

Technische Notiz
DESY Hasylab - F41 79/01
Oktober 1979

Eigentum der	Physik	Bibliothek
Property of	DESY	Library
Zurung:	8. NOV. 1979	
Accession:		
Leihfrist:	7	Tage
Loan period:		days

Eine binäre
speicherprogrammierte Ablaufsteuerung

K. Geske, R. Maschuw^{*)}, H. Riege, R. van Staa
Technische Entwicklungs- und Betriebsgruppe
II. Institut für Experimentalphysik
Universität Hamburg
1979

^{*)} jetzige Adresse: Institut für Kernphysik
Kernforschungszentrum Karlsruhe

Inhaltsverzeichnis

1. Einleitung
2. Steuerungskonzepte
3. Der elektronische Aufbau der Steuerung
 - 3.1 Die zentrale Prozessoreinheit
 - 3.2 Die Ein/Ausgabeeinheit
4. Das Steuerungsprogramm
5. Die HASYLAB-Pumpstandsteuerung
6. Zusammenfassung

1. Einleitung

Ausgangspunkt der Entwicklung der hier beschriebenen Steuerung war das Vorhaben, für das geplante Hamburger Synchrotronstrahlungslabor HASYLAB die Turbomolekularpumpstände zu fertigen sowie das Strahlrohrvakuuminterlock für alle Meßplätze zu erstellen. Da die Strahlrohrführung zu den einzelnen Experimentierplätzen unterschiedlich verläuft und damit individuelle Interlockbedingungen erforderlich sind, ergab sich die Notwendigkeit der Herstellung vieler prinzipiell gleichartiger, im Detail aber unterschiedlicher Steuerungen. Um den hierfür erforderlichen Aufwand in Grenzen zu halten, wurde das Konzept verfolgt, eine generalisierte hochflexible Steuerung zu entwickeln, welche für die genannten Aufgaben eingesetzt und an die individuelle Problemstellung leicht angepaßt werden kann.

Die für das gewählte Konzept ausschlaggebenden Gesichtspunkte werden im nächsten Abschnitt behandelt. Es folgen zwei Kapitel, welche die Hardware und die Programmverwirklichung der Steuerung beschreiben. Abschließend wird als Beispiel die Pumpstandsteuerung vorgestellt.

2. Steuerungskonzepte ¹⁾

Bei der Projektierung einer Steuerung ist zunächst die Frage zu stellen, von welcher Art die Informationen sind, welche verarbeitet werden sollen. Von analoger bzw. digitaler Steuerung spricht man, wenn die Eingangs- und Ausgangssignale überwiegend als Analogimpulse bzw. als Mehrbitwörter vorliegen, wie z.B. bei der Verarbeitung kontinuierlicher Meßwerte oder der Regelung von Stellgliedern. Typische Funktionseinheiten solcher Steuerungen sind Register, Zähler und heute auch schon Mikroprozessoren. Fallen dagegen nur logische Informationen an (jede Variable hat nur die Werte "0" und "1"), so kommt man mit der sog. binären Steuerung aus. Die wichtigsten Funktionseinheiten sind hier Baugruppen für die logischen Operationen, Speicher und Zeitglieder. Da die binäre Steuerung in der Regel technisch weniger aufwendig ist, wird man versuchen, nach Möglichkeit den zu steuernden Prozeß so auszulegen, daß aus dem Prozeß kommende Meßwerte über Diskriminatoren abgefragt werden und Stellglieder mit nur zwei Zuständen eingesetzt werden können, sodaß nur binäre Ein- und Ausgangssignale auftreten. Bei den hier diskutierten Aufgaben "Pumpstandsteuerung" und "Vakuuminterlock" ist das durchführbar.

Nach der Festlegung der Art der zu verarbeitenden Daten gilt es zu entscheiden, wie die Signalverarbeitung erfolgen soll. Man unterscheidet hier im wesentlichen zwei Möglichkeiten:

a) die Verknüpfungssteuerung, bei der den Signalzuständen der Eingangsdaten bestimmte Werte der Ausgangssignale im Sinne von Verknüpfungsfunktionen zugeordnet werden, und

b) die Ablaufsteuerung, bei der aus den möglichen Wertekombinationen der Ausgangssignale bestimmte erlaubte Steuerungszustände definiert werden, die abhängig von den Eingangsdaten sequentiell durchlaufen werden.

Unter Eingangssignalen sollen dabei die Informationen verstanden werden, welche vom Bedienungspult oder vom zu steuernden Prozeß selbst in die Steuerung fließen, während Ausgangssignale entsprechend Steuerbefehle an den Prozeß oder Meldungen an das Panel sind.

Die beiden Steuerungsarten sollen am folgenden Beispiel kurz erläutert werden. Die zu steuernde Anordnung bestehe aus einer Vorvakuumpumpe, einer Turbopumpe mit Ventil am Hochvakuumflansch und einer Druckmeßstelle in der Vorvakuumleitung zwischen Vorpumpe und Turbopumpe. Der Druck wird über zwei Diskriminatorschwellen abgefragt. Die Steuerung soll folgendes leisten: Nach Betätigung eines Schalters soll zunächst die Vorpumpe eingeschaltet werden. Das Unterschreiten der ersten Druckschwelle D1 soll den Start der Turbopumpe auslösen. Nach Erreichen der Druckschwelle D2 soll schließlich die Ventilschließung erteilt werden.

