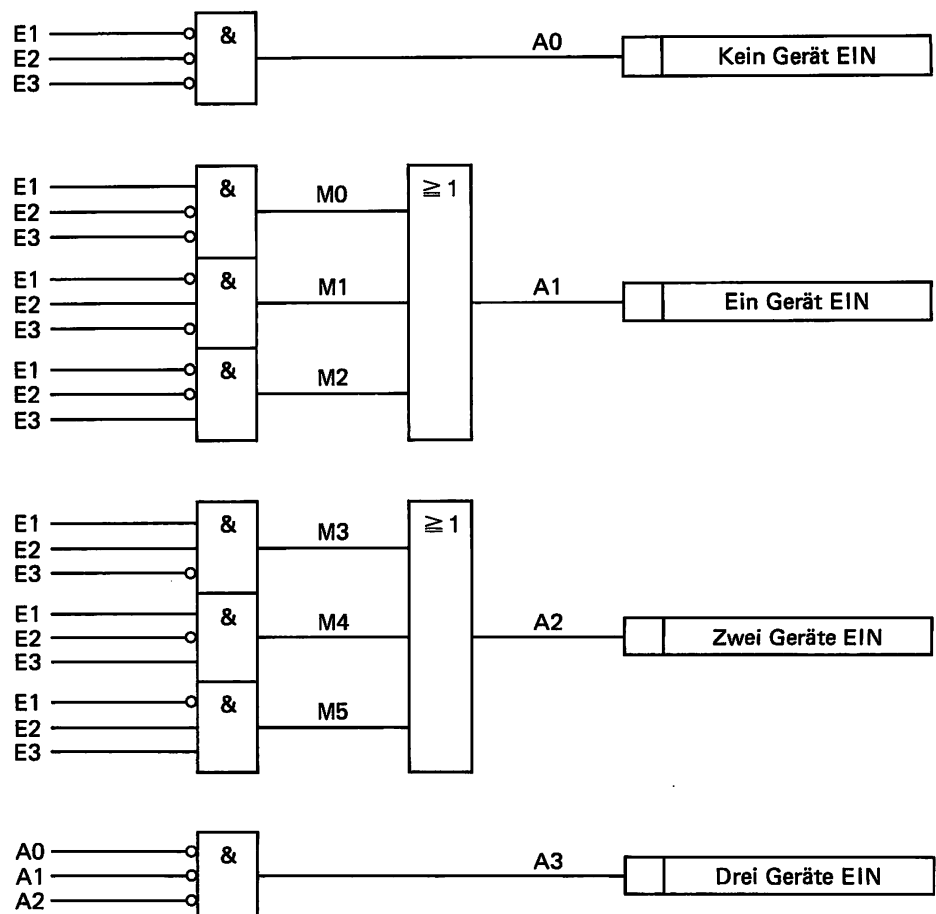


Bild E 32.1
Anzeige des Betriebszustands von
drei elektrischen Geräten, Beispiel
E 30.1 (Funktionsplan).

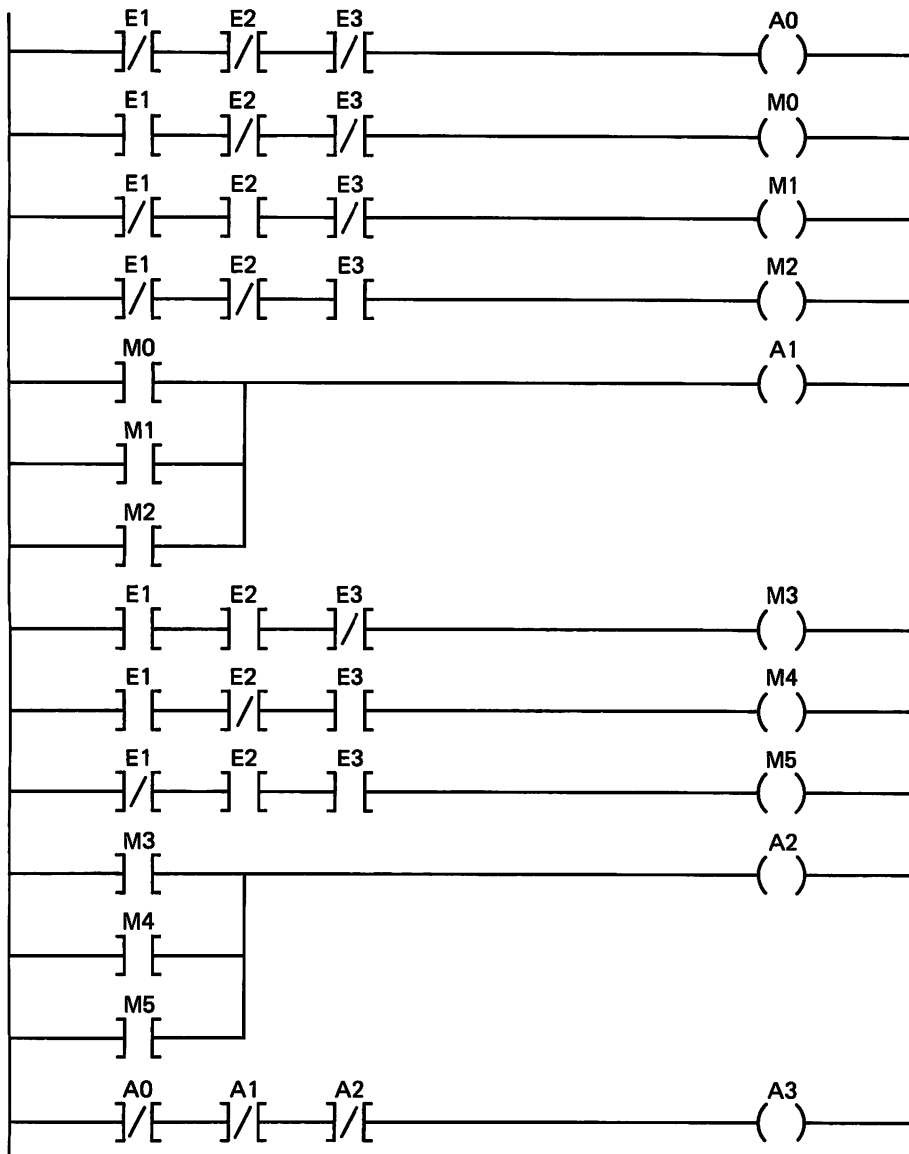


Bild E 33.1
Kontaktplan zur Anzeige des Betriebszustands von drei elektrischen Geräten, Beispiel E 30.1.

Meldung: drei Geräte eingeschaltet. Ausgangssignal A3=1.

```
!NA0&NA1&NA2=A3
!PE
```

- d) Nach dem vorangegangenen Programm haben wir den Funktionsplan und den Kontaktplan in Bild E 32.1 und in Bild E 33.1 aufgezeichnet.
- e) Die auf der nächsten Seite dargestellte Anweisungsliste im Mnemo-Code hat genau die gleiche Gliederung wie das Programm. Eingeschaltet sind:

kein Gerät	ein Gerät	zwei Geräte	drei Geräte
UN E 1	U E 1	U E 1	UN A 0
UN E 2	UN E 2	U E 2	UN A 1
UN E 3	UN E 3	UN E 3	UN A 2
= A 0	= M 0	= M 3	= A 3
	UN E 1	U E 1	= BE
	U E 2	UN E 2	
	UN E 3	U E 3	
	= M 1	= M 4	
	UN E 1	UN E 1	
	UN E 2	U E 2	
	U E 3	U E 3	
	= M 2	= M 5	
	O M 0	O M 3	
	O M 1	O M 4	
	O M 2	O M 5	
	= A 1	= A 2	

- f) Nach der Eingabe des Programms erfolgt der Test durch experimentelle Überprüfung der Funktionstabelle. Die Ergebnisse stimmen mit den Forderungen der Aufgabenstellung überein.

Programmierung einfacher Schaltwerke

Alle bisher programmierten, anspruchsvollen Aufgaben enthalten Schaltnetze. Jeder Eingangskonfiguration sind bestimmte Zustände der Ausgangsvariablen zugeordnet. Zur Programmierung der Schaltnetze haben wir nur die Programme der logischen Funktionen benötigt.

Um bestimmte Aufgaben lösen zu können, müssen zur Signalverarbeitung oft Kombinationen von logischen Verknüpfungsgliedern und Speichergliedern verwendet werden. Derartige Kombinationen bezeichnet man als **logische Schaltwerke** oder **sequentielle Schaltwerke**. Wir nennen diese Schaltungen kurz **Schaltwerke**. Analyse und Synthese der Schaltwerke und auch deren Programmierung setzen Kenntnisse über Funktion und Zusammenwirken voraus.

Wir analysieren und programmieren zunächst leicht durchschaubare Schaltwerke. Die Zusammenhänge in komplizierteren Schaltwerken lassen sich mit unseren bisher zur Verfügung stehenden Mitteln nur schwer formulieren und darstellen. Solche Aufgaben werden wir nach Bearbeitung des Fachgebietes „Methoden zur Beschreibung von Steuerungsvorgängen“ behandeln.

Beispiel E 35.1

Programmierung einer Überwachungsschaltung

Die zu programmierende Überwachungsschaltung kennen Sie schon. Sie wurde im Fachgebiet „Digitaltechnik“ im Beispiel B 41.1 untersucht. Die Schaltung ist noch einmal in Bild E 35.1 dargestellt. Lesen Sie bitte die ausführlich beschriebene Aufgabe und Funktion der Überwachungsschaltung im Beispiel B 41.1 nach. Das zu entwickelnde Programm soll die Hardware-Funktion der vorgegebenen Schaltung eindeutig nachbilden.

Lösung:

Dieses erste Programm eines Schaltwerks erläutern wir Ihnen wieder sehr ausführlich. Die Lösungen der folgenden Beispiele und Aufgaben können dann nach dem vorgeführten Schema erfolgen.

Eine direkte Programmierung des Schaltwerks ist nur bei unkomplizierten Schaltungen möglich, auch muß der Programmierer über einige Erfahrungen verfügen. Es ist besser, die vorgegebene Schaltung in **logische Funktionseinheiten** aufzulösen. Das Problem wird dadurch überschaubarer und die Programmierung wird einfach. Für die Auflösung einer Schaltung in logische Einheiten gibt es keine Vorschriften. Man trennt die Schaltung so auf, daß zwischen den jeweils vorhandenen Eingangs- und Ausgangsvariablen der logische Zusammenhang eindeutig erkennbar ist.

Auch ist es nicht immer möglich, die Hardware-Lösung eines Schaltwerks direkt zu programmieren. Sie müssen immer bedenken, daß

E
35

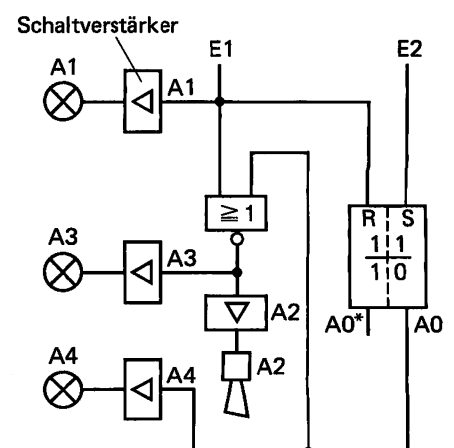


Bild E 35.1
Überwachungsschaltung zum Beispiel E 35.1. Schaltverstärker übernehmen die erforderliche Leistungsverstärkung der Ausgangssignale.

von der SPS im Gegensatz zur VPS eine **sequentielle Programmbearbeitung** erfolgt, d.h. die Anweisungen werden nacheinander Schritt für Schritt ausgeführt.

Nach der Programmierung ist immer ein umfangreicher experimenteller Programmtest erforderlich. Es ist zu überprüfen, ob das Programm auch alle Vorgaben erfüllt. Bei komplexen Programmen lassen sich nicht immer auf dem Papier alle Randbedingungen überprüfen. Aber nun wieder zurück zu unserem Problem.

In Bild E 36.1 sehen Sie die vollständig „auseinandergenommene“ Schaltung von Bild E 35.1. Würden Sie die logischen Funktionseinheiten an den Schnittstellen wieder zusammenfügen, entstünde die Ursprungsschaltung.

Das zentrale Element ist das **RS-FF**, im Programm die **RS-Speicherfunktion**. E1 und E2 steuern den Kippvorgang. Wir programmieren die Rücksetzdominanz. Dafür ist die eingezeichnete Ergänzung erforderlich. Signal E1 ist außerdem noch dem Ausgang A1 zuzuweisen. A1 steuert eine Meldeleuchte an. Diese Anweisung symbolisiert das Rechteck mit dem eingetragenen Zustand 1.

Nun folgt die NOR-Funktion. Sie verknüpft die Eingangsvariable E1 und die Ausgangsvariable A0 der Speicherfunktion. Das Verknüpfungsergebnis ist den Ausgängen A2 und A3 zuzuweisen (Hupe A2 und Meldeleuchte A3). Das Ausgangssignal A0 der Speicherfunktion muß außerdem noch das Signal A4 zur Ansteuerung einer weiteren Meldeleuchte erzeugen.

Wir müssen jetzt nur noch die einzelnen Verarbeitungsschritte programmieren. Durch Anweisungen ausgedrückt lautet das Programm:

!NE1&E2=SA0	Speicherfunktion setzen
!E1=RA0	Speicherfunktion rücksetzen
!E1=A1	Ansteuerung von A1
!E1/A0=NA2=NA3	Ansteuerung von A2 und A3
!A0=A4	Ansteuerung von A4
!PE	

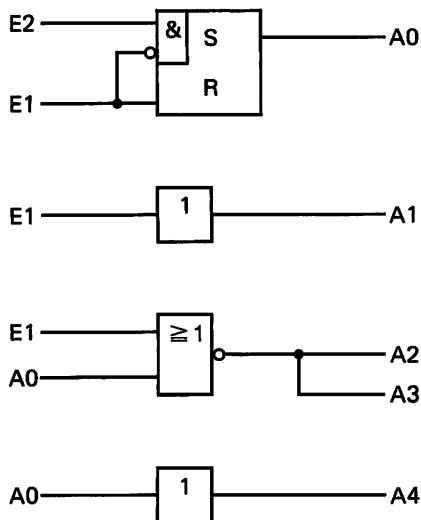


Bild E 36.1
Funktionsblöcke (aufgelöste Schaltung) der Schaltung in Bild E 35.1.

Das Programm ergibt den in Bild E 36.2 gezeichneten Kontaktplan.

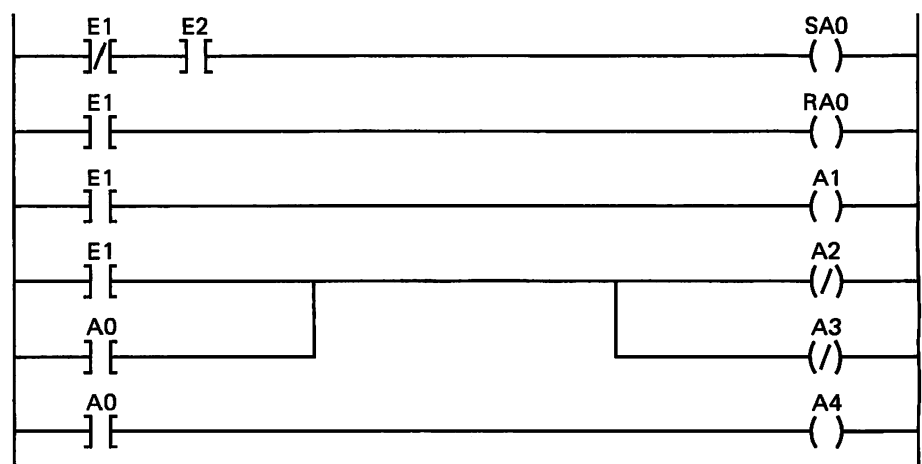


Bild E 36.2
Kontaktplan der Überwachungsschaltung.

Da wir die Dominanz der Speicherfunktion programmiert haben, ist die Reihenfolge der Speicheranweisungen im Programm gleichgültig. Falls keine logischen Fehler enthalten sind und die Regeln der Syntax beachtet wurden, erfüllt das Programm die vorgegebenen Bedingungen. Ob das der Fall ist, sollen Sie überprüfen.

Aufgabe E 37.1

Im Beispiel B 41.1 wurde für die Überwachungsschaltung folgende Funktionstabelle aufgestellt:

Tabelle E 37.1: Funktionstabelle der Überwachungsschaltung zu Aufgabe E 37.1

E1	E2	A0	A1	A2	A3	A4
1	0	0	1	0	0	0
0	0	0	0	1	1	0
0	1	1	0	0	0	1
0	0	1	0	0	0	1
1	0	0	1	0	0	0

E1 und E2 sind die Signaleingaben. Überprüfen Sie bitte experimentell, ob das auf der vorhergehenden Seite aufgestellte Programm die in Tabelle E 37.1 eingetragenen Werte der Ausgänge erzeugt.

Die Lösung dieser Aufgabe finden Sie auf Seite F 17.

Beispiel E 37.1

Steuerung eines elektrischen Antriebs

Die in Bild E 38.1 gezeigte Hardware-Lösung einer Antriebssteuerung soll von einer SPS ausgeführt werden. Für die SPS wollen wir die Anweisungsliste in der Sprache des NDR-Computers, das Programm und den Kontaktplan aufstellen.

Wirkungsweise der Schaltung:

Durch kurzzeitige Betätigung der Taster E1 oder E2 kann ein Elektromotor in die Drehrichtung „Links“ oder „Rechts“ geschaltet werden. Ein weiterer Taster E3 dient zum Ausschalten des Motors. Der Drehrichtungswechsel ist nur über den Schaltzustand AUS möglich. Die Schaltstellungen „Links“ oder „Rechts“ sind gegeneinander verriegelt. Damit wird bei Fehlbedienung ein Kurzschluß auf der Starkstromseite vermieden. Der Motor wird abgeschaltet und die Wiedereinschaltung ist blockiert, solange

- a) der mit einem Druckwächter erfaßte Schmieröldruck nicht vorhanden ist (E4=1) oder
- b) der NOT-AUS-Taster betätigt ist (E5=1).

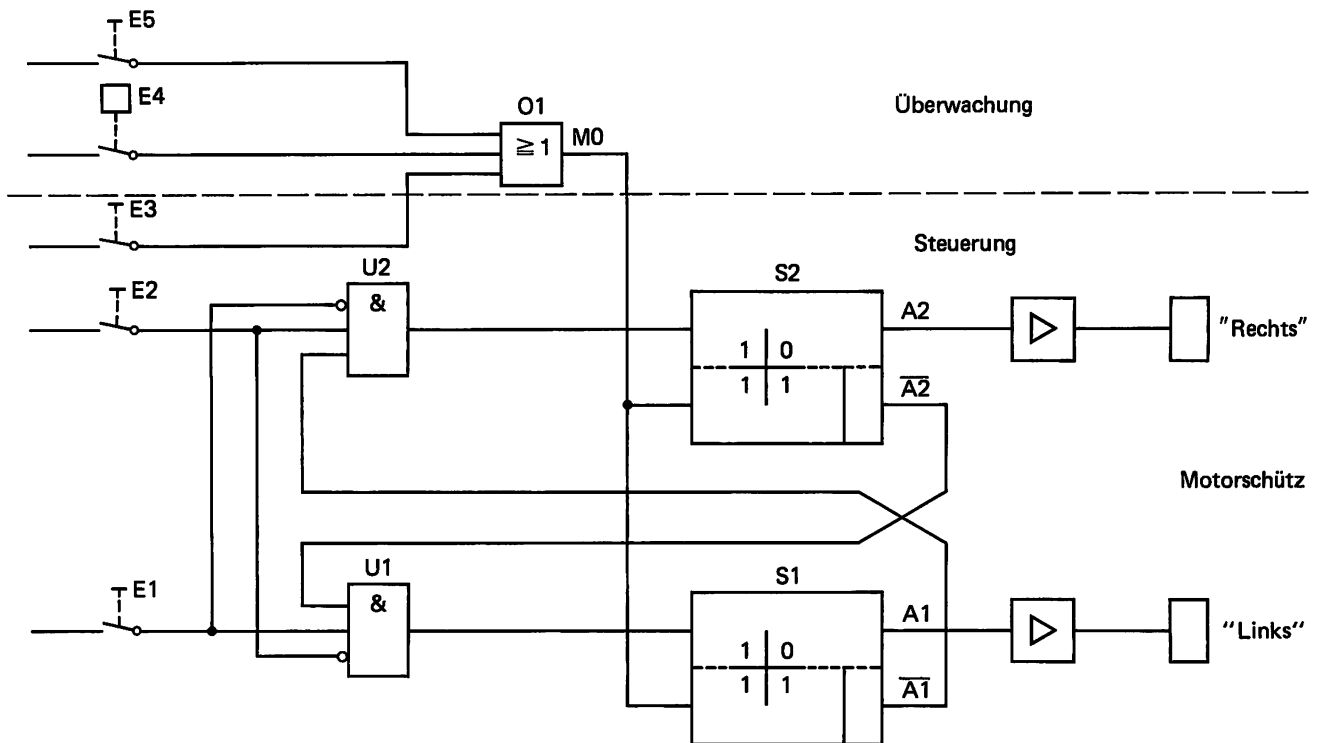


Bild E 38.1
Schaltung einer Antriebssteuerung.

Bei Signaleingabe mit einem Taster entsteht ein Kurzzeitsignal. Es muß gespeichert werden, damit es während der erforderlichen Zeit wirksam ist. Diese Aufgabe übernehmen in der Schaltung die Speicherglieder S1 und S2.

Lösung:

Wir müssen die Schaltung wieder auflösen, damit die Zusammenhänge zwischen Eingangs- und Ausgangsvariablen klar ersichtlich werden. Eine Möglichkeit dafür gibt das Bild E 39.1 wieder. Es zeigt Ihnen eine neue Art der Signaldarstellung, die in der Steuerungstechnik oft Anwendung findet. An die Signaleingänge werden nicht mehr die auslösenden Signalgeber gezeichnet, sondern nur noch deren Signale. Verbal wird angegeben, bei welcher Bedingung der Eingang den Signalzustand 1 annimmt. Zusätzlich kann noch der Operand eingetragen werden. Nach der Darstellung in Bild E 39.1 läßt sich das Programm unmittelbar aufstellen.

Zum funktionellen Verlauf geben wir Ihnen noch einige Anmerkungen. Zur Verriegelung wird der UND-Verknüpfung U2 das negierte Ausgangssignal von A1 und der UND-Verknüpfung U1 das negierte Ausgangssignal von A2 zugeführt. Sobald eine Speicherfunktion den Wert 1 hat, läßt sich die andere nicht mehr setzen. Auch bei gleichzeitiger Betätigung der beiden Eingangstaster kann keine Fehlschaltung entstehen. U1 erhält das negierte Signal von E2, U2 erhält das negierte Signal von E1.

Aus dem Funktionsplan können wir unmittelbar die Anweisungen zur Aufstellung des Programms ablesen.

Programm:

```

!E1&NE2&NA2=SA1
!NE1&E2&NA1=SA2
!E3/E4/E5=M0
!M0=RA1=RA2
!PE

```

Die Rücksetzbedingung ist für beide Speicherfunktionen dieselbe. Sie wurde deshalb nur einmal programmiert und beiden Speicherfunktionen zugewiesen. Da Rücksetzdominanz besteht, steht die betreffende Anweisung hinter den Setzanweisungen. Das Programm als Anweisungsliste in der Sprache des NDR-Computers kann aus den vorstehenden Anweisungen aufgestellt werden.

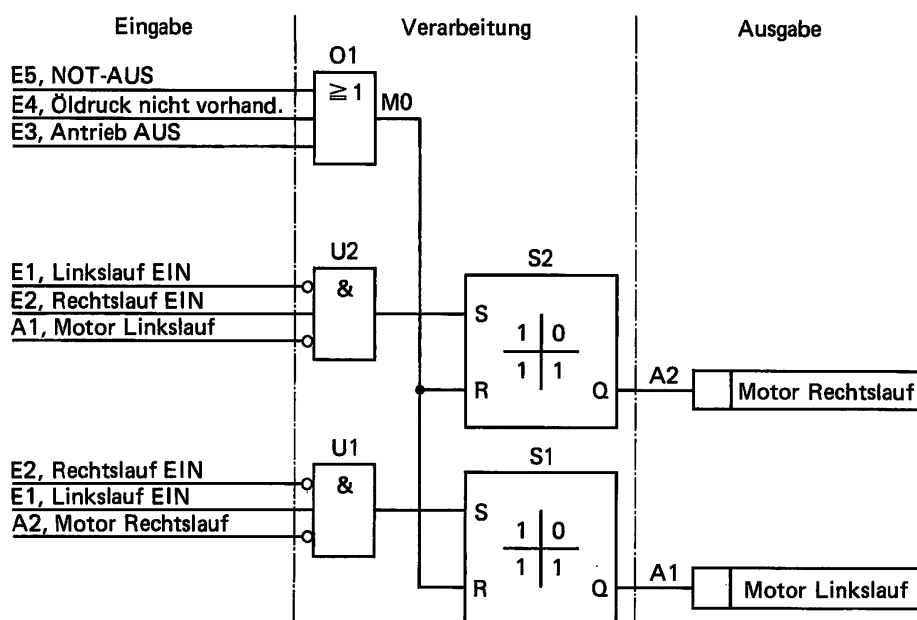


Bild E 39.1
Aufgelöste Schaltung der Antriebs-
steuerung.

Anweisungsliste:

```

!E1
&NE2
&NA2
=SA1  Motor „Links“ EIN
!NE1
&E2
&NA1
=SA2  Motor „Rechts“ EIN
!E3   Motor „AUS“
/E4   kein Öldruck
/E5   NOT-AUS
=M0   Rücksetzbedingung
!M0
=RA1  Motor AUS
=RA2  Motor AUS
!PE

```

Nach dem Programm ist der Kontaktplan aufgezeichnet und in Bild E40.1 wiedergegeben.

