

Elektronische Steuerungstechnik

Steuern und Steuerungen

Sicherlich ist Ihnen das grundsätzliche Prinzip einer Steuerung bekannt. Trotzdem müssen wir noch einmal auf die Grundlagen eingehen, um für die kommenden Darstellungen einheitliche Begriffe festzulegen. Nehmen Sie an, es wäre der Flüssigkeitsstand in einem Vorratsbehälter zu steuern. Zur Erledigung der Aufgabe müssten Ihnen bestimmte Zustandsgrößen bekannt sein. Dazu gehören beispielsweise: vorhandener Flüssigkeitsstand im Vorratsbehälter, Stellung des Zuflußventils, Zustand der Sicherheitseinrichtungen, Flüssigkeitsdruck in der Zuleitung usw. Diese Informationen erhält die Steuerung als **Eingabesignale**, sie werden auch als Prozeßgrößen bezeichnet.

Nach bestimmten Vorschriften, die für die jeweilige Steuerungsaufgabe immer erst festgelegt werden müssen, erfolgt die Verarbeitung der Eingabesignale. Die Gesamtheit der **Verarbeitungsvorschriften** bildet das **Steuerungsprogramm**. Es ist somit für jede Steuerungsaufgabe ein Programm aufzustellen. Die nach programmgemäßer Verarbeitung der Eingabesignale entstehenden neuen Signale sind die Ausgabesignale. Mit diesen wirkt die Steuerung in den Prozeß, also in die zu steuernde Anlage, ein. In unserem Beispiel wird damit das Einlaßventil geöffnet oder geschlossen.

Jede Steuerungsaufgabe läßt sich verbal (mit Worten) beschreiben, zum besseren Verständnis kann man noch ein **Technologieschema** angeben. Das Technologieschema stellt stark vereinfacht die zu steuernde Anlage dar. Es soll die Funktion der vorhandenen Antriebe erläutern und die Meßstellen zeigen, an denen die Prozeßgrößen erfaßt werden. Eine Liste mit den Daten der Signalgeber, der Meßgeräte und der Antriebe sollte das Schema ergänzen. Bild A 2.1 zeigt als Beispiel das Technologieschema einer Füllstandssteuerung.

Nach vorstehender Darstellung läßt sich der grundsätzliche Ablauf eines Steuerungsvorgangs wie folgt beschreiben:

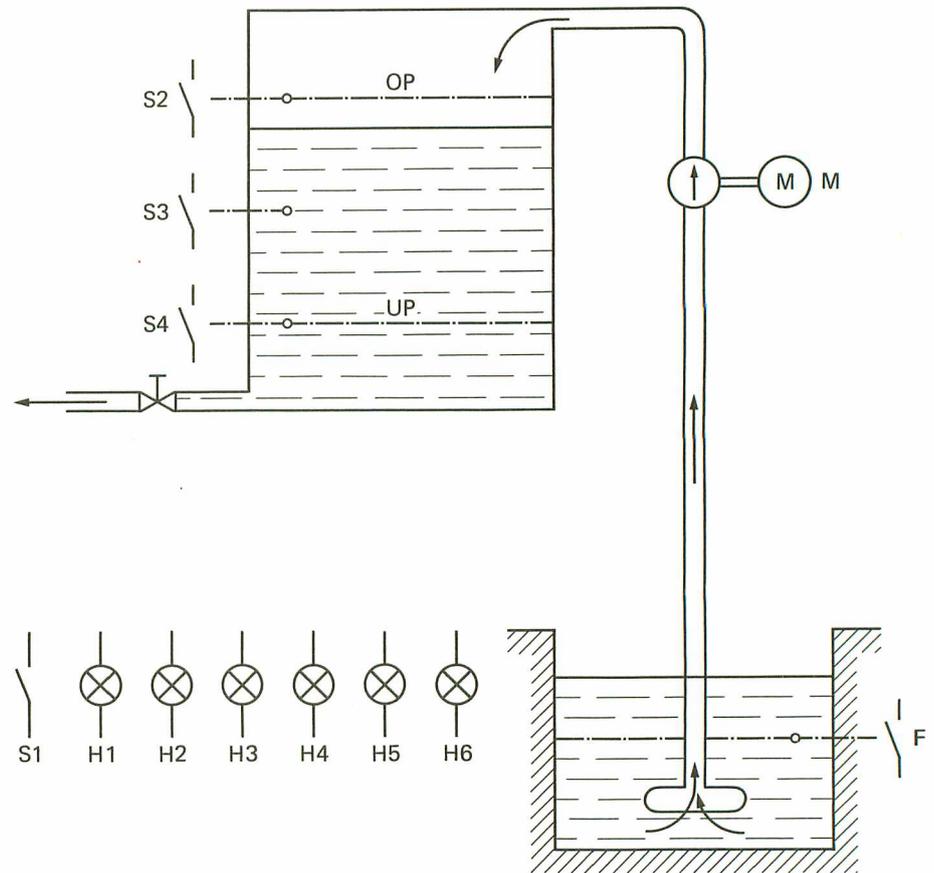
Steuern bedeutet

- Informationen über den Zustand eines Systems (eines Prozesses) entgegennehmen,
- die angenommenen Informationen nach einer Vorschrift (Programm) verarbeiten und
- die neu erzeugten Informationen direkt oder nach Zwischenspeicherung ausgeben.

Die Behälterfüllung erfolgt mit einer elektrisch angetriebenen Wasserpumpe. Bei einer Unterschreitung des unteren Pegels beginnt die Füllung automatisch; bei Erreichen des oberen Pegels wird die Pumpe automatisch abgeschaltet. Die Flüssigkeitspegel „voll“, „halbvoll“ und „leer“ werden ebenfalls angezeigt. Da das Wasser aus einem Brunnen gepumpt wird, ist ein Trockenlaufschutz vorgesehen.

A

2



Betriebsmittel	Funktion	Operand
S1	Hauptschalter	E 0
S2	Schwimmerschalter: OP = „voll“	E 1
S3	Schwimmerschalter: „halbvoll“	E 2
S4	Schwimmerschalter: UP = „leer“	E 3
F	Trockenlaufschutz-Schalter	E 4
H1	Meldung: „Hauptschalter EIN“	A 0
H2	Anzeige: „Behälter voll“	A 1
H3	Anzeige: „Pegel zwischen voll und halbvoll“	A 2
H4	Anzeige: „Pegel zwischen halbvoll und leer“	A 3
H5	Anzeige: „Behälter leer“	A 4
H6	Meldung: „Störung“	A 5
M	Pumpenmotor	A 6

Bild A 2.1
Technologieschema und Zuordnungsliste einer Füllstandssteuerung.

Er schaltet den Pumpenmotor sofort ab, wenn ein bestimmter Wasserpegel im Brunnen unterschritten wird. Mit dem Schalter S1 wird die Anlage in Betrieb genommen – den Betriebszustand zeigt die Meldeleuchte H1 an.

Der Entwurf oder auch die Programmierung der Steuerung wäre nach einer **verbalen Beschreibung** und dem **Technologieschema** möglich. Bei komplexen Steuerungen ginge aber bald die Übersicht verloren. Man stellt deshalb das Zusammenwirken der Signale innerhalb der Steuerung in einem **Blockschema** dar. Bild A 3.1 zeigt als Blockschema eine einfache Steuerung.

Sie sehen zunächst die Blöcke „Steuerungseinrichtung“ und „Steuerstrecke“. Die zu steuernde Anlage bzw. der zu steuernde Prozeß wird als **Steuerstrecke** bezeichnet; sie bildet den einen Block. Im vorher angeführten Beispiel ist der Flüssigkeitsbehälter die Steuerstrecke. Steuerstrecken können u. a. auch Fertigungseinrichtungen, Signalanlagen, motorische Antriebe, Apparate oder Öfen sein.

Der zweite Block umfaßt die **Steuerungseinrichtung**. Dazu gehören alle Geräte, die zur aufgabengemäßen Ausführung des Steuerungsprogramms erforderlich sind. Sie sehen in Bild A 3.1 außerdem verschiedene Pfeile, die Signale darstellen.

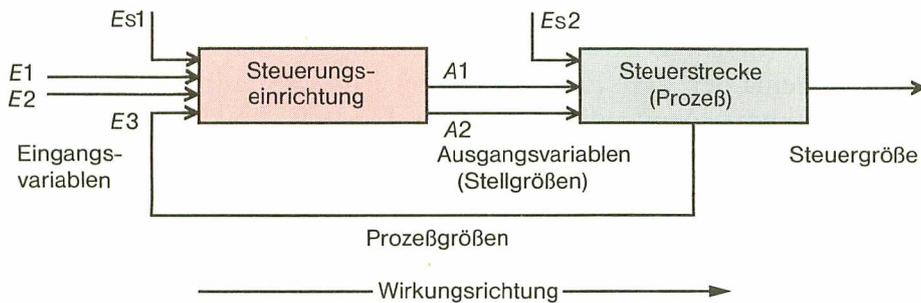


Bild A 3.1
Schema einer Steuerung
E Eingangssignale
Es Signale von Störgrößen
A Ausgangssignale.

Es bedeuten:

- E Eingabesignal oder Eingangsvariable
- A Ausgabesignal oder Ausgangsvariable

Die **Eingangsvariablen E** geben Auskunft über den Zustand der zu steuernden Anlage. Zur Erfassung der Prozeßgrößen sind elektronische Signalgeber erforderlich, man bezeichnet sie auch als **Sensoren**. Die von ihnen erzeugten Signale werden nach dem vorgegebenen Programm verarbeitet.

Als Eingabesignale können auch verschiedene **Störgrößen** wirken, die wir mit **Es** bezeichnen. Darunter versteht man unerwünschte Signale, die den vorprogrammierten Funktionsablauf störend beeinflussen und unterbrechen können. Störgrößen (-signale) entstehen beispielsweise bei einem Leitungsbruch oder durch Einwirkung starker magnetischer Felder (verursacht durch Starkstromleitungen).

Sie betrachten am Fernsehempfänger die Übertragung eines Fußballspiels. Plötzlich klingelt es an der Haustür und Besuch steht unangemeldet vor der Tür. In diesem Fall wäre der Besuch eine typische Störgröße.

Ausgabesignale sind meistens **Stellsignale**, die auf eine Steuerstrecke einer Steuerung einwirken. Die zu steuernde physikalische Größe wird als **Steuergröße** bezeichnet. Im angeführten Beispiel ist die Steuergröße der Flüssigkeitspegel. Zur Untersuchung von Steuerungen werden ausführliche Blockschaltungen entwickelt und die entsprechenden Steuerungssignale eingetragen. Die Pfeilspitzen geben deren Wirkungsrichtung an.

Steuerungskonzepte

Derzeit sind zwei verschiedene Richtungen vertreten. Sie unterscheiden sich in der technischen Realisierung (Verwirklichung) des Steuerungsprogramms. Im Programm ist die Funktion der einzelnen Steuerungsorgane festgelegt. Nach der Art der Programmrealisierung unterscheidet man zwischen

- **verbindungsprogrammierten Steuerung (Abkürzung: VPS)** und
- **speicherprogrammierten Steuerung (Abkürzung: SPS).**

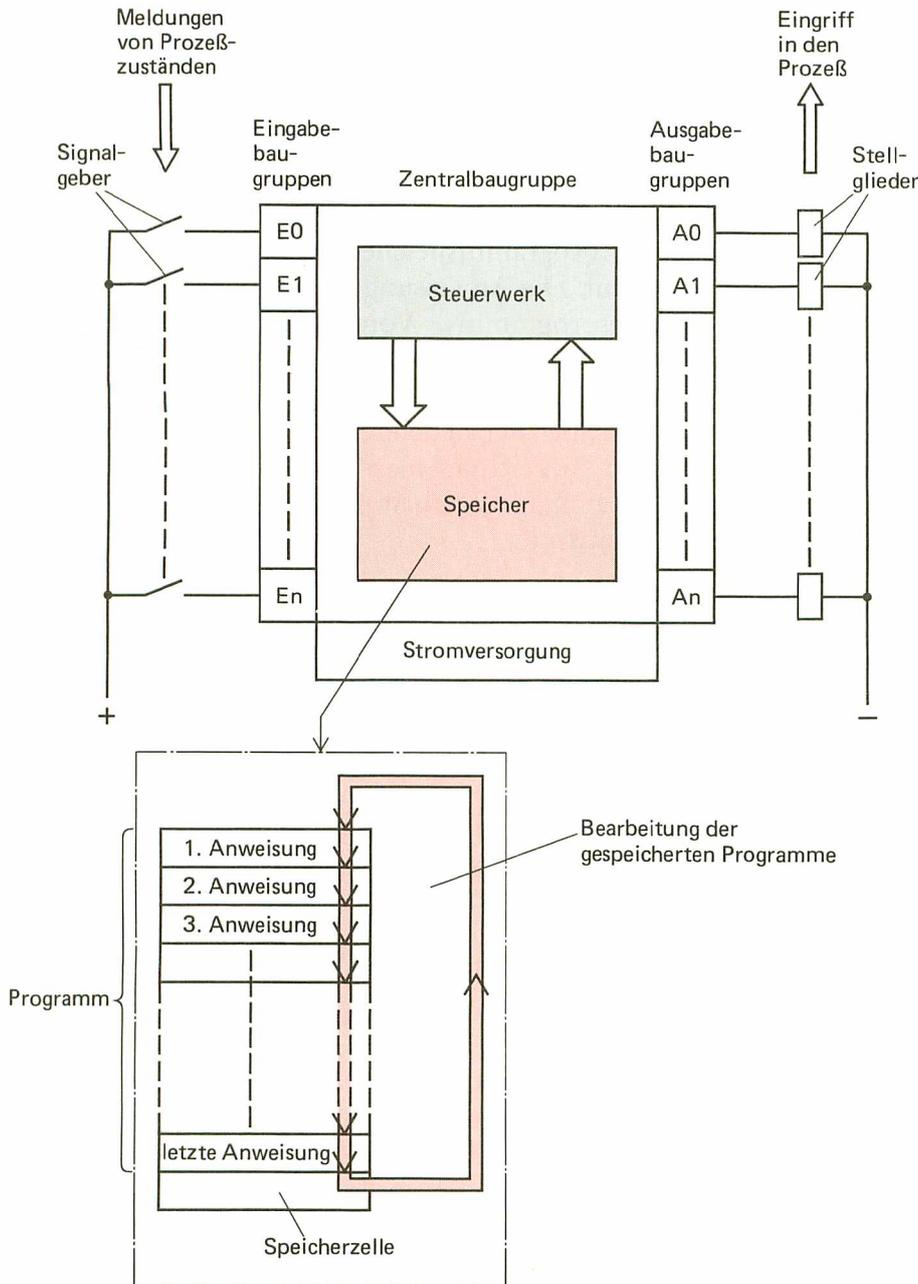
Die VPS ist Ihnen sicher nicht unbekannt. Durch die Verbindung (Verdrahtung) der Funktionsglieder ist der Programmablauf festgelegt. Verbindungsprogrammierte Steuerungen können mit elektromagnetisch betätigten Schaltern (Schütze, Relais) oder elektronischen Schaltern (Schaltkreistechnik) aufgebaut sein. Bild A 4.1 zeigt das Blockscheema der VPS. Eine solche Steuerung ist deshalb nur für eine ganz bestimmte Anwendung ausgelegt. Diese Technik hat unter anderem den Nachteil, daß eine nachträgliche **Anpassung an eine Änderung** des Steuerungsprogramms nur mit einem sehr großen Aufwand vorgenommen werden kann. Oft ist eine Änderung gar nicht mehr möglich, die Steuerung läßt sich nicht mehr verwenden und ein neues Steuerungskonzept mit erneutem Materialaufwand ist notwendig.



Bild A 4.1
Blockscheema einer VPS (VPS =
verbindungsprogrammierte Steuerung).

Bei der SPS dagegen erfordert eine Programmänderung keine Schaltungsänderung innerhalb der Steuerungseinrichtung. Der Programmierer teilt der Steuerung das Programm in Form von Anweisungen mit. Diese programmierten Anweisungen bezeichnen wir auch als Software. Das Programm wird über eine Tastatur oder über ein Programmiergerät in den Programmspeicher der Steuerung eingegeben. Der Speicher kann wieder gelöscht oder umprogrammiert werden. Man nennt deshalb die SPS auch eine **frei programmierbare Steuerung**. Diese freie Programmierbarkeit ist natürlich nur in einem gewissen Umfang möglich. Der Rahmen ist von der vorhandenen Hardware, also dem gerätetechnischen Teil der Steuerungseinrichtung, abgesteckt.

Bild A 5.1 zeigt das Schema einer SPS. Die Steuerungseinrichtung wird vielfach auch als Automatisierungsgerät bezeichnet. Das **Automati-**



A
5

Bild A 5.1
Schema einer SPS
(SPS = speicherprogrammierte Steuerung).

steuerungsgerät besteht im wesentlichen aus einer Zentralbaugruppe mit Steuerwerk und Programmspeicher, den Eingabe- und Ausgabebaugruppen und einer Stromversorgung. Zusätzlich sind **Signalgeber** zur Meldung von Prozeßzuständen sowie Stellgeräte zum Eingriff in den zu steuernden Prozeß erforderlich. Verschiedentlich sind ausgangseitig auch Leuchtmelder angeschlossen. Die von den Signalgebern kommenden Signalspannungen werden auf die Anschlußleiste der Eingabebaugruppen geschaltet. In der Zentralbaugruppe bearbeitet das Steuerwerk das im Speicher stehende Programm. Es fragt dabei die einzelnen Eingänge auf den anliegenden Signalzustand ab.

In Abhängigkeit von den **Eingangssignalzuständen** und vom Steuerungsprogramm im Speicher entstehen Anweisungen, welche als Signalzustände auf die Ausgänge der Ausgabebaugruppen zu schalten sind.

Entsprechend diesen **Ausgangssignalzuständen** werden die angeschlossenen Stellgeräte und Leuchtmelder eingeschaltet oder ausgeschaltet.