Bei diesem Problem gibt es also drei Eingangssignale:

PSE	Schalter "Pumpstand ein" betätigt
D1	Druckschwelle D1 unterschritten
D2	Druckschwelle D2 unterschritten

und drei Ausgangssignale:

VPE	Vorpumpe einschalten
TPE	Turbopumpe einschalten
VFG	Ventilschließung

Bei der Verknüpfungssteuerung hat man die Aufgabe, die Verknüpfungsfunktionen zwischen den Eingangs- und Ausgangsvariablen zu finden. Es sind $2^3 = 8$ Wertekombinationen der Eingangsvariablen möglich. In Abb. 1a sind die zugehörigen Werte der Ausgangsvariablen angegeben. Durch Einführung des Symbols \bar{I} (=: der Wert der Eingangsvariablen I ist bedeutungslos) läßt sich eine Reduktion auf vier Relationen durchführen (Abb. 1b). Aus dieser Liste ergeben sich sofort die Verknüpfungsfunktionen in Form von Boole'schen Gleichungen:

$$\begin{aligned} VPE &= PSE \\ TPE &= PSE \cdot D1 \\ VFG &= PSE \cdot D1 \cdot D2 \end{aligned}$$

Die Steuerung muß also so ausgelegt werden, daß diese drei Gleichungen erfüllt werden. Eine dem heutigen Stand der Technik entsprechende Lösung einer solchen Aufgabe ist der Einsatz eines sog. Field Programmable Logic Arrays (FPLA), eines hochintegrierten Netzwerkes von UND- und ODER-Gattern, in welchem durch Herausbrennen von Verbindungsleitungen logische Verknüpfungsfunktionen zwischen Ein- und Ausgangsvariablen programmiert werden können. Zum jetzigen Zeitpunkt (Frühjahr 1979) sind Bausteine verfügbar, in denen von 16 Eingangsvariablen bis zu 48 Produktterme gebildet und 8 Ausgangsvariablen zugeordnet werden können.

Die Vorzüge einer so realisierten Verknüpfungssteuerung liegen in dem sehr geringen Hardware-Aufwand und in den äußerst kurzen Reaktionszeiten (die Verzögerungszeit zwischen Ein- und Ausgang des FPLA liegt bei 50 nsec). Als nachteilig sind folgende Gesichtspunkte zu werten:

- a) Die Zahl ZE der Eingänge sollte den Wert 16 nicht wesentlich übersteigen, da die Anzahl der benötigten Bausteine wie 2^n steigt, wenn $ZE = 16 + n$.

	Eingang	Ausgang
0	PSE, $\bar{D1}$, $\bar{D2}$	VPE, TPE, VFG
1	PSE, D1, D2	VPE, TPE, VFG
2	PSE, D1, D2	VPE, TPE, VFG
3	PSE, D1, $\bar{D2}$	VPE, TPE, VFG
4	PSE, D1, D2	VPE, TPE, VFG
5	PSE, D1, D2	VPE, TPE, VFG
6	PSE, D1, D2	VPE, TPE, VFG
7	PSE, D1, D2	VPE, TPE, VFG

Abb. 1a: Zuordnung der Ausgangsvariablen zu den möglichen Wertekombinationen der Eingangsvariablen.

Eingang	Ausgang	Zusammenfassung von
PSE, $\bar{D1}$, $\bar{D2}$	VPE, TPE, VFG	0, 2, 4, 6
PSE, D1, $\bar{D2}$	VPE, TPE, VFG	1, 5
PSE, D1, D2	VPE, TPE, VFG	3
PSE, D1, D2	VPE, TPE, VFG	7

Abb. 1b: Reduktion der Tabelle aus Abb. 1a

b) Die Bildung der Verknüpfungsfunktionen kann bei einer größeren Zahl von Eingangsvariablen recht kompliziert werden und sogar Computerunterstützung erfordern. Dieses schränkt den Anwenderkreis ein und beeinträchtigt die Dokumentierbarkeit der fertigen Steuerungsprogramme.

Für Steuerungsaufgaben mit wenigen Eingangsvariablen oder mit hohen Anforderungen an die Reaktionsgeschwindigkeit ist die Verknüpfungssteuerung mit FPLA jedoch eine elegante und kostengünstige Lösung.

Soll das gleiche Problem mit einer Ablaufsteuerung gelöst werden, ist es notwendig, aus den 8 Wertekombinationen der Ausgangssignale die erlaubten Zustände festzulegen. In dem beschriebenen Beispiel sind dies die folgenden:

- Zustand 0: VPE, TPE, VFG
- Zustand 1: VPE, \overline{TPE} , \overline{VFG}
- Zustand 2: VPE, TPE, VFG
- Zustand 4: VPE, TPE, VFG

Die Steuerung befindet sich zu jeder Zeit in einem und nur einem dieser Zustände. Der Ablauf vollzieht sich durch Übergang von einem Zustand in einen anderen, wobei die Übergangsbedingungen durch die Eingangsvariablen festgelegt werden. Die Angabe der Zustände und die Richtung der erlaubten Übergänge erfolgt in einem sog. Zustandsdiagramm. Abb. 2 zeigt ein solches Diagramm für das angegebene Steuerungsbeispiel.

Bei der Programmverwirklichung einer Ablaufsteuerung hat in jüngster Zeit die Speicherprogrammierung immer mehr an Bedeutung gewonnen aufgrund der günstigen Kostenentwicklung der Halbleiterspeicher und Prozessoren. Der Speicher enthält das Steuerprogramm und kann entweder ein Schreib-Lese-Speicher (RAM) oder ein Nur-Lese-Speicher (ROM, PROM, EPROM) sein. Die einzelnen Anweisungen des

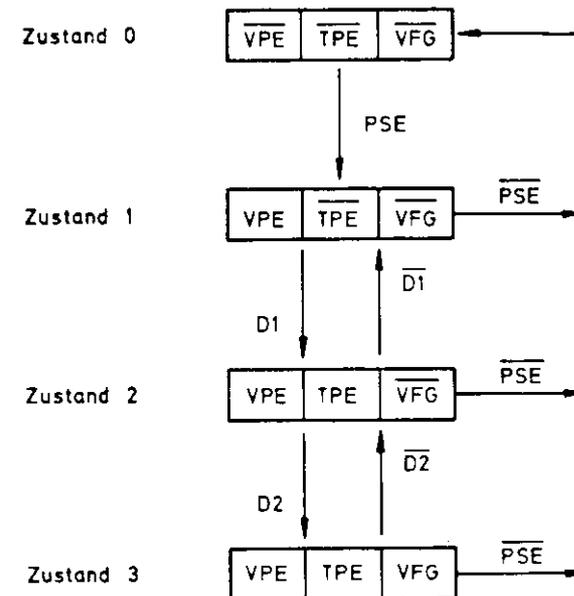


Abb. 2 : Zustandsdiagramm einer Ablaufsteuerung

Programms werden nacheinander vom Prozessor abgearbeitet, welcher den Signaltransfer und die Signalverarbeitung durchführt. Im Falle der Binärsteuerung setzt man einen sog. "Ein-Bit-Prozessor" ein, welcher nur logische Operationen ausführt und besonders einfach zu programmieren ist.