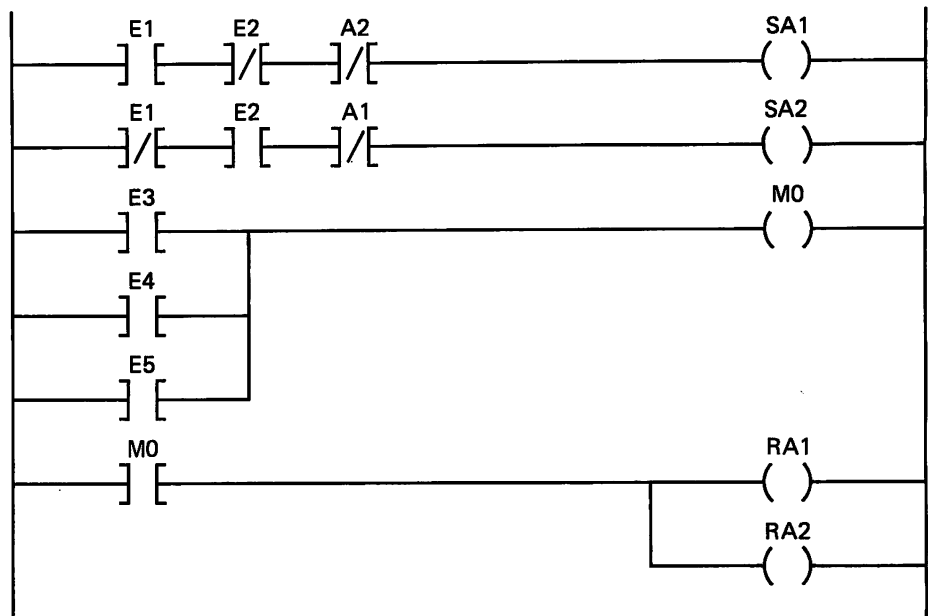


Bild E40.1
Kontaktplan der Antriebssteuerung.

Aufgabe E40.1

Test des Programms „Antriebssteuerung“

Der NDR-Computer ist mit dem soeben aufgestellten Programm zu laden. Sie sollen das Programm testen. Die Funktionstabelle E40.1 enthält eine Folge verschiedener Eingaben. Es ist angegeben, welche Funktionen dadurch aufgerufen werden. Tragen Sie die Signalausgaben in die Tabelle ein und überprüfen Sie, ob unsere SPS richtig reagiert.

Tabelle E40.1: Funktionstabelle zu Aufgabe E40.1

E5	E4	E3	E2	E1	A1	A2	Bemerkung
0	0	0	0	1			Motor „Linkslauf“ EIN
0	0	0	0	0			
0	0	0	1	0			Motor „Rechtslauf“ EIN (E2 betätigt)
0	0	1	0	0			Motor AUS
0	1	0	1	0			Motor „Rechtslauf“ EIN (E2 betätigt, kein Öldruck)
0	0	0	1	0			Motor „Rechtslauf“ EIN (Öldruck o. k.)
1	0	0	0	0			NOT-AUS

Vergleichen Sie außerdem den Kontaktplan von Bild E40.1 mit dem des Programms (Menü 3).

Die Lösung dieser Aufgabe finden Sie auf Seite F17.

Beispiel E 41.1

Programmierung einer Meldeschaltung

Die zu programmierende Schaltung ist in Bild E 41.1 wiedergegeben. Sie soll einen bestimmten Betriebszustand, der durch den Zustand des Signals E1 gemeldet wird, mit Dauerlicht anzeigen.

Wirkungsweise der Schaltung:

Bei $E1=1$ soll die Meldung $A1=1$ erscheinen. Mit $E2=1$ ist die Meldung zu quittieren. Besteht bei der Quittierung noch der Signalzustand $E1=1$, so soll die Meldung $A1=1$ solange bestehen bleiben, bis $E1=0$ wird. Ist dagegen bei der Quittierung Zustand $E1=0$ vorhanden, soll $A1$ sofort in Zustand 0 gehen.

Die vorgegebene Schaltung in Bild E 41.1 ist wieder in überschaubare Funktionseinheiten aufzulösen. Danach sind Kontaktplan und Programm zu entwickeln. Anschließend soll das Programm durch die Aufnahme einer Funktionstabelle getestet werden.

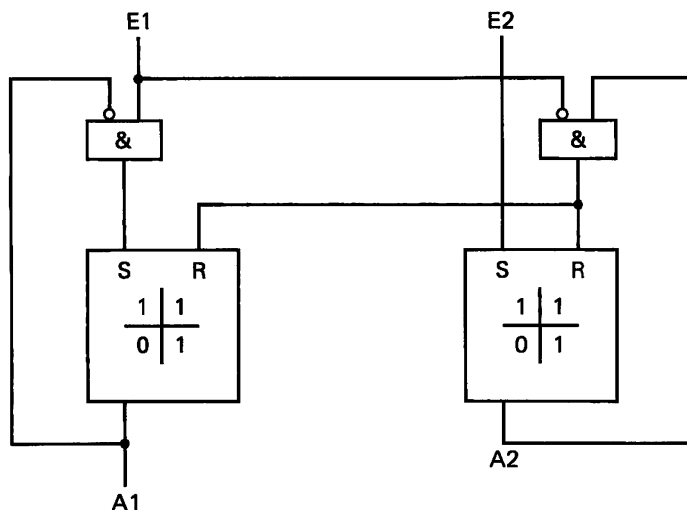


Bild E 41.1
Meldeschtung, Beispiel E 41.1.

Lösung:

Überprüfen Sie zunächst, ob die Schaltung in Bild E 41.1 alle Vorgaben erfüllt. Danach lösen wir die Schaltung in leicht zu programmierende Funktionsblöcke auf, einen Vorschlag dafür enthält das Bild E 41.2.

Die Umsetzung der Funktionsblöcke in den Kontaktplan ergibt das Schema von Bild E 42.1. Die Rücksetzbedingungen sind für beide Speicherfunktionen gleich. Aus dem Kontaktplan lassen sich unmittelbar die Anweisungen für das Programm ablesen.

Die Anweisungen des Programms lauten:

```
!E1&NA1=SA1
!E2=SA2
!NE1&A2=RA1=RA2
!PE
```

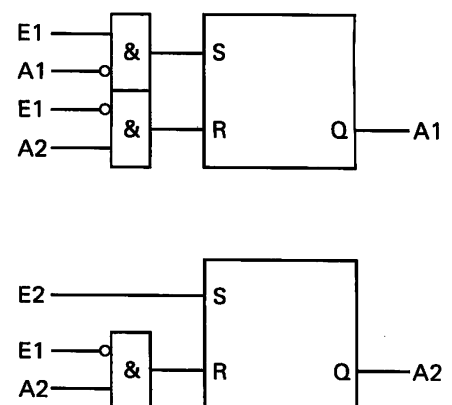
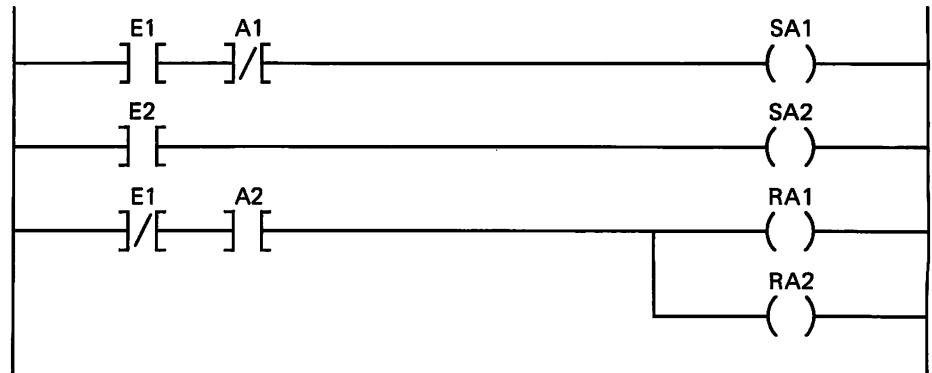


Bild E 41.2
Meldeschtung von Bild E 41.1,
aufgelöst in Funktionsblöcke.

Bild E42.1
Kontaktplan der Meldeschaltung
von Bild E41.1.



Die nachstehende Funktionstabelle E 42.1 enthält die theoretisch an Hand der Schaltung in Bild E 41.1 ermittelten Zustände der Ausgangssignale für verschiedene, praktisch mögliche Kombinationen der Eingangssignale. Überprüfen Sie die Vorgaben bitte experimentell. Vergleichen Sie dabei wieder den entworfenen Kontaktplan mit den vom Computer ausgegebenen Plan.

Tabelle E 42.1: Funktionstabelle zu Beispiel E 41.1

E1	E2	A1	A2	A1e	A2e
1	0	1	0		
1	1	1	1		
1	0	1	1		
0	0	0	0		
1	0	1	0		
0	0	1	0		
0	1	0	0		
0	0	0	0		

A1,2 theoretisch ermittelte
Ergebnisse

A1e,2e experimentell ermittelte
Ergebnisse

Jetzt sollen Sie selbst kontrollieren, ob Sie die Grundlagen der Programmierung von Speicherfunktionen und einfachen Schaltwerken verstanden haben. Lösen Sie bitte die folgenden Aufgaben selbstständig. Wir geben Ihnen selbstverständlich wieder Lösungsvorschläge an. Es kann aber durchaus möglich sein, daß Ihr Programm von unserem Lösungsvorschlag abweicht und dennoch beide Programme zum richtigen Ergebnis führen.

Zunächst eine sehr einfache Aufgabe, die Sie schon im Fachgebiet „Digitaltechnik“ untersucht haben.

Aufgabe E 42.1

Meldung des Zustands eines Signalgebers

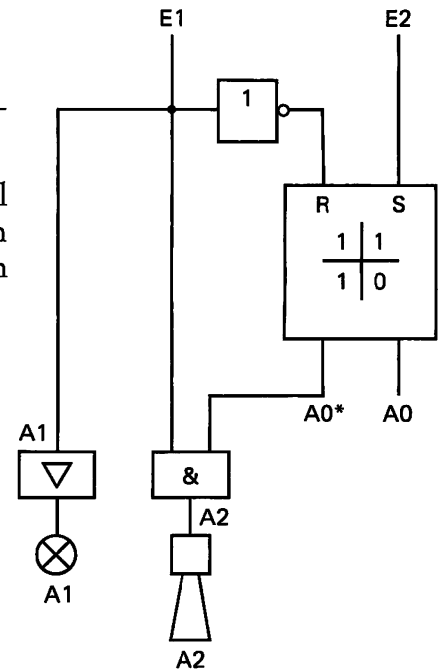
In Aufgabe B 43.1 finden Sie die in Bild E 43.1 wiedergegebene Schaltung beschrieben. Sie haben die Aufgabe:

- die Schaltung von Bild E 43.1 in Funktionsblöcke, die sich leicht programmieren lassen, aufzulösen,
- das Programm als Kontaktplan aufzuzeichnen,
- das Programm in Form von Anweisungen aufzustellen,
- das Programm c) als Anweisungsliste (NDR-Computer) wiederzugeben und
- das Programm c) in Ihre SPS einzugeben. Danach ist experimentell die vorbereitete Funktionstabelle E 43.1 aufzunehmen. Außerdem ist Ihr Kontaktplan b) mit dem vom Computer ausgegebenen zu vergleichen.

Tabelle E 43.1: Funktionstabelle zur Aufgabe E 42.1

E1	E2	A0	A1	A2	Bemerkung
0	0				keine Störung
1	0				Störung
1	1				Meldung quittiert
1	0				Störung noch vorhanden
0	0				Störung beseitigt

Die Lösung dieser Aufgabe finden Sie auf Seite F 18.

Bild E 43.1
Meldeschiung, Aufgabe E 42.1.E
43

Aufgabe E 43.1

Steuerung eines Schieberantriebs

Das Schema der Steuerung geht aus Bild E 44.1 hervor. Die Steuerung soll die zwei Befehle „Schieber-AUF“ oder „Schieber-ZU“ und die Meldung „Schieber-HALT“ ausgeben. Ein Ausgang steuert den Schieberantrieb in Richtung „Schieber AUF“, der andere in Richtung „ZU“. Führt keiner der beiden Ausgänge Zustand 1, so verharrt der Schieber in der momentanen Lage (HALT). Dieser Zustand ist am dritten Ausgang anzuzeigen. Es darf immer nur ein Ausgang den Zustand 1 führen.

Über Taster werden die Eingangssignale AUF, ZU und HALT eingegeben. Sie sind nur bei Zustand 1 wirksam. Die Befehle AUF und ZU werden in **Merkern** gespeichert. Über ein **Freigabesignal** E0 sind die gespeicherten Merkersignale auf die Ausgänge geschaltet. E0=1 gibt die Signalausgabe frei, E0=0 sperrt die Ausgabe.

Sobald der Schieber die Endstellung AUF oder ZU erreicht hat, entstehen die Rückmeldungen „Endlage AUF“ bzw. „Endlage ZU“ mit Signalzustand 1. Diese müssen den betreffenden Merker sofort rücksetzen. Außerdem sind die Speicherfunktionen noch gegeneinander zu verriegeln. Ist beispielsweise Merker M1 gesetzt, so darf keine Möglichkeit bestehen, Merker M2 in die Setzlage zu steuern und umgekehrt.

Das Schema in Bild E 44.1 enthält die notwendigen Operanden. Die Bedeutung aller Signalzustände geht aus dem Text hervor.

- Entwickeln Sie zunächst eine Schaltung. Die Aufgabe sollte mit Bausteinen der Digitaltechnik gelöst werden. Lösen Sie die Schaltung in einfache Funktionseinheiten auf.
- Stellen Sie nach der aufgelösten Schaltung die Anweisungen für das Programm zusammen.
- Zeichnen Sie danach den Kontaktplan auf.
- Programmieren Sie Ihre SPS. Vergleichen Sie Ihren Kontaktplan mit dem vom Computer ausgegebenen Plan.
- Nun ist noch der Funktionstest auszuführen. Die nachstehende Funktionstabelle enthält einige mögliche Eingangssignalkombinationen. Ermitteln Sie experimentell die Ausgangssignale.

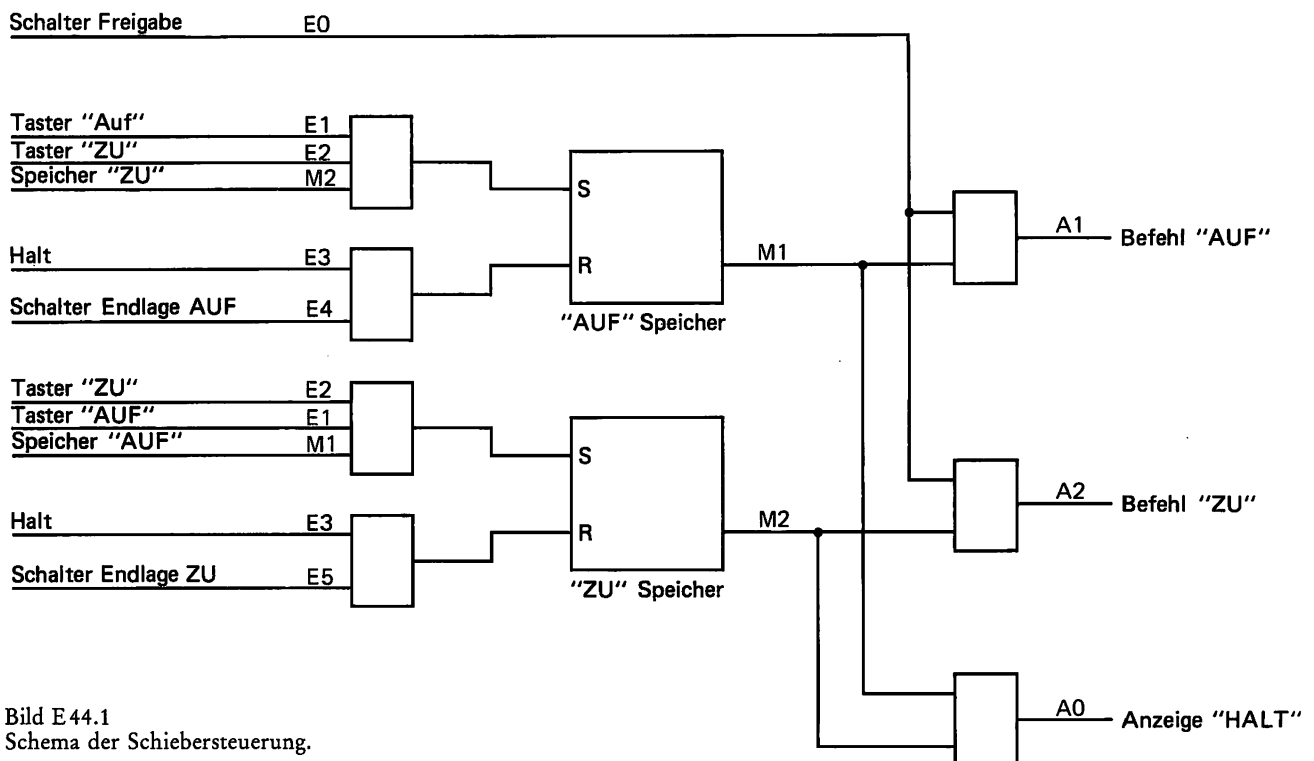


Tabelle E 44.1: Funktionstabelle zu Aufgabe E 43.1

E5	E4	E3	E2	E1	E0	A0	A1	A2	Bemerkung
0	0	1	0	0	0				Antrieb HALT
0	0	0	0	0	1				Freigabe
0	0	0	0	1	0				Antrieb AUF, keine Freigabe
0	0	0	0	1	1				Antrieb AUF, Freigabe
0	0	0	0	0	1				Antrieb AUF gespeichert
0	1	0	0	0	1				Endlage AUF erreicht
0	1	0	0	1	1				Fehlschaltung
0	1	0	1	0	1				Antrieb ZU, Freigabe
0	0	0	0	0	1				Antrieb ZU gespeichert
0	0	1	0	0	1				Antrieb HALT

Die Lösung dieser Aufgabe finden Sie auf Seite F 18.

Anwendung des Funktionsplans als Hilfsmittel zur Programmerstellung

Der Funktionsplan und das Programm stehen im Mittelpunkt des folgenden Lehrstoffs. Bearbeitet werden vorzugsweise Schaltnetze und Schaltwerke aus verschiedenen Anwendungsbereichen. Da oft Zeitfunktionen erforderlich sind, beginnen wir mit einigen Timer-Programmen.

Anwendungen von Zeitfunktionen

Beispiel E45.1

Programmierung einer monostabilen Kippschaltung

Im Abschnitt Speicherglieder (Lehrbrief 2, Seite B35) wurden bereits die Eigenschaften der Kippschaltungen besprochen. In unserem Beispiel soll das Programm für die Erzeugung eines Rechtecksignals (Rechteckimpuls) mit einer bestimmten Zeitdauer aufgestellt werden. Die Zeitdauer des auslösenden Eingangssignals darf keinen Einfluß auf die Dauer des programmierten Rechtecksignals haben. In der Starkstromtechnik und bei VPS wird eine Schaltung mit diesen Eigenschaften als „Blocker“ bezeichnet.

Lösung:

Das Bild E 45.1 zeigt das Funktionssymbol und das geforderte Signal-Zeit-Diagramm. Den Funktionsplan, der die gestellten Forderungen erfüllt, wollen wir besprechen und programmieren. In Bild E 46.1 sehen Sie einen Vorschlag in geschlossener a) und in aufgelöster b) Darstellung. Der aufgelöste Plan b) ist aus dem geschlossenen Plan a) entwickelt. Wenn Sie die im Plan b) in GRÜN eingezeichneten Schnittstellen 1, 2 und 3 miteinander verbinden, entsteht Plan a).

Wirkungsweise: Signal E1 startet den Kippvorgang. Zustand E1=1 setzt die Speicherfunktion (A2=1). Die UND-Verknüpfung gibt den Zustand des Signals E1 weiter, da der Ausgang der Zeitfunktion T2 den Wert 0 hat. Gleichzeitig wird auch die Zeitfunktion, die mit einer Einschaltverzögerung t_1 programmiert ist, gestartet. Sollte in der Zwischenzeit E1 wieder den Wert 0 angenommen haben, dann hält das auf den Eingang der ODER-Verknüpfung geführte Ausgangssignal A2 die Bedingung für die Funktion des Timers T2 aufrecht. Sobald der Ausgang des Timers den Zustand 1 annimmt, wird die Speicherfunktion rückgesetzt, A2 nimmt den Wert 0 an. Die Zeitdauer (Impulsdauer) des Ausgangssignals A2 entspricht somit der Einschaltverzögerung t_1 der Zeitfunktion.

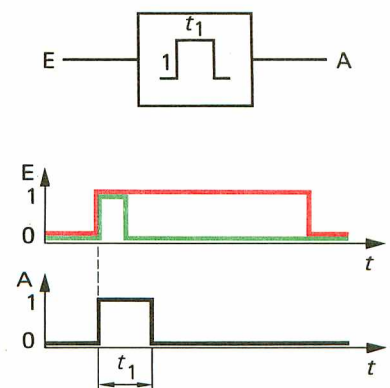


Bild E 45.1
Monostabile Kippschaltung; die Kippzeit t_1 ist unabhängig von der Dauer des Eingangssignals.

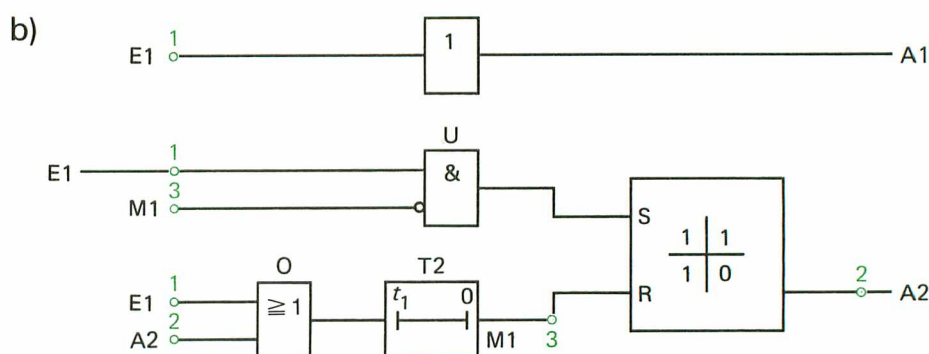
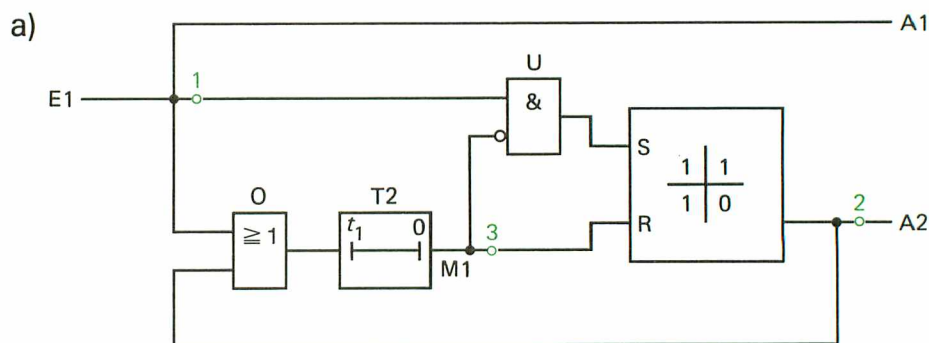


Bild E 46.1
Impulsfunktion, Nachbildung der
monostabilen Kippstufe.
a) geschlossener Funktionsplan,
b) offener Funktionsplan.

Im FUP zeigt die Ausgangsvariable A1 nur das den Kippvorgang auslösende Signal E1 an. Der beschriebene Schaltvorgang kann erst nach Ablauf eines vollständigen Kippvorgangs erneut gestartet werden.

Aus dem aufgelösten FUP können Sie unmittelbar folgendes Programm aufstellen:

Programm:

!E1=A1	Start des Kippvorgangs
!E1/A2=T2	Start des Timers T2
!T2=M1	
!M1=RA2	Rücksetzen der Speicherfunktion
!E1&NM1=SA2	Setzen der Speicherfunktion
!PE	

Wir wollen das Programm sofort laufen lassen. Stellen Sie bitte die Impulszeit von $t_1=5\text{ s}$ ein. Diese Zeit kann man mit dem Sekundenzeiger der Armbanduhr näherungsweise überprüfen. Verwenden Sie die im Programm aufgeführten Operanden.

Timer T2 ist mit einer Einschaltverzögerung $t_1=5\text{ s}=250\text{ ZE}$ und einer Ausschaltverzögerung $t_2=0\text{ ZE}$ zu programmieren.

Der Test muß Übereinstimmung mit der Aufgabenstellung ergeben.

Aufgabe E 47.1

Programmierung einer Ausschaltverzögerung

In einem Programm wird eine Zeitfunktion mit einer Ausschaltverzögerung benötigt. Angenommen, es stehen nur Timer mit programmierbarer Einschaltverzögerung zur Verfügung. Untersuchen Sie bitte, ob der Funktionsplan einer „Trick-Schaltung“ in Bild E 47.1 eine Zeitfunktion mit Ausschaltverzögerung ergibt.

- Entwickeln Sie den aufgelösten FUP mit den in Bild E 47.1 eingetragenen Schnittstellen.
- Nach dem Plan aus a) ist das Programm aufzustellen.
- Entwerfen Sie den KOP.
- Verwenden Sie zum Test Timer T0. T0 ist mit einer Einschaltverzögerung $t_1=4\text{ s}$ und der Ausschaltverzögerung $t_2=0\text{ s}$ zu programmieren.

Anschließend ist das Programm aus b) in die SPS einzugeben und zu testen. Überprüfen Sie Ihren entworfenen KOP mit dem des Programms im Menü 3.

Die Lösung dieser Aufgabe finden Sie auf Seite F 34.

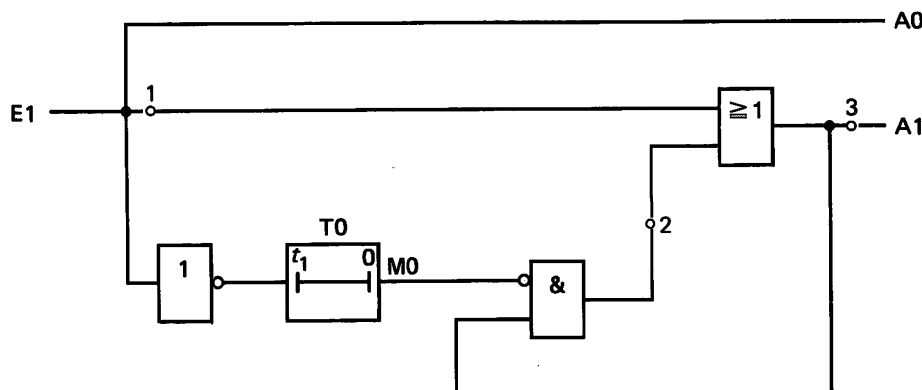


Bild E 47.1
Funktionsplan zur Erzeugung einer Ausschaltverzögerung.
Annahme: Timer T0 ist nur für eine Einschaltverzögerung t_1 programmierbar.