A

6

Der im Bild A 5.1 herausgezeichnete Speicher soll Ihnen folgendes zeigen:

Eine Speicherzelle des Programmspeichers nimmt immer nur eine Steuerungsanweisung auf. Die Anweisung ist die kleinste selbständige Einheit des Steuerungsprogramms. Vom Steuerwerk werden die Anweisungen nacheinander bearbeitet. Nach der Bearbeitung der letzten im Speicher stehenden Anweisung beginnt das Steuerwerk wieder mit der ersten im Speicher stehenden Anweisung. Da sich diese Bearbeitung wiederholt, bezeichnet man sie als **zyklische Bearbeitung**. Die Zeit für die einmalige Bearbeitung aller Anweisungen eines Programms ist die **Zykluszeit**.

Ein Betrieb, der mit modernen und teuren Fertigungsanlagen ausgerüstet ist, benötigt auch eine moderne und flexible Fertigungssteuerung. Mit einer SPS lassen sich u. a.

- wertvolle Maschinen und Anlagen optimal ausnutzen,
- Neukonstruktionen durch Programmänderung schnell einführen,
- kundenspezifische Wünsche berücksichtigen und
- eine kleine Lagerhaltung realisieren.

Ursprünglich sollte die SPS nur die **konventionelle Schützensteuerung** ersetzen. Moderne SPS sind aber bereits soweit entwickelt, daß sie teilweise mit teuren **Prozeßrechenanlagen** konkurrieren können.

Eines der Haupthindernisse bei der Umstellung von herkömmlichen Schützen- oder Relais-Steuerungen liegt darin, daß viele Mitarbeiter der Betriebe, vor allem auf der Ebene der Meister und Facharbeiter, mit der Elektronik und Datenverarbeitung nur wenig vertraut sind. Die Betriebe sind deshalb vielfach auf fremde Spezialisten angewiesen, die wiederum nur geringe Kenntnisse von der innerbetrieblichen Anwendung der jeweiligen SPS haben.

Dieser Lehrgang soll deshalb auch dem Fachpersonal helfen, vorhandene Lücken in elektronischen und programmiertechnischen Kenntnissen zu schließen. Sie werden sehen, daß zur Anwendung der SPS (nur um diese geht es im Lehrgang), keine speziellen Kenntnisse in Computertechnik, Mikroelektronik oder Digitaltechnik erforderlich sind. Selbstverständlich benötigt man auch hier ein bestimmtes Basiswissen, das wir Ihnen aber vermitteln werden. In der Hauptsache werden zur **SPS-Anwendung** genaue technologische Informationen über den zu steuernden Vorgang gebraucht; diese waren aber auch schon bei Schützensteuerungen erforderlich.

Vorab wollen wir Ihnen nun noch einige Kurzinformationen zur SPS geben, später besprechen wir die Funktionsgruppen dann ausführlicher.

Zur **Zentralbaugruppe der SPS** gehören ein Mikroprozessor und elektronische Speicher. Das Steuerungsprogramm, das bei einer VPS durch die **Verdrahtung** festgelegt ist, wird bei der SPS in eine eigene Sprache, die der Mikroprozessor versteht, übersetzt. Das Programm enthält im wesentlichen Anweisungen, wie die Eingangssignale zu bestimmten Ausgangssignalen zu verarbeiten sind. Die Programmeingabe in den Programmspeicher kann beispielsweise über eine Tastatur oder über ein besonderes Programmiergerät erfolgen.

Als Eingangssignale erhält die SPS vorwiegend **Prozeßinformationen**. Das sind Signale, die den Zustand der zu steuernden Anlage melden; sie werden mit Sensoren erzeugt. Sensoren sind in der Regel kontaktlose Signalgeber. Dazu gehören: optoelektronische Bauelemente, Näherungsinitiatoren, Endschalter und andere schaltende Elemente. Die Ausgangssignale der SPS können **Stell- oder Meldesignale** sein. Die Stellsignale betätigen Ventile, Leistungsschütze, Kupplungen und andere Stellglieder. Über die Stellglieder greift die Steuerung in die Steuerstrecke ein. Die Meldesignale geben nur eine Information aus; sie zeigen beispielsweise einen bestimmten Prozeßzustand an.

Die Zahl der Signalein- und -ausgänge, die Aufnahmefähigkeit des Programmspeichers und die programmierbaren Anweisungen legen den Einsatzbereich der speicherprogrammierbaren Steuerungen fest.

In der Verarbeitung der Eingangssignale besteht zwischen der VPS und der SPS ein großer Unterschied. Bei der VPS beeinflussen Zustandsänderungen der Eingangssignale sofort den Zustand der Ausgangssignale. Wir sprechen von einer **parallelen Signalverarbeitung**. Die SPS arbeitet jedoch die Anweisungen im Programmspeicher nacheinander ab. Wir bezeichnen diese Arbeitsweise als **seriell**. Die serielle Abarbeitung des Programms dauert natürlich viel länger als die parallele Verarbeitung. Nur wegen der großen Schaltgeschwindigkeit der SPS bleibt die Verlängerung der Programmbearbeitung in tragbaren Grenzen.

Zusammenfassung

Eine zu steuernde Anlage nennt man Steuerstrecke. In der Steuerungseinrichtung werden die Eingangssignale nach bestimmten Vorschriften zu Ausgangssignalen verarbeitet. Ausgangssignale sind meist Stellsignale zum Betätigen von Stellgliedern. Über die betätigten Stellglieder wirkt die Steuerungseinrichtung auf die Steuerstrecke ein.

Dem Konzept nach ist zwischen verbindungsprogrammierten (VPS) und speicherprogrammierten Steuerungen (SPS) zu unterscheiden. Bei der VPS liegt das Steuerungsprogramm in der Verdrahtung bzw. Verbindung der Bauelemente. Die SPS benötigt für das programmgemäße Arbeiten keine Verdrahtung. Das eingegebene Steuerungsprogramm ist in einem Speicher abgelegt.

VPS verarbeiten die Eingangssignale parallel; SPS verarbeiten die Eingangssignale seriell.

Aufgaben A 8.1

1. Beschreiben Sie den grundsätzlichen Ablauf eines Steuerungsvorgangs.
2. Was verstehen Sie unter einer Steuerungseinrichtung?
3. Welche zwei Möglichkeiten gibt es grundsätzlich, das Programm des Steuerungsablaufs in einer Steuerungseinrichtung technisch zu realisieren?
4. In welcher Steuerungsart wird das Steuerungsprogramm seriell abgearbeitet? Was ist unter einem seriellen Betrieb zu verstehen?
5. Nennen Sie die wichtigsten Baugruppen der SPS.

Die Lösungen zu diesen Fragen finden Sie auf Seite F 3.

A**8**

Steuerungsstrukturen

Die elektronische Steuerungstechnik hat sich in den beiden vergangenen Jahrzehnten enorm fortentwickelt. Die Gründe hierfür liegen hauptsächlich in der Anwendung der Digitaltechnik und der digitalen Bauelemente. Letztere eignen sich sowohl zur **Verknüpfung** als auch zur **Speicherung** großer Datenmengen. Hinzu kommt, daß bereits viele von der Steuerung zu verarbeitende Prozeßgrößen binär oder digital zur Verfügung stehen.

Die verschiedenen Formen der möglichen Steuersignale sowie eine große Zahl unterschiedlicher elektromechanischer und elektronischer Funktionsbausteine ergeben vielfältige Möglichkeiten, eine Steuerungsaufgabe zu lösen und technisch zu realisieren.

Man hat nun versucht, die Lösungsmöglichkeiten für die verschiedenen Steuerungsaufgaben zu **standardisieren**. Mit Hilfe der Standards kann man bereits aus der formulierten Aufgabenstellung auf ein geeignetes Steuerungsschema schließen. Die sich dann anbietenden Lösungsmöglichkeiten bezeichnen wir als **Steuerungsstrukturen**. Diese Steuerungsstrukturen für eine Steuerungsaufgabe lassen sich nach bestimmten Merkmalen ordnen, z. B. nach der Art der

- Informationsdarstellung,
- Signalverarbeitung,
- Programmverwirklichung,
- Aufbauorganisation.

Diese Merkmale sind nicht nur auf die VPS bezogen. Viele Festlegungen haben auch für die SPS Gültigkeit. Sie müssen immer bedenken, daß wir bei der SPS schaltungstechnische Hardware-Lösungen durch **Programme (Software)** nachbilden müssen. Bezüglich der Gesetzmäßigkeit in der Verknüpfung von Steuerungssignalen gibt es keine grundlegenden Unterschiede zwischen VPS und SPS.

Jetzt wollen wir die interessierenden Merkmale von Steuerungsstrukturen etwas ausführlicher betrachten.

Informationsdarstellung

Signal

Bei der Beschreibung von Steuerungsvorgängen wird oft mit den Begriffen Signal und Nachricht gearbeitet. Hierzu müssen wir Ihnen einige Erläuterungen geben.

Ein **Signal** ist die physikalische Darstellung einer Nachricht.

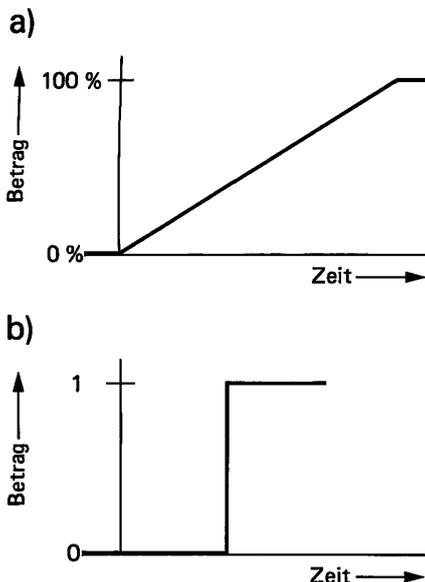
Demnach könnte eine Nachricht wiedergegeben werden durch den Zeigerausschlag eines Meßgeräts, durch Ansteuerung optischer Anzeigeeinheiten oder durch bestimmte Werte einer elektrischen Spannung. Zur Erzeugung eines Signals ist eine Hilfsenergie erforderlich. Damit läßt sich dann das Signal technisch verwirklichen. Wir benutzen als **Hilfsenergie** eine elektrische Gleichspannung. Je nach der Art des zeitlichen Signalverlaufs unterscheiden wir zwischen **analogen** und **binären Signalen**.

Analoge Signalform

Das analoge Signal (Bild A 10.1a) ist ein stetes Abbild der zu verarbeitenden Größe. Es bildet die Steuerungsgröße nach Betrag und Vorzeichen in Abhängigkeit von der Zeit kontinuierlich nach. Bezogen auf die Drehzahlmessung mit einem Tachogenerator als Beispiel heißt das:

- Die vom Tachogenerator abgegebene Gleichspannung ist ein Maß für die Drehzahl, sie ist der Betrag der Steuerungsgröße.
- Die Polarität der Spannung gibt die Drehrichtung wieder, sie ist das Vorzeichen der Steuerungsgröße.
- Betrag und Vorzeichen der Gleichspannung werden während der Betriebszeit ausgewertet, sie dienen dann zur Steuerung der Drehzahl.

Analoge Signale kann die SPS nicht direkt verarbeiten. Sie müssen erst in eine andere Signalform gebracht werden.



Binäre Signalform

In Natur und Technik gibt es außer den stetig veränderlichen Größen auch solche, die nur zwei stabile Betriebszustände annehmen können. Die dazugehörigen Signale bezeichnen wir als „**binäre Signale**“ (Bild A 10.1b). Binär heißt beispielsweise, daß ein Stromkreis oder Kontakt nur geöffnet oder geschlossen, ein Relais oder Schütz nur angezogen oder abgefallen, eine Diode oder ein Transistor nur leitend oder gesperrt sein können. Das Darstellen zweiwertiger oder binär veränderlicher Größen erfolgt mit dem binären Signal. Die beiden möglichen Signalzustände bezeichnen wir mit „1“ oder „0“.

Bild A 10.1
Signalformen
a) analoge Signalform
b) binäre Signalform.

Zustand 1 bedeutet Signal vorhanden,
Zustand 0 bedeutet Signal nicht vorhanden.

Physikalisch bilden wir diese Zustände durch eine elektrische Gleichspannung nach. In der Praxis haben die schaltenden elektronischen Bauelemente bei den Zuständen 1 und 0 keine einheitlichen Spannungswerte. Die Eingangsvariablen der TTL-Bausteine (TTL = Transistor-Transistor-Logik) – unser System enthält diese Bausteine – müssen beispielsweise innerhalb folgender Spannungspegel liegen:

- 0-Zustand: 0 bis maximal 0,8 V
- 1-Zustand: minimal 2,0 V bis maximal 5,0 V

Diese Spannungspegel werden auch als **Logikpegel** (Bild A11.1) bezeichnet. Ist der Spannungswert für den 1-Zustand positiver als für den 0-Zustand, so gehört das Bausteinsystem zur positiven Logik. Für diese gilt folgende Zuordnung:

H (high)-Pegel entspricht Zustand 1,
L (low)-Pegel entspricht Zustand 0.

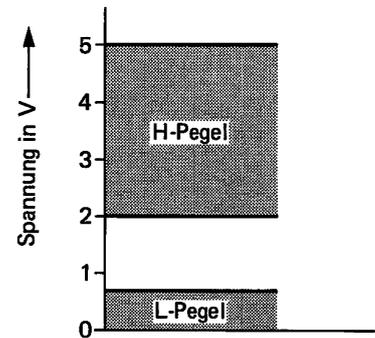


Bild A11.1
Logikpegel.

Darstellung binärer Signalzustände

Tabelle A11.1

Signalgeber	Signalzustand	
	1	0
Kontakt	geschlossen	geöffnet
Relais	angezogen	abgefallen
Spannung	+ 3,2 V	0,2 V

Ein Element, z. B. ein Kontakt, mit dem sich die Signalzustände 1 und 0 technisch verwirklichen lassen, wird als **Binärstelle** bezeichnet. Der Wechsel des Signalzustands einer Binärstelle stellt die Entscheidung zwischen zwei möglichen Werten dar. Diese Entscheidung ist eine Binärentscheidung. Sie erhält im technischen Sprachgebrauch die Einheit **Bit** (Bit = Abkürzung für binary digit).

Ein Bit ist die kleinste mögliche Informationseinheit, sie kann nur den Wert 0 oder 1 haben.

Unsere NDR-SPS verarbeitet **elektrische Binärsignale**. Wir haben diese Signalform deshalb ausführlich dargestellt.

A

12

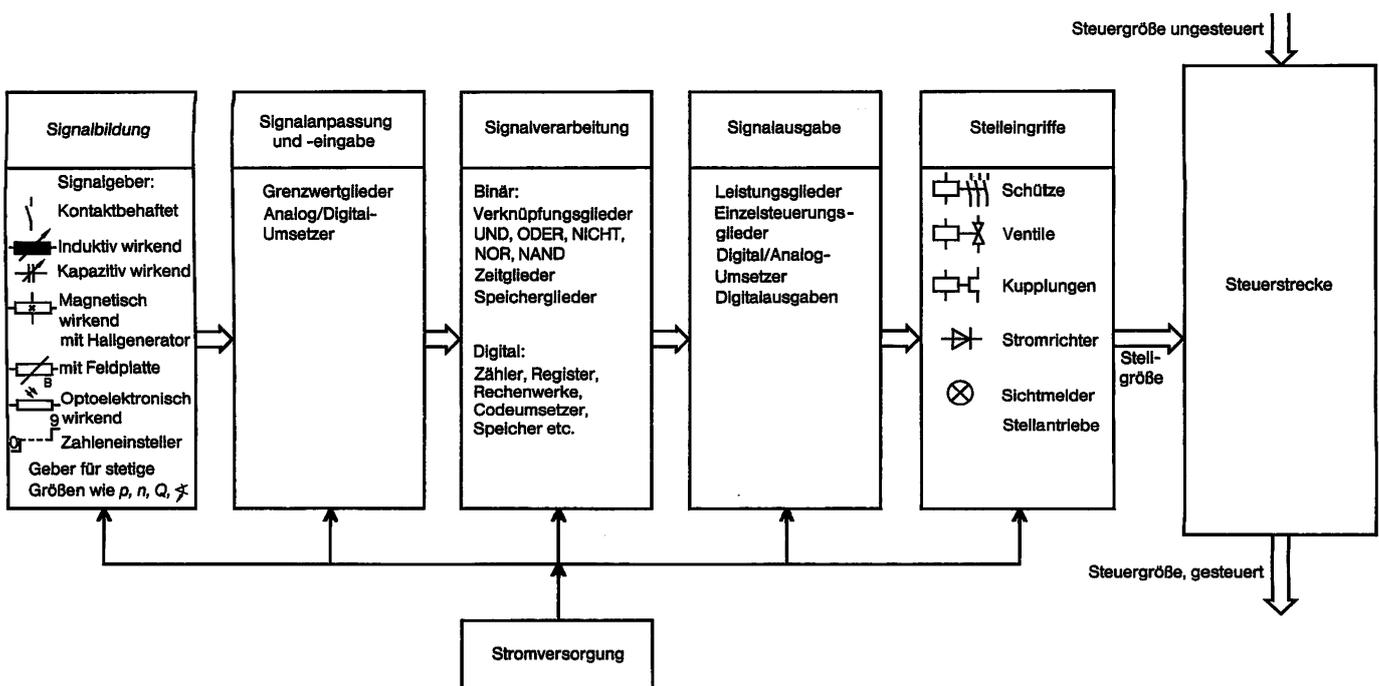
Digitale Informationsdarstellung

Digital ist vom englischen Wort „digit“, zu deutsch „Ziffer“ oder „Zahl“ abgeleitet. Mit diesem Begriff wird die zahlenmäßige, also die **numerische Darstellung** von Informationen bezeichnet. Eine sich stetig ändernde Größe wird digital so dargestellt, daß man ihren Verlauf in kleinen, gleichgroßen Schritten abtastet und die zugehörigen Größenwerte als Ziffern, z.B. von Dualzahlen, ausgibt. Ein **digitales Signal** besteht aus einer bestimmten Anzahl von **Binärzeichen**. Für die Anordnung der Binärzeichen gelten bestimmte Vorschriften. Darüber werden wir Ihnen im Abschnitt „Codes“ die Grundlagen vorstellen. Zur Verarbeitung von Zahlenwerten (z.B. bei Zähl- und Rechenvorgängen) sind digital arbeitende Steuerungen erforderlich, die einen **Automatisierungsgrad** ermöglichen, der über das mit binären Steuerungen erreichbare Maß hinausgeht.