Der große Vorzug einer speicherprogrammierten Anordnung gegenüber der herkömmlichen Verbindungsprogrammierung liegt darin, daß der elektronische Aufbau für eine ganze Klasse von Steuerungsproblemen (z.B. für alle Binärsteuerungen) gleich ist und modular durchgeführt werden kann. Die Anpassung an die individuelle Aufgabe erfolgt über das gespeicherte Programm. Daher sprechen folgende Gründe für eine speicherprogrammierte Ablaufsteuerung:

- a) Geringe Entwicklungskosten bei der Realisation mehrerer ähnlicher Steuerungsaufgaben.
- b) Kurze Produktionszeit und kleine Fehlerwahrscheinlichkeit bei der Bestückung gedruckter Platinen, damit geringe Produktionskosten.
- c) Hohe Flexibilität bei der Problemanpassung, daher auch nachträgliche Änderung der Steuerungsaufgabe mit wenig Aufwand möglich.
- d) Bei Ausfall der Steuerung kurze Reparaturzeit durch Auswechseln von Platinen.
- e) Gute Dokumentierbarkeit des Steuerungsprogramms.
- f) Der Hardware-Aufwand steigt nur unwesentlich mit der Zahl der Ein- und Ausgangskanäle.

In wenigen Anwendungen könnte sich als nachteilig erweisen, daß die Steuerung je nach Programmumfang Reaktionszeiten in der Größenordnung von hundert Mikrosekunden haben kann.

Außerdem muß berücksichtigt werden, daß der Entwickler bzw. Anwender eine apparative Infrastruktur in Höhe von rund 10 kDM errichten muß, um ein Programm in den Speicher laden zu können, wenn er nicht von der Möglichkeit Gebrauch machen will, die Programmspeicherung im Auftragswege durchführen zu lassen.

3. Der elektronische Aufbau der Steuerung

Für die Lösung der in der Einleitung genannten Steuerungsaufgaben wurde das Konzept einer binären, speicherprogrammierten Ablaufsteuerung verfolgt. Der elektronische Aufbau ist modular und besteht aus der zentralen Prozesseinheit und der Ein/Ausgabe-Einheit. Sie sollen im folgenden beschrieben werden.

3.1 Die zentrale Prozesseinheit

Das Blockschaltbild dieser Schaltung zeigt Abb. 3. Die wesentlichen Bausteine sind der Programmschrittzähler, der Programmspeicher und der Ein-Bit-Prozessor. Aus Gründen der Störsicherheit werden nur CMOS-Komponenten bei einer Betriebsspannung von 15 V eingesetzt mit Ausnahme des Speichers, welcher zur Zeit in der erforderlichen Größe nur TTL-kompatibel auf dem Markt ist und daher über Level-Konverter mit der übrigen Elektronik verbunden ist.

Als Ein-Bit-Prozessor findet die ICU (Industrial Control Unit) MC 14500 B von Motorola Verwendung. Sie ist mit nur 16 Befehlen zu programmieren, sodaß ihr Befehlscode die vier Bits B₀ - B₃ umfaßt.

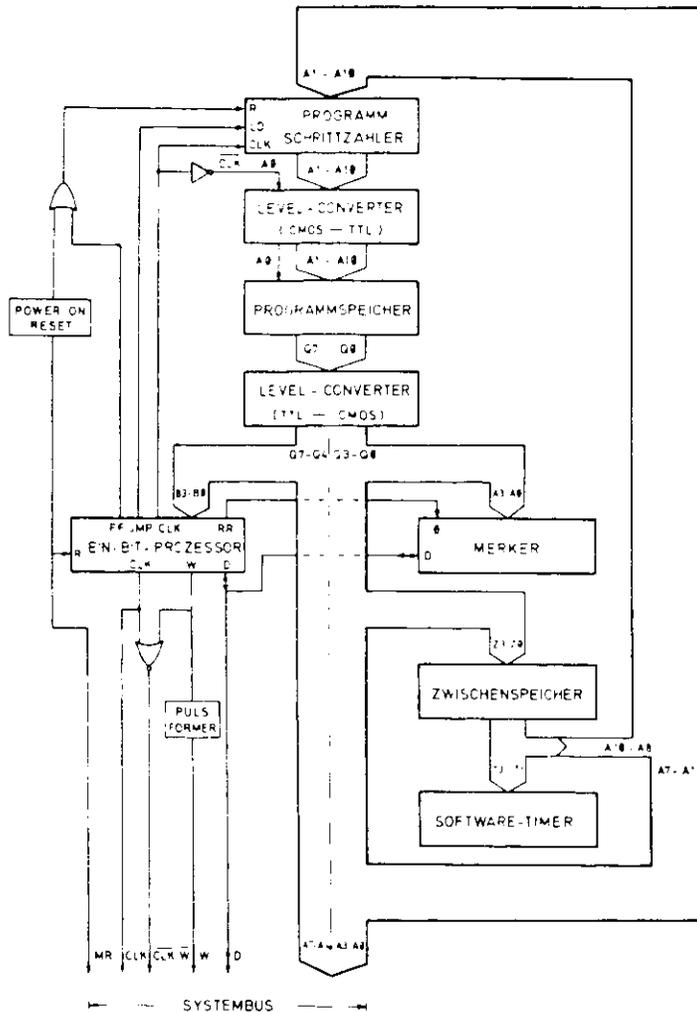


Abb. 3 . Blockschaltbild der zentralen Prozessoreinheit

Der Programmspeicher ist ein EPRCM (Erasable Programmable Read Only Memory) mit einer Kapazität von 2 K Bytes zu je 8 Bit. Je zwei Bytes bilden eine Programminstruktion, wobei die signifikanteren vier Bits $Q_{17} - Q_{14}$ des ersten Byte den Befehlscode für die ICU darstellen, während die restlichen zwölf Bits (Q_{13} bis Q_{10} des ersten Byte sowie Q_{27} bis Q_{20} des zweiten Byte) das Argument der Instruktion sind. In diesem Argument können auftreten:

a) Die Ein/Ausgabeadresse

Dabei bilden die Bits $Q_{27} - Q_{24}$ die sog. Stationsadresse, also die Nummer der angesprochenen Ein/Ausgabereinheit, während die Bits $Q_{23} - Q_{20}$ die Subadresse bestimmen. Die Stationsadresse 0 ist reserviert für die Merker, ein internes Datenregister, welches keine Verbindung zu den Ein/Ausgabekanälen hat und zum Ablegen von Zwischenergebnissen vorgesehen ist. Es können also im Prinzip 15 Stationen mit je 16 Ein/Ausgaberegistern adressiert werden.