Zeitfunktionen werden wir in den kommenden Beispielen und Aufgaben noch oft einsetzen. Die in den folgenden Beispielen zu programmierenden Schaltnetze sind etwas aufwendiger als die bisher untersuchten. Bei diesen treten bereits Kombinationen von Verknüpfungs- und Zeitfunktionen auf. Auch die Schaltungsgebra kommt wieder zur Anwendung.

Das erste Beispiel bearbeiten wir zur Einführung gemeinsam.

Programme allgemeiner Steuer- und Überwachungsschaltungen

Beispiel E 47.1

Programm zur Anzeige des Schaltzustands von zwei Lüftern.

Zwei Lüfter werden in Abhängigkeit von der Raumtemperatur über Thermoschalter selbständig ein- und ausgeschaltet. Eine Melde-

leuchte soll den Schaltzustand der Lüfter wie folgt anzeigen:

- Kein Lüfter eingeschaltet: Dauerlicht
- Ein Lüfter eingeschaltet: Blinklicht mit der Frequenz 1 Hz
- Zwei Lüfter eingeschaltet: Blinklicht mit der Frequenz 3 Hz.

Für die Betriebsmittel gilt folgende Zuordnung:

Betriebsmittel-Kennzeichen	Funktion	Operand
Eingänge:		
B1	Thermoschalter Lüfter 1	E1
B2	Thermoschalter Lüfter 2	E2
Ausgang:		
H1	Meldeleuchte	A1
Timer:		
	Taktgenerator G1, 1 Hz	T0, T1
	Taktgenerator G2, 3 Hz	T2, T3

Bei eingeschaltetem Lüfter hat die jeweilige Variable E den Zustand 1, im anderen Fall Zustand 0. Hat A1 den Zustand 1, dann befindet sich die Meldeleuchte im Zustand EIN, Wert 0 bedeutet Zustand AUS.

Falls erforderlich, ist das Zwischenergebnis einer Verknüpfung in einen Merker zu setzen. Die Merker stehen ab M0 zur Verfügung.

Lösen Sie das Beispiel in folgenden Schritten:

- a) Die Schaltfunktionen für die verschiedenen Schaltzustände der Lüfter sind aufzustellen.
- b) Mit Hilfe der Schaltfunktionen und den Bedingungen laut Aufgabenstellung ist der FUP zu entwickeln.
- c) Programmieren Sie die Taktgeneratoren.
- d) Stellen Sie das Gesamtprogramm der Meldeschaltung auf.
- e) Nunmehr ist nach dem Gesamtprogramm der KOP aufzuzeichnen.
- f) Das Programm ist mit unserer SPS zu testen. Überprüfen Sie, ob Ausgabe A1 die Meldungen gemäß Aufgabenstellung abgibt. Die LED muß ununterbrochen leuchten (Dauerlicht) oder mit einer bestimmten Frequenz blinken (Blinklicht).

Lösung:

- a) Schaltfunktionen:

Wir wollen in diesem Beispiel die Schaltfunktionen so formulieren, daß sie ohne Umstellung programmierbar sind. Zur Aufbewahrung von Zwischenergebnissen sind Merker erforderlich.

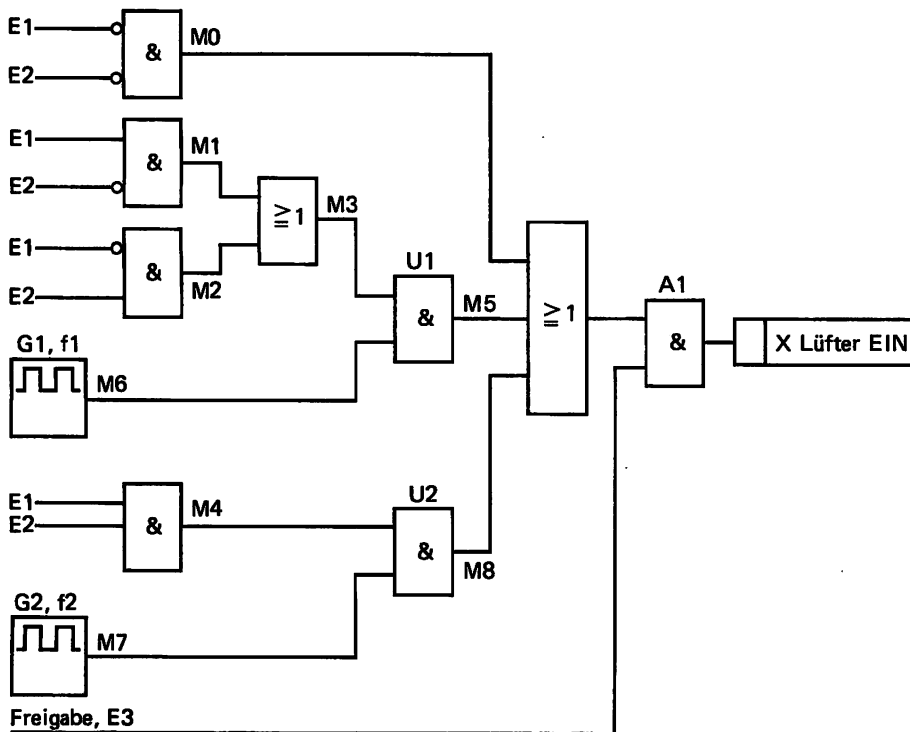


Bild E49.1
Funktionsplan zur Anzeige des
Schaltzustands von zwei Lüftern,
Beispiel E47.1.

Kein Lüfter eingeschaltet: $\overline{E1} \wedge \overline{E2} = M0$

Ein Lüfter eingeschaltet: $E1 \wedge \overline{E2} = M1$

$\overline{E1} \wedge E2 = M2$

$M1 \vee M2 = M3$

Zwei Lüfter eingeschaltet: $E1 \wedge E2 = M4$

b) Den Funktionsplan zeigt das Bild E49.1.

Zur Vereinfachung sind im FUP die Eingänge der Verknüpfungsfunktionen nur mit den Operanden bezeichnet. Die Bedeutung der Operanden sehen Sie in der Zuordnungsliste. Im Plan erfolgt die Signalverknüpfung nach den aufgestellten Schaltfunktionen. Das Taktsignal von Generator G1 bzw. G2 öffnet und schließt periodisch die UND-Verknüpfungen U1 bzw. U2. Bei Erfüllung der jeweiligen Schaltbedingung entsteht am Meldeausgang ein Blinklichtsignal. Mit Signal E3=1 wird der Ausgang freigegeben.

c) Programmierung der Taktgeneratoren

Wir verwenden das Programm: „Taktgeneratoren mit zwei Zeitfunktionen“ (siehe ab Seite D 39).

Taktgenerator G1:

Das Taktsignal mit $f_1=1\text{ Hz}$ hat eine Periodendauer $T_1=1/f_1=1/1\text{ Hz}=1\text{ s}$. Bei der Definition „Impulszeit=Pausenzeit“ muß die Einschaltverzögerung je Zeitfunktion $t_1=1/2T=0,5\text{ s}$ betragen. $t_1=500\text{ ms}$ entsprechen 25 ZE. Das Taktsignal soll nicht dauernd am

Ausgang zu sehen sein, wir verwenden deshalb im Programm Merker M6 als Speicherfunktion.

Programm für G1:

```
!NM6=T0
!T0=SM6
!M6=T1
!T1=RM6
```

Taktgenerator G2:

Für die Taktfrequenz $f_2 = 3 \text{ Hz}$ ergeben sich folgende Werte:

$T_2 = 1/f = 1/3 \text{ Hz} = 0,33 \text{ s}$
 $t_1 = T_2/2 \approx 0,165 \text{ s} \approx 8 \text{ ZE}$

Das Programm wird mit Merker M7 aufgestellt.

Programm für G2:

```
!NM7=T2
!T2=SM7
!M7=T3
!T3=RM7
```

d) Gesamtprogramm:

Das folgende Programm ist direkt nach dem Funktionsplan aufgestellt. Vergleichen Sie bitte die folgenden Anweisungen mit dem FUP.

Am Anfang stehen die Programme für die Taktgeneratoren. Diese Teilprogramme könnten auch am Ende des Gesamtprogramms stehen. Auf die Verarbeitung der Meldesignale haben diese Programme keinen Einfluß.

!NM6=T0	
!T0=SM6	
!M6=T1	
!T1=RM6	Taktsignal der Frequenz f_1
!NM7=T2	
!T2=SM7	
!M7=T3	
!T3=RM7	Taktsignal der Frequenz f_2
!NE1&NE2=M0	kein Lüfter eingeschaltet
!E1&NE2=M1	
!NE1&E2=M2	
!M1/M2=M3	
!M3&M6=M5	ein Lüfter eingeschaltet
!E1&E2=M4	
!M4&M7=M8	zwei Lüfter eingeschaltet
!(M0/M5/M8)&E3=A1	Ausgabe des Meldesignals
!PE	

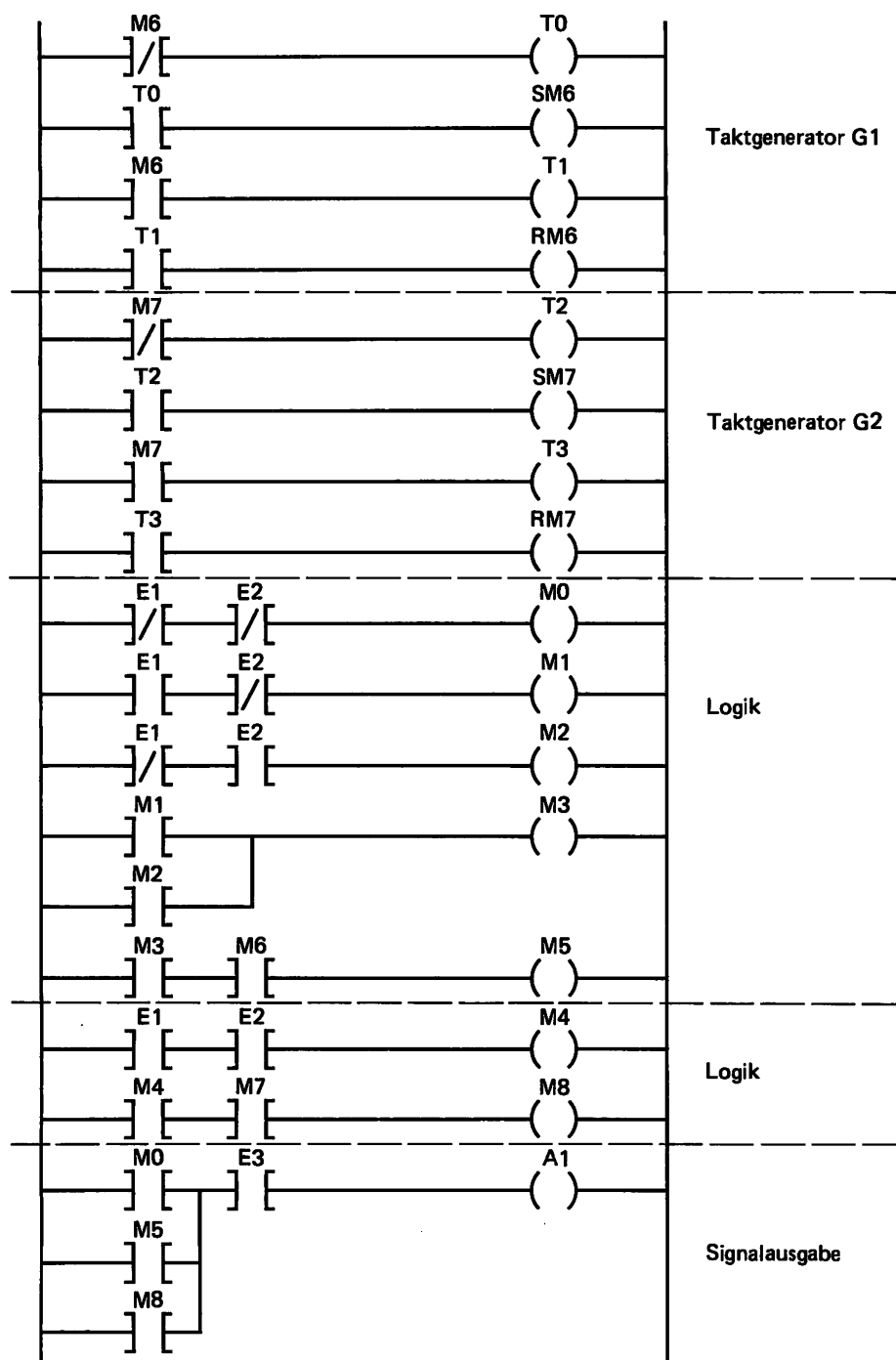


Bild E 51.1
Kontaktplan zum Beispiel E47.1.

- e) Den nach dem Programm aufgestellten KOP zeigt das Bild E 51.1.
f) Der Programmtest ergibt Übereinstimmung mit der Aufgabenstellung.

Mit Ihren bisher erworbenen Kenntnissen haben Sie sicher den Lösungsweg des letzten Beispiels gut verfolgen können. In diesem wie auch in den folgenden Programmen lassen sich manche Anweisungen einsparen oder auch anders formulieren. Weniger Anweisungen ergeben eine kürzere Zykluszeit. Von uns wurde ein Lösungsweg vorgeschlagen, der Ihnen anschaulich zeigt, wie man für eine gestellte Aufgabe ein Programm findet. Wir haben nicht nach einer optimalen Lösung gesucht.

Aufgabe E52.1

Programm einer Auswahlhaltung

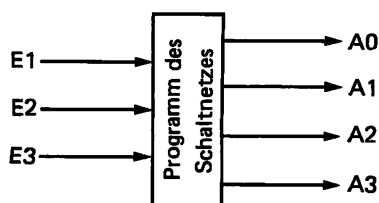


Bild E 52.1
Blockschaltung zur Aufgabe E 52.1

Es ist ein Programm aufzustellen, das die Eingangsvariablen E1, E2 und E3 nach den Vorgaben der Funktionstabelle E 52.1 verknüpft. Das Bild E 52.1 zeigt die Aufgabe als Blockschaltung. Es darf immer nur eine Ausgangsvariable den Zustand 1 haben.

- Stellen Sie die Schaltfunktionen für die Ausgangsvariablen A0 bis A3 auf. Sie sind, falls möglich, mit der Schaltalgebra zu vereinfachen.
- Mit den vereinfachten Schaltfunktionen ist das Programm aufzustellen. Verwenden Sie die angegebenen Operanden.

Tabelle E 52.1: Funktionstabelle zur Aufgabe E 52.1

E3	E2	E1	A0	A1	A2	A3
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	1	0	0
1	0	0	0	0	0	1
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	1	0	0

- Nach dem Programm sind FUP und KOP zu entwerfen.
- Überprüfen Sie experimentell, ob die Ergebnisse Ihres Programms mit der vorgegebenen Funktionstabelle E 52.1 übereinstimmen.

Einen Lösungsvorschlag finden Sie auf Seite F 34 und F 35.

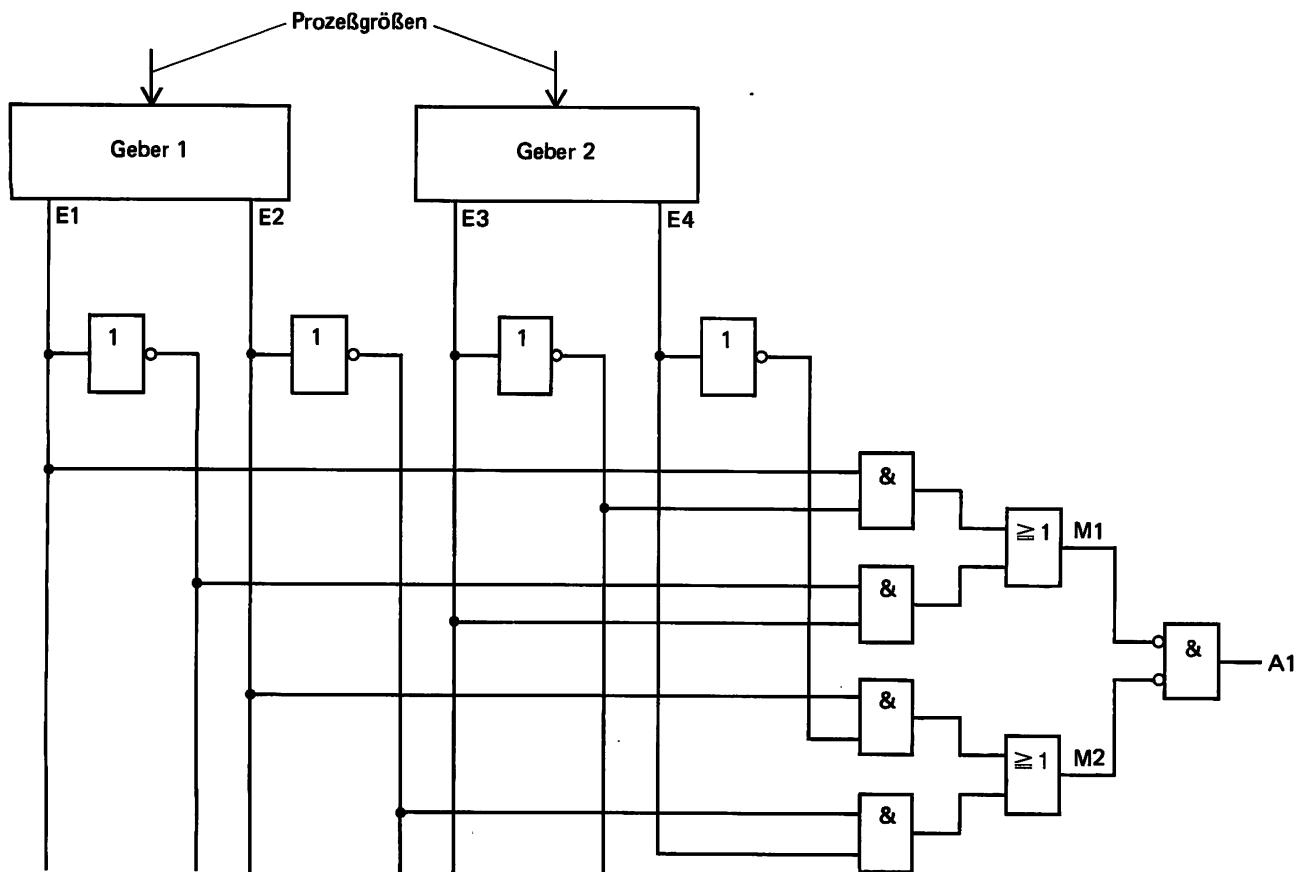
Aufgabe E52.2

Programm eines 2-Bit-Komparators

Ein Komparator ist ein Vergleicher. Es gibt Vergleicherschaltungen für analoge und für binäre Signale. Der nachstehend beschriebene Komparator soll zwei Signale, die aus je zwei Bit bestehen, miteinander vergleichen. Das Ergebnis des Vergleichs erzeugt am Komparatorausgang ein Signal mit dem Wert 1 oder 0.

Die Hardwarelösung des 2-Bit-Komparators zeigt das Bild E 53.1. Binäre Signalgeber erzeugen die prozeßabhängigen Signalkombinationen E1E2 und E3E4. Diese Kombination wird von der Logik verglichen. Nur bei bestimmten Signalzuständen der Kombinationen entsteht am Ausgang der Logik der Zustand A1=1.

- Entwerfen Sie die aufgelösten Funktionspläne für die Zweige:
Eingang bis Ausgang M1; Eingang bis Ausgang M2;
von M1, M2 bis A1.


E
53

- b) Mit den angegebenen Operanden ist das Programm aufzustellen.
- c) Ermitteln Sie nach Eingabe des Programms in die SPS experimentell, bei welchen Eingangssignalkombinationen der Zustand $A1=1$ entsteht. Diese Signalzustände sind tabellarisch zu erfassen.

Bild E 53.1
Schaltung eines 2-Bit-Komparators, Aufgabe E 52.2.

Tabelle E 53.1: Erfassung der Signalkombinationen für $A1=1$.

E4	E3	E2	E1	A1
				1
				1
				1
				1

Die Lösung finden Sie auf Seite F 35.

Aufgabe E 53.1

Das Bild E 53.2 enthält den Auszug aus einem FUP. Beantworten Sie zunächst folgende Fragen:

- a) Welche Bedingungen müssen erfüllt sein, damit am Ausgang der Zeitfunktion Signalzustand 1 entsteht?
- b) Der Ausgang A1 hat den Wert 1. Welchen Signalzustand müssen die Eingangsvariablen E1 bis E4 haben?

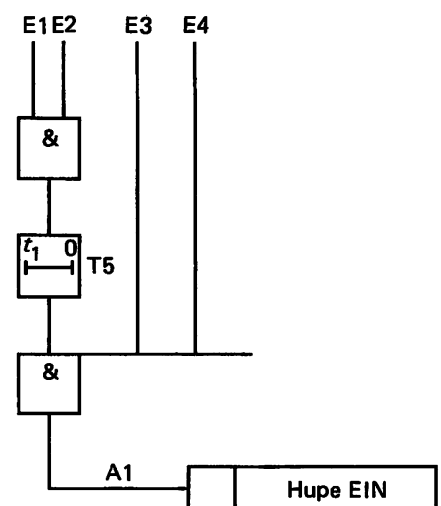


Bild E 53.2
Funktionsplan zur Aufgabe E 53.1.

- c) Ausgang A1 befindet sich im Zustand 1. Was passiert, wenn die Eingangsvariable E1 vom Zustand 1 in den Zustand 0 geht?

Ob Ihre Antworten richtig sind, können Sie selbst überprüfen.

- d) Stellen Sie für den FUP das Programm auf. Timer T5 ist mit den Verzögerungen $t_1=3\text{ s}$ und $t_2=0\text{ s}$ als Zeitfunktion zu programmieren. Nach Eingabe des Programms in die SPS können Sie Ihre theoretischen Überlegungen experimentell überprüfen.

Die Lösung finden Sie auf den Seiten F35 und F36.

Zum Abschluß der Programmierung reiner Schaltnetze wollen wir noch ein umfangreiches Beispiel aus der Steuerungstechnik bearbeiten. Das Programm für eine Verknüpfungssteuerung ist zu entwickeln.

E

54

Beispiel E 54.1

Steuerung eines chemischen Reaktors

Das Bild E 54.1 zeigt das Technologieschema. Mit binären Signalgebern werden Druck und Temperatur im Behälter erfaßt. In Abhängigkeit von den Zuständen der Gebersignale erfolgt die Betätigung der Stellglieder. Vom Entwickler der zu steuernden Anlage ist die Funktionstabelle bereits vorgegeben, nach der das Steuerungsprogramm zu entwerfen ist (Tabelle E 55.1).

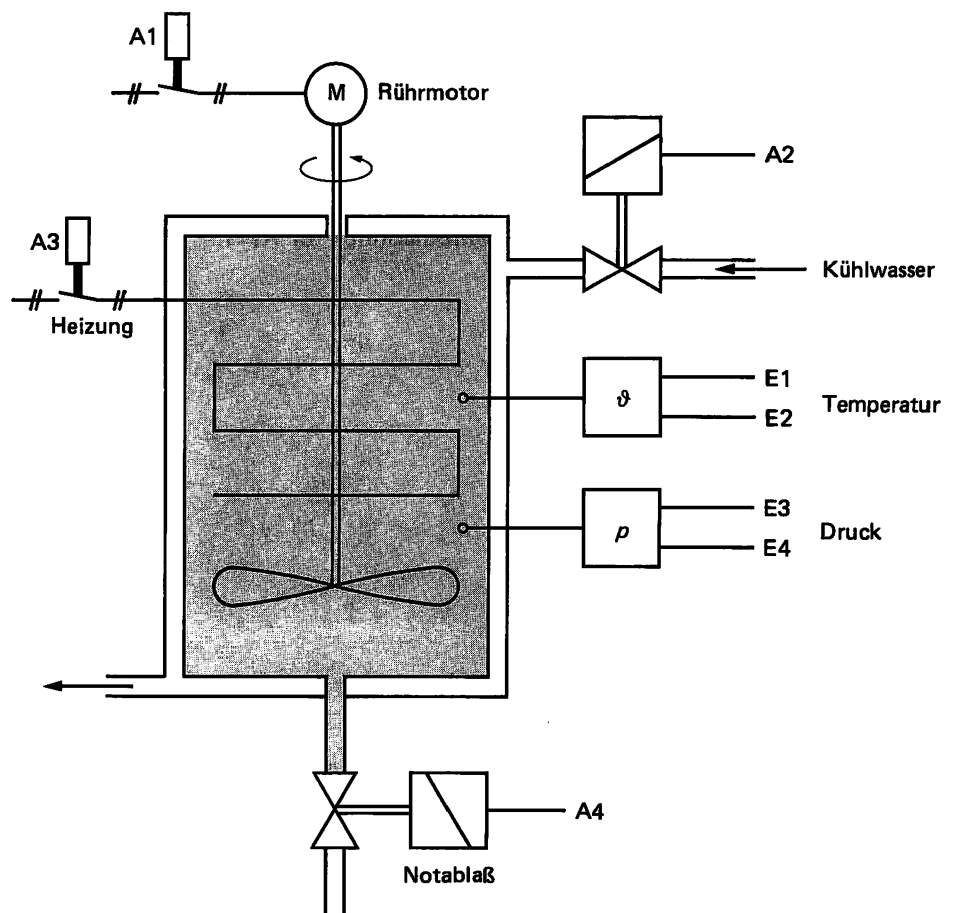


Bild E 54.1
Technologieschema: Steuerung eines chemischen Reaktors, Beispiel E 54.1.

Tabelle E55.1: Funktionstabelle der Steuerung

E4	E3	E2	E1	A1	A2	A3	A4	Bemerkung
0	0	0	0	1	0	1	0	Inbetriebnahme
0	0	1	0	0	0	1	0	
0	0	1	1	1	0	0	0	
1	0	0	0	0	0	1	0	
1	0	1	0	0	0	0	0	Sollwertzustand
1	0	1	1	1	1	0	0	
1	1	0	0	1	0	0	0	
1	1	1	0	1	1	0	0	
1	1	1	1	1	1	0	1	Gefahrenzustand
1	1	1	1	1	1	0	1	
Druck		Temp.						