Zusammenfassung

Ein Signal ist die physikalische Darstellung einer Nachricht. Wir benutzen die Gleichspannung zur Signaldarstellung. Bei der **analogen Signalform** ändert sich die physikalische Größe in Abhängigkeit von der Zeit stetig. **Binäre Signale** haben nur zwei stabile Betriebszustände. Man ordnet den beiden Betriebszuständen die Symbole „1“ (Signal vorhanden) und „0“ (Signal nicht vorhanden) zu. Statt der 1-0-Zustände werden, speziell auf die Hardware bezogen, auch **Logikpegel** angegeben. Bei positiver Logik gilt die Zuordnung „1 = H(High)“ und „0 = L(Low)“. Werden Binärsignale nach bestimmten Vorschriften kombiniert, kann man z.B. Ziffern und Zahlen verschlüsseln. Wir erhalten eine digitale Information.

Bild A 12.1
Struktur einer binären oder digitalen Steuerung.



Signalverarbeitung

Die folgenden Abschnitte müssen Sie besonders aufmerksam studieren. Wir besprechen die **grundsätzlichen Strukturen** der verschiedenen Steuerungsarten. Diese Strukturen finden Sie in späteren Abschnitten innerhalb der Programme für Steuerungsaufgaben wieder. Die Einteilung entspricht der nach **DIN 19237**. Wir beschäftigen uns nur mit Steuerungen, die ausschließlich mit binären Signalen arbeiten. Das Bild A 12.1 zeigt die Struktur einer binären oder digitalen elektronischen Steuerung. Die einzelnen **Funktionsgruppen** sind wirkungsmäßig zusammengestellt.

Verknüpfungssteuerung

Dazu zählt auch die als **Verriegelungs- und Führungssteuerung** bezeichnete Schaltung. Diese Steuerung ist dadurch gekennzeichnet, daß den Werten der Eingabesignale immer ganz bestimmte Zustände der Ausgabesignale zugeordnet sind. Die Verarbeitung der Eingabesignale erfolgt durch **logische Verknüpfungen**, z.B. mit UND-, ODER- und NICHT-Verknüpfungen. Die Lösung soll möglichst wenige Zeit- und Speicherfunktionen enthalten. Die verschiedenen Verknüpfungs- und Speicherfunktionen stellen wir Ihnen im Laufe des Lehrgangs vor.

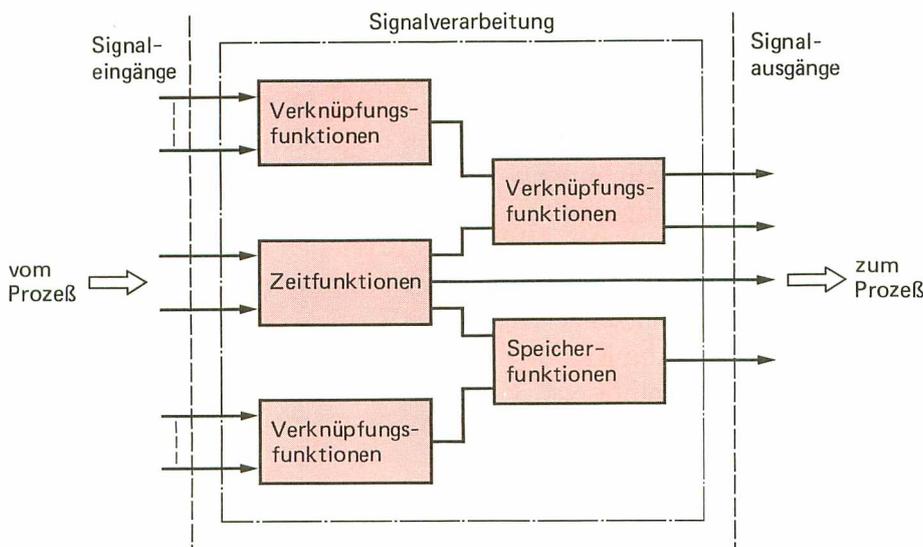


Bild A 13.1
Signalverarbeitung in der Verknüpfungssteuerung.

Typisch für diese Struktur ist, daß die am Ausgang entstehenden Steuerbefehle „spontan“ den Eingabedaten folgen. Zwischen den Ein- und Ausgabesignalen gilt eine statische Zuordnung. Bild A 13.1 zeigt als Beispiel die **Struktur einer Verknüpfungssteuerung** im Blockschema. Diese Struktur kann durch eine mathematische Funktion, die sogenannte **Schaltfunktion**, beschrieben werden. Das Arbeiten mit Schaltfunktionen wird im Fachteil „Digitaltechnik“ ausführlich geübt.

Ablaufsteuerung

Bei vielen Steuerungsaufgaben dominiert nicht wie bei der Verknüpfungssteuerung die statische Zuordnung der Ein- und Ausgabesignale, sondern deren **zeitliche Folge**. Deshalb wird bei industriellen Anwendungen die Ablaufsteuerung bevorzugt eingesetzt. Die **Ablaufsteuerung** wird auch als Folge-, Programm- oder Taktsteuerung bezeichnet.

Ablaufsteuerungen zählen zu den binären Steuerungen. Zum Entwurf der Schaltung und zur Darstellung der Funktion können deshalb die gleichen Schaltzeichen wie bei Verknüpfungssteuerungen Verwendung finden. Sie werden ergänzt durch einige Funktionssymbole, die das schrittweise Abarbeiten des Steuerungsprogramms wiedergeben.

Ablaufsteuerungen sind durch ein zwangsläufig schrittweises Abarbeiten des Programms gekennzeichnet. Das Weiterschalten von einem Schritt zum programmgemäß folgenden setzt die Erfüllung von **Weiterschaltbedingungen** voraus. Diese können prozeßabhängig oder zeitabhängig sein.

Prozeßgeführte Ablaufsteuerung

Bei der Steuerung von Vorgängen, die in einer ganz bestimmten Reihenfolge ablaufen müssen, darf beispielsweise der nächste Vorgang erst eingeleitet werden, wenn der vorangehende abgeschlossen ist. Der Abschluß wird der Steuerung mit einem Signalgeber gemeldet (**Rückmeldung**). Es liegt also hier eine Steuerung vor, bei der der Steuerungsablauf zwangsläufig schrittweise erfolgt. Das Weiterschalten von einem Schritt auf den programmgemäß folgenden ist an die Erfüllung von Weiterschaltbedingungen gebunden. Weiterschaltbedingungen können prozeßabhängig sein. Die Rückmeldungen kommen aus der zu steuernden Anlage; sie melden das Erreichen bestimmter Prozeßzustände wie z. B. das Erreichen einer bestimmten Temperatur, einer Lage, eines Flüssigkeitspegels usw. Die Weiterschaltbedingung ist erst dann erfüllt, wenn die Rückmeldung aus dem Prozeß, daß der zuvor erteilte Befehl ausgeführt ist, eintrifft.

Zeitgeführte Ablaufsteuerung

Den Ablauf des Steuerungsvorgangs bestimmt ein fest **vorgegebener Zeitplan**. Ein typisches Beispiel ist die Verkehrsampelsteuerung. Mit Taktsignalen erfolgt das Weiterschalten von einem Schritt zum programmgemäß folgenden.

Ablaufsteuerungen sind wie Verknüpfungssteuerungen binäre Steuerungen. Die kleinste Funktionseinheit ist die **Schrittfunktion**. Meh-

re in Kette angeordnete Schritte ergeben die Ablaufsteuerung. Das Schema hierfür zeigt Bild A 15.1. Die Schritte werden bei Erfüllung der Weberschaltbedingungen nacheinander gesetzt. Es ist immer nur ein Schritt wirksam. Der jeweils gesetzte Schritt n gibt Befehle aus. Gleichzeitig setzt Schritt n den vorher wirksamen Schritt $(n-1)$ zurück und bereitet den folgenden Schritt $(n+1)$ vor.

Die einfache Störungserkennung ist mit ein Grund dafür, daß anstelle von Verknüpfungssteuerungen bevorzugt Ablaufsteuerungen eingesetzt werden. Bei einer Störung bleibt nämlich die Steuerung bei dem zuletzt ausgeführten Programmschritt stehen. Die Fehlersuche beschränkt sich deshalb nur auf die Funktion dieses Schrittes.

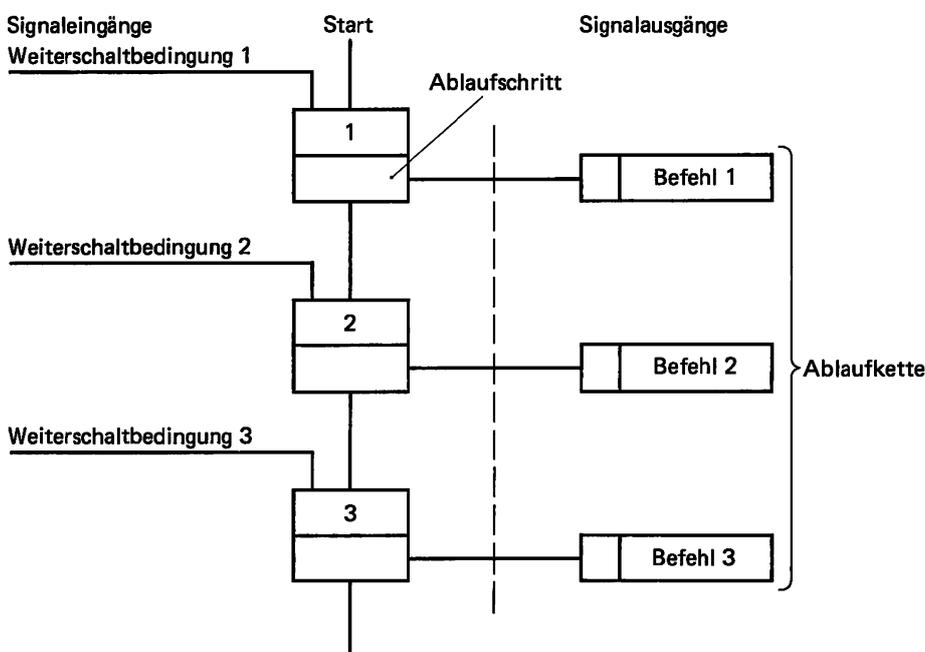


Bild A 15.1
Schema einer Ablaufsteuerung mit Schrittsymbolen. Die Weberschaltbedingungen können prozeß- oder zeitabhängig sein.

Zusammenfassung

Bei Verknüpfungssteuerungen sind zu jedem beliebigen Zeitpunkt den Eingabesignalen bestimmte Ausgabesignale zugeordnet. Die Informationsverarbeitung erfolgt vorwiegend mit Verknüpfungsfunktionen.

Bei der Ablaufsteuerung werden die Schritte **nacheinander** in strenger Reihenfolge durchlaufen. Es ist immer nur ein Schritt der Ablaufkette gesetzt. Das Schalten zum nächsten Schritt ist an die Erfüllung von Weberschaltbedingungen gebunden. Diese können prozeßabhängig oder zeitabhängig sein. Die gesetzten Schritte geben **Befehle (Anweisungen)** aus.

Zu den Merkmalen der Strukturen gehören noch die **Programmverwirklichung** und die **Aufbauorganisation**. Beide Punkte können wir kurz abhandeln. Über die Programmverwirklichung haben wir schon mehrfach gesprochen. Es gibt die verbindungsprogrammierten und die speicherprogrammierten Steuerungen. Thema dieses Lehr-

gangs ist ausschließlich die **speicherprogrammierte Steuerung**. Die Aufbauorganisation gibt Empfehlungen über die Anordnung der Steuerungsblöcke innerhalb der Gesamtsteuerung. Dieser Punkt hat kaum einen Einfluß auf die Programmgestaltung. Natürlich sind auch bei der Programmgestaltung bestimmte Regeln zu beachten. Darüber erhalten Sie in den Übungsaufgaben entsprechende Hinweise.

Aufgaben A16.1

1. Was versteht man unter einem Signal?
2. Geben Sie für die positive Logik die Zuordnung der Pegel H und L zu den Signalzuständen 1 und 0 an.
3. Nennen Sie die Merkmale einer Verknüpfungssteuerung.
4. Wie erfolgt die Signalverarbeitung bei der Ablaufsteuerung?
5. Was versteht man unter einer Weiterschaltbedingung?
6. Tragen Sie die Signalform (analog, binär oder digital) des betreffenden Signalgebers in die folgende Tabelle ein:

Tabelle A16.1

Darstellung durch	Signalform
Zeigerausschlag eines Spannungsmessers	
Aufleuchten einer Signallampe, die über ein Relais geschaltet wird	
Rollenzählwerk am Tonbandgerät	
Anzeige des Taschenrechners	
Aussteuerungsanzeige mit einem Leuchtbalken	

Die Lösungen finden Sie auf Seite F3.

Programmierung der speicherprogrammierbaren Steuerungen

Über die grundsätzliche Arbeitsweise der SPS haben Sie schon das Wichtigste in Lehrbrief 1 erfahren. Verschiedene Baugruppen, die auch für den NDR-Computer als SPS von Bedeutung sind, werden wir noch ausführlich besprechen. Außerdem sollen Sie in diesem Lehrgang auch Informationen über die industriell eingesetzten, speicherprogrammierten Steuerungen erhalten.

Bevor wir mit dem NDR-Computer speicherprogrammierbare Steuerungen entwickeln, geben wir Ihnen noch einige allgemeingültige Informationen. Wir werden sie anschließend auch auf den NDR-Computer beziehen.

Die Anwender der SPS verlangen von den Steuerungsherstellern, daß zur Programmierung der SPS keine speziellen Kenntnisse aus der Datenverarbeitung erforderlich sind. Die Programmierung sollte der üblichen Beschreibung von Steuerungsaufgaben angepaßt sein. Das Programmieren muß demnach unmittelbar möglich sein nach

- dem Kontaktplan oder
- einer Schaltfunktion oder
- einer Anweisungsliste oder
- einem Funktionsplan.

In Bild A 17.1 sind die angeführten Möglichkeiten wiedergegeben. Zur Aufnahme des Programms ist ein **Programmspeicher** erforderlich. Darüber erhalten Sie in einem der folgenden Abschnitte noch ausführliche Informationen. Die Programmeingabe in den Speicher erfolgt

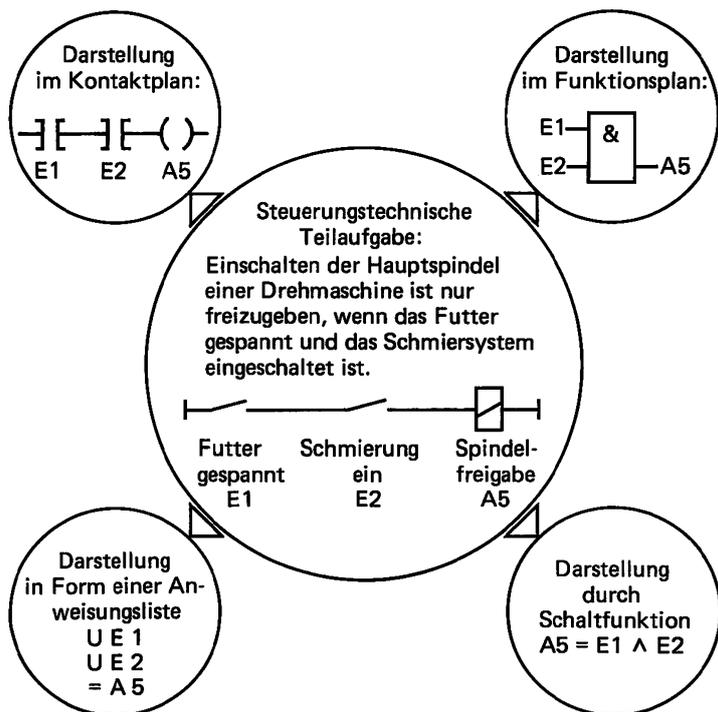


Bild A 17.1
 Programmierverfahren für speicherprogrammierte Steuerungen (Siemens AG).

A

18

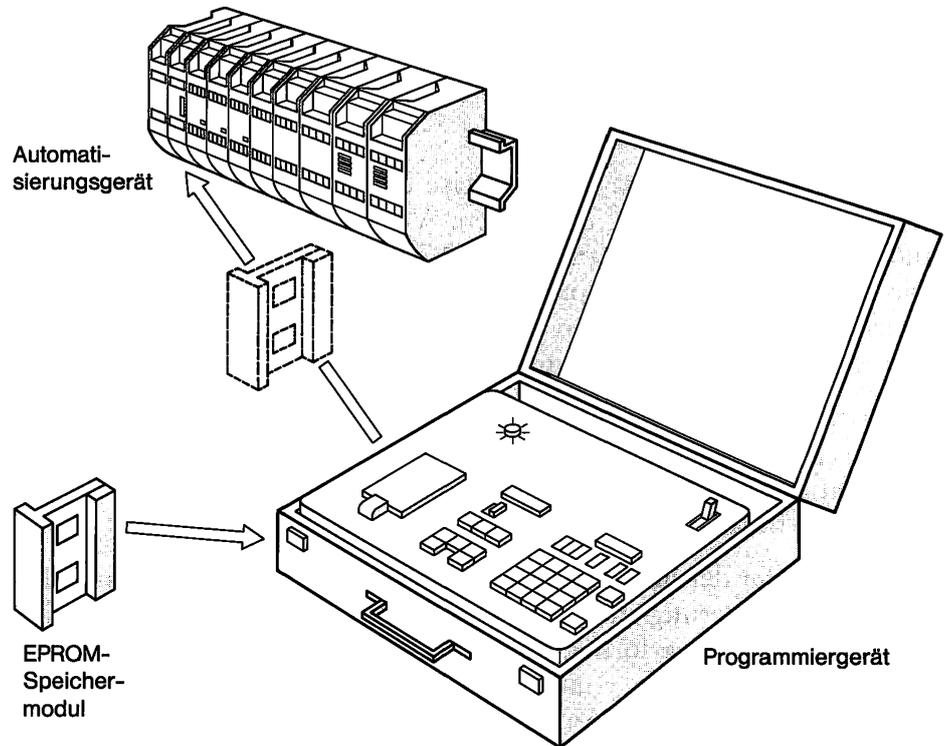


Bild A 18.1
Baugruppen und Geräte einer speicherprogrammierten Steuerung (Siemens AG).