b) Die Sprungadresse

Da der Programmspeicher eine Kapazität von 2 K Bytes besitzt, kann er 1024 Instruktionen aufnehmen. Daher muß die Sprungadresse aus 10 Bits bestehen. Sie setzt sich zusammen aus den sieben Bits $Q_{27} - Q_{21}$ sowie den drei Bits $Q_{12} - Q_{10}$, welche den signifikanteren Teil der Adresse bilden und bis zum Anliegen des zweiten Instruktionwortes im Zwischenspeicher abgelegt werden. Die Sprungadresse wird durch den Impuls der JMP-Flagge der ICU in den Programmschrittzähler geladen.

Hierbei handelt es sich um einen synchron ladbaren 10-Bit-Binärzähler, welcher durch die Systemclock inkrementiert wird. Diese Clock wird von der ICU bereitgestellt und synchronisiert den Programmablauf. Ihre Frequenz ist durch einen externen Widerstand auf rund 256 kHz festgelegt worden, kann aber im Prinzip zwischen 10 kHz und 1 MHz eingestellt werden.

Da in jeder Clock-Periode eine Programminstruktion mit zwei Bytes abgearbeitet werden muß, ist das Speicheradressbit A_0 mit der Clock verbunden, während die Bits $A_1 - A_{10}$ vom Programmschrittzähler geliefert werden. Eine Clock-Periode läuft dann folgendermaßen ab: Die positiv gehende Flanke der Clock schaltet den Programmschrittzähler weiter, während $A_0 = \text{CLK} = "0"$ wird, so daß der Programmspeicher das erste Byte der nächsten Instruktion bereitstellt. Mit der negativ gehenden Flanke der Clock übernimmt die ICU die Bits $Q_{17} - Q_{14}$ als Befehlscode $B_3 - B_0$, während die übrigen Bits $Q_{13} - Q_{10}$ im Zwischenspeicher abgelegt werden. Außerdem ist $A_0 = "1"$ geworden, sodaß mit einer gewissen Durchlaufverzögerung am Ausgang des Programmspeichers das zweite Byte der Instruktion erscheint und damit das Argument der Instruktion vollständig vorliegt. Mit der positiv gehenden Flanke der Clock führt die ICU den Befehl aus. Gleichzeitig wird der Programmschrittzähler weitergeschaltet, und ein neuer Zyklus beginnt.

Zur Bereitstellung von variablen Zeitintervallen verfügt die Prozessoreinheit über einen sog. Software-Timer, dessen Name daher rührt, daß die Intervalllänge vom Programm her über die vier Bits $Q_{13} - Q_{10}$ angewählt werden kann. Es lassen sich also Zeiten zwischen $15 \cdot T_b$ und $1 \cdot T_b$ anfordern, wobei die Zeitbasis T_b auf die Werte 2^n Sek. ($0 \leq n \leq 6$) hardwaremäßig festgelegt werden kann.