Für die Signale gelten folgende Festlegungen:

Eingabesignale:

Temperaturgeber

E2	E1	Temperatur
0	0	zu niedrig
1	0	Sollwert
1	1	zu hoch

Druckgeber

E4	E3	Druck
0	0	zu niedrig
1	0	Sollwert
1	1	zu hoch

Ausgabe- bzw. Stellsignale:

Rührmotor

A1=0 bedeutet Motor AUS
A1=1 bedeutet Motor EIN

Kühlwasserventil

A2=0 bedeutet Ventil ZU
A2=1 bedeutet Ventil AUF

Heizung

A3=0 bedeutet Heizung AUS
A3=1 bedeutet Heizung EIN

Notablaß

A4=0 bedeutet Ventil ZU
A4=1 bedeutet Ventil AUF

- Für die Signale zur Betätigung der Stellglieder sind nach Tabelle E55.1 die Schaltfunktionen aufzustellen. Diese sind mit Hilfe der Schaltalgebra zu vereinfachen.
- Entwerfen Sie mit den vereinfachten Schaltfunktionen den FUP und den KOP.
- Nach dem FUP ist die Steuerung zu programmieren.
- Überprüfen Sie experimentell, ob das Programm für die vorgegebenen Eingangssignal-Kombinationen (Tabelle E 55.1) die geforderten Ausgangssignale liefert.

Lösung:

a) Schaltfunktionen:

$$A1 = \overline{E1} \overline{E2} \overline{E3} \overline{E4} \vee \overline{E1} \overline{E2} \overline{E3} E4 \vee \overline{E1} \overline{E2} \overline{E3} E4 \vee \overline{E1} \overline{E2} E3 \overline{E4} \vee \overline{E1} \overline{E2} E3 E4 \vee \overline{E1} E2 \overline{E3} \overline{E4} \vee \overline{E1} E2 \overline{E3} E4$$

$$A2 = \overline{E1} \overline{E2} \overline{E3} E4 \vee \overline{E1} \overline{E2} E3 \overline{E4} \vee \overline{E1} \overline{E2} E3 E4$$

$$A3 = \overline{E1} \overline{E2} \overline{E3} \overline{E4} \vee \overline{E1} \overline{E2} \overline{E3} E4 \vee \overline{E1} \overline{E2} E3 \overline{E4}$$

$$A4 = E1 E2 E3 E4$$

Die gleichfarbig markierten Terme unterscheiden sich nur in einer Variablen. Nach dem Ausklammern gleichartiger Ausdrücke und der Anwendung der Gesetzmäßigkeit $(E \vee \overline{E})=1$ entstehen folgende vereinfachte Schaltfunktionen:

$$A1 = E1 E2 \overline{E3} \vee \overline{E1} E3 E4 \vee E2 E3 E4 \vee \overline{E1} \overline{E2} \overline{E3} \overline{E4}$$

$$A2 = E1 E2 E4 \vee E2 E3 E4$$

$$A3 = \overline{E1} \overline{E2} \overline{E3} \vee \overline{E1} \overline{E3} \overline{E4}$$

$$A4 = E1 E2 E3 E4$$

Die Anwendung dieser einfachen algebraischen Regeln reduziert den Umfang der Schaltfunktionen doch schon recht erheblich.

b) FUP und KOP sind in Bild E 57.1 wiedergegeben. Setzen Sie die Schaltfunktionen unmittelbar in die symbolische Darstellung um. Zur Vereinfachung sind im FUP die Eingänge mit den Operanden bezeichnet.

c) Nach dem FUP ergeben sich für das Programm folgende Anweisungen:

Programm:

```
!E1&E2&NE3
/NE1&E3&E4
/E2&E3&E4
/NE1&NE2&NE3&NE4=A1    Rührmotor
!E1&E2&E4
/E2&E3&E4=A2            Kühlwasserventil
!NE1&NE2&NE3
/NE1&NE3&NE4=A3          Heizung
!E1&E2&E3&E4=A4          Notablaß
!PE
```

Zur besseren Übersicht ist das Programm in Form einer Liste aufgestellt. Sie können es auch als Liste in die SPS eintasten. Nach dem Einspeichern erscheint die Liste auf dem Bildschirm des Monitors bei einer erneuten Ausgabe wieder zeilenweise geschrieben.

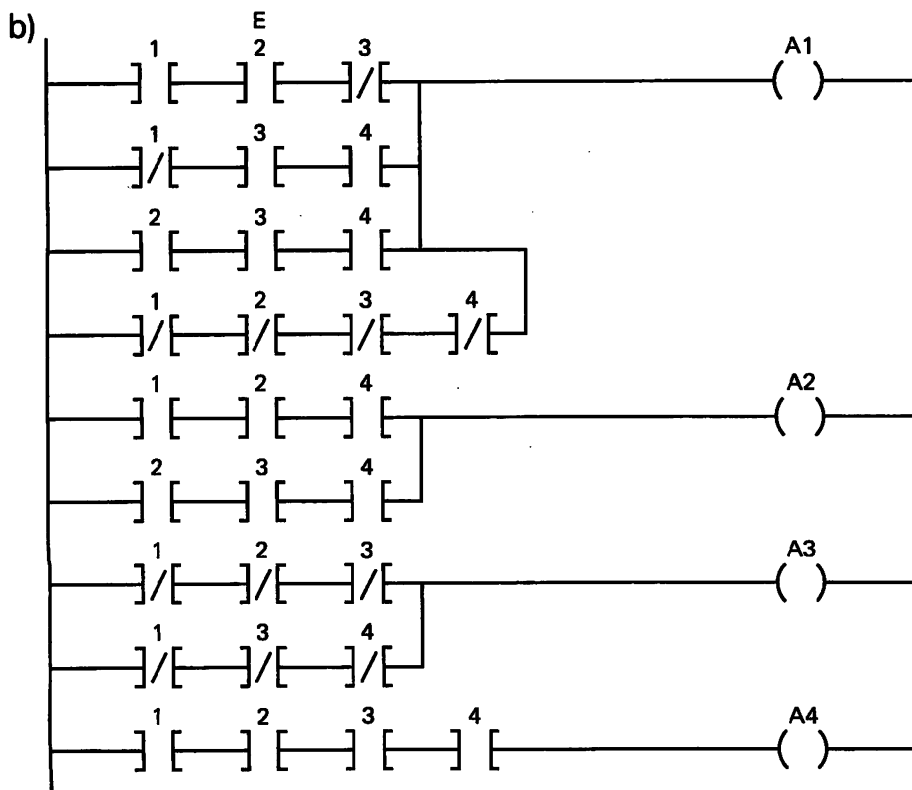
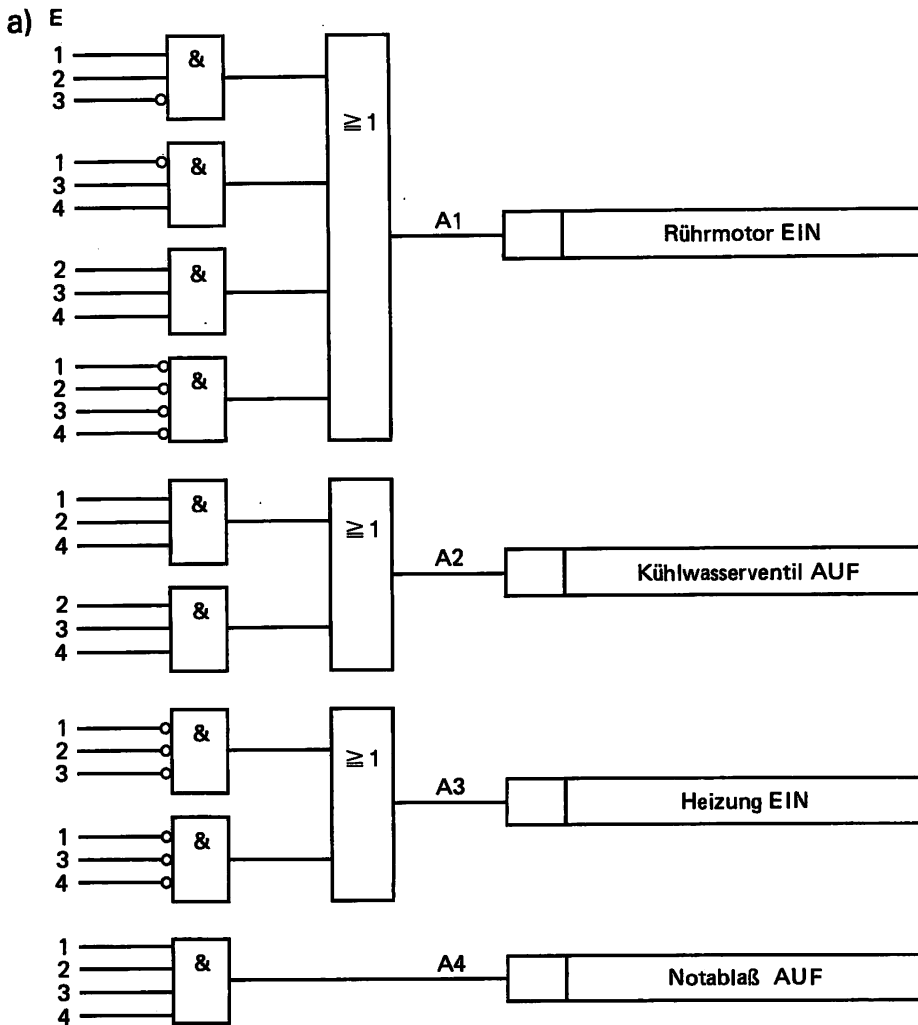


Bild E57.1
Chemischer Reaktor, Beispiel
E54.1.
a) Funktionsplan,
b) Kontaktplan.

Besonders bei der Eingabe langer Programme kann man leicht eine falsche Taste drücken. Sie wissen doch noch, wie man fehlerhafte Eingaben korrigiert?

Aufgabe E58.1

Zusatzaufgabe zu Beispiel E54.1

- 1) Bei der Ausgabe „Notablaßventil AUF“ hat Ausgang A4 Zustand 1. Zusätzlich soll als Warnung der Ausgang A5 noch Blinklicht mit der Frequenz $f_1=1\text{ Hz}$ ausgeben.
- 2) Bei 4 Eingangsvariablen E1 bis E4 gibt es 16 verschiedene Signalkombinationen. Laut Funktionstabelle E55.1 werden nur 9 Kombinationen ausgenutzt. Tritt eine nicht in der Tabelle enthaltene Kombination auf, entsteht eine Störung in der Meßwerterfassung. Diese Störung soll am Ausgang A0 ein Blinklicht mit der Frequenz $f_2=1/2\text{ Hz}$ in Betrieb setzen.
 - a) Die Schaltfunktion für den Störfall ist aufzustellen und mit der Schaltalgebra zu vereinfachen. Weisen Sie das Ergebnis dem Merker M0 zu.
 - b) Der Funktionsplan in Bild E57.1a) des Beispiels E54.1 ist gemäß der Zusatzaufgabe zu ergänzen.
 - c) Programmieren Sie die Taktgeneratoren. Für das Blinklicht „Störung“ sind die Timer T1 und T2 und Merker M1 als Speicherfunktion, für das Blinklicht „Notablaß“ sind die Timer T5 und T6 und Merker M5 als Speicherfunktion zu verwenden.
 - d) Erweitern Sie den KOP von Beispiel E54.1 (Bild E57.1b)) entsprechend der Zusatzaufgabe. Die Ergänzungen sind am Ende des Schaltnetzes in der Reihenfolge „Blinklicht Notablaß“ und „Blinklicht Störung“ anzufügen.
 - e) Stellen Sie nun das Gesamtprogramm auf. Die Programmerweiterung ist ebenfalls an das bereits vorhandene Programm des Schaltnetzes von Beispiel E54.1 anzuhängen.
 - f) Zum Abschluß sollen Sie das Gesamtprogramm testen. Es ist zu überprüfen, ob die 16 möglichen Eingabekombinationen die richtigen Antworten erzeugen.

Die Lösung dieser Aufgabe finden Sie auf den Seiten F36, F37, F38 und F39.

Unseren Glückwunsch, wenn Sie den Lehrstoff bis hierher erfolgreich bearbeitet und die Lösungen zu den Aufgaben gefunden haben. Sie sind damit in der Lage, doch schon recht anspruchsvolle Schaltnetze zu programmieren. Es kommt dabei nicht auf die optimale Lösung an. Eine gewisse Routine werden Sie sich bei der Bearbeitung der noch folgenden Aufgaben sowie in der späteren Praxis automatisch aneignen.

Zur Erweiterung der Programmierkenntnisse enthalten die folgenden Beispiele und Aufgaben zusätzlich noch Speicherfunktionen.

Programmierung einfacher Schaltwerke

Beispiel E 59.1

Programm zur Verriegelung von drei Speicherfunktionen, Auswahl-schaltung 1 aus 3

Unter Verriegelung verstehen wir in diesem Fall, daß sich die Speicherfunktionen nur in bestimmte, vorgegebene Signalzustände steuern lassen. Im Beispiel ist folgende Aufgabe zu lösen:

Drei RS-Speicherfunktionen arbeiten als Signalspeicher. Die Verriegelung soll bewirken, daß durch Steuersignale immer nur eine der Speicherfunktionen gesetzt werden kann. Befindet sich z. B. Speicherfunktion S1 im Setzzustand (Signalausgang führt Zustand 1), dann müssen die Setzschnale für die Speicherfunktionen S2 und S3 wirkungslos bleiben (Signalausgänge führen Zustand 0). Erst nach dem Rücksetzen von S1 läßt sich eine andere Speicherfunktion setzen. Weiterhin wird festgelegt, daß die Setzschnale für die Speicherfunktionen nacheinander einzugeben sind.

Das Rücksetzen erfolgt mit einem Richtimpuls. Dieser bringt alle Speicherfunktionen in die Grundstellung (alle Signalausgänge führen Zustand 0). Der Richtimpuls ist ein 1-Signal kurzer Zeitdauer. Wir erzeugen diesen Impuls durch eine schnelle EIN/AUS-Betätigung des entsprechenden Eingabeschalters.

- Entwerfen und erläutern Sie den Funktionsplan.
- Stellen Sie das Programm nach dem aufgelösten FUP auf.
- Es ist experimentell zu untersuchen, ob das aufgestellte Programm den Forderungen der Aufgabenstellung genügt.

Lösung:

- Der geschlossene FUP in Bild E 59.1 zeigt einen Lösungsvorschlag.

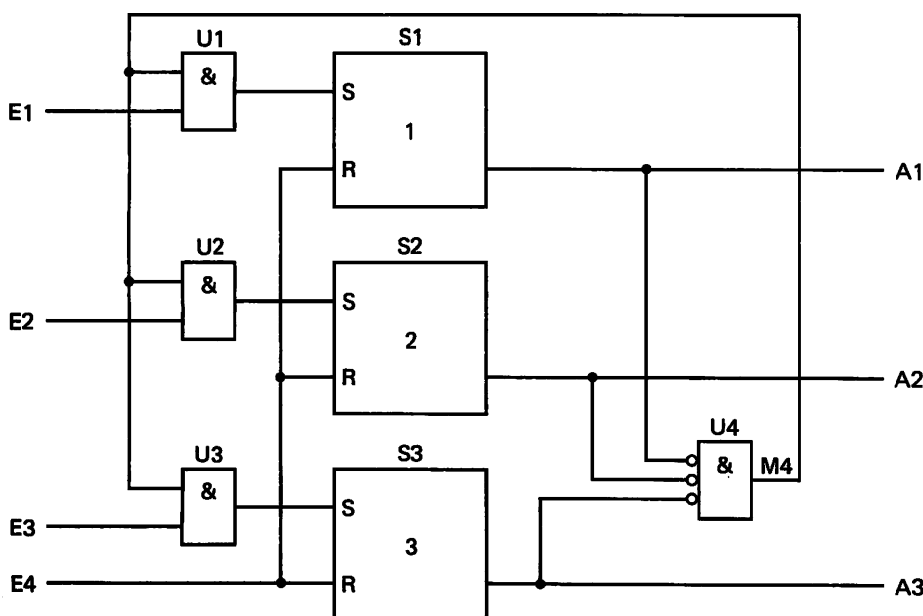


Bild E 59.1
Funktionsplan der Auswahl-schaltung 1 aus 3, Beispiel E 59.1.

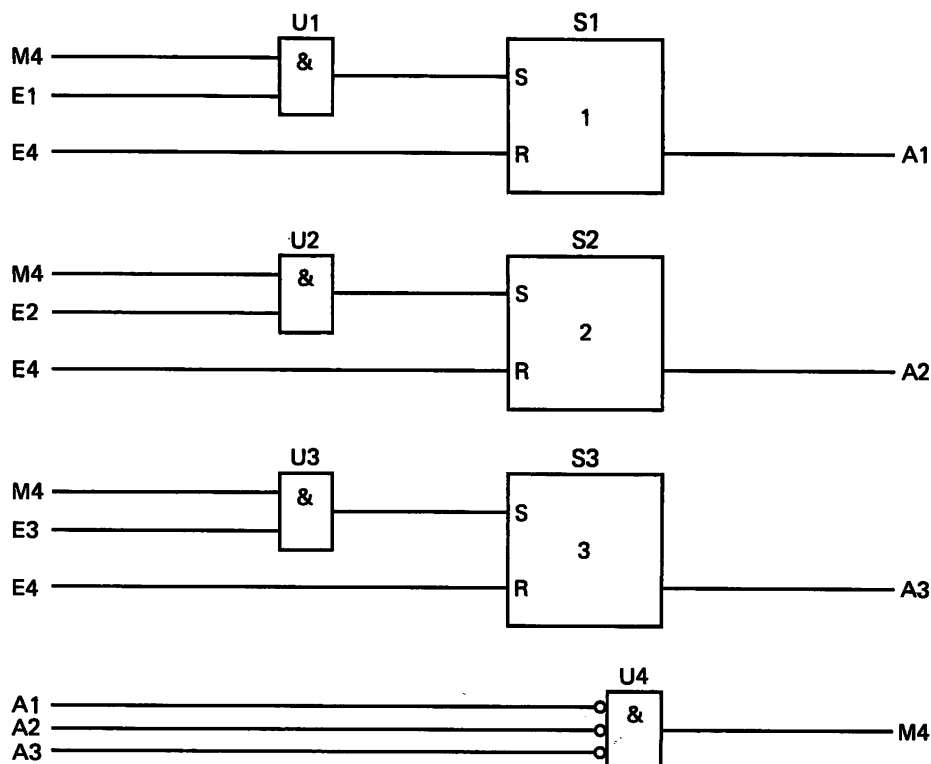


Bild E 60.1
Aufgelöster Funktionsplan der
Auswahlschaltung 1 aus 3, Beispiel
E 59.1.

Wirkungsweise:

Wir gehen davon aus, daß der Richtimpuls $E4=1$ alle Speicherfunktionen in die Grundstellung gebracht hat (A1 bis A3 haben Zustand 0). Der Ausgang der UND-Verknüpfung U4 hat den Wert 1, er bereitet die UND-Verknüpfungen U1 bis U3 zur Signalweitergabe vor. Nach dem Setzen einer Speicherfunktion ist die UND-Verknüpfung U4 nicht mehr erfüllt. Zu den nicht gesetzten Speicherfunktionen kann keine Setzanweisung gelangen. Erst nach dem Rücksetzen der gesetzten Speicherfunktion über E4 läßt sich eine andere Signalkombination eingeben.

- b) Als Vorbereitung zur Programmierung wird aus dem FUP von Bild E 59.1 der offene FUP (Bild E 60.1) entwickelt. Diesen können wir wieder direkt programmieren.

Programm:

```

!M4&E1=SA1
!M4&E2=SA2
!M4&E3=SA3
!E4=RA1=RA2=RA3
!NA1&NA2&NA3=M4
!PE
  
```

Für alle Speicherfunktionen hat die Rücksetzbedingung ($E4=1$) Gültigkeit.

Aufgabe E 61.1**Programm zur Verriegelung von drei Speicherfunktionen, Auswahlhaltung 2 aus 3**

Das Programm des letzten Beispiels ist so abzuändern, daß von den drei Speicherfunktionen maximal nur zwei in den Setzzustand gelangen können. Die Setzsignale für die Speicherfunktionen werden wieder nacheinander über Schalter eingegeben.

Das Rücksetzen der Speicherfunktionen erfolgt über eine gemeinsame Anweisung. Erst nach dem Rücksetzen aller Speicherfunktionen soll eine neue Signaleingabe möglich sein.

Für die Signalein- und ausgabe sind die Operanden des letzten Beispiels zu verwenden. Weitere zur Signalverarbeitung erforderliche Operanden wählen Sie bitte selbst.

- a) Entwerfen Sie für diese Aufgabe den FUP.
- b) Nach dem FUP ist das Programm aufzustellen.
- c) Wandeln Sie den FUP in einen KOP um.
- d) Überprüfen Sie bitte das Programm experimentell und vergleichen Sie den KOP mit der Bildschirmausgabe (Menü 3).

Einen Lösungsvorschlag finden Sie auf den Seiten F40 und F41.

In einem einfachen Beispiel sollen nun alle bisher besprochenen Funktionen unserer SPS zur Anwendung kommen.

Beispiel E 61.1**Automatische Steuerung des Füllvorgangs eines Flüssigkeitsbehälters mit einem Magnetventil**

Das Bild E 62.1 zeigt das Technologieschema. Mit Schalter S1 wird der Betrieb „Automatik“ eingestellt. Den Schaltzustand von S1 zeigt Meldeleuchte H1 an. Das Füllen erfolgt über das Magnetventil B1. Sinkt der Flüssigkeitspegel unter „UP“ ab, spricht Schwimmerschalter B2 an und setzt eine Speicherfunktion, die den Füllvorgang einleitet. Erreicht der Flüssigkeitsstand Pegel „OP“, spricht Schwimmerschalter B3 an. Durch Rücksetzen der Speicherfunktion wird der Füllvorgang beendet.

Wellen und Schwallwasser dürfen kein vorzeitiges Abschalten des Füllvorgangs bewirken. Der Füllvorgang soll deshalb erst beendet werden, wenn B3 nach einer Frist ($t_1=3\text{ s}$) noch den Zustand 1 hat.

Der Befehl „Füllen“ soll nur ausgeführt werden, wenn zwei Zusatzbedingungen, simuliert durch die Schalter S2 und S3, erfüllt sind. Bereits eine nicht erfüllte Bedingung muß den Füllvorgang solange unterbrechen, bis die Bedingung wieder der Vorgabe entspricht.

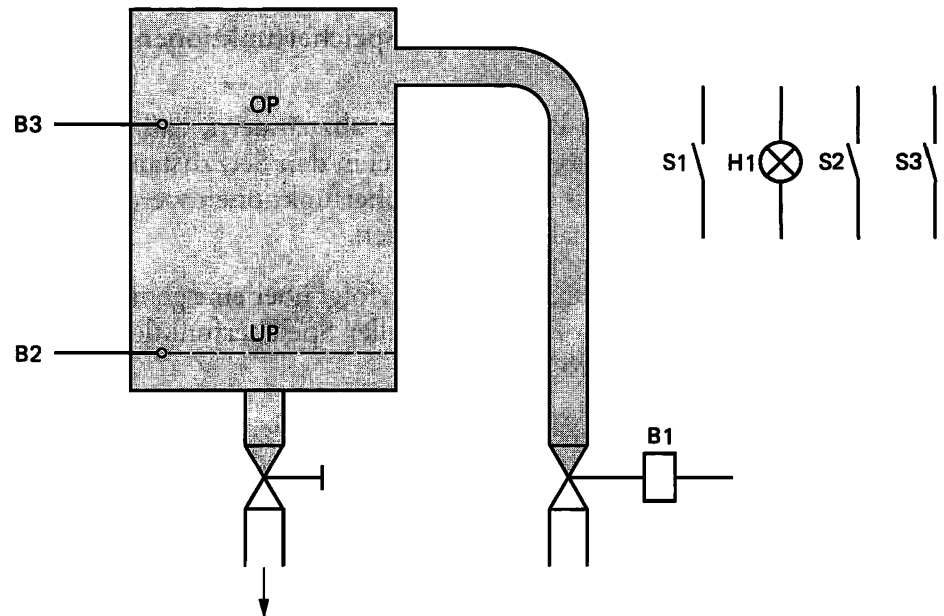


Bild E 62.1
Steuerung der Behälterfüllung mit
Magnetventil, Technologieschema
zum Beispiel E 61.1.

Die Lösung wird in folgenden Schritten demonstriert:

- Für die Betriebsmittel ist die Zuordnungstabelle aufzustellen.
- Entwerfen Sie einen Funktionsplan, der die Bedingungen der Aufgabenstellung erfüllt.
- Stellen Sie nach dem FUP das Programm auf.
- Beim Programmtest sind die Eingangssignale in der Folge einzugeben, wie sie auch im praktischen Betrieb innerhalb einer gesteuerten Anlage auftreten. Wir können hierfür wieder eine Funktionstabelle (E 62.1) aufstellen. Wenn Sie die Signalkombinationen E1 bis E5 in Ihre SPS-Anlage eingeben, müssen die eingetragenen Zustände der Ausgangssignale entstehen.