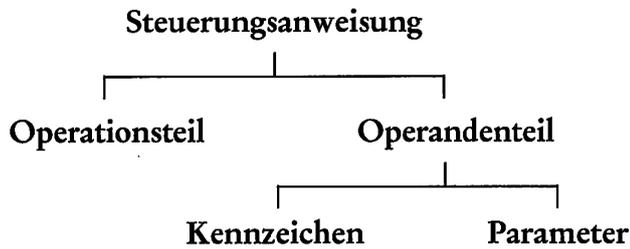
mit Hilfe eines **Programmiergeräts** über eine Tastatur. Die Tasten sind mit Programmiersymbolen belegt, die in der Programmieranleitung des Herstellers festgelegt sind. Im Programmiergerät werden die eingetasteten Informationen in digitale Signale umgesetzt. Der elektronische Teil verarbeitet nur binär codierte Informationen.

Die zur Programmierung erforderliche Geräteausrüstung zeigt das Bild A 18.1. Im Programmiergerät wird ein Halbleiterspeicher in der Sprache programmiert, die die SPS „versteht“. Man bezeichnet diese Sprache als **Maschinensprache**. Im Bild A 18.1 ist die modular aufgebaute speicherprogrammierte Steuerung als **Automatisierungsgerät** bezeichnet.

Programmdarstellung

Zum Verständnis der Funktion einer speicherprogrammierten Steuerung ist schon jetzt ein kleiner Einstieg in die Programmdarstellung und Programmierung erforderlich. Uns interessiert nur das Erstellen und Verarbeiten der **Anwenderprogramme**. Darunter sind, vereinfacht ausgedrückt, alle Anweisungen und Vereinbarungen zu verstehen, mit denen die zu steuernde Anlage nach der Aufgabenstellung beeinflusst wird.

Die kleinste selbständige Einheit eines Programms wird als **Steuerungsanweisung** bezeichnet. Sie stellt eine Arbeitsvorschrift für das Steuerwerk (Prozessor) dar. Das Programm besteht aus einer Folge von Steuerungsanweisungen. Eine Anweisung ist wie folgt aufgebaut:



Die **Operation** sagt dem Steuerwerk, was es tun soll. Sie drückt aus, wie eine binäre Variable zu verarbeiten ist (z. B. Ausführung einer UND-Verknüpfung). Der **Operand** enthält die zur Ausführung der Operation erforderlichen Daten. Er sagt dem Steuerwerk, womit etwas geschehen soll. Operanden können z. B. die logischen Zustände der Eingangsvariablen sein. Das Kennzeichen wäre dann Buchstabe E und als Parameter ist die Adresse des Eingangs anzugeben.

Wegen stetig sinkender Hardwarekosten im Elektronikbereich dominieren immer mehr die Kosten der personalintensiven Arbeiten, zu denen die Programmerstellung gehört. Mit geeigneten Programmierhilfen lassen sich diese Kosten jedoch erheblich reduzieren. Moderne Programmiergeräte erlauben deshalb die **direkte Programmeingabe** nach dem Kontaktplan, der Schaltfunktion, der Anweisungsliste oder dem Funktionsplan.

Das einfache Beispiel in Bild A 19.1 soll Ihnen die verschiedenen Möglichkeiten demonstrieren. Programmieren Sie bitte die Schaltfunktion:

$$A1 = (E1 \overline{VE2}) \wedge E3$$

Programm als Kontaktplan

Die Symbole der Kontaktplandarstellung sind Ihnen ja bereits bekannt. Das Bild A 19.2 zeigt den Kontaktplan für vorstehende Schaltfunktion. Verschiedene Programmiergeräte enthalten bereits Tasten für die Kontaktplansymbole. Über diese Tasten kann man den Kontaktplan auf dem Bildschirm zusammensetzen. Die Bildschirmdarstellung wird im Gerät decodiert und in Form von Anweisungen in Maschinensprache in den Programmspeicher geladen.

Programm als mathematische Funktion

Verwendet werden hierzu die mathematischen Zeichen nach DIN 19 239 (Speicherprogrammierte Steuerungen, Programmierung). Der Struktur nach ist die mathematische Funktion eine Schaltfunktion, wie sie in der Digitaltechnik angewendet wird. Im Unterschied zur Schaltfunktion sind jedoch die Operationen in der Reihenfolge aufgeführt, wie sie von der Steuerung abgearbeitet werden. Die mathematische Funktion beinhaltet meist eine logische Zusammenfassung von Anweisungen.

Schaltfunktion unseres Beispiels: $A1 = (E1 \overline{VE2}) \wedge E3$
 Mathematische Funktion: $!E1 / NE2 = M1$
 $!M1 \& E3 = A1$

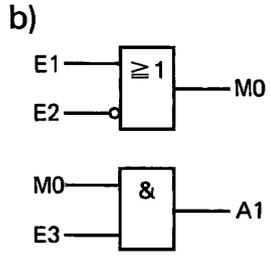
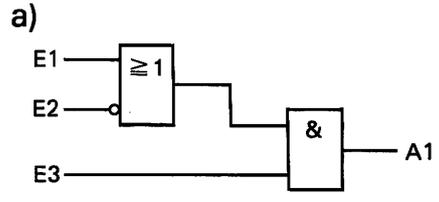


Bild A 19.1 Funktionsplan der Schaltfunktion: $A1 = (E1 \overline{VE2}) \wedge E3$
 Plan b) wird als „aufgelöster Funktionsplan“ bezeichnet. Das Zwischenergebnis E1VE2 ist in Merker M0 gesetzt.

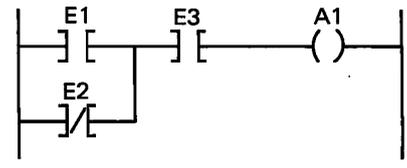


Bild A 19.2 Kontaktplan der Schaltfunktion $A1 = (E1 \overline{VE2}) \wedge E3$.

Programm als Anweisungsliste

Die Anweisungen werden in einem mnemonischen Code geschrieben. Dieser Code verwendet leicht merkbare Verschlüsselungssymbole, die gleichzeitig eine Brücke zu ihrem Verständnis bilden. Die Symbole sind ebenfalls nach DIN 19 239 genormt. Mit „U“ wird beispielsweise die logische Verknüpfung „UND“ symbolisiert. Ein spezielles Programm übersetzt den über die Tastatur des Programmiergeräts eingegebenen mnemonischen Code in den Maschinencode. Die Anweisungsliste für die Schaltfunktion unseres einfachen Beispiels lautet:

```
O E 1
ONE 2
= M 1
U M 1
U E 3
= A 1
= BE
```

Mit „**mathematischen Funktionssymbolen**“ hat die Anweisungsliste folgende Form:

```
!E1
/NE2
=M1
!M1
&E3
=A1
!PE
```

Programm als Funktionsplan

Hier erfolgt die Funktionsbeschreibung der Steuerungsaufgabe mit **Funktionssymbolen**, die von den Schaltzeichen der Digitaltechnik abgeleitet sind. Das Verfahren wird vorwiegend zur Lösung von Steuerungsaufgaben eingesetzt. An Hand des Funktionsplans lassen sich leicht die Schaltfunktion, die Anweisungsliste und das Programm in Form von Anweisungen aufstellen. Anspruchsvolle Programmiergeräte erlauben auch die unmittelbare Programmierung mit Funktionssymbolen. Der auf dem Bildschirm des Programmiergeräts dargestellte Funktionsplan wird direkt in die Maschinensprache der Steuerung übersetzt. In einem eigenen Abschnitt wird der Funktionsplan noch ausführlich besprochen. Für die gestellte Aufgabe ist der Funktionsplan in Bild A 19.1 aufgezeichnet.

Bei komfortablen Programmiereinrichtungen lassen sich die in den Programmspeicher eingegebenen Befehle über ein Sichtgerät kontrollieren. Sie werden auf dem Bildschirm des Programmiergeräts z. B. als logische Verknüpfungen dargestellt. Bei der Programmentwicklung wird als Programmspeicher ein Schreib-Lese-Speicher (RAM) benutzt, damit fehlerhafte Anweisungen noch korrigiert werden kön-

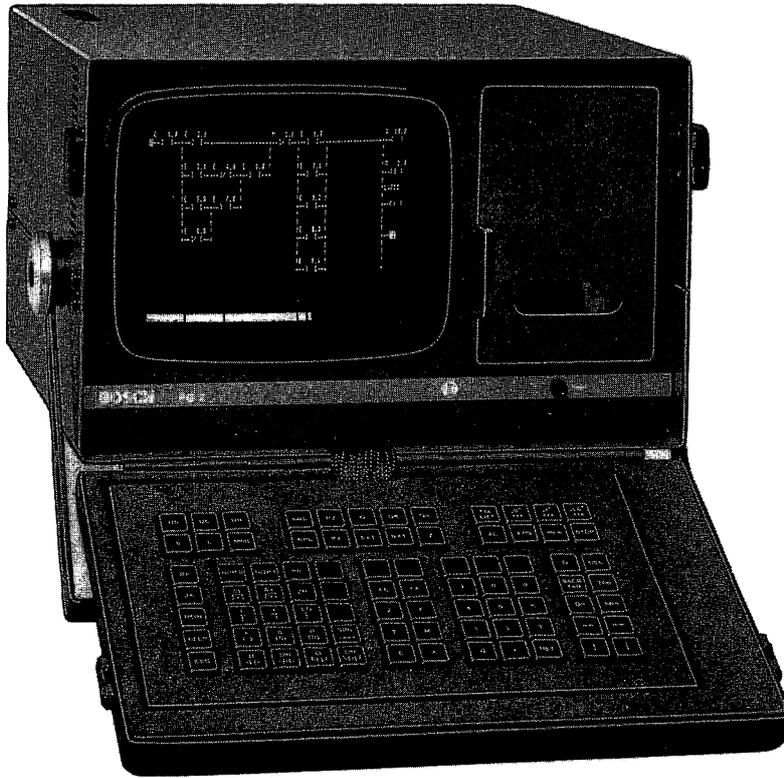


Bild A21.1
Bildschirmprogrammiergerät für Programmierung in den Darstellungsarten Kontaktplan und Anweisungsliste.
(Werkbild Robert Bosch GmbH)

nen. Erst nach Abschluß des Programmiervorgangs kopiert man den Inhalt des RAM in den eigentlichen Programmspeicher, z. B. in ein EPROM (Nur-Lese-Speicher). Im Bild A 21.1 sehen Sie ein **Bildschirmprogrammiergerät**. Der Bildschirm zeigt einen Kontaktplan des Steuerungsablaufs, der gerade programmiert wird.

Das Bild A 21.2 zeigt die übliche Darstellung eines Funktionsplans auf dem Bildschirm eines Programmiergeräts. Die programmierten Funktionen werden als **logische Funktionen** wiedergegeben. Links vom

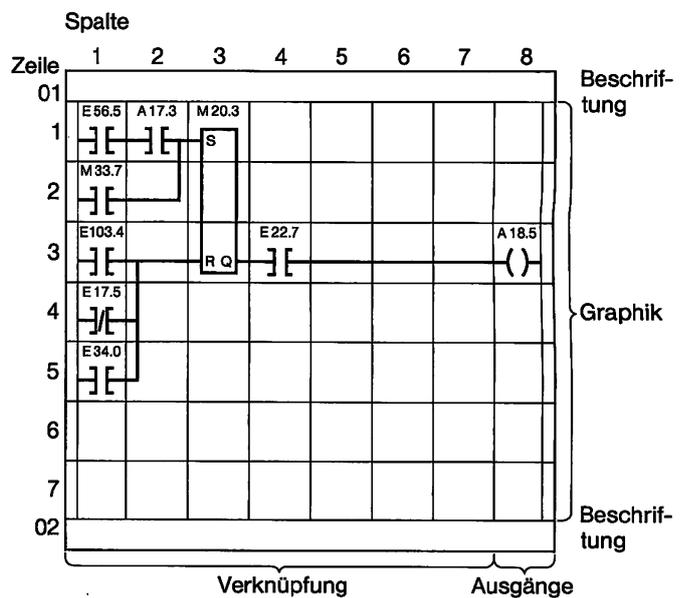
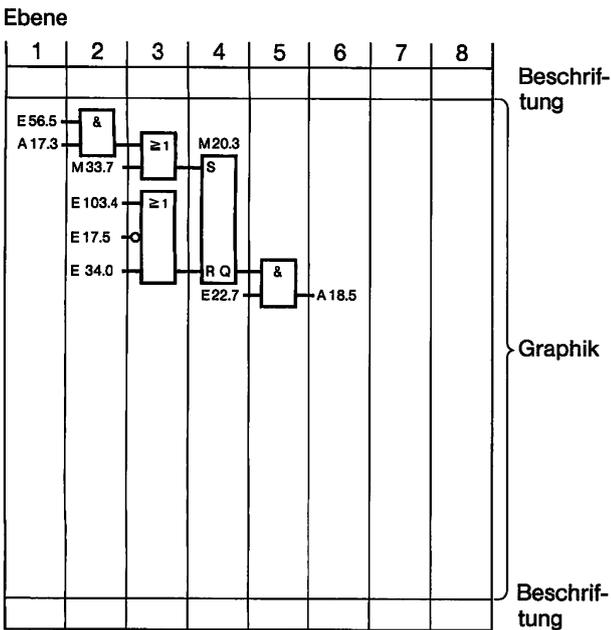


Bild A21.2
Darstellung eines Funktionsplans auf dem Bildschirm eines Programmiergeräts.

Bild A21.3
Darstellung des entsprechenden Kontaktplans auf dem Bildschirm eines Programmiergeräts.

Symbol stehen die Bezeichnungen der Eingänge, rechts davon die der Ausgänge.

Den Kontaktplan der im Bild A 21.2 gezeigten Funktionsdarstellung sehen Sie im Bild A 21.3. Die senkrechten Striche links und rechts davon deuten die „Spannungsschienen“ an. Zwischen den beiden waagrecht verlaufenden Strichen liegt das **Kontaktnetzwerk**. Die hier gezeigten Kontaktsymbole entsprechen nicht der DIN, sondern der amerikanischen Norm. Zur Wiedergabe der Symbole nach DIN müssten erst neue Zeichengeneratoren entwickelt und in die Geräte eingebaut werden. Auf dem Bildschirm eines Programmiergeräts kann auch eine **Anweisungsliste** wiedergegeben werden (Bild A 22.1). Die Anweisungsliste enthält die Steuerungsanweisungen in der Reihenfolge, in der sie auszuführen sind. Die Basis der Anweisungsliste ist ein skizzierter Funktionsplan.

Die Operanden, z. B. E56.5, bestehen aus Kennzeichen und Parameter. Der **Parameter** ist eine Zahlengruppe, die durch einen Punkt aufgeteilt ist. Der Punkt ist das Trennzeichen zwischen der Byteadresse und der Bitadresse. Die Zahl vor dem Punkt ist die Nummer eines bestimmten Bytes (Baugruppe) und die Zahl hinter dem Punkt ist die Nummer des anzusprechenden Bits innerhalb des Bytes. In den meisten Automatisierungsgeräten (SPS) sind die Eingänge, Ausgänge und Merker byteweise organisiert.

U	E	56.5	Beschriftung
U	A	17.3	
O	M	33.7	
S	M	20.3	Anweisungsliste
O	E	103.4	
ON	E	17.5	
O	E	34.0	
R	M	20.3	Steuerungsanweisung
U	M	20.3	
=	E	22.7	
	A	18.5	

Operationsteil Kennzeichen Parameter Operandenteil

Bild A 22.1
Darstellung der Anweisungsliste auf dem Bildschirm eines Programmiergeräts.

Zusammenfassung

Beim Programmieren in Maschinsprache werden erhebliche Programmieranforderungen gestellt. Deshalb erfolgt die Umsetzung des Programms in die Maschinsprache mit einem Programmiergerät. Die Programmerstellung und Eingabe kann in einer einfachen Programmiersprache als Kontaktplan, Funktionsplan, Anweisungsliste oder mathematische Funktion erfolgen. Ein spezielles Programm übersetzt im Programmiergerät die Anweisungen der Programmiersprache in die Maschinsprache.