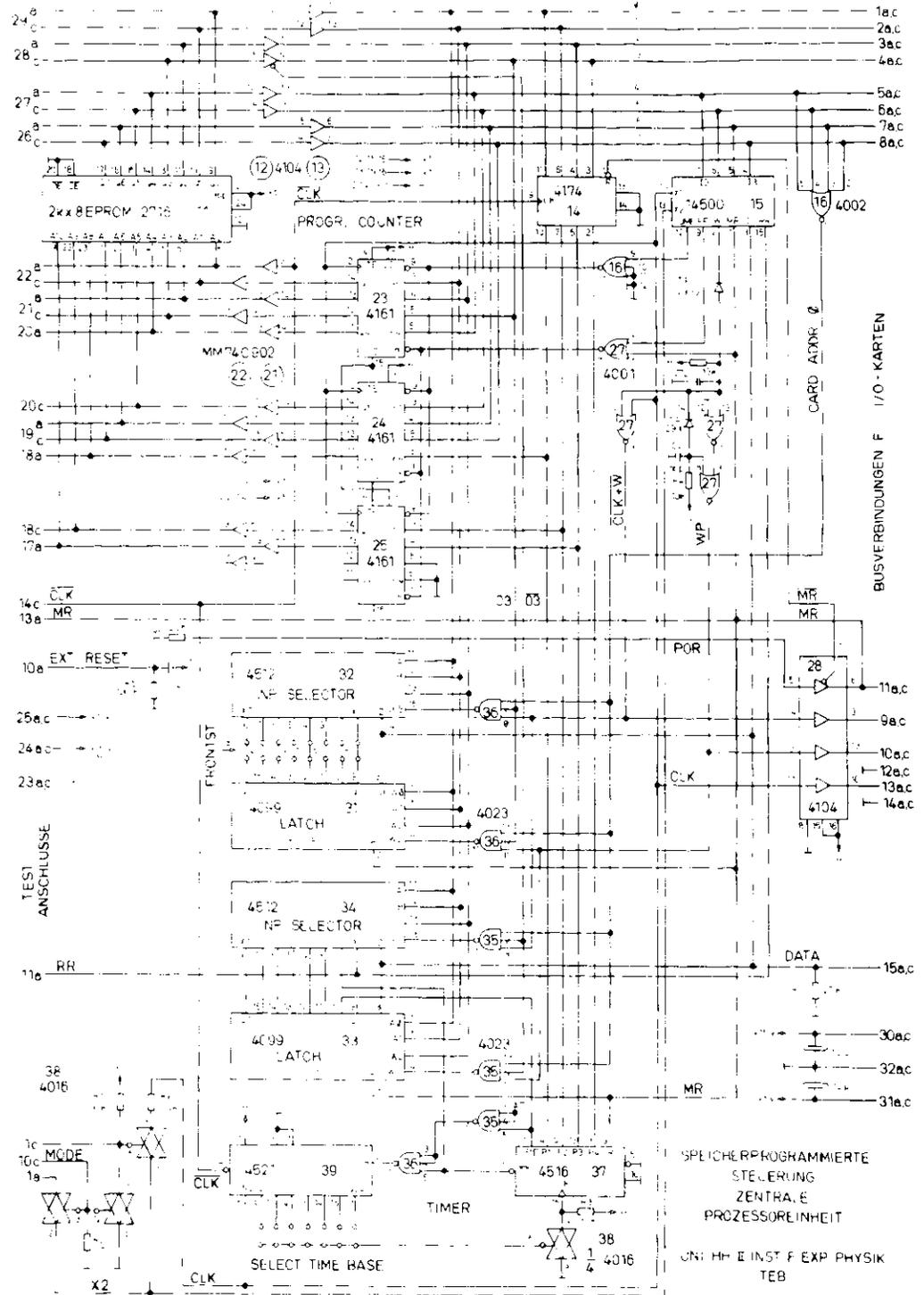


Abb. 4a

Die Verbindung der zentralen Prozessoreinheit mit den Ein-/Ausgabekarten erfolgt über einen Systembus. Dieser Bus enthält 8 Leitungen für die Adressierung der Ein-/Ausgaberegister, eine bidirektionale Datenleitung, zwei Leitungen für die Organisation des Schreib- und Lesevorganges sowie die Clock- und die Resetleitung. Ein Resetimpuls wird jeweils beim Einschalten des Systems durch die sog. Power-On-Reset-Einheit veranlaßt. Damit wird gewährleistet, daß das Programm mit der Speicheradresse 000 definiert startet.

Das Schaltbild der zentralen Prozessoreinheit zeigt die Abb. 4a. Man erkennt die drei Binärzähler 4161 des Programmschrittzählers, den Programmspeicher 2716 und den Prozessor 14500. Die Merker sind aus dem Input Selector 4512 und dem Output Latch 4099 aufgebaut, der Timer aus dem Untersetzer 4521 und dem Zähler 4516, welcher aus dem Zwischenspeicher 4174 parallel ladbar ist. Als Levelkonverter werden 74C902 (CMOS → TTL) und 4104 (TTL → CMOS) eingesetzt. Die letzteren dienen gleichzeitig als Treiber für den Systembus, der auf der Zeichnung nach rechts herausgeführt ist. Auf der gegenüberliegenden Seite führen die Leitungen auf einen Stecker, an den eine Diagnose-Karte angeschlossen werden kann. Diese Karte (Abb. 4b) erlaubt es, die Systemclock abzuschalten und das Programm über einen Taster im Einzelschrittbetrieb ablaufen zu lassen, wobei die Speicheradressen und die Instruktionsbytes hexadezimal dargestellt werden. Auf diese Weise ist eine schnelle Diagnose etwaiger Fehler sowohl der Hardware wie auch der Software möglich.

Der Strombedarf der Prozessorkarte beträgt 10 mA aus + 15 V sowie 65 mA aus + 5 V. Die Diagnosekarte belastet die 5 V-Spannung zusätzlich mit 430 mA.