Tabelle E 62.1: Funktionstabelle zum Beispiel E 61.1

E5	E4	E3	E2	E1	A1	A2	Bemerkung
0	0	1	1	1	1	1	Behälter leer, „Füllen“ EIN
0	1	1	1	1	1	1	Pegel zwischen leer und voll
0	1	1	0	1	1	0	Bedingung E2 nicht erfüllt
0	1	1	1	1	1	1	Bedingung E2 erfüllt
1	1	1	1	1	1	0	Behälter voll, „Füllen“ AUS
0	1	1	1	1	1	0	Pegel zwischen voll und leer
0	0	1	1	0	0	0	Behälter leer, Automatik AUS

Lösung:

a) Zuordnungsliste

Betriebsmittel-Kennzeichen	Funktion	Operand
Eingänge:		
S1	Schalter für Automatikbetrieb	E1
S2	Schalter für Zusatzbedingung 1	E2
S3	Schalter für Zusatzbedingung 2	E3
B2	Schwimmerschalter für „UP“	E4
B3	Schwimmerschalter für „OP“	E5
Ausgänge:		
H1	Meldeleuchte „Automatik“	A1
B1	Magnetventil	A2
Zeiten:	Abschaltverzögerung	T1

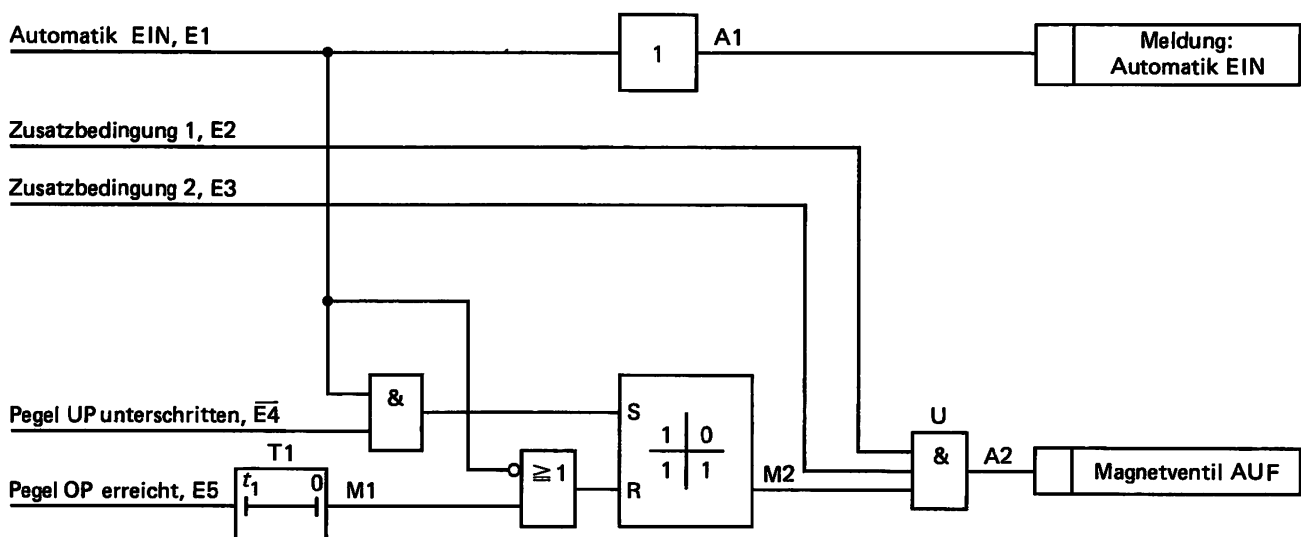
b) Funktionsplan

Zunächst vereinbaren wir die Signalzustände der Betriebsmittel. Es bedeuten:

Eingänge:

E1=1	Automatikbetrieb EIN
E1=0	Automatikbetrieb AUS
E2,E3=1	Zusatzbedingung erfüllt
E2,E3=0	Zusatzbedingung nicht erfüllt
E4,E5=1	Schwimmerschalter in Flüssigkeit
E4,E5=0	Schwimmerschalter nicht in Flüssigkeit

Bild E 63.1
Funktionsplan zum Beispiel E 61.1.



Ausgänge:

A1=1	Meldeleuchte „Automatik EIN“
A1=0	Meldeleuchte „Automatik AUS“
A2=1	Magnetventil AUF
A2=0	Magnetventil ZU

Einen FUP, der die Bedingungen der Aufgabenstellung erfüllt, zeigt das Bild E 63.1. Auf den Ablauf der Signalverarbeitung kommen wir im folgenden Punkt zu sprechen.

c) Programm:

Das Programm läßt sich unmittelbar nach dem FUP in Bild E 63.1 aufstellen. Die Steuerung hat folgende Funktion:

Das Setzen der Speicherfunktion und die damit verbundene Ausgabe des Befehls „Füllen“ erfolgt, wenn der Schwimmerschalter B2 den Signalzustand $E4=0$ hat. Bei $E4=0$ ist der Pegel „UP“ unterschritten. Zum Setzen der Speicherfunktion benötigen wir jedoch ein 1-Signal. Im FUP finden Sie deshalb das invertierte Signal E4 eingetragen, denn $\overline{E4}=\overline{0}=1$. Das Setzsignal kommt nur beim Betriebszustand „Automatik EIN“ ($E1=1$) zur Auswirkung.

Im FUP hat die Speicherfunktion Rücksetzdominanz. Das Rücksetzsignal kommt bei „OP“ erreicht bzw. überschritten ($E5=1$) oder „Automatik AUS“ ($E1=0 \rightarrow \overline{E1}=\overline{0}=1$) zur Wirkung.

Falls eine der beiden Zusatzbedingungen nicht erfüllt sein sollte ($E2=0$ oder $E3=0$), wird die Weitergabe der Füllanweisung unterbrochen. Die Speicherfunktion bleibt jedoch gesetzt. Bei $E2=E3 \rightarrow 1$ wird der Füllvorgang automatisch fortgesetzt.

Nach diesen Erläuterungen wird Ihnen der FUP sicherlich verständlich sein.

Programm:

Der Timer T1 ist mit einer Einschaltverzögerung $t_1=3\text{ s}$ zu programmieren ($t_2=0\text{ s}$).

!E1=A1	Meldung: Automatik EIN
!E1&NE4=SM2	Anweisung „Füllen“ speichern
!NE1/M1=RM2	Anweisung „Füllen“ löschen
!E2&E3&M2=A2	Befehl „Füllen“ ausgeben
!E5=T1	Timer starten
!T1=M1	„Füllen“ verzögert beenden
!PE	

Als Befehlsspeicher wurde ein Merker verwendet, weil das gespeicherte Signal an keinem Ausgang benötigt wird.

d) Nach Eingabe des Programms in den Arbeitsspeicher erfolgt der Programmtest. Die experimentell ermittelten Zustände der Ausgangssignale stimmen mit den theoretisch ermittelten Werten in Tabelle E 62.1 überein.

Zum Abschluß des Abschnitts „Programmierung einfacher Schaltwerke“ stellen wir Ihnen noch eine Aufgabe, die Sie selbst bearbeiten sollen. Wir bleiben bei unserem Thema „Füllstandssteuerung“. Das Füllen des Behälters erfolgt mit einer elektromotorisch angetriebenen Pumpe.

Aufgabe E 65.1

Automatische Steuerung des Füllvorgangs eines Flüssigkeitsbehälters mit einer Pumpe

Es ist ein Programm für die automatische Füllstandssteuerung eines Flüssigkeitsbehälters aufzustellen und zu testen.

Das Technologieschema der Steuerung zeigt das Bild E 65.1. Der Behälter wird mit einer elektrisch angetriebenen Wasserpumpe gefüllt. Bei Unterschreiten des unteren Pegels „UP“ soll die Füllung automatisch beginnen, bei Erreichen des oberen Pegels „OP“ soll der Pumpenmotor automatisch abschalten. Die Signalgeber B1 bis B3 melden den Stand des Flüssigkeitspegels. Solange sich die Signalgeber in der Flüssigkeit befinden, geben sie ein Signal mit dem Wert 1, im anderen Fall mit dem Wert 0 ab. Die Bedeutung der Gebersignale erkennen Sie in Tabelle E 66.1.

Da das Wasser aus einem Brunnen gepumpt wird, ist ein Trockenlaufschutz F vorzusehen. Er schaltet den Pumpenmotor bei Unterschreiten eines bestimmten Wasserpegels im Brunnen sofort ab. Mit Schalter S1 wird die Steuerung in Betrieb genommen. Den Betriebszustand zeigt die Meldeleuchte H1 an. Die Bedeutung der weiteren Meldelampen enthält Tabelle E 66.1.

Bevor Sie die Aufgabe bearbeiten können, ist noch eine Zuordnungsliste der zur Steuerung erforderlichen Betriebsmittel aufzustellen.

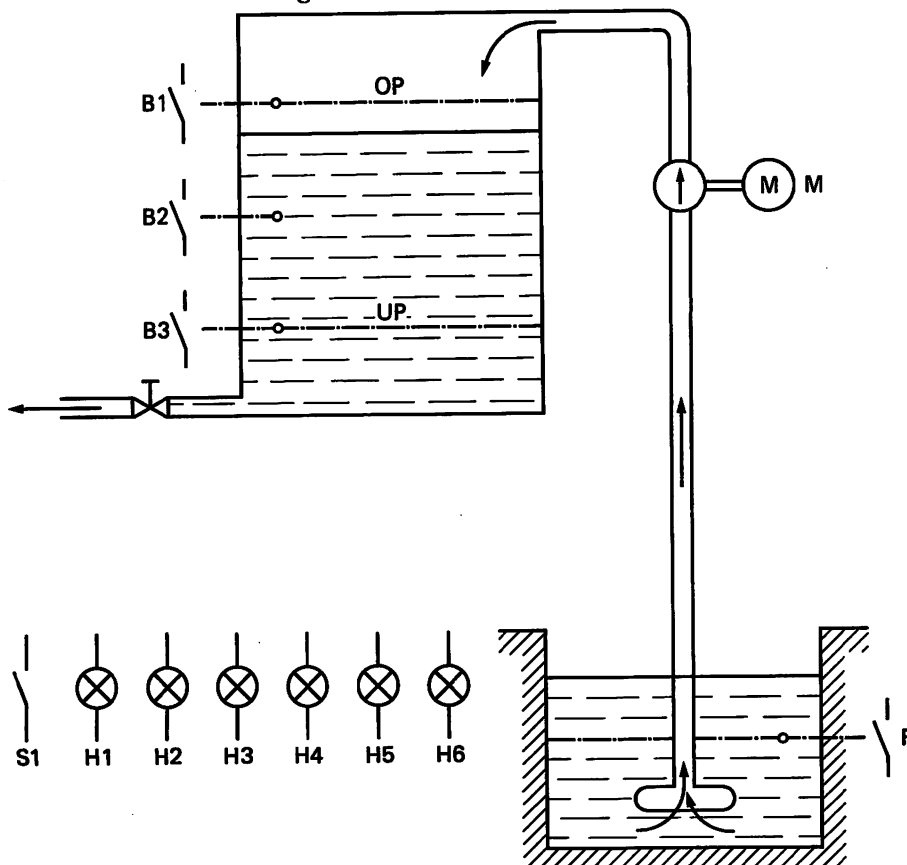


Bild E 65.1
Steuerung der Behälterfüllung,
Technologieschema zur Aufgabe
E 65.1

Tabelle E 66.1: Zuordnungsliste zur Aufgabe E 65.1

Betriebsmittel-Kennzeichen	Funktion	Operand
Eingänge:		
S1	Hauptschalter	E0
B1	Schwimmerschalter „OP“=„voll“	E1
B2	Schwimmerschalter „halbvoll“	E2
B3	Schwimmerschalter „UP“=„leer“	E3
F	Trockenlaufschutz	E4
Ausgänge:		
H1	Meldung: „Hauptschalter EIN“	A0
H2	Anzeige: Behälter voll	A1
H3	Anzeige: Pegel zwischen halbvoll und voll	A2
H4	Anzeige: Pegel zwischen halbvoll und leer	A3
H5	Anzeige: Behälter leer	A4
H6	Meldung: „Störung“	A5
M	Pumpenmotor	A6

Nun sind noch die Schaltzustände zu vereinbaren. Es gelten folgende Festlegungen:

Schalter oder Motor EIN entspricht Zustand 1
 Schalter oder Motor AUS entspricht Zustand 0

Schwimmerschalter in Flüssigkeit entspricht Zustand 1
 Schwimmerschalter nicht in Flüssigkeit entspricht Zustand 0

Anzeige oder Meldung vorhanden entspricht Zustand 1
 Anzeige oder Meldung nicht vorhanden entspricht Zustand 0

- a) Am Behälter befinden sich drei Schwimmerschalter als Signalgeber. Somit sind acht verschiedene Meldungen möglich. Von diesen sind einige Signalkombinationen bei störungsfreiem Betrieb nicht möglich.

Die auf der nächsten Seite dargestellte Funktionstabelle E 67.1 enthält alle möglichen Signalgeberkombinationen der Schwimmerschalter. Tragen Sie für diese Möglichkeiten bitte die Signalfunktion der Ausgangssignale A1 bis A5 ein. In die Spalte „Bemerkung“ können Sie einen kurzen Hinweis auf die Bedeutung der Signalkombination eintragen, wie z.B. „Behälter leer“, „Störung“ usw. „Störung“ gilt für solche Signalkombinationen, die bei störungsfreiem Betrieb nicht auftreten können.

Tabelle E 67.1: Funktionstabelle zur Aufgabe E 65.1

E3	E2	E1	A1	A2	A3	A4	A5	Bemerkung
0	0	0						
0	0	1						
0	1	0						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

Bevor Sie den Lehrstoff weiter durcharbeiten, vergleichen Sie bitte Ihre Lösung erst mit der ausgefüllten Tabelle im Lösungsvorschlag auf Seite F41.

- b) Für die Ausgabesignale A1 bis A5 sind nach der ausgefüllten Tabelle E 67.1 die Schaltfunktionen aufzustellen und mit der Schaltalgebra zu vereinfachen.
- c) Der Entwurf des Funktionsplans ist unter Berücksichtigung folgender Bedingungen aufzustellen:

Einschaltbedingungen für den Pumpenmotor:

Hauptschalter EIN und
Trockenlaufschutz nicht angesprochen und
„Behälter leer“.

Ausschaltbedingungen für den Pumpenmotor:

Hauptschalter AUS oder
„Behälter voll“ oder
Trockenlaufschutz angesprochen oder
„Störung“.

Vergleichen Sie Ihren FUP mit dem im Lösungsvorschlag auf Seite F42, Bild F42.1.

- d) Programmieren Sie die Füllstandssteuerung. Das Programm ist unmittelbar nach dem FUP und den Schaltfunktionen aufzustellen.
- 1) Stellen Sie das Programm mit Anweisungen auf.
 - 2) Stellen Sie das Programm als Anweisungsliste auf.
- e) Entwickeln Sie nach Ihrem Programm den Kontaktplan.
- f) Nun ist das Programm zu testen. Die erforderlichen Schritte hierfür wurden schon oft besprochen. Wir werden sie deshalb ab sofort nicht mehr wiederholen, sondern nur bei Bedarf auf wichtige Punkte eingehen.

Beim Test müssen die verschiedenen Betriebszustände in der richtigen Reihenfolge (so, wie sie in der technisch realen Steuerung auftreten würden) eingegeben werden. Ausgehend vom Zustand „Behälter voll“ gilt für die Signale E1 bis E3 die Folge:

	E3	E2	E1	Bemerkung
„Entleeren“	1	1	1	Behälter voll
	1	1	0	Pegel zwischen voll und halbvoll
	1	0	0	Pegel zwischen halbvoll und leer
	0	0	0	Behälter leer
„Füllen“	1	0	0	} wie vorher
	1	1	0	
	1	1	1	

Alle anderen Signalkombinationen sind Fehlmeldungen, die als Störungen erkannt werden müssen und zum Abschalten des Pumpenmotors führen.

Testen Sie deshalb Ihr Programm mit der vorgegebenen Signalfolge und simulieren Sie anschließend Störungen. Vergessen Sie nicht, den „Hauptschalter“ einzuschalten!

Die Lösung dieser Aufgabe finden Sie auf den Seiten F41, F42 und F43.

Zusammenfassung

Sie haben jetzt eine ganze Reihe von Aufgaben gelöst, die in den verschiedensten Anwendungsbereichen von SPS-Anlagen eingesetzt werden. Bearbeitet wurden Schaltnetze und Schaltwerke, Kippschaltungen und Zeitfunktionen. Sie sehen schon jetzt, daß sich bereits recht umfangreiche Aufgaben mit einem verhältnismäßig kleinen SPS-System ganz gut lösen lassen. Bei einigen Aufgaben ging es uns nicht um eine optimale Lösung – es sollte vielmehr ein leicht verständlicher Lösungsvorschlag gezeigt werden.

Programmierung von Sonderfunktionen

Die bisherigen Programme waren vielfach schon auf die Lösung bestimmter Aufgaben ausgerichtet. Wir haben Ihnen gezeigt, wie man mit den Grundfunktionen Schaltnetze und Schaltwerke mit bestimmten Eigenschaften programmiert.

Die nun folgenden Programme dienen mehr der Ergänzung. Sie sind teilweise allein kaum anwendbar und ergeben erst in Kombination mit anderen Programmen einen Sinn. Ausführliche Anwendungen dieser Sonderfunktionsprogramme folgen später.

Binäruntersetzter

Das Programm bildet die Hardwareschaltung von Bild E 69.1 nach. Gezeichnet ist ein JK-Speicherglied, dessen vorbereitende Eingänge den Zustand 1 haben. Bei jedem Signalwechsel von 0 auf 1 (steigende Flanke) am Takteingang E wechselt der Ausgang A seinen Signalzustand. Das Ausgangssignal hat die halbe Frequenz des Eingangssignals. Eine Schaltung mit diesen Eigenschaften bezeichnet man als **Binäruntersetzter**. Sie wird als Ausgangsschaltung für Frequenzteiler und Dualzähler eingesetzt.

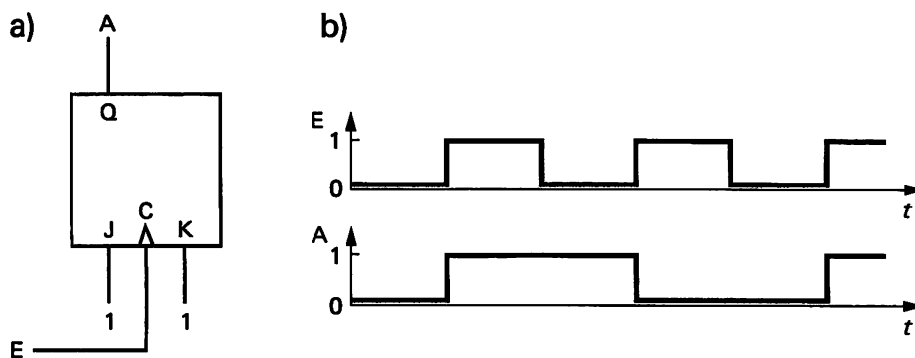


Bild E 69.1
JK-Speicherglied als Binäruntersetz-
zer.
a) Schaltzeichen,
b) Signal-Zeit-Diagramm.

Den Funktionsplan, der die Eigenschaften der Hardwareschaltung nachbildet, zeigt das Bild E 70.1. Der Merker M1 hat die Funktion eines Speichers. Es wird angenommen, daß beide Speicherfunktionen den Signalzustand 0 haben ($A1=M1=0$). E1 ist das Eingangssignal mit der Frequenz f_1 , an A1 erscheint das Ausgangssignal mit der Frequenz $f_2=f_1/2$. Signalzustand $E1=1$ setzt die RS-Speicherfunktion (U1 erfüllt), A1 geht in Zustand 1. Wechselt E1 vom Zustand 1 in Zustand 0, erfolgt das Setzen des Merkers M1 (U3 erfüllt). Erscheint am Eingang E1 wieder ein 1-Signal, wird die RS-Speicherfunktion rückgesetzt (U2 erfüllt), A1 geht in Zustand 0. Bei dem folgenden Signalwechsel $E1=0$ schaltet auch der Merker wieder in den Ausgangszustand $M1=0$. Der beschriebene Vorgang wiederholt sich periodisch. Zwei Eingangsimpulse ergeben einen Ausgangsimpuls. Sie können den beschriebenen Vorgang am **Signal-Zeit-Diagramm** verfolgen.

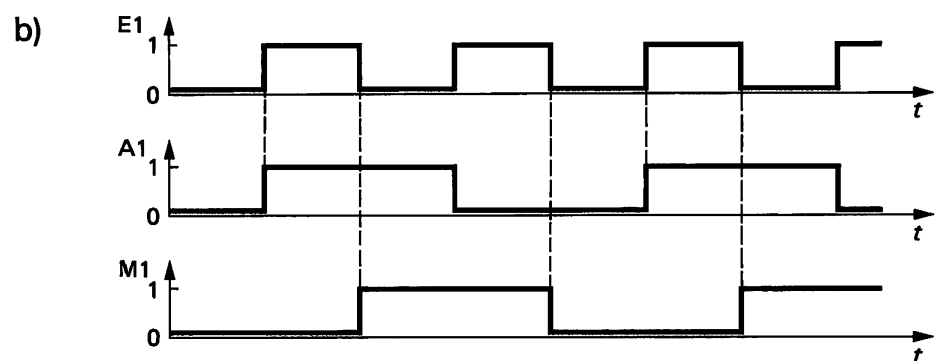
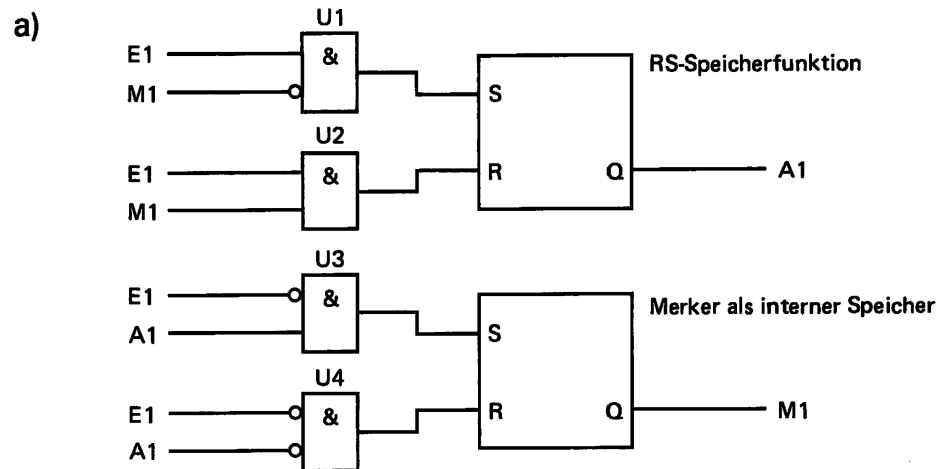


Bild E70.1
Binäruntersetzter.
a) Funktionsplan,
b) Signal-Zeit-Diagramm.

Nach dem Funktionsplan lauten die Anweisungen des Programms:

!E1=A0	Anzeige des Eingangssignals
!E1&NM1=SA1	Setzen der RS-Speicherfunktion
!E1&M1=RA1	Rücksetzen der Speicherfunktion
!NE1&A1=SM1	Setzen des Merkers als Speicher
!NE1&NA1=RM1	Rücksetzen des Merkers als Speicher
!PE	

Das Programm ergibt den in Bild E70.2 gezeichneten Kontaktplan.

Wenn Sie mit einer beliebigen Frequenz den Schalter E1 ein- und ausschalten, können Sie das Programm testen. Die LED am Ausgang A1 hat eine halb so große Blinkfrequenz wie die LED am Ausgang A0.

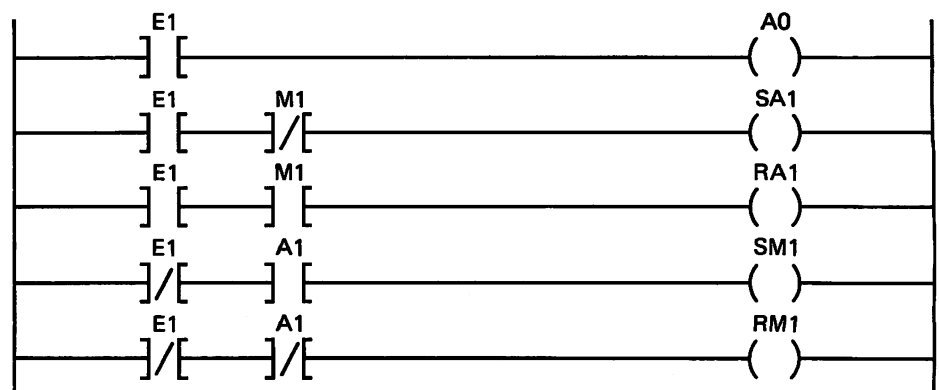


Bild E70.2
Binäruntersetzter (Kontaktplan).

Aufgabe E 71.1

Programmierung eines Taktgenerators mit anschließendem Binäruntersetzer.

- a) Benutzen Sie für den Taktgenerator das bereits besprochene Programm mit zwei Zeitfunktionen. Die Taktfrequenz soll 1Hz betragen. Ordnen Sie das Taktsignal dem Ausgang A1 zu.
- b) Die Frequenz des Taktsignals ist mit einem Binäruntersetzer im Verhältnis 2:1 zu teilen. Schalten Sie das Ausgangssignal auf Ausgang A2.
- c) Die Teilprogramme a) und b) sind zu einem Gesamtprogramm zusammenzusetzen und mit der SPS zu testen. Wählen Sie bitte die erforderlichen Operanden selbst.

Einen Lösungsvorschlag finden Sie auf den Seiten F43 und F44.

E**71****Flankenbewertung**

Verschiedentlich wird ein Steuerungsvorgang von einer Signalflanke ausgelöst. Wir unterscheiden zwischen steigender und fallender Flanke. Eine **steigende Flanke** liegt bei einem Signalwechsel von Zustand 0 nach Zustand 1 vor. Im umgekehrten Fall spricht man von einer **fallenden Flanke**.

Ein Programm zur Erkennung und Auswertung von Signalflanken wird als Flankenmerker bezeichnet. Es untersucht in jedem Bearbeitungszyklus, ob sich der Signalzustand, z.B. eines Eingangssignals gegenüber dem vorangegangenen Zyklus geändert hat. Beim Auftreten einer Flanke, also eines Signalwechsels, wird ein Merker in den Zustand 1 gesetzt. Das Signal steht nur während einer Zykluszeit zur Verfügung. Es kann abgefragt und für weitere Steuerungsaufgaben ausgewertet werden.

Dem Programm liegt folgende Überlegung zugrunde:

Der abgefragte Signalzustand, z.B. des Eingangssignals, wird gespeichert. Diese Aufgabe übernimmt der Flankenmerker. Im folgenden Zyklus erfolgt eine erneute Abfrage des Eingangs. Der Zustand des abgefragten Signals wird mit dem im Flankenmerker gespeicherten Signal verglichen. Stimmen die Signalzustände überein, so hat keine Signaländerung stattgefunden, es war keine Flanke wirksam. Bei nicht übereinstimmenden Signalzuständen war eine Signalflanke vorhanden. Nur in diesem Fall wird ein zweiter Merker, der sogenannte **Impulsmerker**, in Zustand 1 gesetzt. Dieser Merker dient als Zwischenspeicher für das Vergleichsergebnis, sein Zustand kann abgefragt werden.

Nach dem Erkennen der Flanke muß der Flankenmerker dem Eingangssignal nachgeführt werden, damit beim erneuten Abfragezyklus die gleiche Signaländerung nicht ein zweites Mal ausgewertet wird. Die Nachführung des Flankenmerkers übernimmt der Impulsmerker.

Flankenmerker zur Auswertung einer steigenden Flanke

Der Flankenmerker soll den 0-1 Sprung eines Signals, z.B. des Eingangssignals E1, auswerten. Wir betrachten den Vorgang mit Hilfe des Funktionsplans, der in Bild E 72.1. dargestellt ist. Folgender Anfangszustand wird angenommen:

Eingangssignal E1 und Ausgang M2 des Flankenmerkers haben Zustand 0. Eine steigende Flanke ist vorhanden, wenn E1 in Zustand 1 schaltet. Bei M2=0 und E1=1 wird der Impulsmerker gesetzt, er hat den Zustand M1=1. Dieser Zustand führt auch zum Setzen des Flankenmerkers M2=1 und zum Rücksetzen des Impulsmerkers. Nimmt E1 wieder Zustand 0 an, dann wird der Flankenmerker zurückgesetzt, M2=0. Der Zustand des Flankenmerkers wird somit dem Zustand des Eingangssignals E1 nachgeführt. Im folgenden Abfragezyklus ist bei E1=0 die UND-Bedingung nicht mehr erfüllt. Mit M2=1 ist die UND-Bedingung ebenfalls nicht mehr erfüllt; der Impulsmerker wird zurückgesetzt. M1 nimmt sofort wieder den Zustand 0 an. Erst bei einem erneuten Wechsel des Eingangssignals E1 von 0 nach 1 wiederholt sich der geschilderte Ablauf.