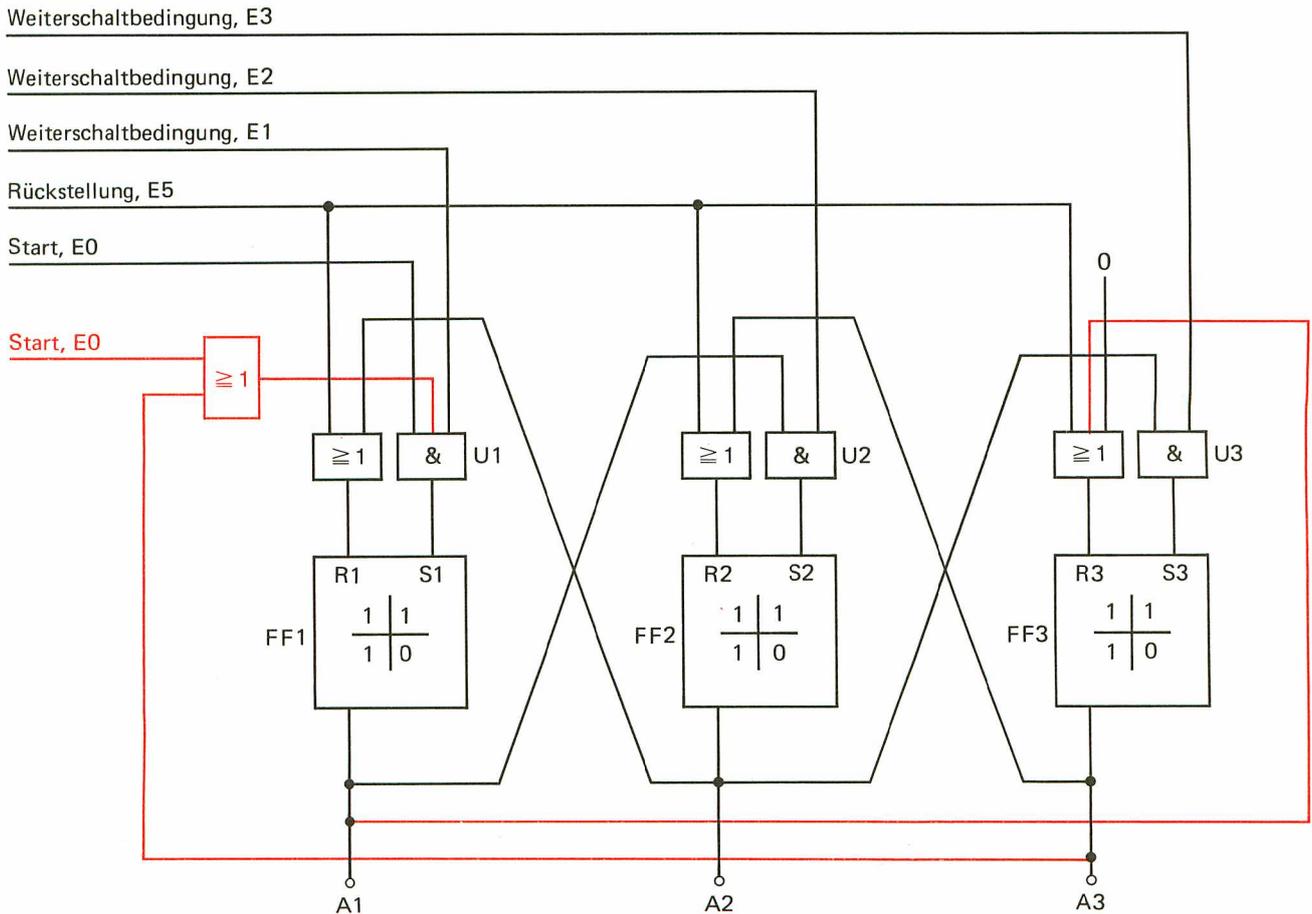
Ablaufsteuerung

Die grundsätzlichen Erläuterungen zur Arbeitsweise einer Ablaufsteuerung haben wir Ihnen bereits im Lehrbrief 1 gegeben. Wir wollen nunmehr das Gelernte in einigen Beispielen und Programmen anwenden. Zunächst erfolgt die schrittweise Entwicklung dieser binären Steuerung. Für den FUP können die gleichen Symbole wie für Verknüpfungssteuerungen verwendet werden. Sie sind zusätzlich durch **Funktionsymbole**, die das schrittweise Abarbeiten des Steuerungsprogramms charakterisieren, ergänzt.

Erinnern wir uns noch einmal an die typische Arbeitsweise aller Ablaufsteuerungen. Sie bearbeiten das Programm in Schritten. Das Weiterschalten von einem Schritt zum programmgemäß folgenden Schritt setzt die Erfüllung von **Weiterschaltbedingungen** voraus. Diese können prozeßabhängig oder zeitabhängig sein. Vielfach sind Weiterschaltbedingungen Rückmeldungen aus der zu steuernden Anlage. Sie signalisieren einen bestimmten Prozeßzustand oder die beendete Ausführung eines Befehls.

Das Bild A 23.1 zeigt einen Ausschnitt des FUP einer einfachen Ablaufsteuerung. Die Steuerung besteht aus einer Kettenschaltung von Speicherfunktionen. Eine Speicherfunktion mit den erforderlichen zusätzlichen Verknüpfungsfunktionen bildet eine **Ablauffunktion** (in der Schaltungstechnik als Ablaufglied bezeichnet). Die Kettenschaltung der Ablauffunktionen wird auch als **Ablaufkette** bezeichnet.

Bild A 23.1
FUP einer Ablaufkette mit 3
Ablaufschritten, Beispiel A 24.1.



Für jeden Schritt des Steuerungsprogramms ist eine Ablauffunktion erforderlich.

Aus dem FUP in Bild A23.1 kann man weiter folgende wichtige Verknüpfungsregeln ableiten:

Das Ausgangssignal 1 einer Speicherfunktion in der Ablaufkette bereitet das Setzen der jeweils folgenden Speicherfunktion vor und setzt die vorhergehende Speicherfunktion zurück.

Nach dem FUP in Bild A23.1 erzeugt die Ausgangsvariable $A2=1$ von FF2 das Rücksetzsignal für FF1. Gleichzeitig bereitet A2 die UND-Verknüpfung von U3 vor. Nach Erfüllung der UND-Bedingung durch die prozeßabhängige Rückmeldung E3 wird FF3 gesetzt.

Das Weiterschalten der Ablaufkette erfolgt in Bild A23.1 über die Signale E1, E2 und E3. Bei der prozeßabhängigen Ablaufsteuerung sind es Rückmeldungen aus der zu steuernden Anlage. Ist z. B. der vom Ausgangssignal A2 ausgelöste Befehl ausgeführt, könnte eine Rückmeldung mit $E3=1$ erfolgen. Damit ist die Setzbedingung für Schritt 3 erfüllt, ein neuer Befehl wird ausgegeben. Gleichzeitig erfolgt das Rücksetzen von Schritt 2.

Beispiel A24.1

Programmierung einer Ablaufsteuerung mit drei Ablauffunktionen

- a) Der FUP von Bild A23.1 ist in die aufgelöste Darstellung einzusetzen.
- b) Stellen Sie nach dem aufgelösten FUP für die Ablaufsteuerung das Programm auf.
- c) Testen Sie das Programm und gehen Sie zur Überprüfung wie folgt vor:
 1. Alle Speicherfunktionen sind in die Grundstellung zu bringen; schalten Sie E5 kurzzeitig auf 1-Signal.
 2. Zur Vorbereitung des Starts muß E0 solange im Zustand 1 bleiben, bis auch die Weiterschaltbedingung E1 für den ersten Schritt Zustand 1 hat.
 3. Nun erfolgt der Steuerungsablauf. Es wird vereinbart, daß die prozeßabhängigen Rückmeldungen E1 bis E3 aus einem kurzen 1-Impuls (0-1-0 Signal) bestehen.

Überprüfen Sie bitte die Funktion der Ablaufsteuerung durch die Aufnahme der Funktionstabelle A25.1.

Tabelle A25.1: Funktionstabelle zum Beispiel A24.1

E5	E0	E1	E2	E3	A1	A2	A3
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	┐	0	0	1	0	0
0	0	0	0	0	1	0	0
0	0	0	┐	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	┐	0	0	1
0	0	0	0	0	0	0	1
0	0	┐	0	0	0	0	1

(┐ Symbol für ein kurzes 1-Signal)

Lösung:

a) Im FUP bedeuten:

- E1 bis E3: prozeßabhängige Weiterschaltbedingungen
- E5: Rücksetzsignal
- E0: Startsignal

Den aufgelösten FUP der Ablaufsteuerung von Bild A23.1 zeigt das Bild A26.1.

b) Nach dem aufgelösten FUP können Sie das folgende Programm aufstellen:

Programm:

- !E0&E1=SA1
- !E5/A2=RA1
- !A1&E2=SA2
- !E5/A3=RA2
- !A2&E3=SA3
- !E5=RA3
- !PE

c) Der Test muß die in Tabelle A25.1 ROT eingetragenen Signalzustände ergeben. Die Ablaufkette wird nur einmal durchlaufen. Sie muß anschließend mit E0 erneut gestartet werden.

Die Ablaufkette läßt sich nach dem Schema von Bild A23.1 leicht auf eine größere Zahl von Ablaufschritten erweitern.

Beispiel A25.1

Änderung der Ablaufsteuerung von Beispiel A24.1

Durch Änderung des Programms soll erreicht werden, daß die Steuerung nach Abarbeitung von Schritt 3 selbständig wieder mit Schritt 1 beginnt.

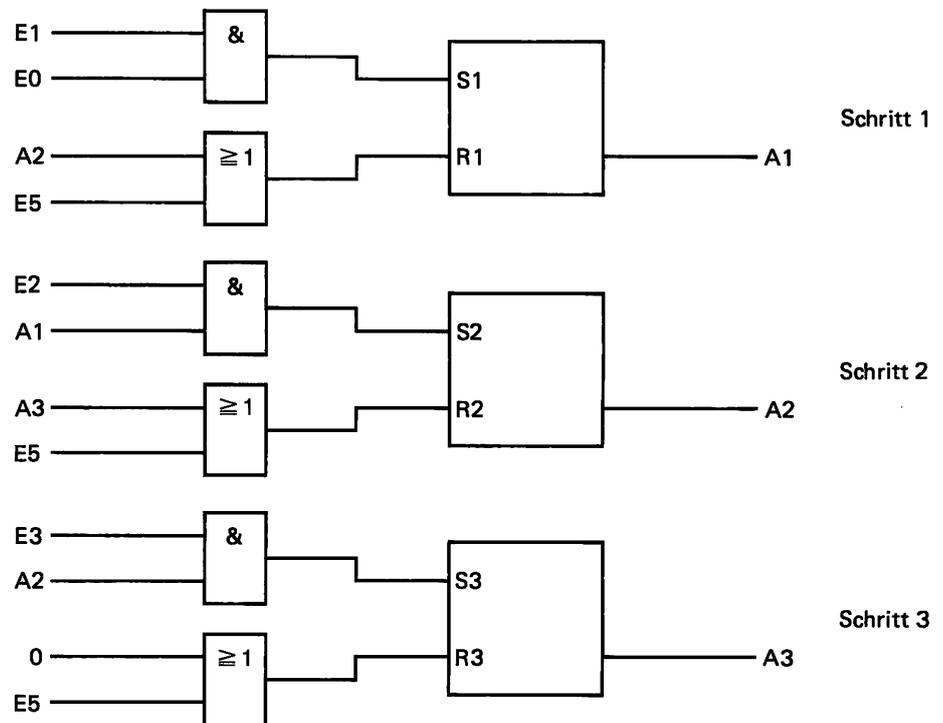
A**26**

Bild A26.1
Aufgelöster FUP von Bild A23.1,
Beispiel A24.1.

- Der FUP von Beispiel A24.1 ist so abzuändern, daß er die neuen Vorgaben erfüllt.
- Übertragen Sie die FUP-Änderungen in das Programm.
- Testen Sie das abgeänderte Programm. Dabei ist zu beachten, daß die prozeßabhängigen Weiterschaltbedingungen in der richtigen Reihenfolge in die SPS eingegeben werden.

Lösung:

- Im FUP von Bild A26.1 müssen die Ablaufverknüpfungen der Schritte 1 und 3 geändert werden. Einen Vorschlag hierfür haben wir ROT in den FUP von Bild A23.1 eingetragen. Der Start erfolgt wieder mit $E0=1$ und $E1=1$. Nach Durchlaufen der Kette hat A3 den Zustand 1. Da zwischenzeitlich das Startsignal E0 den Wert 0 angenommen hat, bereitet $A3=1$ die Verknüpfung zum Setzen von FF1 (Schritt 1) vor, E0 und A3 sind deshalb mit ODER verknüpft. Nach dem Setzen mit $E1=1$ wird $A1=1$. Dieser Zustand setzt Ablaufschritt 3 (FF3) zurück ($A3=0$).

Für die Schritte 1 und 3 ergeben sich die im FUP von Bild A27.1 eingetragenen Änderungen.

- Programm:

```

!(E0/A3)&E1=SA1
!E5/A2=RA1
!A1&E2=SA2
!E5/A3=RA2
!A2&E3=SA3
!E5/A1=RA3
!PE

```

c) Testen Sie das Programm mit Hilfe der Funktionstabelle A 25.1. Beachten Sie, daß die letzte Zahl in der Tabelle nun nicht mehr zutrifft. Bei folgerichtiger Eingabe der Weiterschaltbedingungen wiederholt sich der Ablauf beliebig oft. Wir haben durch die Rückführung des Ausgangs auf den Eingang eine Schleife gebildet.

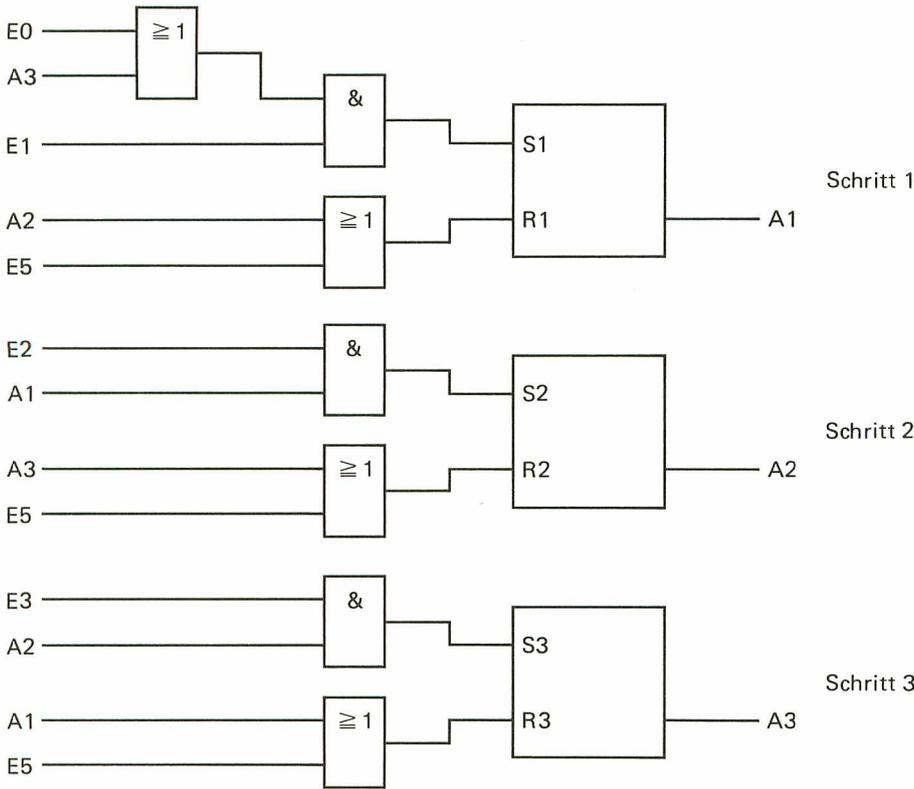


Bild A27.1
Aufgelöster FUP vom Beispiel A 25.1, Bildung einer Schleife.

Aufgabe A27.1

Ändern Sie die Rückführung der dreigliedrigen Ablaufkette von Bild A 23.1 nach Bild A 27.2.

In der Rückführung von Ausgang A3 zum Setzeingang des FF1 liegt die UND-Verknüpfung U. Der Zustand des Signals E6 beeinflusst die Weitergabe des Signalzustands A3 und damit auch die Funktion der Ablaufsteuerung. Beziehen Sie die Antworten zu nachstehenden Fragen auf die Funktion der Ablaufsteuerung von Beispiel A 25.1.

a) Wie wirken sich die Signalzustände
 1) E6=1 und 2) E6=0
 auf die Funktion der Ablaufsteuerung von Beispiel A 25.1 aus?

b) Übernehmen Sie die Änderung in das Programm von Beispiel A 25.1.

c) Überprüfen Sie experimentell Ihre Antworten zu Punkt a).

Die Lösung dieser Aufgabe finden Sie auf Seite F51.

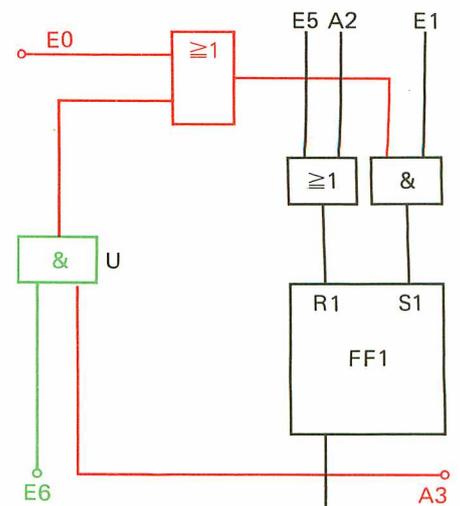


Bild A27.2
Änderung der Eingangsschaltung, Einbau einer Rückführung, Aufgabe A27.1.

Vereinfachter Funktionsplan für die Ablaufsteuerung

Der bisher in allen Einzelheiten aufgezeichnete FUP einer Ablaufsteuerung kann bei umfangreicheren Steuerungen sehr aufwendig und dadurch unübersichtlich werden. In der Praxis wird deshalb eine sehr vereinfachte Darstellung bevorzugt. Verwendung findet ein spezielles, nur für die Ablaufsteuerung gültiges **Funktionsplansymbol**. Es beinhaltet alle den Steuerungsablauf bestimmenden Schaltbedingungen.

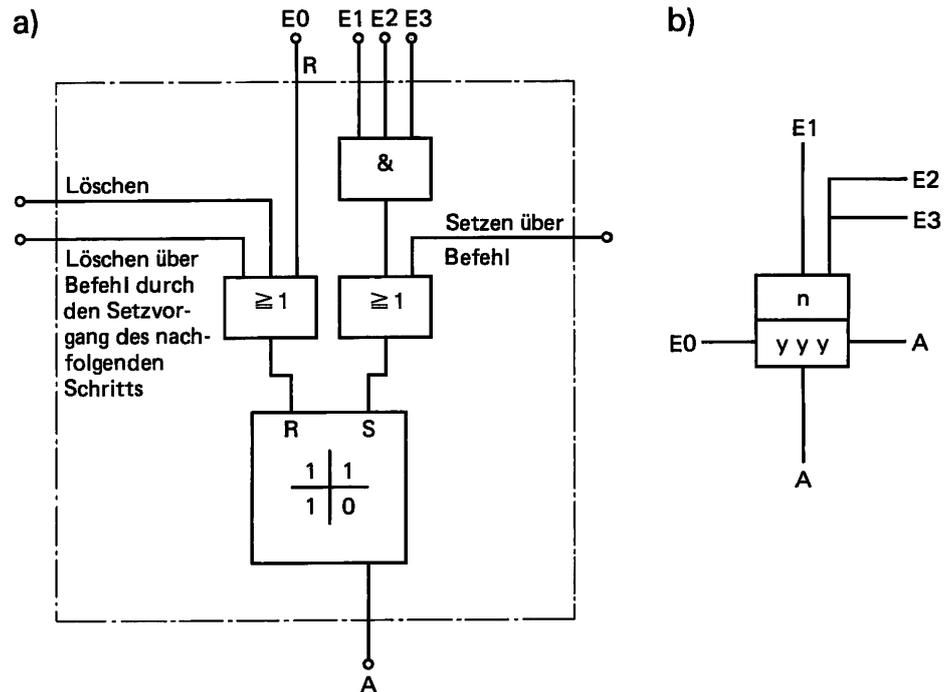


Bild A 28.1
Schritt einer Ablaufkette.
a) ausführlicher FUP eines
Ablaufschrittes,
b) vereinfachter FUP eines
Ablaufschrittes.