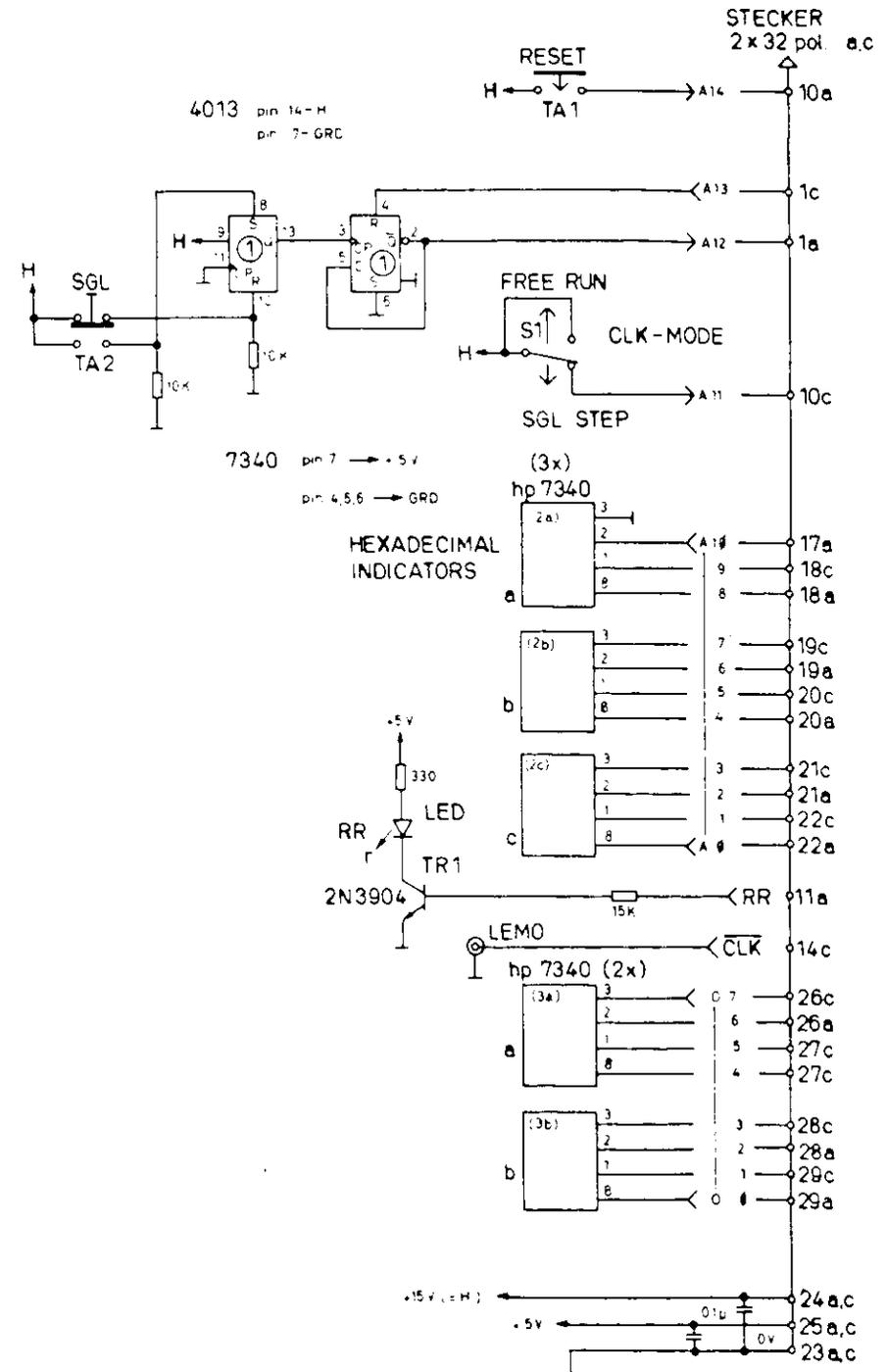


Abb. 4b: Schaltbild der Diagnose-Karte

3.2 Die Ein/Ausgabeeinheit

Die Ein/Ausgabeeinheit hat die Aufgabe, die Verbindung zwischen der zentralenessoreinheit einerseits und den Datenquellen und -senken des zu steuernden Prozesses andererseits herzustellen. Die wesentlichen Bausteine dieser Einheit sind daher die Ein- und Ausgaberegister, an deren Ein- bzw. Ausgängen die Informationen permanent anliegen (siehe Blockschaltbild Abb. 5). Die Zentrale kann auf einen bestimmten Kanal zugreifen, indem sie über die Adressierbits A7 - A4 die betreffende Karte anwählt und über die Bits A3 - A0 den gewünschten Kanal auf die Datenleitung schaltet. Ob es sich bei dem Datentransfer um einen Schreibvorgang (in das Ausgaberegister) oder einen Lesevorgang (aus dem Eingaberegister) handelt, wird über die beiden Leitungen "W" bzw. " \bar{W} · CLK" gesteuert. Ein Impuls auf der Resetleitung setzt bei jedem Einschaltvorgang sämtliche Ausgaberegister auf "0".

Prinzipiell können die Eingaberegister direkt und die Ausgaberegister über Treiberstufen an die Datenquellen und -senken des Prozesses angeschlossen werden. Zur Erhöhung der Störsicherheit ist jedoch auch die Möglichkeit der galvanischen Trennung der Zuleitung durch Optokoppler gegeben.

Im Decoder wird von Hand eine Zahl zwischen 1 und 15 eingestellt als Stationsadresse, unter der die betreffende Karte vom Programm aufgerufen werden soll. Er entschlüsselt die vier Bits A7 - A4 und aktiviert bei Übereinstimmung mit der vorgegebenen Adresse die Register. Zusätzlich ist die Einheit mit je zwei Timern ausgerüstet, deren Zeitintervalle hardwaremäßig unabhängig voneinander auf Werte zwischen 1 Sekunde und 15 Minuten eingestellt werden können.