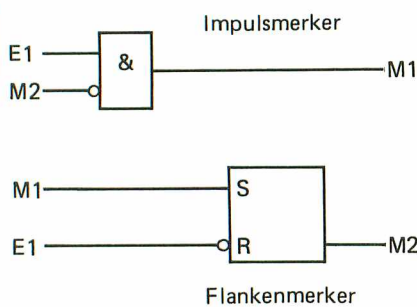


Bild E 72.1
Funktionsplan des Flankenmerkers
für die Auswertung einer anstei-
genden Flanke.

Nach dem Funktionsplan von Bild E 72.1 erhalten wir folgendes Programm:

```

!E1&NM2=M1
!M1=SM2
!NE1=RM2
!PE

```

Den Kontaktplan sehen Sie in Bild E 72.2.

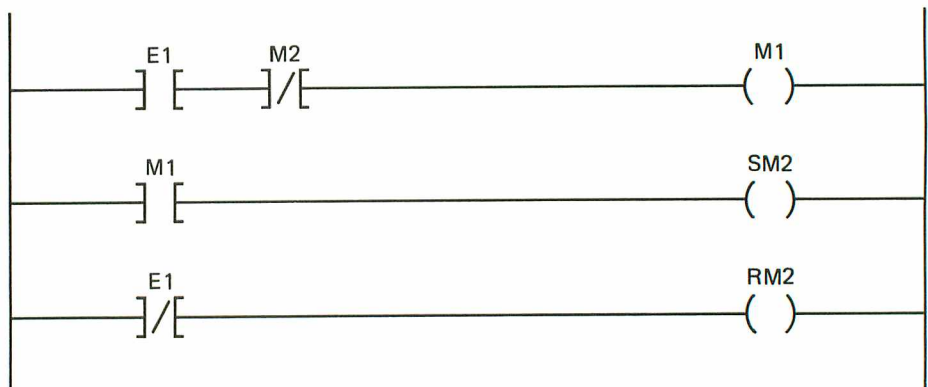


Bild E 72.2
Kontaktplan des Flankenmerkers
für die Auswertung einer anstei-
genden Flanke.

Das Ausgangssignal des Impulsmerkers wird zur weiteren Signalverarbeitung ausgewertet. Wir können das Programm jedoch nicht unmittelbar testen, da dieses Signal nur während einer Zykluszeit zur Verfügung steht. Die Zykluszeit beträgt nur wenige ms. Während dieser Zeit spricht keine LED an. Ein Test ist aber mit dem folgenden Programm möglich.

D-Speicherfunktion

In der elektronischen Schaltungstechnik gibt es ein D-Speicherglied (D-Flip-Flop = D-FF). Sein Schaltzeichen sehen Sie im Bild E 73.1. Anschluß D ist der **Informationseingang**. Der Signalzustand von E1

wird jedoch erst mit der 0-1 Flanke des Taktsignals am C-Eingang übernommen. Ein Programm, das die Funktion des D-FF nachbildet, wollen wir aufstellen. Als Signalspeicher steht uns nur die RS-Speicherfunktion zur Verfügung. Die Programmstruktur können wir aus dem Schema von Bild E73.2a ableiten.

Das dynamische Taktsignal erzeugt ein Flankenauswerter, dessen Programm Sie ja schon kennen. Er wertet die ansteigende Flanke von Taktsignal E0 aus. Bei jeder aktiven Flanke entsteht an M1 ein kurzes 1-Signal. Dieses steuert in Verbindung mit dem Signalzustand von E1 den Zustand der Speicherfunktion. E1 entspricht dem Signal am D-Eingang des Speicherglieds von Bild E73.1.

Bei $E1=1$ UND $M1=1$ wird ein Signal **gespeichert** ($A1=1$), bei $E1=0$ UND $M1=1$ wird das Signal **gelöscht** ($A1=0$). Damit ist die Wirkung des D-FF eindeutig nachgebildet. Den aus Bild E73.2a abgeleiteten Funktionsplan gibt das Bild E73.2b wieder. Mit den Operanden in Bild E73.2b lautet das Programm:

```

!E0&NM2=M1
!M1=SM2
!NE0=RM2      Flankenmerker
!M1&E1=SA1
!M1&NE1=RA1   D-FF
!PE
  
```

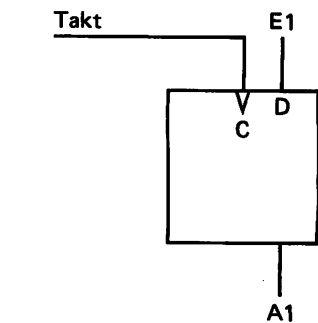
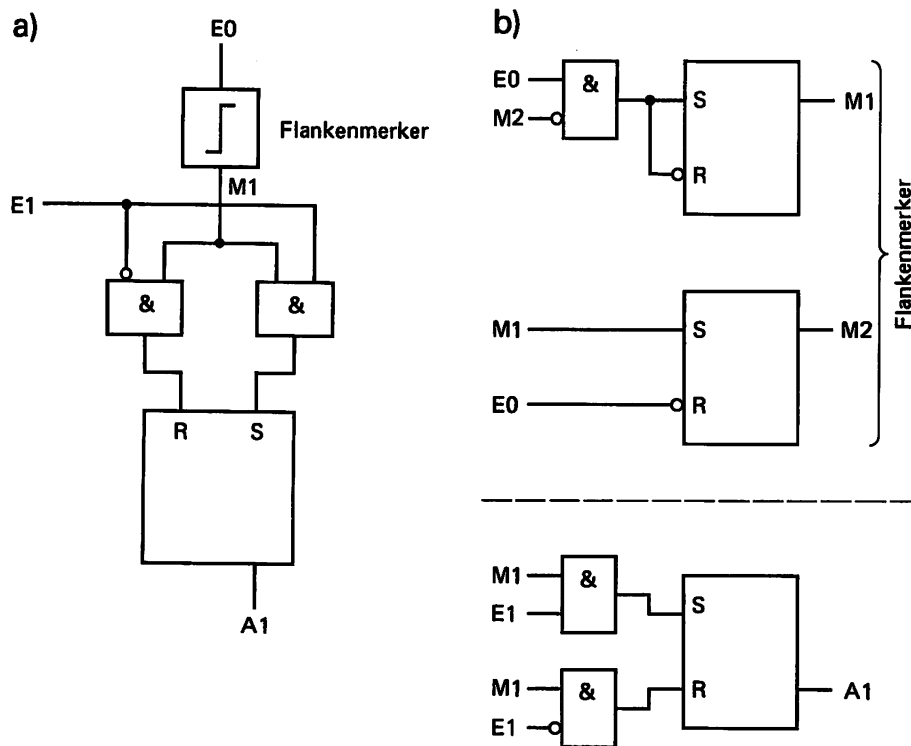


Bild E73.1
Schaltzeichen des D-Speicherglieds.

Aufgabe E73.1

D-Speicherfunktion

- a) Geben Sie bitte das aufgestellte Programm des D-FF in die SPS ein. Durch Schalterbetätigung von E0 wird die Signalfanke 0-1 nachgebildet. Stellen Sie mit Schalter E1 die Signalzustände 1 bzw. 0

ein. Überprüfen Sie, ob beim Schaltvorgang 0-1 der Variablen E0 der Ausgang A1 den Zustand der Variablen E1 übernimmt.

- b) Das Taktsignal soll von einem Taktgenerator erzeugt werden. Benutzen Sie den in Aufgabe E71.1a programmierten Taktgenerator. Folgende Änderungen sind erforderlich:

Verwenden Sie statt Merker M1 die Bezeichnung Merker M0 und schalten Sie das erzeugte Taktsignal auf den Ausgang A0. Nehmen Sie die gleiche Funktionsprüfung wie unter a) vor.

Die Lösung dieser Aufgabe finden Sie auf Seite F44.

Zusammenfassung

Ein komplettes SPS-Programm wird in der Regel aus verschiedenen Teilprogrammen entwickelt. Diese Module sind meistens allein nicht lauffähig. Erst in Verbindung mit dem eigentlichen Hauptprogramm entsteht ein lauffähiges Gesamtprogramm. Folgende Schaltungen, als SPS-Programm entwickelt, sind oft Bestandteile des Gesamtprogramms:

Frequenzteiler, Taktgeneratoren, Dualzähler, Flankenauswerter.

Programmierung von Zählfunktionen

Dualzähler

Im Signal-Zeit-Diagramm des asynchronen Dualzählers in Bild B 92.1b haben wir Ihnen gezeigt, daß jede Zählstufe als **Frequenzteiler** 2 : 1 arbeitet. Ein Programm, das die Frequenz der Eingangsimpulse im Verhältnis 2 : 1 teilt, kennen Sie bereits: es ist das Programm des Binäruntersetzers (siehe Seite E 69). Schauen Sie sich diesen Abschnitt bitte noch einmal an — das dort besprochene Programm benötigen wir jetzt wieder.

Beispiel E 75.1

Programmierung eines 4-Bit-Dualzählers

Entwickeln Sie für diesen Zähler

- den Funktionsplan,
- das Programm und
- den Kontaktplan.

Lösung:

- Für jede Zählstufe ist ein Binäruntersetzer erforderlich. Die Zählfunktion besteht somit aus einer **Kettenschaltung** von vier Binäruntersetzern. Er zählt im Dualcode von 0 bis 15. Das zu programmierende Zählschema zeigt das Bild E 75.1. Der zeitliche Verlauf der Ausgangssignale entspricht dem Signal-Zeit-Diagramm von Bild B 92.1b.

Den aufgelösten FUP zeigt das Bild E 76.1. Jede Stufe besteht aus einem Binäruntersetzer. Damit ein Vorwärtszähler entsteht, muß (wie in den FUP eingetragen) der Folgestufe (z.B. Stufe 2) das negierte Ausgangssignal der vorangehenden Stufe (z.B. Stufe 1) als Zählsignal zugeführt werden. Bei nichtnegiertem Ausgangs-

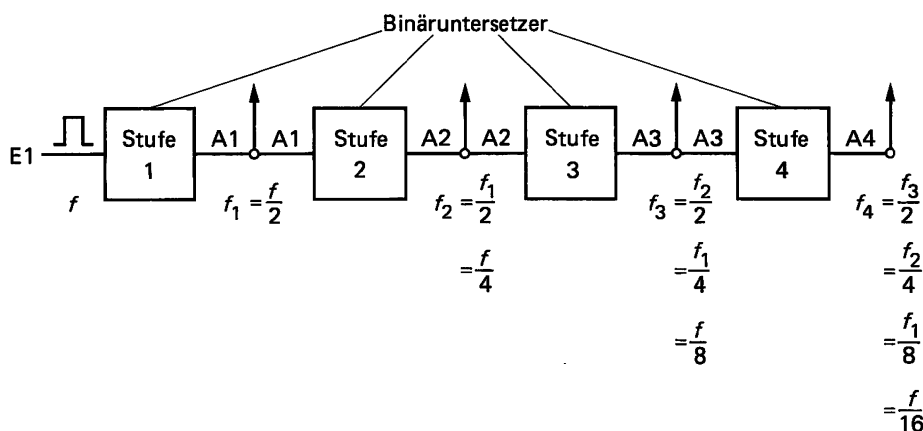


Bild E 75.1
Schema des 4-Bit-Dualzählers mit Binäruntersetzern.

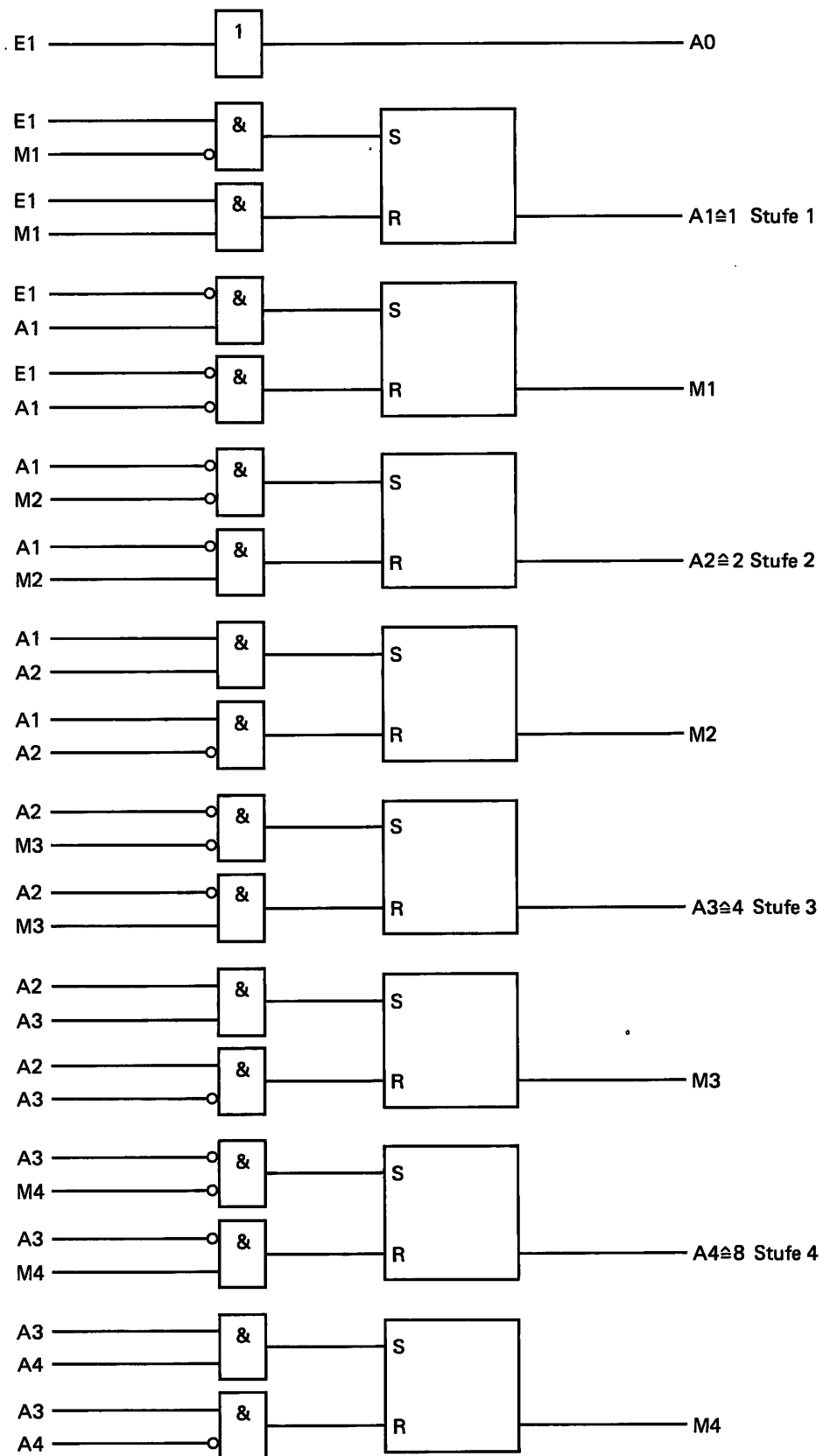


Bild E76.1
Aufgelöster FUP des 4-Bit-Dual-
zählers mit Binäruntersetzern.

signal entsteht ein Rückwärtszähler. Probieren Sie diese Wirkung bitte selbst aus.

Die Anzeige A0 kontrolliert den Zustand des Zählsignals.

- b) Nach dem FUP von Bild E 76.1 lässt sich unmittelbar das Programm der Zählfunktion aufstellen. Den Ausgangsvariablen sind folgende Wertigkeiten zugeordnet:

$$A1 \triangleq 1 \quad A2 \triangleq 2 \quad A3 \triangleq 4 \quad A4 \triangleq 8$$

Beim Einschalten kann ein von $A1, A2, A3, A4 = 0, 0, 0, 0$ abweichender Signalzustand entstehen. Betätigen Sie Schalter E1 solange (0-1-0...), bis alle Signalausgänge auf 0 gesetzt sind. Wir zeigen Ihnen später, wie man den Zähler auch per Software rücksetzen kann.

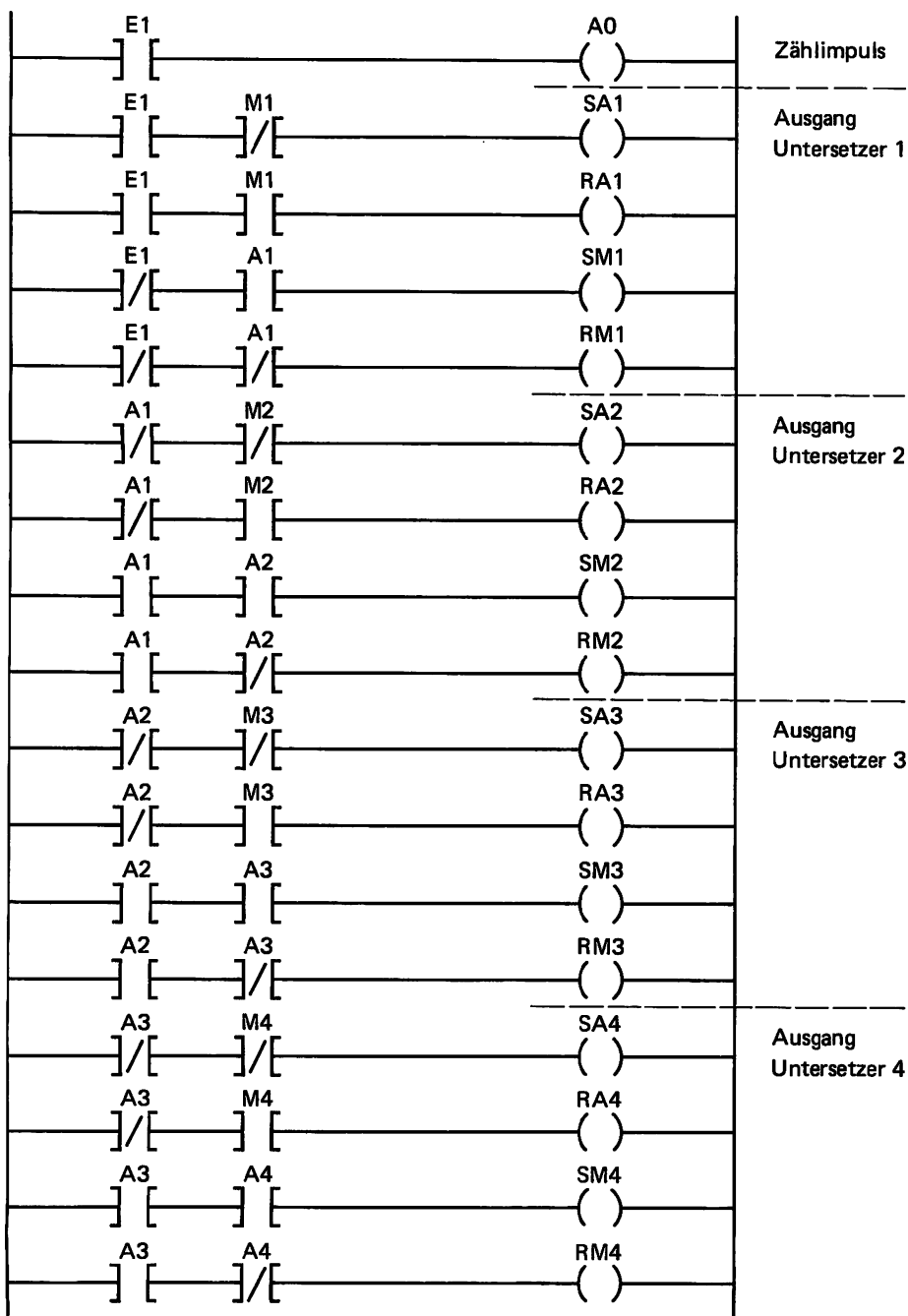


Bild E77.1
KOP des 4-Bit-Dualzählers mit
Binäruntersetzern, Beispiel E75.1.

Programm:

!E1=A0	Zählimpuls	!NA2&NM3=SA3	Ausgang 3. Stufe
!E1&NM1=SA1	Ausgang 1. Stufe	!NA2&M3=RA3	
!E1&M1=RA1		!A2&A3=SM3	
!NE1&A1=SM1		!A2&NA3=RM3	
!NE1&NA1=RM1		!NA3&NM4=SA4	Ausgang 4. Stufe
!NA1&NM2=SA2	Ausgang 2. Stufe	!NA3&M4=RA4	
!NA1&M2=RA2		!A3&A4=SM4	
!A1&A2=SM2		!A3&NA4=RM4	
!A1&NA2=RM2		!PE	

Testen Sie Ihr Programm und überprüfen Sie, ob die Ausgangssignale der Zählstufen dem Dualcode entsprechen. Beim 16-ten Zähltakt müssen die Anzeigen von 1,1,1,1 auf 0,0,0,0 springen.

- c) Nach dem Programm ist der in Bild E77.1 wiedergegebene KOP entwickelt.

E**78****Steuerung einer Ampelanlage**

In den folgenden Beispielen und Aufgaben stehen die Zählfunktionen im Mittelpunkt. Die Programme nehmen teilweise schon einen erheblichen Umfang an. Fehler beim Programmieren und Fehleingaben lassen sich auch bei konzentriertem Arbeiten nicht immer vermeiden. Es ist deshalb sinnvoll, ein Gesamtprogramm in Teilprogramme aufzulösen, zu programmieren und zu testen. Die Fehlersuche wird einfacher. Anschließend müssen Sie nur noch die Teilprogramme miteinander verbinden, um das Gesamtprogramm zu erhalten.

Wir besprechen zunächst die Arbeitsweise der Steuerung. Das Schema sehen Sie im Bild E79.1. Anschließend wird die Steuerung in Funktionsgruppen aufgelöst und programmiert.

Schritte		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Rot	H1,(A5)																
Gelb	H2,(A6)																
Grün	H3,(A7)																
Rot	H4,(A13)																
Gelb	H5,(A14)																
Grün	H6,(A15)																

Bild E78.1
Ablaufdiagramm der Ampelsteuerung.

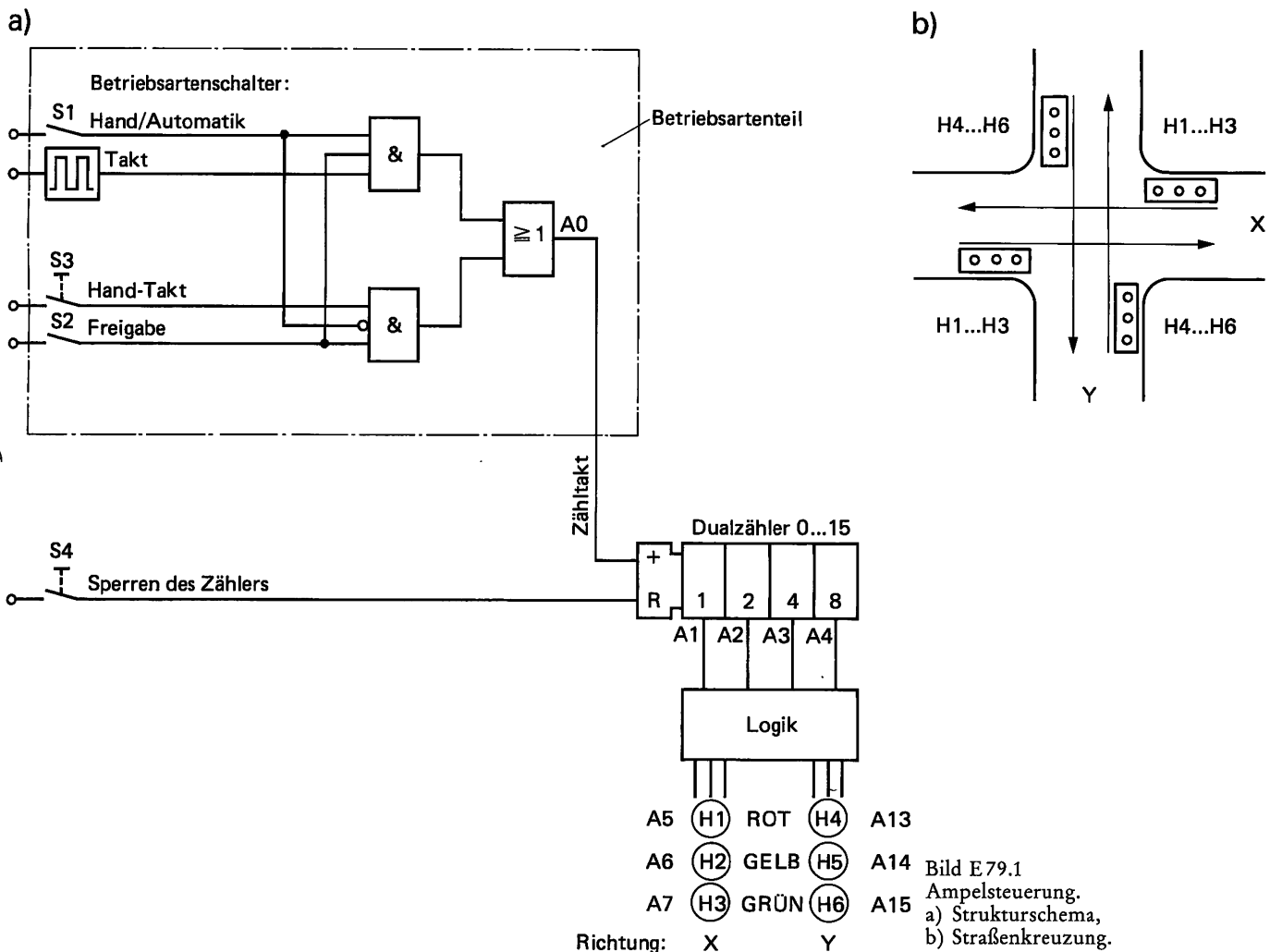


Bild E 79.1
Ampelsteuerung.
a) Strukturschema,
b) Straßenkreuzung.