In Bild A 28.1 haben wir den ausführlichen FUP eines Ablaufschrittes der vereinfachten Darstellung gegenübergestellt. In der vereinfachten Darstellung (**Schrittsymbol** genannt) ist mit n die Schrittnummer innerhalb der Ablaufkette bezeichnet (z. B. $n=5$). Bei YYY ist im Klartext einzutragen, welche Funktion der Schritt auslösen soll, z. B. $YYY =$ „Füllen“. Das Setzen des Schritts erfolgt, wenn die Variablen an den Eingängen $E1$ bis $E3$ alle den Wert 1 haben. Bei gesetztem Signalspeicher nimmt Ausgang A Zustand 1 an. Mit $A=1$ werden Befehle ausgegeben (z. B.: „Ventil AUF“). Gleichzeitig wird der vorangegangene Schritt rückgesetzt und der folgende Schritt wird zum Setzen vorbereitet. In Sonderfällen lässt sich ein Schritt noch über besondere Befehle setzen bzw. rücksetzen (in der Steuerungstechnik vielfach auch als Löschen bezeichnet, Eingang R). Im Bild A 29.1 ist der FUP des Beispiels A 24.1 (Bild A 23.1) mit vereinfachten Symbolen dargestellt.

Vom Ausgang A des jeweiligen Schritts werden Befehle an die Stellgeräte oder Signale für Meldeinrichtungen ausgegeben. Nach Bild A 29.2 ist der Befehl im Klartext in das Befehlsymbol eingetragen. Mit der Befehlsausgabe kann auch eine Zeitbedingung gestartet werden. Die Befehlsausgabe von Schritt 4 in Bild A 29.2 drückt beispielsweise aus, daß der folgende Schritt 5 erst nach Ablauf der Wartezeit $t=120$ s

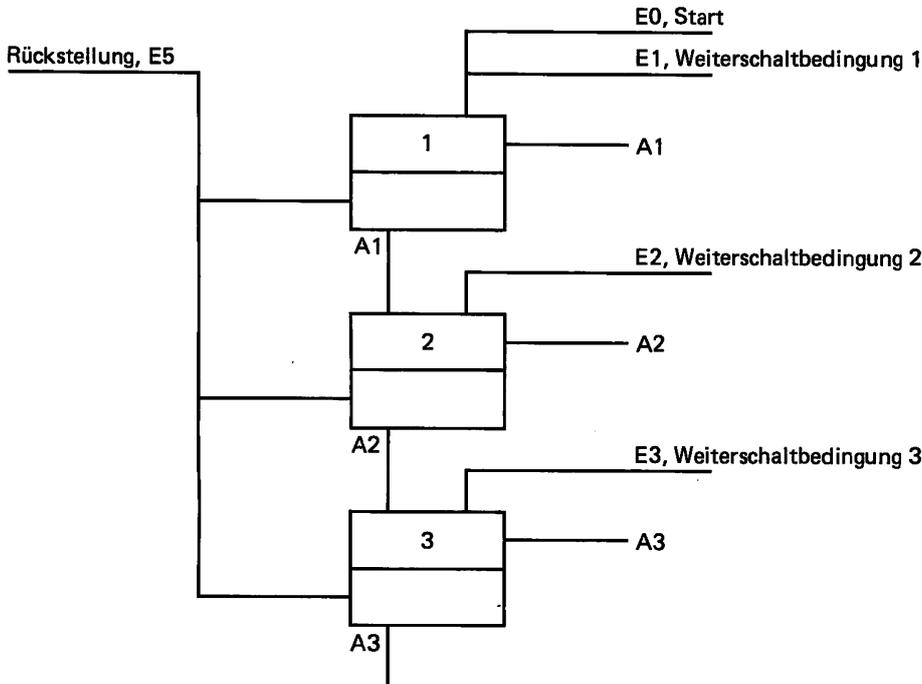


Bild A 29.1
Ablaufkette mit 3 Ablaufschritten
von Bild A 23.1, dargestellt mit vereinfachten Symbolen.

gesetzt werden kann. Wie schon mehrfach erwähnt, wird ein Schritt allerdings erst dann gesetzt, wenn **alle Weichschaltbedingungen** erfüllt sind. Diese Bedingungen können im Klartext oder durch Operanden (E1, E2,..) angegeben sein. In Bild A 29.2, Schritt 6 bedeuten die Ziffern 1, 2, und 3, daß der Schritt erst dann gesetzt wird, wenn die im Schritt 5 erteilten Befehle mit gleichen Kennziffern ausgeführt sind. Zur vereinfachten Darstellung sind dem Befehl und der von der Befehlsausführung abhängigen Rückmeldung (Weichschaltbedingung) gleiche Kennziffern zugeordnet.

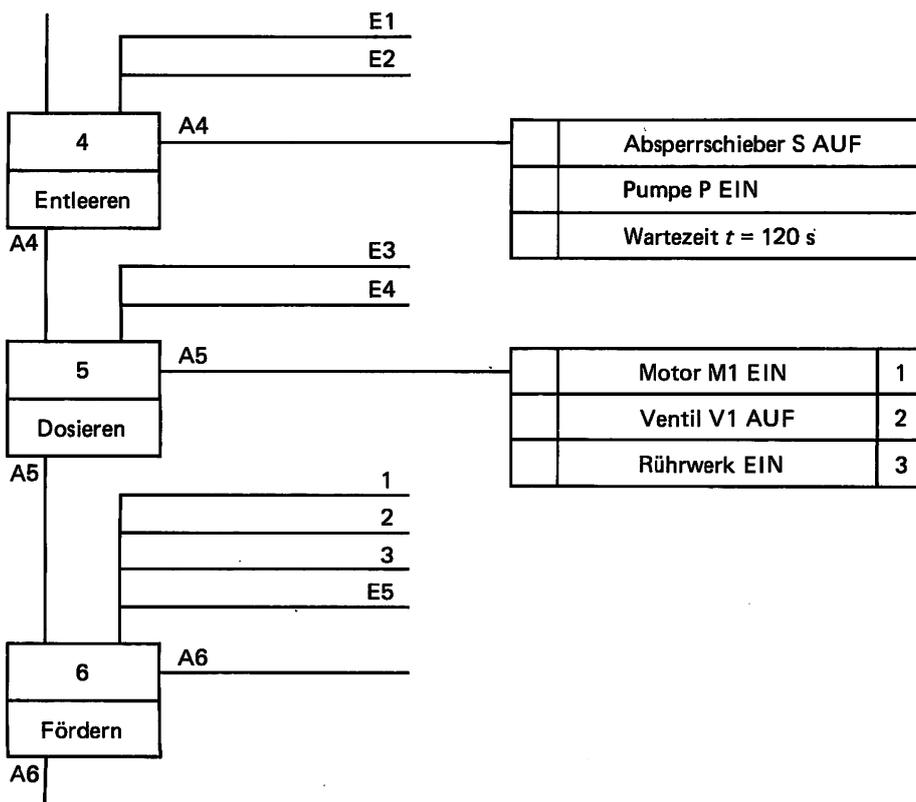


Bild A 29.2
Auszug aus dem FUP einer Ablaufsteuerung.

Aufgabe A 30.1

Prozeßabhängige Ablaufsteuerung

Das Bild A 30.1 zeigt den FUP einer Ablaufsteuerung mit vier Ablauf-
funktionen. Als Speicher werden Merker verwendet. Mit dem Richt-

A
30

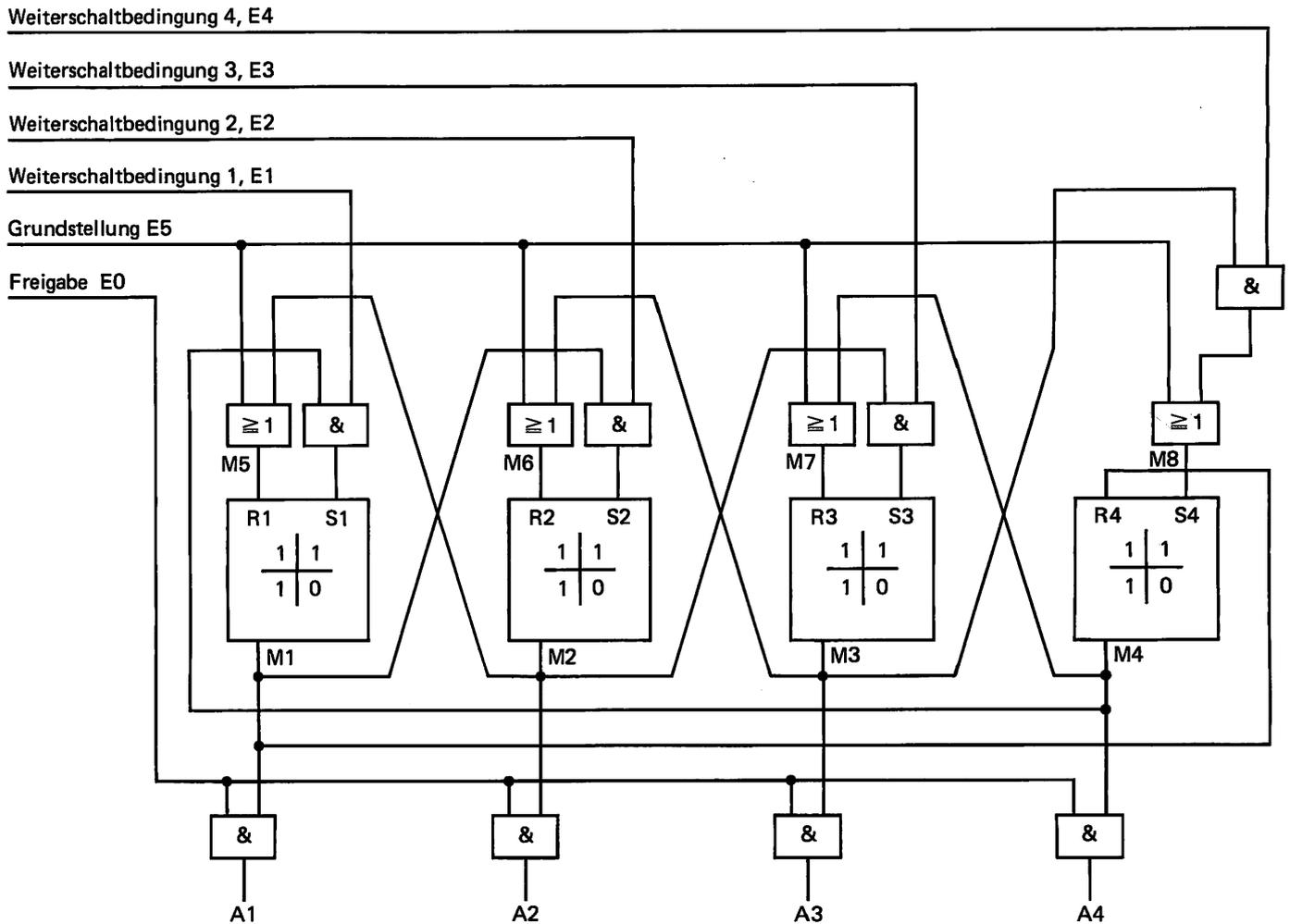


Bild A 30.1
FUP zur Aufgabe S 30.1

impuls E5 werden alle Merker in den Anfangszustand gebracht (M1 bis M3 in Zustand 0, M4 in Zustand 1). Die prozeßabhängigen Weiterschaltbedingungen E1 bis E4 bewirken das Schalten von Schritt zu Schritt, sie sind als kurze 1-Signale einzugeben.

Erst nach dem Setzen von Schritt 1 (M1=1) soll das Freigabesignal E0 den Wert 1 erhalten. Mit E0=1 werden die Signalzustände der Ablaufschritte auf die Ausgänge A1 bis A4 geschaltet. In der Praxis lösen die A-Signale Befehle aus.

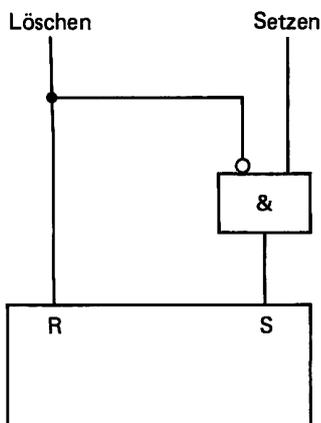


Bild A 30.2
Rücksetzdominanz der Speicherfunktion nach Bild A 30.1.

- Zeigen Sie die aufgelöste Darstellung des geschlossenen FUP von Bild A 30.1.
- Stellen Sie für die Ablaufsteuerung das Programm auf und programmieren Sie die Rücksetzdominanz der Speicherfunktionen nach Bild A 30.2. Verwenden Sie die in Bild A 30.1 eingetragenen Operanden. Falls weitere Merker erforderlich sind, wählen Sie deren Operanden bitte selbst.

- c) Testen Sie Ihr Programm und überprüfen Sie bitte, ob es die Vorgaben von Funktionstabelle A 31.1 erfüllt.

Tabelle A 31.1: Funktionstabelle zur Aufgabe A 30.1

E5	E0	E1	E2	E3	E4	A1	A2	A3	A4
1	0	0	0	0	0	0	0	0	0
0	0	⌋	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	⌋	0	0	0	1	0	0
0	1	0	0	⌋	0	0	0	1	0
0	1	0	0	0	⌋	0	0	0	1
0	1	⌋	0	0	0	1	0	0	0
0	1	0	⌋	0	0	0	1	0	0
1	1	⌋	0	0	0	0	0	0	1

(⌋ = 1-Signale kurzer Zeitdauer)

Beantworten Sie bitte nachstehende Fragen. Die Fragen beziehen sich auf die prozeßgeführte Ablaufsteuerung von Aufgabe A 30.1. Sie können Ihre Antworten experimentell überprüfen.

Aufgabe A 31.1

- a) Was passiert, wenn die Weiterschaltbedingungen nicht in der prozeßbedingten Reihenfolge eingegeben werden?
- b) Für die Weiterschaltbedingungen wurden kurze 1-Impulse vereinbart. Statt der Impulse werden Dauersignale mit Zustand 1 eingegeben. Wie wirken sich diese auf den Steuerungsablauf aus ?
- c) Während des Steuerungsablaufs in Aufgabe A 30.1 geht E5 vom Zustand 0 in Zustand 1. Was bewirkt die Signaländerung?

Die Lösungen dieser Aufgaben finden Sie auf den Seiten F51 und F52.

Zeitabhängige Ablaufsteuerung

Wir wollen jetzt eine Ablaufsteuerung mit zeitabhängigen Weiterschaltbedingungen untersuchen. Die Festlegung der Zeitbedingungen erfolgt mit **Zeitfunktionen**. Wie lange sich ein Schritt im Setzzustand befindet, bestimmt die programmierte Zeitverzögerung. Während dieser Zeit hat der Schrittausgang A Zustand 1.

Das Bild A 32.1 zeigt im Prinzip den FUP eines Ablaufschritts. Wir können daraus die grundsätzliche Wirkungsweise ableiten. Bei gesetzter Speicherfunktion 1 hat M1 Zustand 1. Mit dem Freigabesignal E0=1 wird dieser Zustand auf A1 übertragen. M1=1 bereitet die UND-Verknüpfung U2 vor und startet die Zeitfunktion T1. Erst nach Ablauf der programmierten Einschaltverzögerung t_1 geht der Ausgang von T1 in

Zustand 1 über. Damit ist die UND-Bedingung U2 erfüllt und die Speicherfunktion 2 wird gesetzt. M2 nimmt den Wert 1 an und setzt u. a. die Speicherfunktion 1 zurück.

Ordnet man zwischen den Speicherfunktionen einer Ablaufkette Zeitfunktionen als **Koppelfunktionen** an, dann läuft der Steuerungsvorgang nach dem Start von Schritt zu Schritt selbständig ab. Mit der Einschaltverzögerung t_1 wird die Verweilzeit für den Zustand 1 der Ablauffunktion festgelegt. Im folgenden Beispiel wollen wir eine zeitabhängige Ablaufsteuerung untersuchen.

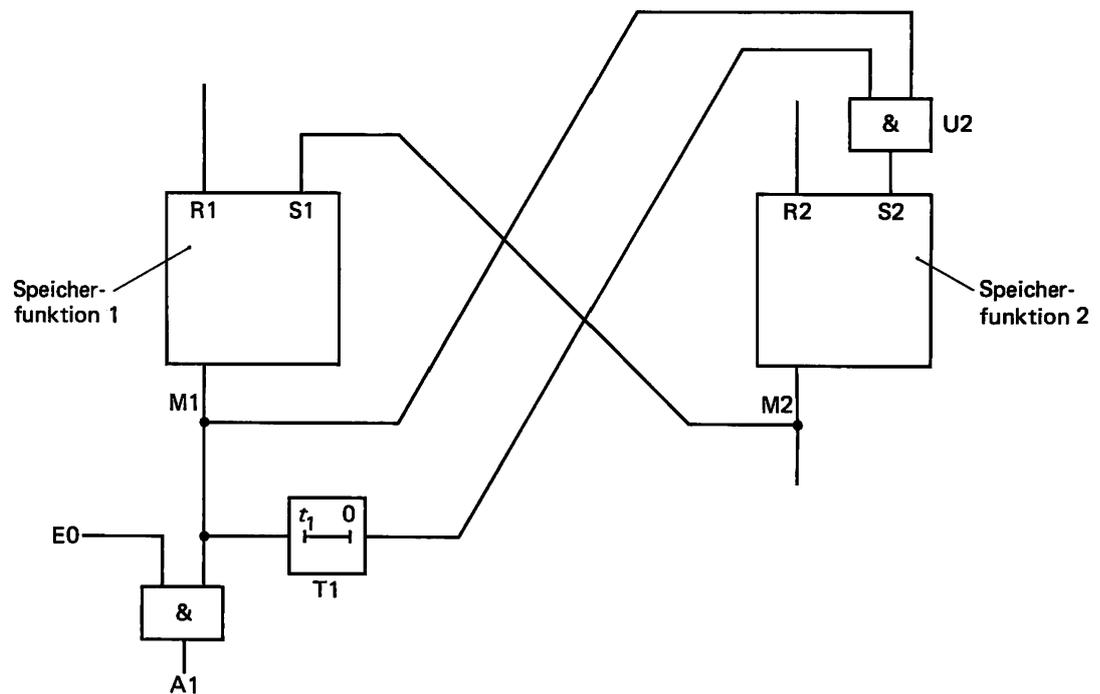


Bild A 32.1
Kopplung zweier Ablauffunktionen über eine Zeitfunktion.