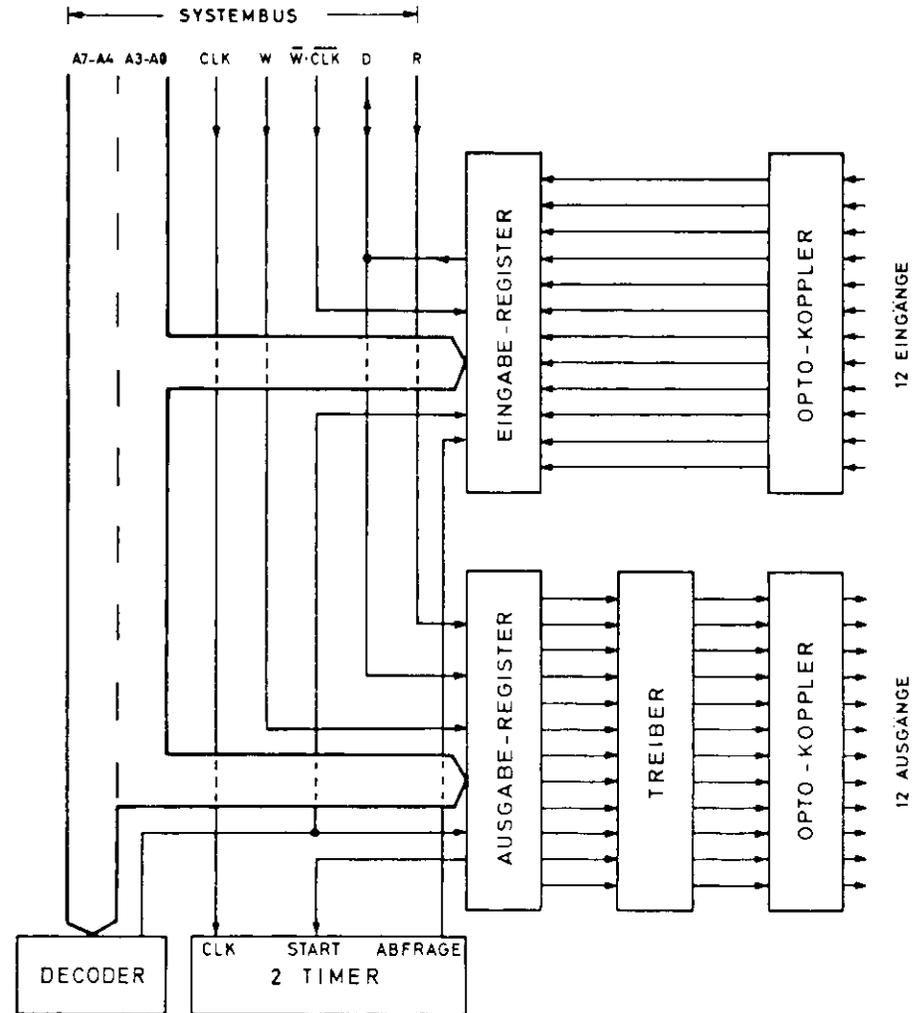


Abb. 5 : Blockschaltbild der Ein / Ausgabeeinheit

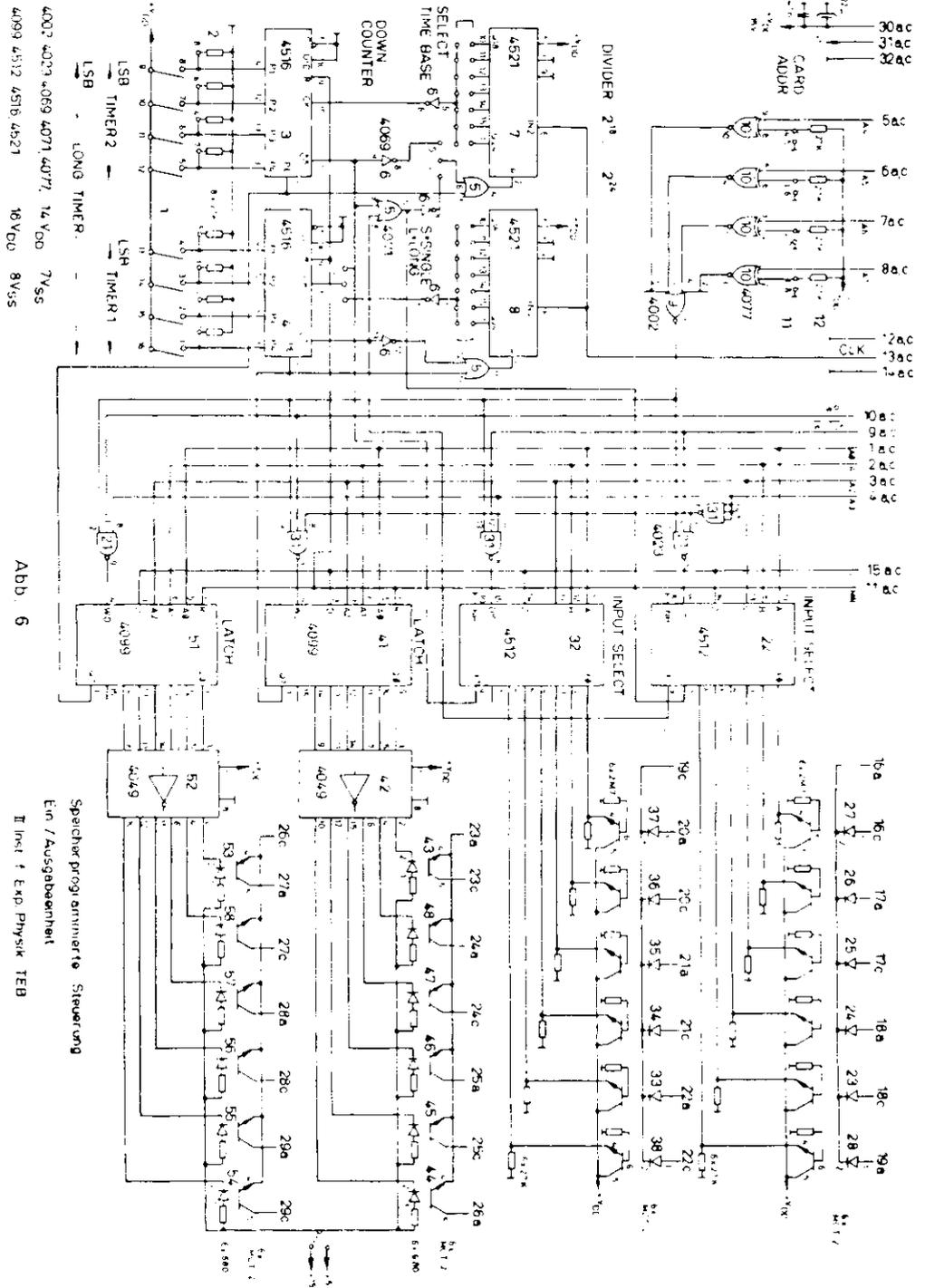
Der Start jeder Uhr erfolgt über einen Ausgabekanal, während die Abfrage der Zeit über einen Eingabekanal durchgeführt wird. Für größere Zeitintervalle ist es möglich, die zwei Timer einer Karte zu einem sog. Long Timer zusammenzuschalten. In diesem Fall kann die Zeitdauer bis auf 256 Minuten ausgedehnt werden.

Die Schaltung der Ein/Ausgabeeinheit ist vollständig aus Bausteinen der CMOS-Reihe aufgebaut (Abb. 6). Die Betriebsspannung beträgt wiederum + 15 V. Von den 16 Eingängen der Input Selektoren 4512 und den 16 Ausgängen der Output Latches 4099 werden je 12 für die Ein- und Ausgabe sowie 3 bzw. 2 für die Timer verwendet. Die übrigen bleiben unbenutzt.

Die Treiber 4049 können am Ausgang im Zustand "0" mit 20 mA belastet werden. Die verwendeten Optokoppler MCT 2 haben ein Stromübertragungsverhältnis von 65%, sodaß bei Bestückung mit Optokopplern am Ausgang rund 13 mA geschaltet werden können. Dieses ist ausreichend, um Leuchtdioden zu betreiben und Halbleiterrelais durchzusteuern. Auf der Eingangsseite verlangen die Optokoppler einen Diodenstrom von 1 mA, um den nachfolgenden Input Selector in den logischen Zustand "1" zu versetzen. Beim Betrieb ohne Koppler ist dagegen ein Spannungslevel von + 15 V notwendig.

Der Kartenaddress-Decoder ist durch vier Exklusiv-Oder-Gatter (4077) und ein 4-fach-NOR(4002) realisiert. Die Einstellung der Adresse erfolgt über Dual-Inline-Schalter.