Arbeitsweise:

Die Schaltphasen der Lichtsignale steuert ein 4-Bit-Dualzähler. Er erhält die Taktsignale (Zählsignale) vom Betriebsartenteil. Nach Freigabe der Steuerung mit Schalter S2 gelangen die von einem Generator (S1=1) oder von Hand (S1=0) erzeugten Taktsignale zum Dualzähler. Der Hand-Takt ist zum Programmtest erforderlich. Der Dualzähler erzeugt die Schaltschritte 0 bis 15 (0,0,0,0 bis 1,1,1,1). In Abhängigkeit vom Schaltschritt und damit vom Bitmuster der Zählerausgangssignale soll die Logik (ein Schaltnetz) die Steuersignale für die Signallampen H1 bis H6 erzeugen. Vorgegeben ist das Ablaufdiagramm in Bild E 78.1. Die Schrittnummer entspricht der Dualzahl des Zählerstands.

Zur weiteren Vorbereitung wollen wir noch eine Tabelle aufstellen, die uns die Zuordnung der Zählerausgangskombinationen zu den Schaltschritten zeigt. Außerdem stellen wir noch eine Liste der Betriebsmittel auf.

Tabelle E 80.1: Zuordnung der Zählerausgangskombinationen zu den Schaltschritten

A4	A3	A2	A1	Schritt
8	4	2	1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
.....				..
1	1	1	0	14
1	1	1	1	15

E

80

Tabelle E 80.2: Zuordnungsliste der Betriebsmittel

Betriebsmittel-Kennzeichen	Funktion	Operand
Eingang:		
S1	Betriebsartenschalter	E1
S2	Freigabeschalter	E2
S3	Taster für Handtakt	E3
S4	Taster zum Anhalten des Zählers	E4
Ausgang:		
H1	Signallampe ROT X	A5
H2	Signallampe GELB X	A6
H3	Signallampe GRÜN X	A7
H4	Signallampe ROT Y	A13
H5	Signallampe GELB Y	A14
H6	Signallampe GRÜN Y	A15
Timer:	Taktgenerator	T1,T2

Aufgabe E 80.1

Programmierung des Betriebsartenteils

Dieses Programm können Sie nach dem ausführlichen Schema von Bild E 79.1a selbst aufstellen.

Entwickeln Sie als Teilprogramme

- den Taktgenerator und
- die Logik des Betriebsartenteils.

Das Ausgangssignal der Logik ist zur Kontrolle auf Ausgang A0 zu legen. Die LED zeigt an, ob Taktsignale ausgegeben werden. Im Experiment arbeiten wir mit einer größeren Taktfrequenz als in der Praxis. Programmieren Sie eine Taktfrequenz von ca. 0,25 Hz.

- c) Testen Sie Ihr entwickeltes Programm und überprüfen Sie, ob der Betriebsartenschalter S1 seine Aufgabe erfüllt. An A0 dürfen nur nach Betätigung der Freigabe (S2=1) Taktsignale erscheinen.

Hinweis

Löschen Sie bitte Ihr Programm nach erfolgreichem Test noch nicht; es wird in der folgenden Aufgabe wieder benötigt.

Einen Lösungsvorschlag finden Sie auf Seite F55.

Aufgabe E81.1

Programmierung des Dualzählers

Das Programm des Dualzählers entspricht dem vom Beispiel E75.1. Sie brauchen nur das Eingangssignal (Zähltakt) von E1 in A0 ändern. Außerdem soll mit dem Stoppsignal E4 der Zählvorgang abgebrochen und der Zähler in die Anfangsstellung rückgesetzt werden.

Hoffentlich haben Sie Ihr Programm für den Betriebsartenteil noch nicht gelöscht. Sie können das Programm der Zählfunktion unmittelbar an den Betriebsartenteil anschließen. Es ist nur der Programmabschluß !PE am Ende des Betriebsartenteils zu löschen und an das Ende der Zählfunktion zu setzen.

Nach Eingabe des Programms ist die Gesamtfunktion zu testen. Überprüfen Sie bitte, ob der Dualzähler die Zählimpulse bei Hand- bzw. Automatikbetrieb nach dem vorgegebenen Code verarbeitet. Überprüfen Sie auch die Wirkung des Befehls E4.

Einen Lösungsvorschlag finden Sie auf Seite F55.

Beispiel E81.1

Programmierung der Logik zur Ansteuerung der Signalleuchten.

Nach dem Schema in Bild E79.1 setzt die Logik die Ausgangssignale der Zählfunktion in Signale zur Ansteuerung der Signalleuchten um. Das Programm zur Umsetzung ist im Ablaufdiagramm von Bild E78.1 eindeutig festgelegt. Mit den Ausgangsvariablen A1 bis A4 des Zählers sind zur Ansteuerung der Leuchten H Schaltfunktionen nach den Regeln der Digitaltechnik aufzustellen. Wir haben folgende Zuordnung getroffen:

H=A=1 Signalleuchte EIN
H=A=0 Signalleuchte AUS

- a) Stellen Sie die Funktionstabelle für die Logik auf. Daraus muß eindeutig hervorgehen, welchen Signalzustand die Ausgangsvariablen der Logik bei den Schaltschritten 0 bis 15 annehmen müssen. Tabelle E82.1 enthält die vorbereiteten Funktionen.
- b) Für die Ausgangsvariablen A5 bis A7 und A13 bis A15 sind die Schaltfunktionen aufzustellen und mit der Schaltalgebra optimal zu vereinfachen.

- c) Programmieren Sie die Schaltfunktionen aus b).
 d) Testen Sie anschließend das Programm.

Lösung:

- a) Die Eintragungen in den Spalten „Ausgang X“ und „Ausgang Y“ der Tabelle E 82.1 legen fest, bei welchen Eingabekombinationen die Ausgangsvariablen den Zustand 1 annehmen müssen.
 b) Die Funktionstabelle ergibt die folgenden Schaltfunktionen. Die Vereinfachung wurde nach den bisher schon oft demonstrierten Regeln vorgenommen. Überprüfen Sie bitte selbst die Ergebnisse.

Tabelle E 82.1: Funktionstabelle zum Beispiel E 81.1

Schritt	Ein- (Ausgang)				Ausgang X			Ausgang Y		
	A4	A3	A2	A1	A5	A6	A7	A13	A14	A15
0	0	0	0	0	1			1		
1	0	0	0	1	1	1		1		
2	0	0	1	0			1	1		
3	0	0	1	1			1	1		
4	0	1	0	0			1	1		
5	0	1	0	1			1	1		
6	0	1	1	0		1		1		
7	0	1	1	1	1			1		
8	1	0	0	0	1			1	1	
9	1	0	0	1	1					1
10	1	0	1	0	1					1
11	1	0	1	1	1					1
12	1	1	0	0	1					1
13	1	1	0	1	1					1
14	1	1	1	0	1					1
15	1	1	1	1	1				1	

Zur besseren Übersicht wurden für die Ausgangsvariablen nur die Signalzustände markiert, bei denen sie den Wert 1 annehmen müssen. Gedanklich sind die nicht ausgefüllten Stellen mit dem Wert 0 zu besetzen.

Schaltfunktionen und deren Vereinfachung:

A5=

$$\begin{aligned} & \overline{A1} \overline{A2} \overline{A3} \overline{A4} \vee \overline{A1} \overline{A2} \overline{A3} A4 \vee \overline{A1} A2 \overline{A3} \overline{A4} \vee \overline{A1} \overline{A2} \overline{A3} A4 \vee \\ & \overline{A1} \overline{A2} \overline{A3} A4 \vee \overline{A1} A2 \overline{A3} A4 \vee \overline{A1} A2 \overline{A3} A4 \vee \overline{A1} \overline{A2} \overline{A3} A4 \vee \\ & \overline{A1} \overline{A2} \overline{A3} A4 \vee \overline{A1} A2 \overline{A3} A4 \vee \overline{A1} \overline{A2} \overline{A3} A4 \end{aligned}$$

Nach der Regel $A1 A2 A4 (\overline{A3} \vee A3) = A1 A2 A4$ lassen sich die farbig markierten Terme zusammenfassen. Die erste Vereinfachung ergibt:

$$A5 = \overline{A2} \overline{A3} \overline{A4} \vee A1 A2 A3 \vee \overline{A2} \overline{A3} A4 \vee \overline{A2} A3 A4 \vee \overline{A2} A3 A4 \vee \overline{A2} A3 A4$$

Die nochmalige Ausklammerung führt zu folgender Lösung:

$$A5 = A1 A2 A3 \vee \overline{A2} \overline{A3} \vee A4$$

$$A6 = A1 \overline{A2} \overline{A3} \overline{A4} \vee \overline{A1} A2 A3 \overline{A4} \quad (\text{keine Vereinfachung möglich})$$

$$A7 = \overline{A1} A2 \overline{A3} \overline{A4} \vee \overline{A1} A2 \overline{A3} A4 \vee \overline{A1} A2 A3 \overline{A4} \vee \overline{A1} A2 A3 A4$$

$$A7 = A2 \overline{A3} \overline{A4} \vee \overline{A2} A3 \overline{A4}$$

$$A13 = \overline{A1} \overline{A2} \overline{A3} \overline{A4} \vee \overline{A1} \overline{A2} \overline{A3} A4 \vee \overline{A1} \overline{A2} A3 \overline{A4} \vee \overline{A1} \overline{A2} A3 A4 \vee \overline{A1} A2 \overline{A3} \overline{A4} \vee \overline{A1} A2 \overline{A3} A4 \vee \overline{A1} A2 A3 \overline{A4} \vee \overline{A1} A2 A3 A4$$

$$A13 = \overline{A1} \overline{A2} \overline{A3} \vee \overline{A4}$$

$$A14 = \overline{A1} \overline{A2} \overline{A3} A4 \vee A1 A2 A3 A4 \quad (\text{keine Vereinfachung möglich})$$

$$A15 = \overline{A1} \overline{A2} \overline{A3} A4 \vee \overline{A1} \overline{A2} \overline{A3} A4 \vee \overline{A1} \overline{A2} \overline{A3} A4 \vee \overline{A1} \overline{A2} A3 A4 \vee \overline{A1} \overline{A2} A3 A4 \vee \overline{A1} \overline{A2} A3 A4$$

$$A15 = \overline{A2} A3 A4 \vee A1 \overline{A3} A4 \vee \overline{A1} A2 A4$$

Den FUP der Logik sehen Sie in Bild E 84.1.

- c) Die Schaltfunktionen können Sie unmittelbar programmieren. Wir erhalten:

Programm:

!A1&A2&A3/NA2&NA3/A4=A5	ROT X
!A1&NA2&NA3&NA4/NA1&A2&A3&NA4=A6	GELB X
!A2&NA3&NA4/NA2&A3&NA4=A7	GRÜN X
!NA1&NA2&NA3/NA4=A13	ROT Y
!NA1&NA2&NA3&A4/A1&A2&A3&A4=A14	GELB Y
!NA2&A3&A4/A1&NA3&A4/NA1&A2&A4=A15	GRÜN Y
!PE	

- d) Zum Test des Logik-Programms ist es empfehlenswert, den Variablen A1 bis A4 die Variablen E1 bis E4 zuzuweisen. Sie können dann durch Betätigung der Schalter E1 bis E4 den Zähler simulieren und überprüfen, ob bei den verschiedenen Schritten die richtigen Signale ausgegeben werden. Setzen Sie !E1=A1 ... !E4=A4.

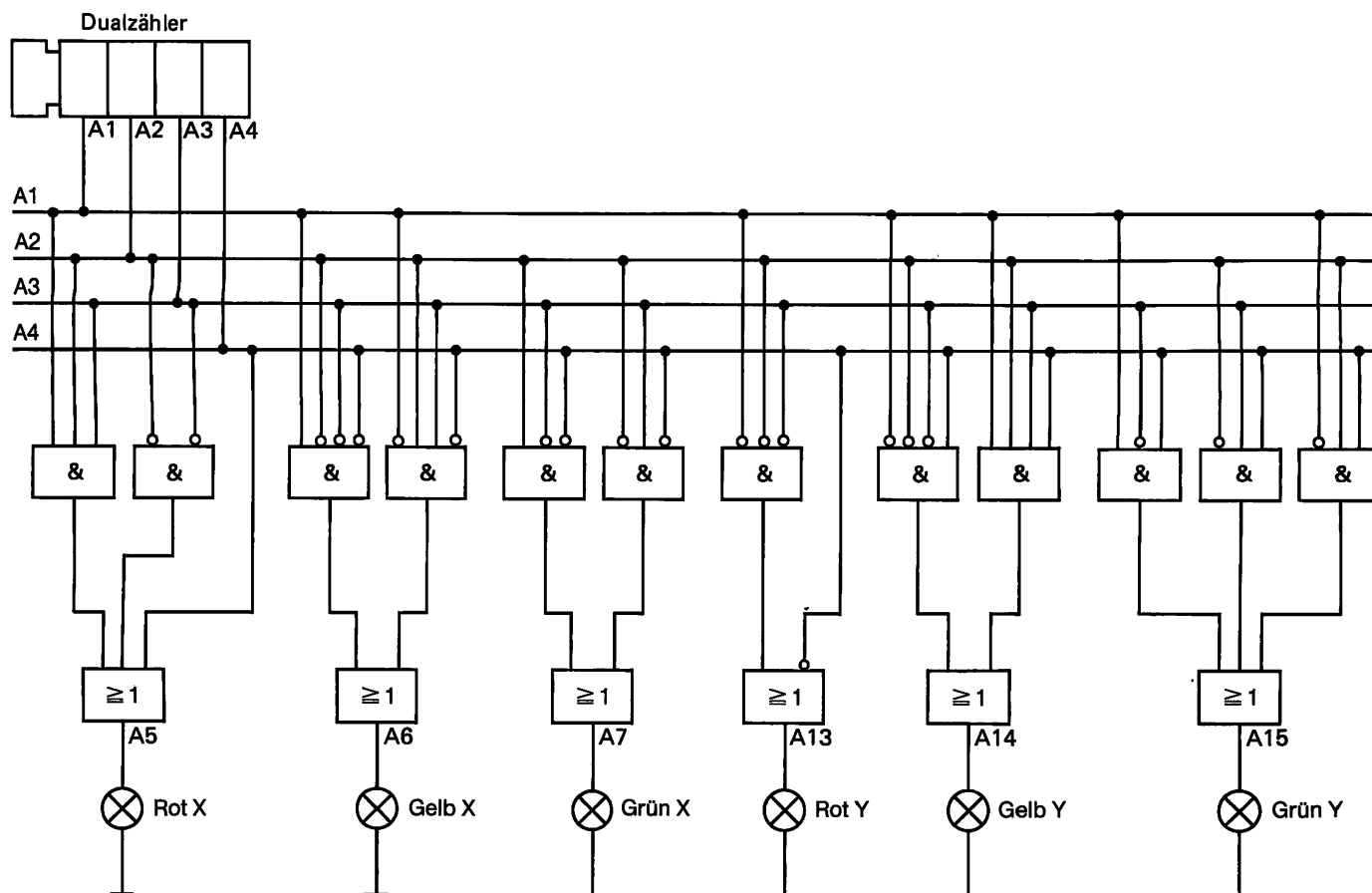


Bild E 84.1
FUP der Logik.

Nun sind alle Teilprogramme aufgestellt und getestet. Mit diesen Teilprogrammen kann das Gesamtprogramm der Ampelsteuerung nach dem Schema in Bild E 85.1a zusammengesetzt werden.

Dieses Programm erreicht bereits die Einsatzgrenze unserer SPS. Es überschreitet die Aufnahmekapazität des Arbeitsspeichers. Dieser ist nur für 24 Bildschirmzeilen ausgelegt, das Gesamtprogramm besteht aber aus mehr als 24 Anweisungen.

Damit wir trotzdem das Zusammenwirken von Dualzähler, Logik und Anzeige demonstrieren können, wird nach dem Schema in Bild E 85.1b mit einem reduzierten Programm gearbeitet; wir betreiben die Steuerung mit einem „Handtakt“. Taktgenerator und Logik des Betriebsartenteils entfallen. Unser Zähltakt wird mit Schalter E3 als Taster (0-1-0-..) erzeugt und unmittelbar in den Dualzähler eingegeben. Der Handtakt hat für uns auch noch den Vorteil, daß wir in Ruhe die Schaltphasen betrachten können.

Das Beispiel sollte Ihnen u. a. zeigen: Die Kapazität des Speichers begrenzt den Umfang des Programms und damit auch den Einsatzbereich unseres SPS-Systems .

In industriellen Automatisierungsgeräten ist der Speicher meist aus Speichermodulen zusammengesetzt. Durch Hinzufügen weiterer Module läßt sich die Kapazität somit den Erfordernissen anpassen.

Testprogramm:

```

IE3&NM1=SA1
IE3&M1=RA1
INE3&A1=SM1
INE3&NA1=RM1
INA1&NM2=SA2
INA1&M2=RA2
IA1&A2=SM2
IA1&NA2=RM2
INA2&NM3=SA3
INA2&M3=RA3
IA2&A3=SM3
IA2&NA3=RM3
INA3&NM4=SA4
INA3&M4=RA4
IA3&A4=SM4
IA3&NA4=RM4
IE4=RA1=RA2=RA3=RA4
IA1&A2&A3/NA2&NA3/A4=A5
IA1&A2&A3/NA2&NA3/NA4=A6
IA1&NA2&NA3&NA4/NA1&A2&A3&NA4=A6
IA2&NA3&NA4/NA2&A3&NA4=A7
INA1&NA2&NA3/NA4=A13
INA1&NA2&NA3&A4/A1&A2&A3&A4=A14
INA2&A3&A4/A1&NA3&A4/NA1&A2&A4=A15

```

Damit eine Anweisung auch nur eine Zeile auf dem Bildschirm bean-
sprucht, darf sie maximal 30 Zeichen lang sein. Im Testprogramm
haben die Anweisungen für die Ausgangsvariablen A6, A14 und A15
mehr als 30 Zeichen, sie beanspruchen deshalb zwei Zeilen. Unter die-
sen Voraussetzungen würden wir auch das reduzierte Programm nicht
im Arbeitsspeicher unterbringen. Hier hilft uns die Schaltungsge-
bra weiter. Nach den Klammeregeln dürfen gleiche Variable der Terme
ausgekammert werden. Dadurch reduziert sich die erforderliche
Zeichenzahl pro Anweisung.

Ändern und ersetzen Sie die folgenden Anweisungen des Testpro-
gramms:

A6: $IA1 \& NA2 \& NA3 \& NA4 / NA1 \& A2 \& A3 \& NA4 = A6$

Variable NA4 wird ausgeklammert:

$INA4 \& (A1 \& NA2 \& NA3 / NA1 \& A2 \& A3) = A6$

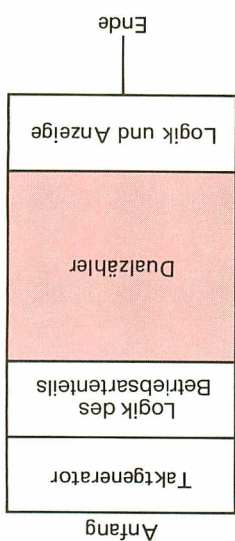
A14: Variable A4 wird ausgeklammert:

$IA4 \& (NA1 \& NA2 \& NA3 / A1 \& A2 \& A3) = A14$

A15: Variable A4 wird ausgeklammert:

$IA4 \& (NA2 \& A3 / A1 \& NA3 / NA1 \& A2) = A15$

a)



b)

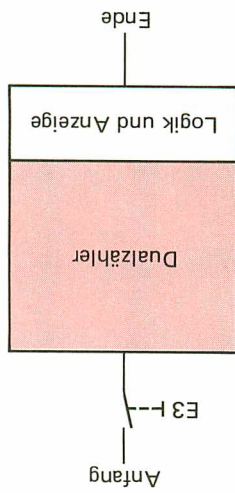


Bild E 85.1

Zusammensetzung des Gesamt-
programms.
a) vollständiges Programm,
b) reduziertes Programm.

Mit diesen Änderungen bringen wir unser Programm in den Arbeitsspeicher. Ein anschließender Test ergibt eine Übereinstimmung mit den Signalfolgen laut unserer Aufgabenstellung.

Dualzähler mit Flankenauswertung

Wir haben Ihnen gezeigt, wie man eine Zählfunktion mit dem Binär-untersetzer verwirklicht. Der Untersetzer spricht auf den Wechsel zum 1-Zustand des Zähltaktes an. In der Praxis kann als Aufgabe gestellt sein, den Zählvorgang mit einer Signalfanke auszulösen. In diesem Fall muß das Programm der Zählfunktion mit einer Flankenauswertung beginnen. Den Flankenauswerter haben wir in Lehrbrief 3 (Seite E 71) unter „Programmierung von Sonderfunktionen“ besprochen. Das Strukturschema für das Gesamtprogramm zeigt das Bild E 86.1. Der Flankenauswerter ist nur einmal erforderlich. Danach folgen die Zählstufen des Dualzählers.

Das folgende Programm eines 4-Bit-Dualzählers mit Flankenauswerter für die steigende Signalfanke des Zähltakts entspricht dem Strukturschema von Bild E 86.1.

Programm:

!E1&NM0=M1	Flankenauswerter
!M1=SM0	
!NE1=RM0	
!M1&NA1=SA1=RM1	A1 \triangleq 1, Ausgang Zählstufe 1
!M1&A1=RA1	
!M1&NA2=SA2=RM1	A2 \triangleq 2, Ausgang Zählstufe 2
!M1&A2=RA2	
!M1&NA3=SA3=RM1	A3 \triangleq 4, Ausgang Zählstufe 3
!M1&A3=RA3	
!M1&NA4=SA4=RM1	A4 \triangleq 8, Ausgang Zählstufe 4
!M1&A4=RA4	
!PE	

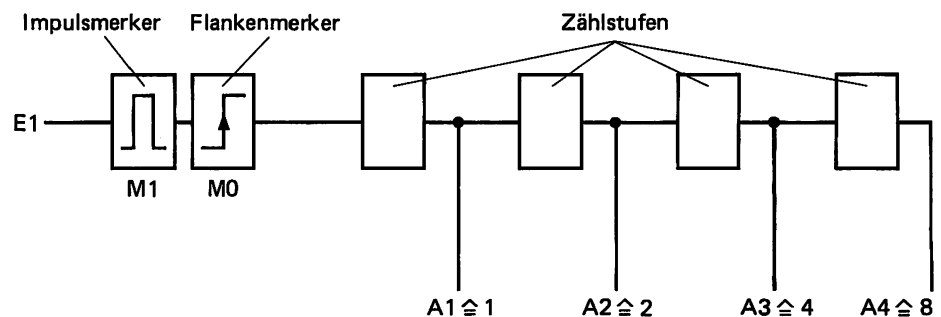


Bild E 86.1
Struktur eines 4-Bit-Dualzählers mit Flankenauswertung für die ansteigende Signalfanke.

Aufgabe E 86.1

Test des Dualzählers mit Flankenauswerter

a) Zeichnen Sie nach dem Programm den aufgelöste KOP auf.

- b) Testen Sie das vorstehende Programm. Mit Schalter E1 sind die Zählimpulse nachzubilden. Der Zählerstand muß sich bei steigender Flanke, also beim Schalten AUS → EIN ($0 \rightarrow 1$) ändern. Nach Inbetriebnahme ist, falls erforderlich, durch Schalterbetätigung die Anfangsstellung $A1, A2, A3, A4 = 0, 0, 0, 0$ herzustellen. Überprüfen Sie danach die Zählfunktion durch experimentelle Aufnahme der Funktionstabelle E 87.1.

Tabelle E 87.1: Funktionstabelle zur Aufgabe E 86.1

E1 Takt	A1 1	A2 2	A3 4	A4 8	Dezimaler Wert	E1 Takt	A1 1	A2 2	A3 4	A4 8	Dezimaler Wert
1						9					
2						10					
3						11					
4						12					
5						13					
6						14					
7						15					
8						16					

Die Lösung dieser Aufgabe finden Sie auf Seite F56.

Aufgabe E 87.1

Erweitern Sie die Zählfunktion von Aufgabe E 86.1 um eine Zählstufe. Die neue Funktion erhält die Ausgangsvariable A5. Da wir einen Dualzähler nachbilden, muß A5 die dezimale Wertigkeit 16 haben.

- a) Erweitern Sie das Programm des vorgegebenen Dualzählers um eine Zählstufe.
- b) Testen sie das Programm. Nach der Programmeingabe muß die Anfangsstellung $A1, A2, A3, A4, A5 = 0, 0, 0, 0, 0$ eingestellt werden. Überprüfen Sie nur die in Tabelle E 87.2 aufgeführten Zählerzustände.

Tabelle E 87.2: Testtabelle zur Aufgabe E 87.1

E1 Takt	A1 1	A2 2	A3 4	A4 8	A5 16	Dezimal- wert
	0	0	0	0	0	0
1						
7						
15						
16						
23						
24						
31						
32						

Die Lösung dieser Aufgabe finden Sie auf Seite F57.

Rücksetzen des 4-Bit-Dualzählers auf den Anfangswert

Mit der Eingabe E2=1 soll die Zählfunktion auf den Anfangswert Null gesetzt werden. E2=1 muß die Speicherfunktionen der Zählstufen und auch den Impulsmerker M1 rücksetzen. Danach befindet sich das Programm wieder im Anfangszustand. Der folgende Programmteil ist nach dem Schema in Bild E 88.1 zwischen Flankenauswerter und die erste Zählstufe zu setzen.

Programmerweiterung:

```

..... Flankenauswerter

!E2=RA1=RA2=RA3=RA4=RM1

..... 1. Zählstufe
  
```

Aufgabe E 88.1

Mit den vorstehenden Rücksetzanweisungen wird das Zählerprogramm von Aufgabe E 86.1 erweitert.

- Überprüfen Sie, ob bei beliebigem Zählerstand die Zählfunktion durch ein kurzes 1-Signal E2 auf den Anfangszustand Null gesetzt wird.
- Was passiert, wenn ein Dauersignal E2=1 eingegeben wird?

Die Lösung dieser Aufgabe finden Sie auf Seite F 58.