Beispiel A 32.1

Programmierung einer zeitabhängigen Ablaufsteuerung

Die prozeßabhängige Ablaufsteuerung von Aufgabe A 30.1 und Bild A 30.1 ist in eine zeitabhängige Ablaufsteuerung umzustellen. Den Ablaufschritten 1 bis 4 sind die Timer T1 bis T4 zugeordnet. Sie sind für den Programmtest mit folgenden Verzögerungszeiten zu programmieren:

Einschaltverzögerung $t_1=250$ ZE, Ausschaltverzögerung $t_2=0$ ZE.

- Der geschlossene FUP der prozeßabhängigen Ablaufsteuerung nach Bild A 30.1 ist gemäß der Aufgabenstellung in eine zeitabhängige Ablaufsteuerung zu ändern.
- Entwickeln Sie aus dem geschlossenen FUP den aufgelösten FUP.
- Programmieren Sie die zeitabhängige Ablaufsteuerung. Wir benutzen als Grundlage das bereits von Ihnen aufgestellte Programm der Aufgabe A 30.1. Es sind nur kleine Programmänderungen erforderlich.

Testen Sie anschließend das Programm.

Lösung:

- a) Den geänderten und der Aufgabenstellung entsprechenden FUP zeigt das Bild A33.1.
- b) In Bild A34.1 ist der aufgelöste FUP wiedergegeben.
- c) Der FUP läßt sich aus folgendem Programm entwickeln:

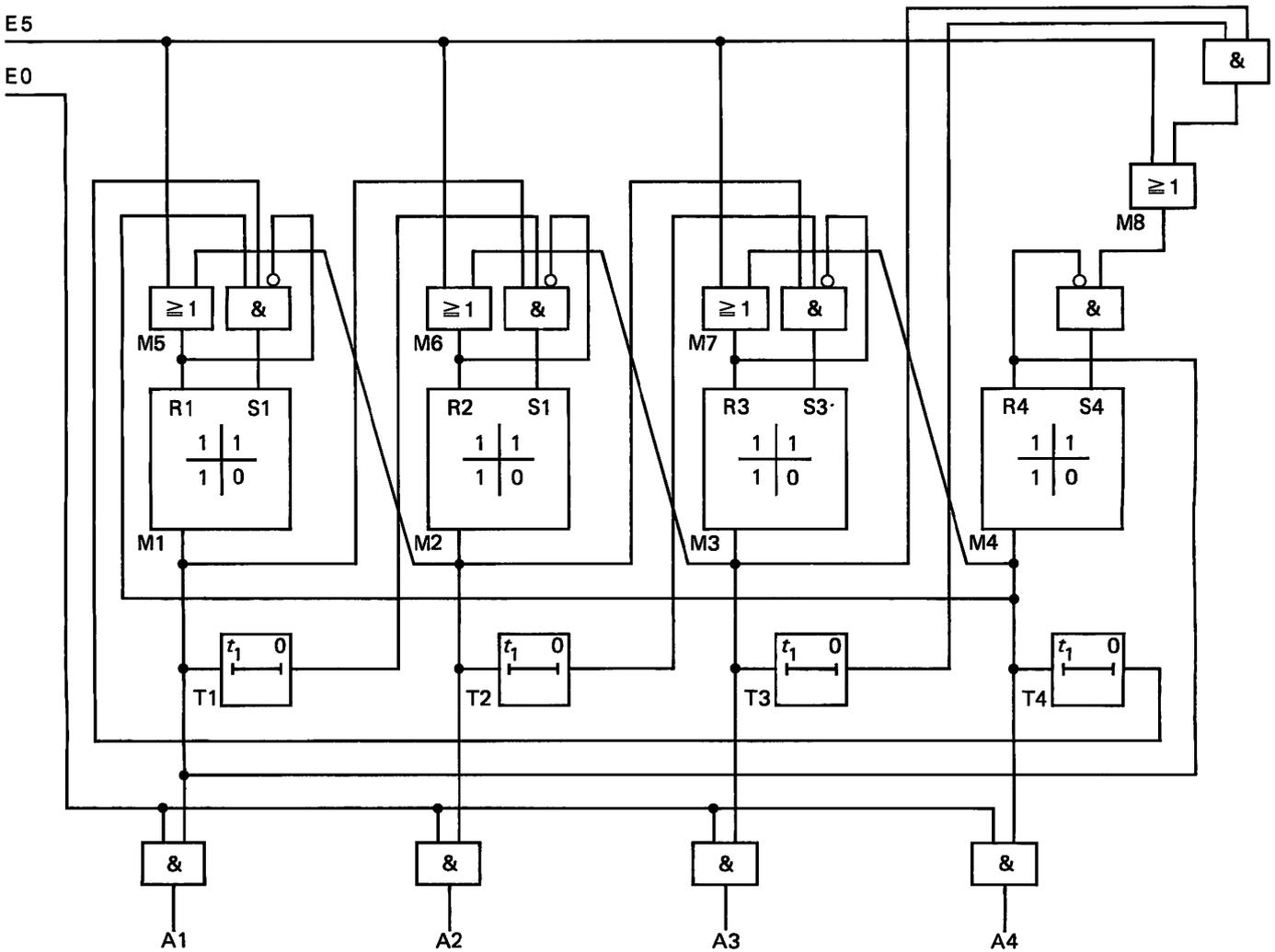
Programm:

!T4&M4&NM5=SM1	!T3&M3/E5=M8
!M2/E5=M5	!M8&NM1=SM4
!M5=RM1	!M1=RM4
!M1=T1	!M4=T4

!T1&M1&NM6=SM2	!E0&M1=A1
!M3/E5=M6	!E0&M2=A2
!M6=RM2	!E0&M3=A3
!M2=T2	!E0&M4=A4
	!PE

!T2&M2&NM7=SM3
!M4/E5=M7
!M7=RM3
!M3=T3

Bild A33.1
Blockschaltung einer Ablaufsteuerung.



A

34

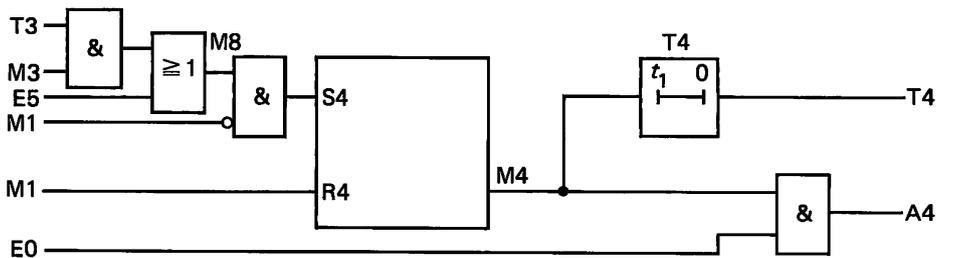
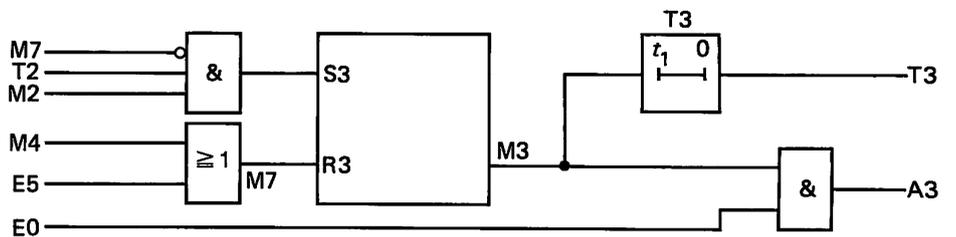
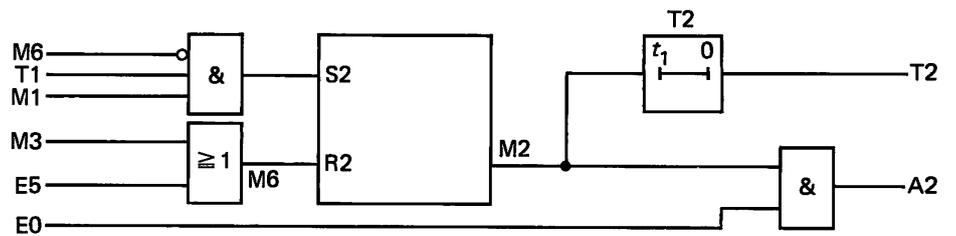
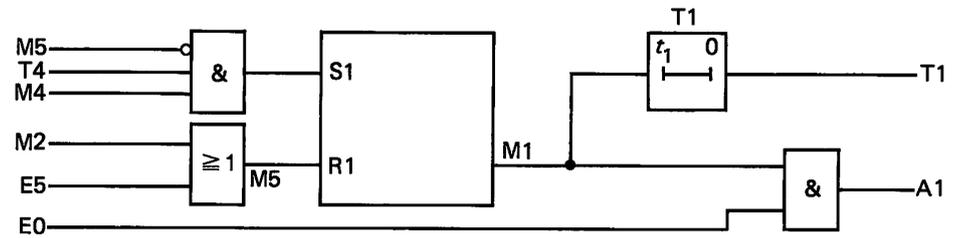


Bild A 34.1
FUP der Ablaufsteuerung zu Beispiel A 32.1.

Zusammenfassung

Bei den Ablaufsteuerungen werden die Programmschritte nacheinander durchgeführt. Weichschaltbedingungen können sowohl prozess- oder zeitabhängig sein. Speicherfunktionen mit zusätzlichen Verknüpfungen bezeichnet man als Ablauffunktionen (Ablaufglieder).

Umfangreiche FUP-Darstellungen lassen sich auch vereinfacht realisieren. Diese vereinfachten Darstellungen verwendet bei Ablaufsteuerungen das Schrittsymbol. Darin sind alle den Steuerungsablauf beeinflussenden Schaltbedingungen enthalten.

Ablaufsteuerung mit Betriebsartenteil und Ablaufkette mit Makrobefehlen

Wir kommen in diesem Abschnitt wieder auf die Ablaufsteuerung zurück. Einmal sollen die Makrobefehle Anwendung finden, zum anderen wollen wir eine Steuerungsaufgabe möglichst praxisnah mit verschiedenen Nebenbedingungen bearbeiten. Allerdings müssen wir die Nebenbedingungen etwas einschränken. Sie wissen ja bereits, daß die Kapazität des Arbeitsspeichers begrenzt ist. Die allgemeine Blockschaltung der Steuerung mit zusätzlichen Betriebseinheiten zeigt das Bild A 35.1. Der eigentlichen Steuerung (Ablaufkette) ist ein Betriebsartenteil vorgeschaltet. Die meisten Steuerungen lassen verschiedene Betriebsarten zu. Diese können über Eingabebefehle gewählt werden.

a) **Automatikbetrieb:** Die Betätigung der Stellgeräte oder Anzeigen erfolgt ausschließlich durch die Steuerung. Eine Handbetätigung (z. B. zum Testen) ist nicht möglich. Nach dem Start wird die Ablaufkette zyklisch durchlaufen.

b) **Einzelschrittbetrieb:** Diese Betriebsart wird zum Prüfen des Programms bei Inbetriebnahme der Steuerung benötigt. Die Ablaufkette läßt sich von Hand immer um einen Schritt weiterschalten.

c) **Tippbetrieb:** In dieser Betriebsart können die einzelnen Stellgeräte von Hand geschaltet werden. Vielfach sind „Tipp- und Einzelschritt-Betrieb“ kombiniert.

Zu wichtigen, vom Betriebsartenteil ausgegebenen Anweisungen gehören:

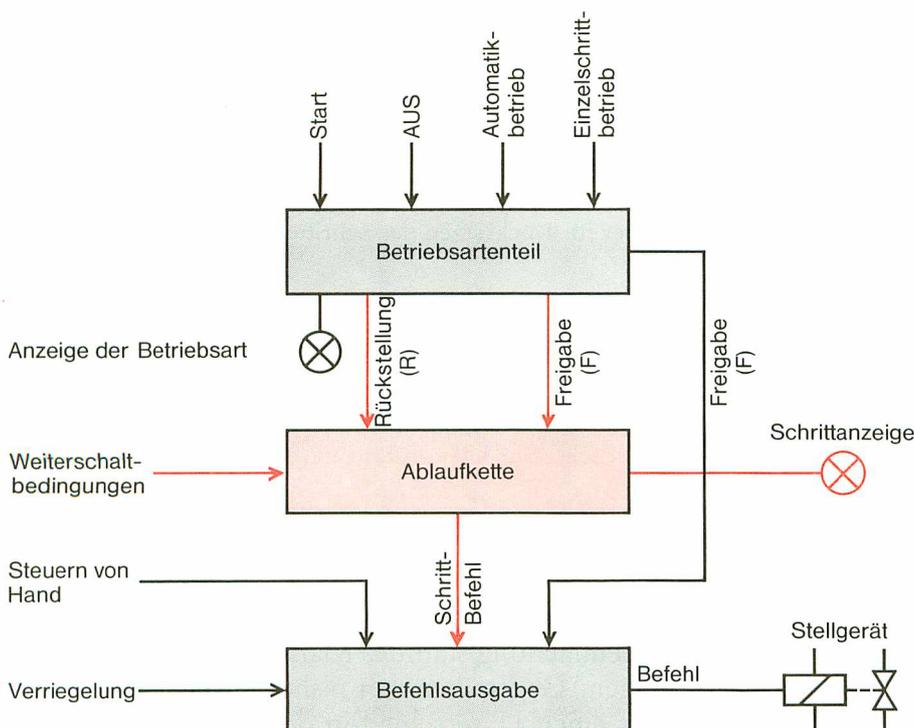


Bild A 35.1
Blockschaltung einer Ablaufsteuerung.

a) **Freigabesignal:** Nur wenn es den Zustand 1 hat, kommen z. B. die Weiterschaltbedingungen von einem Schritt zum nächsten zur Wirkung. Bei Zustand 0 ist der Funktionsablauf gesperrt.

b) **Rücksetzsignal:** Es setzt alle Ablaufschritte in die Grundstellung (Startstellung) zurück. Meist wird der letzte Schritt gesetzt, um den Schaltzustand zum Start eines erneuten Ablaufs herzustellen.

Die Betriebsarten können über Eingabebefehle gewählt werden. Das Weiterschalten von Schritt zu Schritt erfolgt wie in den bisherigen Beispielen und Aufgaben besprochen. Die Befehlsausgabe kann zusätzlich noch von **Verriegelungsbedingungen** abhängig sein.

Beispiel A36.1

Programmierung einer Ablaufsteuerung mit Makrobefehlen

Die Analyse einer verbindungsprogrammierten Ablaufsteuerung sehen Sie im FUP von Bild A 37.1. Die VPS soll durch eine SPS ersetzt werden. Für diese Steuerung ist das Programm aufzustellen. Die Programmstruktur soll sich in Betriebsartenteil, Ablaufkette und Befehlsausgabe gliedern.

Wir fassen zunächst die erforderlichen Betriebsmittel zusammen. Sie sind mit den gewählten Operanden in Tabelle A36.1 zusammengestellt.

Tabelle A36.1: Zuordnungsliste zu Beispiel A36.1

Betriebsmittel-Kennzeichen	Funktion	Operand
Eingaben:		
S1	Taster für Start	E1
S2	Taster für Stop	E2
S3	Taster für Grundstellung	E3
S4	Schalter für Freigabe der Steuerung	E4
S5	Schalter für Freigabe der Befehlsausgabe	E5
S6	Schalter für Rücksetzen der Schritte	E6
Ausgaben:		
H1 bis H5	Meldeleuchten zum Anzeigen des Zustands der Ablaufschritte	A1 bis A5
H6	Meldeleuchte zur Anzeige des Betriebszustands der Steuerung	A6
H7	Meldeleuchte zur Überprüfung der Grundstellung	A7

Zur Abgrenzung der Aufgabenstellung und des Programmumfangs ist nur der **Automatikbetrieb** zu programmieren. Für die auszugebenden Befehle wurde zur Vereinfachung statt des Klartextes nur die Adresse der Ausgabe eingetragen. Gegenüber den bisher besprochenen Beispielen haben wir hier einen Unterschied in der Befehlsausgabe.

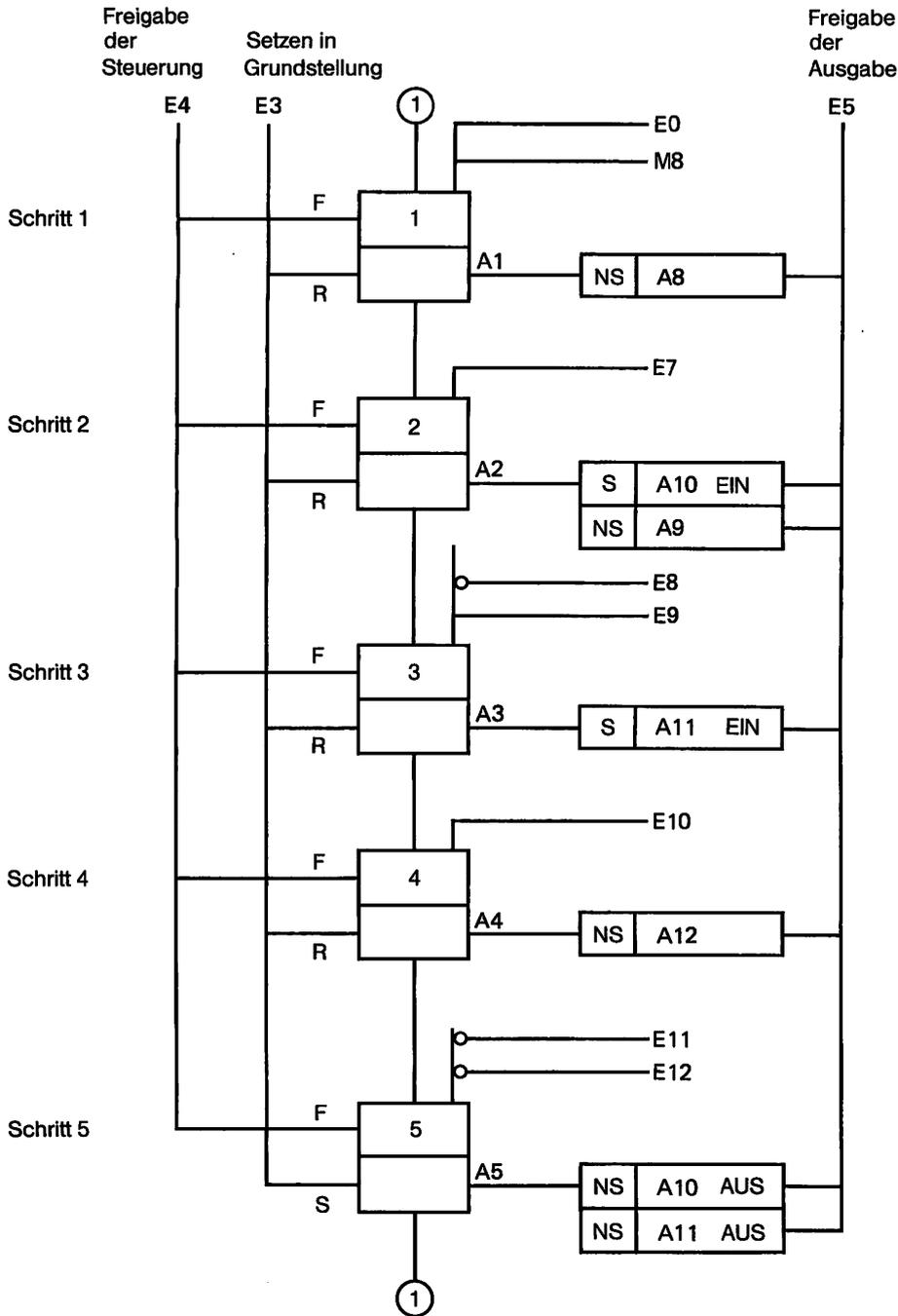


Bild A 37.1
Funktionsplan der Ablaufsteuerung zu Beispiel A 36.1.

Betrachten Sie zunächst Schritt 2 in Bild A 37.1. Die Ausgabe ist wie folgt zu lesen: Bei gesetztem Schritt hat der Schrittausgang A2 den Wert 1. Dieser Zustand setzt den S-Befehl A10 (A10 ist als Speicherfunktion zu programmieren). Außerdem ist noch die nichtspeichernde Befehlsausgabe (NS) A9 vorhanden. Die Befehle werden nur in den Prozeß eingegeben, wenn die Freigabe E5 den Zustand 1 hat. Ausgabe A10 bleibt auch dann noch bestehen, wenn Schritt 2 bereits wieder rückgesetzt ist. Mit dem Rücksetzen geht der NS-Befehl A9 in Zustand 0 über. In einem späteren Schritt wird der S-Befehl wieder gelöscht (Schritt 5 im Beispiel). Einen weiteren S-Befehl gibt Schritt 3 aus. Wir lösen die gestellte Aufgabe jetzt in folgenden Schritten:

- Programmierung des Betriebsartenteils
- Programmierung der Ablaufkette
- Programmierung der Ausgabe
- Gesamtprogramm und Programmtest

Lösung:

a) Programmierung des Betriebsartenteils

Wir müssen zunächst auflisten, welche Funktionen der Betriebsartenteil übernehmen soll. Daraus lassen sich die erforderlichen Anweisungen für die verschiedenen Programmteile ableiten. Diese werden anschließend miteinander zum Gesamtprogramm des Betriebsartenteils verbunden.

Funktionen des Betriebsartenteils

Er soll folgende Aufgaben übernehmen:

- Setzen der Speicherfunktionen in die Grundstellung,
- Überprüfung der Grundstellung,
- Ausgabe der Start- und Stop-Anweisung für die Steuerkette,
- Freigabe der Steuerung,
- Rücksetzen der Ablaufschritte (NOT-AUS) und
- Freigabe der Befehlsausgabe.

Wir stellen nun mit diesen Punkten die Anweisungen auf.

- Anweisung für das Setzen der Speicherfunktionen in die Grundstellung:

Vor Inbetriebnahme der Steuerung müssen die Speicherfunktionen der Schritte eine bestimmte Grundstellung einnehmen. In unserem Beispiel sind die Schritte 1 bis 4 in den rückgesetzten, Schritt 5 in den gesetzten Zustand zu bringen. Diese Anweisung wird mit S3 als Taster eingegeben (kurzes 1-Signal von E3).

Anweisung: $!E3=RA1=RA2=RA3=RA4=SA5$

Wegen der gesetzten Speicherfunktion von Schritt 5 ($A5=1$) wird das Setzen von Schritt 1 vorbereitet. Während des Steuerungsablaufs erfolgt durch die Anweisung „SA5“ gleichzeitig das Rücksetzen der gespeicherten Befehle A10 (Schritt 2) und A11 (Schritt 3).

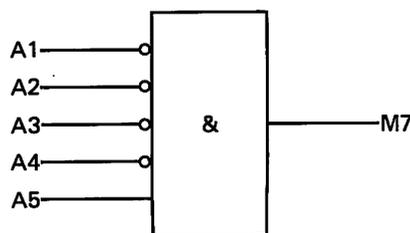


Bild A38.1
Abfrage des Grundzustands.

- Vor dem Start der Steuerung wird oftmals überprüft, ob die den Schritten zugeordneten Speicherfunktionen auch wirklich den Zustand der Grundstellung eingenommen haben. Die Abfrage der Grundstellung kann nach dem FUP von Bild A38.1 erfolgen.

Anweisung: $!NA1\&NA2\&NA3\&NA4\&A5=M7$

3) Nun sind die Start- und die Stop-Anweisung zu programmieren. Der Start mit S1 (E1=1) darf nur erfolgen, wenn sich alle Speicherfunktionen in der Grundstellung (M7=1) befinden. Die Startanweisung wird in Merker M8 gespeichert. Auch der Stop-Befehl (E2=1) soll nur wirksam werden, wenn sich die Ablaufkette gerade im Zustand der Grundstellung (M7=1) befindet. Mit E2=1 wird die Speicherfunktion M8 dominierend rückgesetzt. Beide Forderungen zusammen ergeben den FUP von Bild A 39.1.

Teilprogramm: $!E1 \& M7 = S M8$
 $!E2 \& M7 = R M8$
 $!M7 = A7$ Kontrolle der Signalzustände
 $!M8 = A6$ für Testzwecke

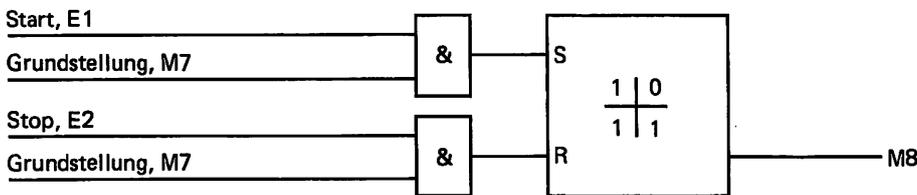


Bild A 39.1
FUP für die Start-Stop-Anweisung.

4) Der Steuerungsablauf wird mit Schalter S4 (E4) freigegeben oder gesperrt. Es gelten folgende Zustände:

- E4 hat Zustand 1: Steuerung ist freigegeben.
- E4 hat Zustand 0: Steuerung ist gesperrt.

Signal E4 wird vom Betriebsartenteil ausgegeben und unmittelbar im Programm der Ablaufschritte berücksichtigt.

5) Während des Betriebs werden bei Betätigung von Schalter S6 (E6) alle Schritte in den rückgesetzten Zustand gebracht. Gleichzeitig soll der Steuerungsablauf blockiert werden. Damit könnte beispielsweise die NOT-AUS-Funktion realisiert werden. E6=1 sperrt den Steuerungsablauf, wenn alle Speicherfunktionen der Ablaufschritte in Rücksetzdominanz programmiert sind. Es gelten folgende Zustände:

- E6=1: Speicherfunktionen werden rückgesetzt.
- E6=0: Speicherfunktionen werden nicht rückgesetzt.

Das Signal E6 wird vom Betriebsartenteil ausgegeben und unmittelbar im Programm der Ablaufschritte berücksichtigt.

6) Die von den Ablaufschritten kommenden Befehle A1 bis A5 dürfen nur zur Ausführung gelangen, wenn von Schalter S5 (E5) die Signalweitergabe freigegeben wird.

- E5=1 bedeutet: Befehlsausgabe freigegeben.
- E5=0 bedeutet: Befehlsausgabe gesperrt.

b) Programmierung der Ablaufkette:

Die Programmierung kann unmittelbar nach dem FUP erfolgen. Im Bild A 41.1 ist der aufgelöste FUP wiedergegeben. Mit Ihrem bisher erworbenen Wissen sollten Sie nun selbst in der Lage sein, die Zusammenhänge zu erkennen und den FUP zu lesen. Ein ähnliches Beispiel haben wir ja bereits schon programmiert.

Programm:

!E0&E4&A5&M8=SA1		!E4&E10&A3=SA4	
!E3/E6/A2=RA1	Schritt 1	!E3/E6/A5=RA4	Schritt 4
!E4&E7&A1=SA2		!E4&NE11&NE12&A4=M11	
!E3/E6/A3=RA2	Schritt 2	!E3/M11=SA5	
!E4&NE8&E9&A2=SA3		!E6/A1=RA5	Schritt 5
!E3/E6/A4=RA3	Schritt 3		

c) Programm der Befehlsausgabe:

Aus unserem FUP in Bild A 41.1 können Sie unmittelbar folgende Anweisungen ablesen:

Programm:

!E5&A1=A8	!E5&A4=A12
!E5&A2=A9=SA10	!E5&A5=RA10=RA11
!E5&A3=SA11	

d) Gesamtprogramm:

Die erforderlichen Teilprogramme sind entwickelt und das Gesamtprogramm kann somit aufgestellt werden.

!E3=RA1=RA2=RA3=RA4=SA5	Setzen in Grundstellung
!NA1&NA2&NA3&NA4&A5=M7	Grundstellung überprüfen
!M7=A7	Anzeige Grundstellung
!E1&M7=SM8	Start
!E2&M7=RM8	Stop
!M8=A6	Anzeige Betriebszustand
!E0&E4&A5&M8=SA1	
!E3/E6/A2=RA1	Schritt 1
!E4&E7&A1=SA2	
!E3/E6/A3=RA2	Schritt 2
!E4&NE8&E9&A2=SA3	
!E3/E6/A4=RA3	Schritt 3
!E4&E10&A3=SA4	
!E3/E6/A5=RA4	Schritt 4
!E4&NE11&NE12&A4=M11	
!E3/M11=SA5	
!E6/A1=RA5	Schritt 5
!E5&A1=A8	
!E5&A2=A9=SA10	
!E5&A3=SA11	
!E5&A4=A12	
!E5&A5=RA10=RA11	Befehlsausgabe
!PE	

Bedeutung der Signale in Bild A 41.1

E1	Start der Ablaufsteuerung
E2	Stop der Ablaufsteuerung
E3	Grundstellung
E4	Freigabe der Steuerung
E5	Freigabe der Befehlsausgabe
E6	NOT-AUS, Rücksetzen aller Schritte

Weiterschaltbedingungen

E0	Schritt 1
E7	Schritt 2
NE8	Schritt 3
E9	
E10	Schritt 4
NE11	Schritt 5
NE12	

Schaltzustand von

A1	Schritt 1
A2	Schritt 2
A3	Schritt 3
A4	Schritt 4
A5	Schritt 5

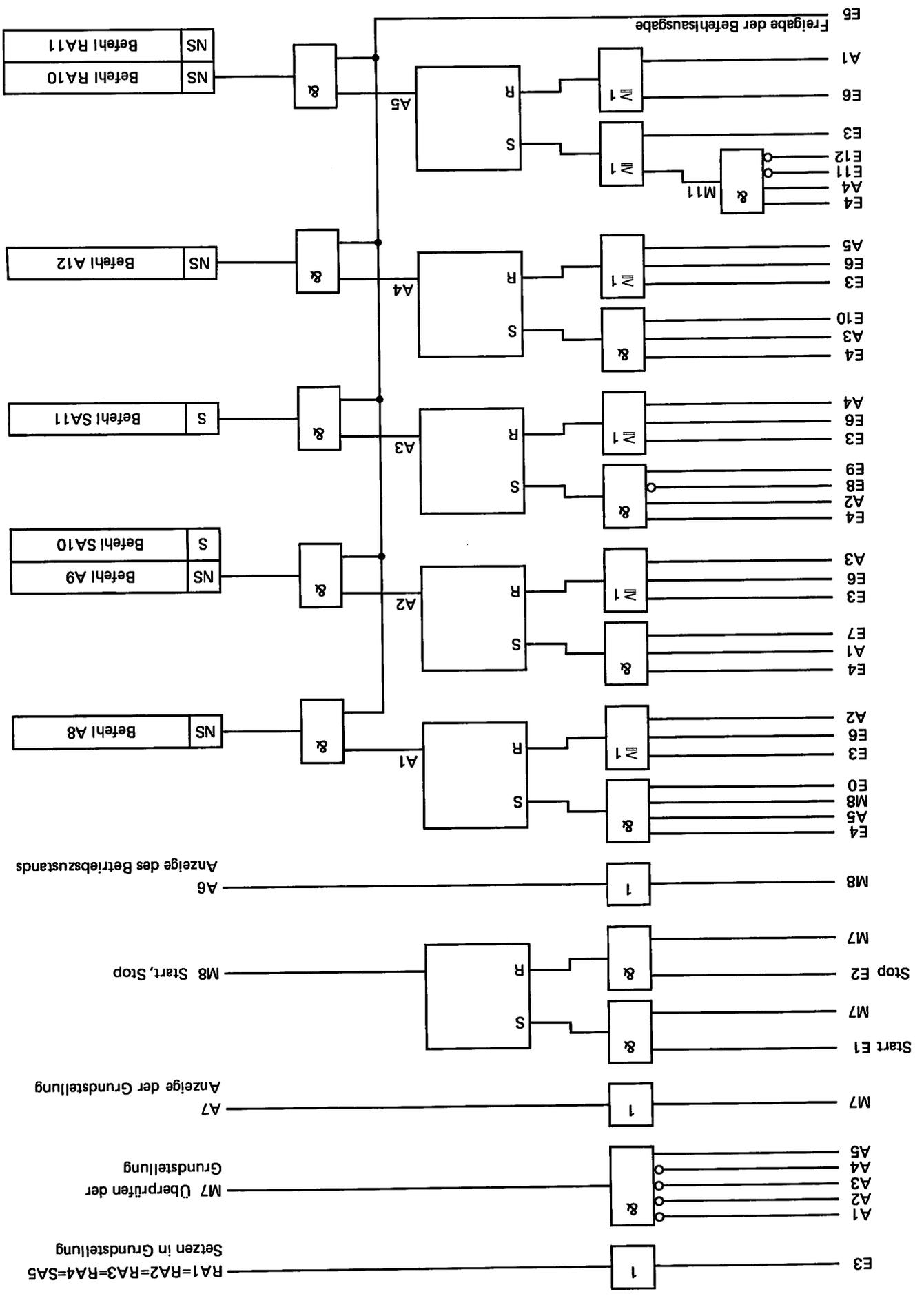


Bild A41.1
 FUP der Ablaufsteuerung von Beispiel A 36.1. Bedeutung der Signale siehe Seite A 40.

Nachstehend sind noch einmal die Signale zusammengestellt, die vom Betriebsartenteil ausgegeben werden. Sie müssen im Programm der Ablaufkette entsprechend berücksichtigt werden.

- M8 Start- bzw. Stop-Signal
- E4 Freigabe der Ablaufsteuerung
- E5 Freigabe der Befehlsausgabe
- E6 Rücksetzen der Ablaufschritte

Der FUP der Gesamtsteuerung (Bild A 41.1) enthält auch die Signalverarbeitung im Betriebsartenteil.

Nun folgt der Programmtest, der einige Konzentration erfordert. Da das Programm sehr viele Signalein- und ausgaben enthält, ist es etwas schwierig, die Übersicht zu behalten. Außerdem müssen die Signale in der Reihenfolge gegeben werden, wie sie in der praktisch ausgeführten Steuerung auftreten würden. Grundlage für den Test ist der FUP nach Bild A 37.1. Wir geben Ihnen hier einige Hinweise.

a) Inbetriebnahme der Steuerung:

- Weiterschaltbedingungen auf „nicht erfüllt“ stellen,
- Grundstellung der Speicherfunktionen mit S3 (E3) durch kurzes 1-Signal (nicht eingeschaltet lassen!) einstellen,
- Freigabe der Steuerung mit S4 (E4=1 – eingeschaltet lassen),
- Startanweisung mit S1 (E1=1),
- Die Ausgänge A5, A6 und A7 zeigen 1-Zustand.

b) Betrieb der Ablaufsteuerung:

- Weiterschaltbedingungen in der richtigen Reihenfolge **nacheinander** eingeben (kurze 1- bzw. 0-Signale). Den jeweiligen Schaltzustand zeigen die Ausgänge A1 bis A5 an.

c) Befehlsausgabe:

- Sie erfolgt nur bei vorhandenem Freigabesignal, mit S5 wird das Signal E5=1 eingestellt.

d) Anhalten der Steuerung:

- Diese Anweisung wird bei folgender Signalkombination ausgeführt: S2=E2=1. Nach dem Durchlaufen der Kette hält die Steuerung an.

Dieses Beispiel zeigt Ihnen, daß die verhältnismäßig einfache Lösung doch schon eine erhebliche Zahl von Anweisungen enthält. Die erforderliche Speicherkapazität für das Programm erhöht sich mit weiteren Anforderungen an die Steuerung sehr schnell.

Zusammenfassung

Demonstriert wurde mit diesem Beispiel der Programmaufbau in Programmblöcken. Jeder Block bearbeitet eine bestimmte Funktion. Das Gesamtprogramm wird dadurch übersichtlicher. Auch Programmänderungen lassen sich leichter ausführen. Häufige Anwendung findet diese Art der Programmierung in der industriellen Praxis.