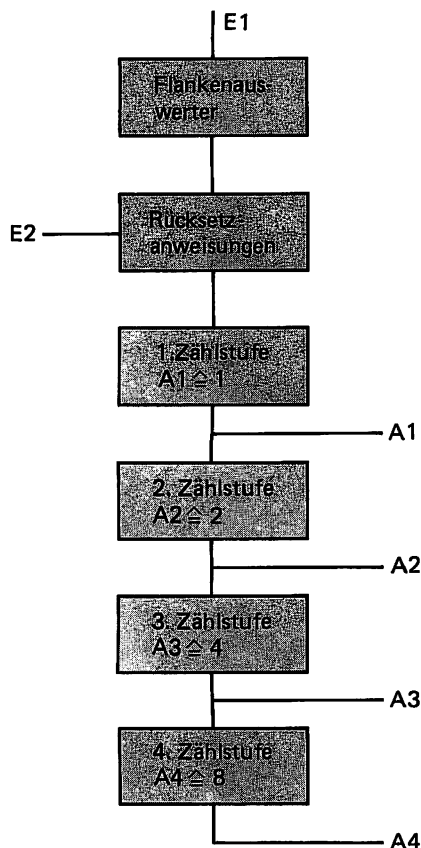


Bild E 88.1
Programmstruktur eines Dualzählers mit Rücksetzanweisung.
E1 Zählimpuls,
E2 Rücksetzsignal.

Zähldekade im BCD-Code, BCD-Korrektur

Die bisher programmierten Dualzähler arbeiten als Vorwärtzzähler. Da wir das Grundprogramm weiter verwenden, entsteht daraus eine **Vorwärtzzähldekade** im BCD- (8-4-2-1-) Code. Das Bild E 89.1 enthält die Programmstruktur.

Eine Zähldekade darf nur von 0 bis 9 zählen. Beim 10-ten Zählimpuls muß die Dekade wieder in den Zustand 0 schalten. Ein vom Rücksetzen abgeleiteter Zählimpuls wird beim mehrstelligen Zähler der nächsthöheren Dekade zugeführt. Man bezeichnet diesen Impuls als **Übertrag**. Bei mehrstelligen Zählern sind die Dekaden in der Reihenfolge: „Einer – Zehner – Hunderter – ...“ angeordnet.

Wir wollen den bisher besprochenen 4-Bit-Dualzähler in eine **Zähldekade** umprogrammieren. Die Schaltzustände der Ausgangsvariablen A1,A2,A3,A4 sind von 0,0,0,0 = 0 bis 1,0,0,1 = 9 erlaubt. Der dann folgende Zustand 0,1,0,1 = 10 ist nicht mehr erlaubt; mit ihm wird das Rücksetzen auf den Anfangswert 0,0,0,0 programmiert. Der Zähler schaltet ganz kurz in den Zustand 0,1,0,1. Hierbei befinden sich die Variablen A1 und A3 bereits im Zustand 0. Es sind nur die 1-Zustände der Variablen A2 und A4 auf den Wert 0 zu bringen.

Die Abfrage auf den Wert 10 wird zum Rücksetzen in den Zustand 0 ausgenutzt. Diesen Vorgang bezeichnet man als **BCD-Korrektur**.

Programm der BCD-Korrektur:

!A2&A4=RA2=RA4

=M7 Übertragssignal

Aufgabe E 89.1

Das Programm des Dualzählers mit Flankenauswertung (Seite E 86) ist in ein Programm für eine BCD-Zähldekade abzuändern.

Ein Übertrag ist nicht notwendig, da unser Programm nur eine Dekade umfaßt.

- Bauen Sie in das Programm von Aufgabe E 86.1 eine BCD-Korrektur nach dem Schema von Bild E 89.1 ein.
- Überprüfen Sie Ihr Programm durch Aufnahme der Funktionstabelle E 89.1. Beim 10-ten Takt muß der Zähler wieder den Stand 0,0,0,0 annehmen.

Tabelle E 89.1: Testtabelle zu Aufgabe E 89.1

E1 Takt	A1 1	A2 2	A3 4	A4 8	Dezimalwert
	0	0	0	0	
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

Die Lösung dieser Aufgabe finden Sie auf Seite F 58.

Beispiel E 89.1**Programmierung eines Dualzählers „1 bis 6“ mit Flankenauswertung und Rückstellung**

Nach dem Schema in Bild E 90.1 soll die Zählfunktion nur bis zum Dezimalwert 6 zählen und anschließend auf den Dezimalwert 1 zurückspringen. Der dezimale Zählbereich umfaßt die Werte 1 bis 6.

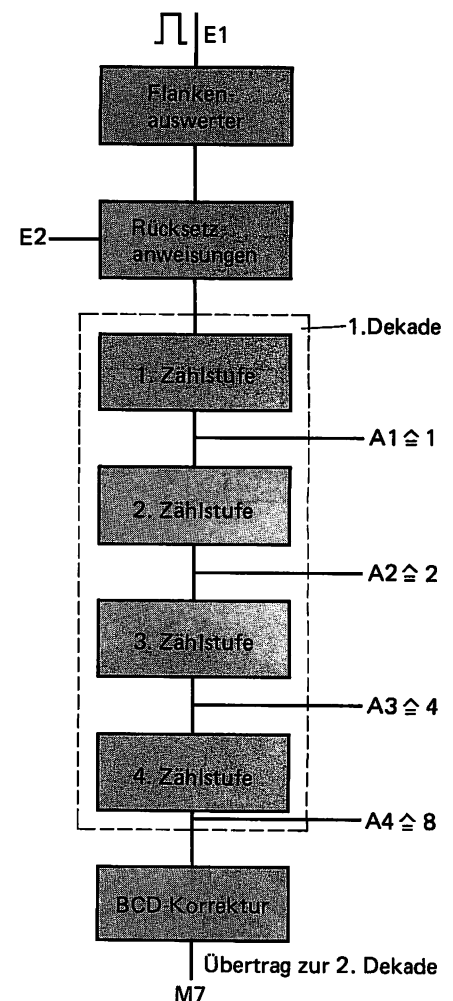


Bild E 89.1
Programmstruktur der BCD-Zähl-
funktion.

- Entwerfen Sie ein geeignetes Korrekturprogramm und begrenzen Sie das Programm des Dualzählers mit dem Korrekturprogramm auf den geforderten Zählbereich.
- Stellen Sie das Gesamtprogramm auf.
- Der Programmtest erfolgt wieder durch experimentelle Aufnahme der Funktionstabelle E 90.1.

Tabelle E 90.1: Testtabelle zum Beispiel E 89.1

E1 Takt	A1 1	A2 2	A3 4	Dezimalwert
	0	0	0	0
1	1	0	0	1
2	0	1	0	2
3	1	1	0	3
4	0	0	1	4
5	1	0	1	5
6	0	1	1	6
7	1	0	0	1
8	0	1	0	2

E

90

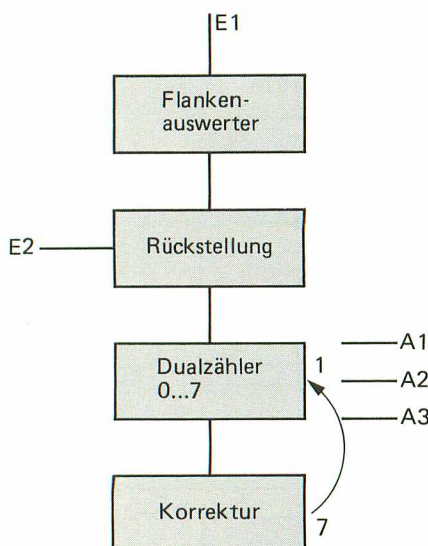


Bild E 90.1
Zählfunktion 1 bis 7 für den Dual-
code mit Flankenbewertung und
Rückstellung.

Lösung:

- Wir benötigen noch die Zählstufen A1 bis A3 mit den dezimalen Wertigkeiten $A1 \triangleq 1$, $A2 \triangleq 2$ und $A3 \triangleq 4$. Der Zähler darf nur bis zum Wert $A1, A2, A3 = 0, 1, 1 \triangleq 6$ zählen. Der Zustand 1,1,1 ist nicht mehr erlaubt, er wird deshalb zur Auslösung des Rücksprungs ausgenutzt. Laut Aufgabenstellung muß der Rücksprung in den Zustand $A1, A2, A3 = 1, 0, 0$ erfolgen. Mit dem Korrekturprogramm sind deshalb nur die Zählstufen A2 und A3 auf den Wert 0 zu setzen.

Aus dieser Überlegung folgt die nachstehende Korrekturanweisung:

$$!A1 \& A2 \& A3 = RA2 = RA3$$

b) Gesamtprogramm:

$$\begin{aligned}
 &!E1 \& NM0 = M1 && !M1 \& A2 = RA2 \\
 &!M1 = SM0 && !M1 \& NA3 = SA3 = RM1 \\
 &!NE1 = RM0 && !M1 \& A3 = RA3 \\
 &!E1 = RA1 = RA2 = RA3 = RM1 && !A1 \& A2 \& A3 = RA2 = RA3 \\
 &!M1 \& NA1 = SA1 = RM1 && !PE \\
 &!M1 \& A1 = RA1 \\
 &!M1 \& NA2 = SA2 = RM1
 \end{aligned}$$

- Der Programmtest ergibt die in Tabelle E 90.1 ROT eingetragenen Ergebnisse.

Programmier-Praxis

Wir wollen Ihre Programmierkenntnisse wieder um einige Feinheiten erweitern und verschiedene Beispiele aus der Praxis vorstellen. Vorher erhalten Sie zur Wiederholung des Gelernten eine relativ umfangreiche Programmieraufgabe aus der Steuerungstechnik.

Aufgabe E 91.1

Transportbandsteuerung als Verknüpfungssteuerung.

Das Technologieschema der Steuerungsaufgabe zeigt das Bild E 91.1. Als verbale Beschreibung können wir sagen: Der Vorratsbehälter 2 wird über Behälter 1 nachgefüllt. Beim Unterschreiten eines bestimmten Füllstands im Behälter 2, das vom Minimum-Niveau-Schalter E4 gemeldet wird, soll der Füllvorgang beginnen. Der Behälter 2 wird vom Behälter 1 solange gefüllt, bis der Maximum-Niveau-Schalter E3 anspricht. Die Anweisung „Füllen“ (gemeldet durch E4) soll folgende Operationen in der angeführten Reihenfolge auslösen:

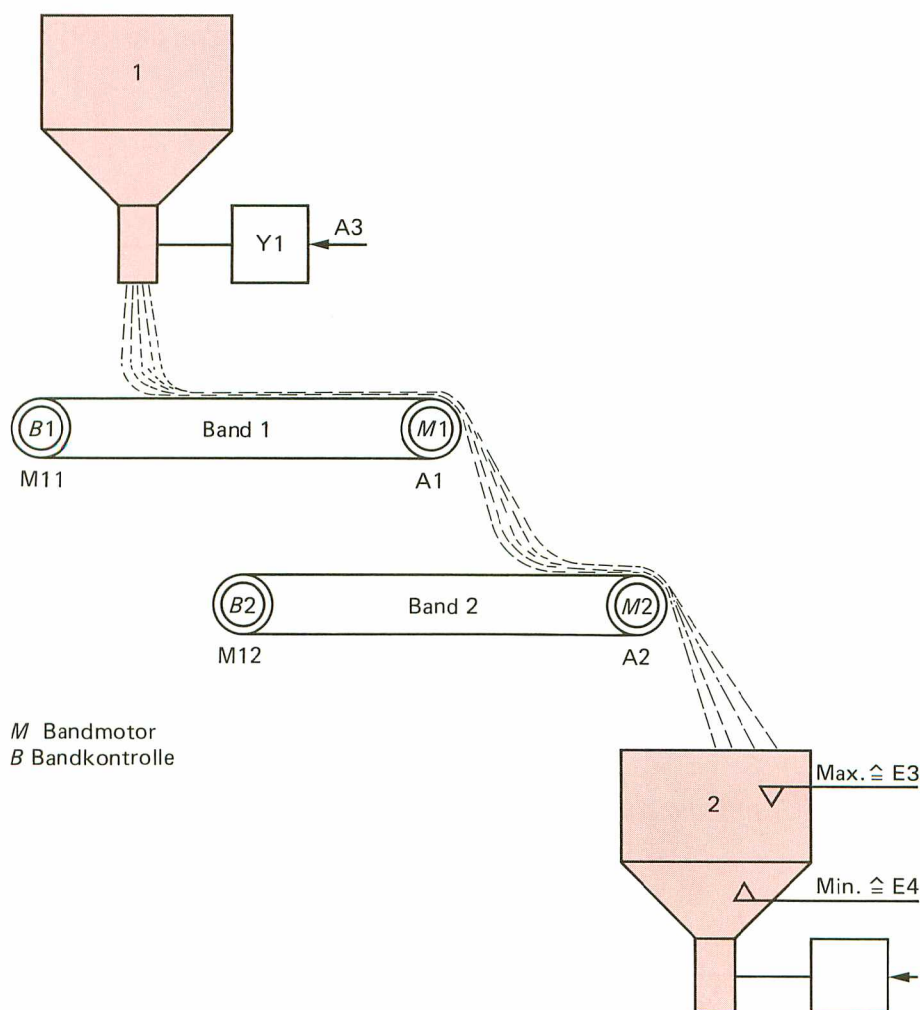


Bild E 91.1
Technologieschema einer Transportbandsteuerung, Aufgabe E 91.1.

Wir legen fest:

A1,2=1 bedeutet: Motor EIN A3=1 bedeutet Schieber AUF
 A1,2=0 bedeutet: Motor AUS A3=0 bedeutet Schieber ZU

M11,12=1 bedeutet: Band hat die richtige Arbeitsgeschwindigkeit,
 M11,12=0 bedeutet: Band hat noch nicht die richtige Arbeits-
 geschwindigkeit.

E3=1 bedeutet: Behälter 2 voll (Auslösung der Stop-Anweisung).
 E3=0 bedeutet: Behälter 2 noch nicht voll.

E4=1 bedeutet: Behälter 2 nicht leer.
 E4=0 bedeutet: Behälter 2 leer (Start).

Zwischen der Erteilung des Befehls „Motor M2 EIN“ (A2=1) und der Meldung von B2 „richtige Bandgeschwindigkeit erreicht“ (E2=1) vergeht eine gewisse Zeit. Diese ist von den elektrischen und mechanischen Daten der verwendeten Geräte abhängig. Wir bilden diese Verzögerung durch ein Zeitglied nach. Das gleiche gilt auch für den Anlauf von Band 1.

a) Führen Sie die Transportbandsteuerung als Verknüpfungssteuerung mit den RS-Speicherfunktionen für A1, A2 und A3 aus. Entwerfen Sie dafür den FUP in aufgelöster Darstellung.

b) Nach dem FUP ist das Programm aufzustellen. Wählen Sie für die Zeitfunktionen folgende Verzögerungszeiten:

Einschaltverzögerung $t_{1,1}=50$ ZE
 Ausschaltverzögerung $t_{1,2}=100$ ZE

Benutzen Sie die im Technologieschema und im Signal-Zeit-Diagramm eingetragenen Operanden. Weitere Operanden sind, falls erforderlich, selbst zu wählen.

c) Durch folgerichtige Betätigung der Eingaben ist das Programm zu überprüfen. Eine Testmöglichkeit bietet die aufgeführte Funktionstabelle E 93.1. Während des Tests haben die Ausgänge A1, A2 und A3 1-Signal, wenn E3=0 und E4=1.

Tabelle E 93.1: Funktionstabelle zur Aufgabe E 91.1

E4	E3	A2	A1	A3	
1	1	0	0	0	Behälter 2 voll
1	0	0	0	0	
0	0	1	0→1	0→1	Behälter 2 leer
1	0	1	1	1	
1	1	1→0	1→0	0	Behälter 2 voll

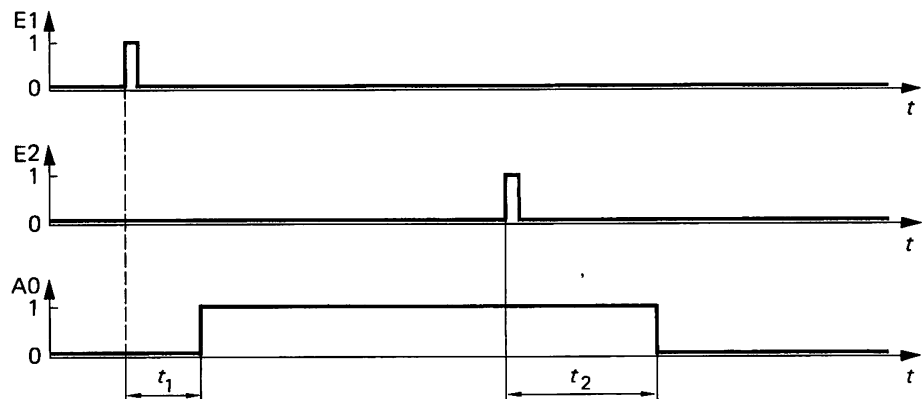
0→1 verzögerter Übergang von Zustand 0 in 1
 1→0 verzögerter Übergang von Zustand 1 in 0

Einen Lösungsvorschlag mit Erläuterungen finden Sie auf Seite F 59.

Erweiterung der Anweisungen für Zeitfunktionen

In Verbindung mit den Operationen S und R kann die Wirkung der Zeitfunktionen nach dem Signal-Zeit-Diagramm von Bild E 94.1 erweitert werden. Danach sind zum Starten und Rücksetzen des Timers nur noch kurze Signalimpulse erforderlich. Realisiert wird dieses Verhalten durch das folgende Programm:

Bild E 94.1
Zeitfunktion, Signal-Zeit-Diagramm mit S- und R-Anweisungen.
 $t_1=000$ bis 999 ZE Einschaltverzögerung,
 $t_2=000$ bis 999 ZE Ausschaltverzögerung.



Programm:

```
!E1=ST0    !T0=A0    !E2=RT0    !PE
```

Der ausgewählte Timer T0 wird natürlich wie bisher mit den Verzögerungszeiten t_1 und t_2 programmiert. Laut Programm erfolgt der Start der Zeitfunktion T0 mit dem Impuls E1. Diese Kurzzeitanweisung wird gespeichert (ST0). Nach Ablauf der Einschaltverzögerung t_1 nimmt der Ausgang des Timers Zustand 1 an. Er ist in unserem Beispiel auf Ausgang A0 geschaltet. Mit E2 ist die Ausschaltanweisung ebenfalls als Kurzzeitsignal eingegeben. Nach Ablauf der Ausschaltverzögerung t_2 nimmt Timer T0 Zustand 0 an.

Testen Sie bitte das vorstehende Programm. T0 ist mit $t_1=t_2=100$ ZE zu programmieren. Die Schalter E1 und E2 werden nur für kurze Zeit in Stellung EIN gebracht. Die Ansteuerung der Timer mit Kurzzeitsignalen bzw. -impulsen eröffnet uns weitere, bisher nur sehr umständlich oder überhaupt nicht realisierbare Anwendungen.

Beispiel E 94.1

Programmierung einer Signalfolge

Es ist ein Programm zu entwerfen, das die Signalfolge des Signal-Zeit-Diagramms von Bild E 95.1 erzeugt.

Lösung:

Eingang E0 startet Timer T1. Nach Ablauf der Einschaltverzögerung $t_{1,1}$ schaltet T1 den Ausgang A1 in Zustand 1. T1=1 startet Timer T2. Bei T2=1 nimmt A2 Zustand 1 an. Timer T1 wird mit E1 und Timer T2 mit E2 unverzüglich zurückgesetzt. A0 meldet das Startsignal E0.

Programmieren Sie die Timer mit folgenden Verzögerungszeiten:

$$t_{1,1}=t_{1,2}=100 \text{ ZE} \quad t_{2,1}=t_{2,2}=0 \text{ ZE}$$

Programm: !E0=A0 impulsförmiges Startsignal
 !E0=ST1 Start von Timer T1
 !T1=A1
 !E1=RT1 Rücksetzen von Timer T1
 !T1=ST2 Start von Timer T2
 !T2=A2
 !E2=RT2 Rücksetzen von Timer T2
 !PE

Der Programmtest zeigt das Signal-Zeit-Verhalten von Bild E95.1.

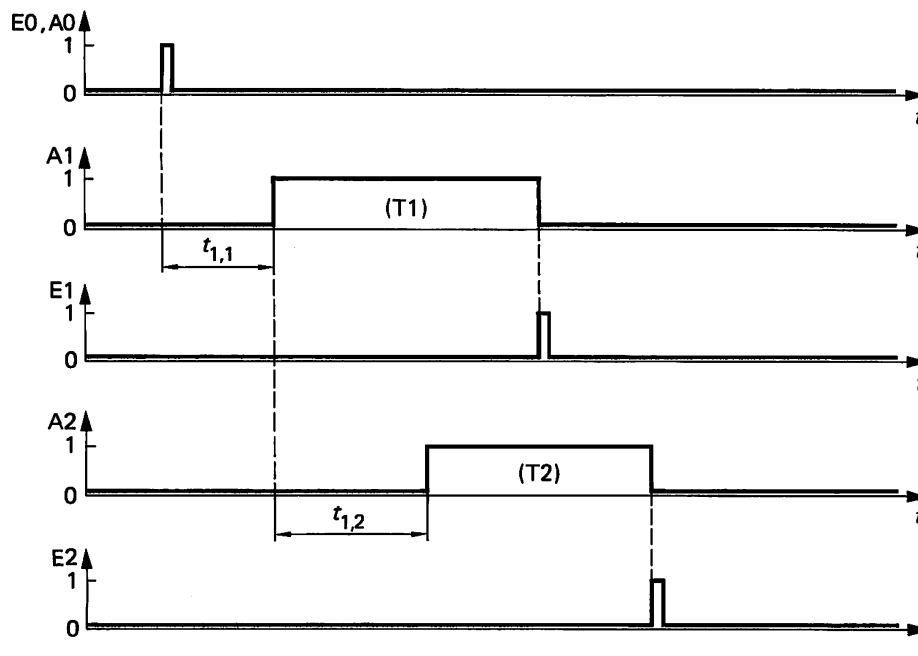


Bild E 95.1
Signal-Zeit-Diagramm,
Beispiel E 94.1.

Aufgabe E 95.1

Programmierung eines Lauflichtes

Das zu programmierende Signal-Zeit-Diagramm gibt Bild E 96.1 wieder. Verwenden Sie für die Zeitfunktionen die Operationen STx und RTx. Für x ist ein Wert als Parameter (Adresse) des Timers einzusetzen.

Zum Programm noch einige Hinweise: Der Start erfolgt mit dem Signalimpuls E0. Wählen Sie für alle Timer die gleiche Einschaltverzögerung, z. B. $t_1=100$ ZE. Die Signalzustände der Timer sind auf die Ausgänge A0 bis A7 zu legen (T0 auf A0 usw.). Der Signalzustand 1 von A7 soll $t_1=100$ ZE betragen.

- Entwickeln Sie ein Programm, das nach einem Programmablauf endet (nach Rücksprung in den Anfangszustand).
- Erweitern Sie Ihr Programm a). Mit Signal E1=1 soll Dauerbetrieb entstehen. Erst mit E1=0 wird der Dauerbetrieb gestoppt (nach Rücksprung in den Anfangszustand). Dauerbetrieb bedeutet, daß sich das Programm laufend wiederholt.

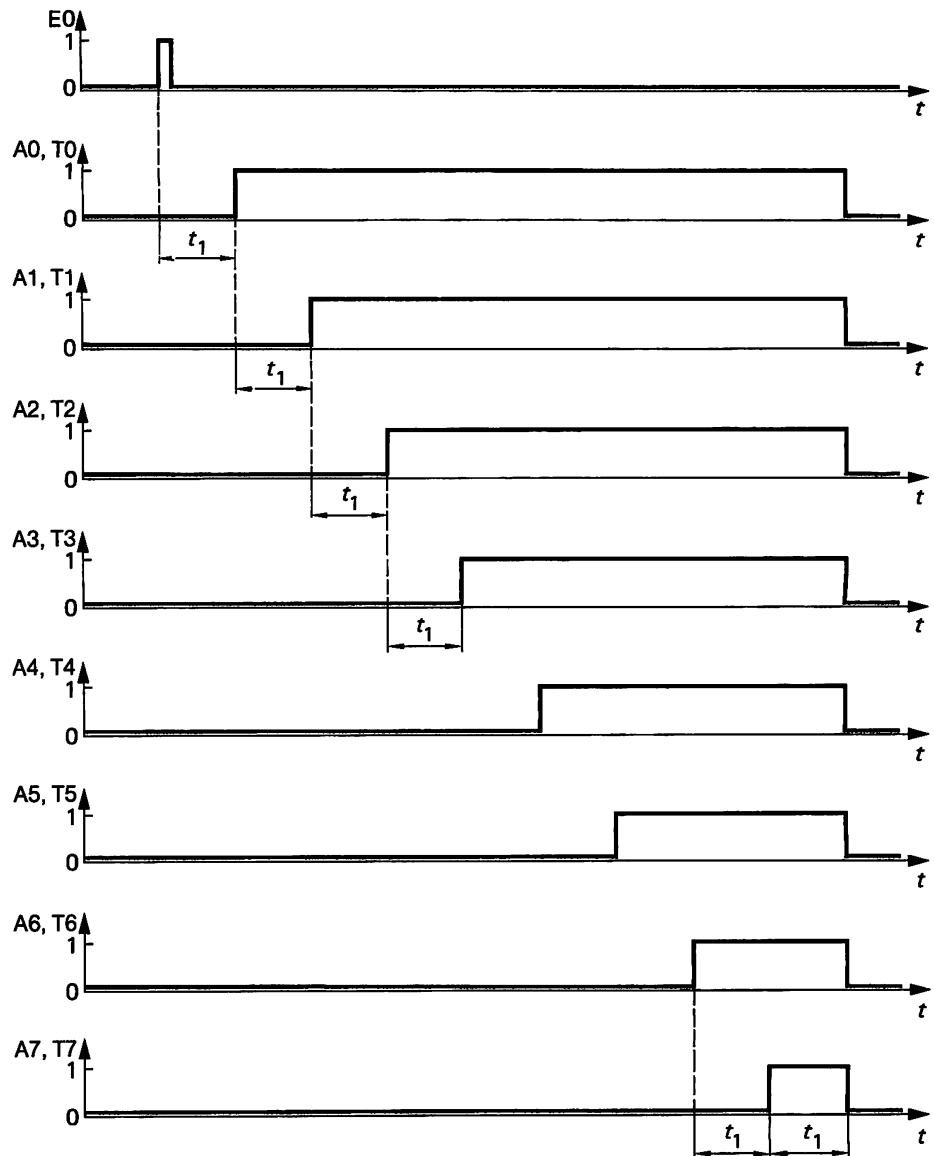


Bild E96.1
Signal-Zeit-Diagramm des Lauf-
lichts, Aufgabe E95.1.

- c) Entwerfen Sie den KOP. Er soll die Umschaltung Hand/Automatik (a/b) enthalten.
- d) Den Programmtest können Sie selbst begutachten. Die Leuchtanzeigen geben Ihnen den Programmablauf wieder.

Einen Lösungsvorschlag finden Sie auf Seite F61.