Die beiden Timer sind wie auf der Prozessorkarte aus je einem Untersetzter 4521 und einem Abwärtszähler 4516 aufgebaut. Die vorgewählte Zeit ergibt sich aus der Zeitbasis,



4002 4077 4069 4071 4077 14 V_{DD} 7Y_{SS}
4099 4512 4516 4521 16 V_{DD} 8V_{SS}

die durch Verbindung der entsprechenden Untersetzungsstufe mit dem folgenden Baustein in den Grenzen von 1 Sekunde bis 64 Sekunden eingestellt wird, und einem Multiplikator zwischen 1 und 15, der über Miniatorschalter binär gesetzt und beim Starten der Uhr in den Zähler geladen wird. Nachdem dieser auf 0 heruntergezählt ist, geht der Ausgang CO auf logisch "0" und signalisiert über den Input Selector dem Prozessor, daß die eingestellte Zeit abgelaufen ist. Die beiden Brücke "SL" vermitteln in der Stellung "S" den Einzelbetrieb der beiden Timer und in der Stellung "L" den Long-Timer-Mode. In dieser Betriebsart liefert der Baustein 7 die Zeitbasis, während die beiden Zähler zu 8 Binärstufen zusammengefaßt sind.

Der Strombedarf einer Karte aus 15 V beträgt 15 mA zusätzlich 20 mA pro angeschlossenen Ausgangskanal.

4. Das Steuerungsprogramm

Die Entwicklung des Steuerungsprogramms und damit die Anpassung der Steuerung an ein individuelles Problem erfolgt in drei Stufen. Im ersten Schritt, der I/O-Deklaration, werden sämtliche Eingangsvariablen (Informationen, welche vom Prozeß und vom Bedienungspult in die Steuerung hineinlaufen) und Ausgangsvariablen (Daten, die von der Steuerung in die Datensinken laufen) zusammengestellt und in Listen den Kanälen der Ein/Ausgabekarten zugeordnet, sodaß auf diese Weise die Adressen der Variablen festgelegt sind.

Im zweiten Schritt wird das Zustandsdiagramm des Steuerungsproblems erstellt. Als Zustand wird eine beliebige Wertekombination sämtlicher Ausgangsvariablen definiert.

Da jede Variable die Werte "0" und "1" annehmen kann, sind bei n Variablen 2^n Zustände möglich. Aus dieser in der Regel recht großen Zahl müssen diejenigen Zustände herausgesucht werden, welche von der Steuerung realisiert werden sollen. Im Beispiel Abb. 2 ist die Menge der acht möglichen Zustände auf vier reduziert worden.

Das Zustandsdiagramm wird vervollständigt durch die Angabe der erlaubten Übergänge zwischen den einzelnen Zuständen. Sie werden charakterisiert durch Pfeile, welche die Richtung des Überganges festlegen, und durch eine logische Verknüpfung der Eingangsvariablen, welche die Bedingung für den Übergang stellt.

Mit der Angabe des Zustandsdiagramms ist die Steuerungsaufgabe eindeutig beschrieben, vollständig dokumentiert und jederzeit leicht nachvollziehbar. Aus ihm ergibt sich zwanglos der dritte Schritt, die eigentliche Programmierung. Das Programm wird aus einzelnen Blöcken aufgebaut, in denen jeweils ein Zustand programmiert ist. Ein einzelner Block läßt sich in einem Flußdiagramm schematisieren (Abb. 7):

Nach dem Einsprung in den Block wird zunächst allen Ausgangsvariablen der Wert zugeordnet, den das Zustandsdiagramm vorschreibt, unabhängig davon, ob gegenüber dem vorherigen Zustand eine Änderung eingetreten ist oder nicht. Danach werden die Aussprungsbedingungen abgefragt, und bei positivem Ergebnis erfolgt der Übergang in den der erfüllten Bedingung zugeordneten Zustand. Andernfalls geht das Programm an den Anfang des Blockes zurück und setzt wieder alle Ausgangsvariablen. Auf diese Weise wird eine hohe Störsicherheit erreicht, da der Blockzyklus permanent durchlaufen wird, solange kein Aussprung erfolgt.

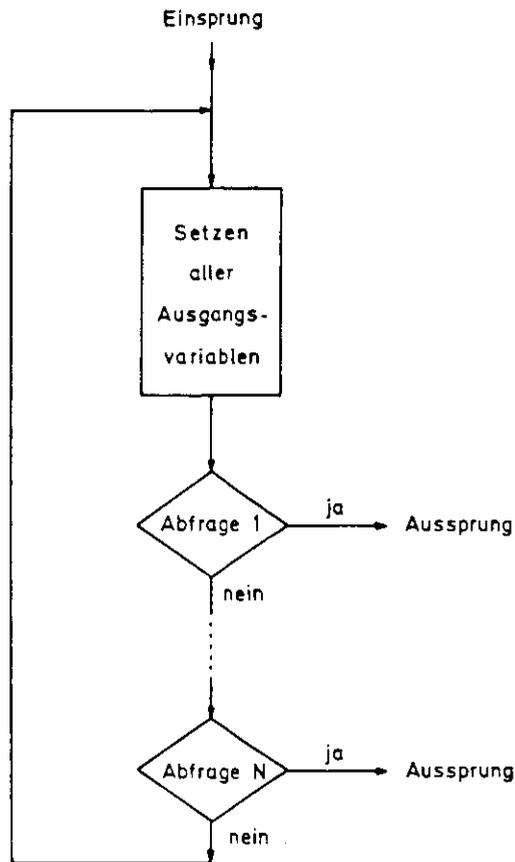


Abb. 7 : Flußdiagramm eines Zustandsblockes

Die maximale Reaktionszeit, d.h. die Zeit, welche die Steuerung im ungünstigsten Falle benötigt, um auf eine Änderung der Eingangssignale zu reagieren, errechnet sich aus der Blocklänge (= Zahl der Instruktionen) multipliziert mit der Instruktionszeit von rund 4 µsec.

Für die Umsetzung des Flußdiagramm in das eigentliche Programm ist die Kenntnis des Befehlscodes der ICU notwendig. Abb. 8 zeigt eine Liste der 16 möglichen Befehle mit den mnemonischen Abkürzungen und dem zugehörigen hexadezimalen Code²⁾.

Es gibt die vier Datentransferbefehle STO, STOC, LD, LDC, die den Transport von Daten zwischen dem Result Register RR des Prozessors und einem externen Speicher veranlassen. Aber auch die fünf logischen Verknüpfungen AND, ANDC, OR, ORC und XNOR beinhaltet einen Ladebefehl, da zunächst eine Information in den Prozessor transferiert wird, ehe sie mit dem Inhalt des Result Register verknüpft wird. Alle diese Befehle benötigen daher im Argument eine Speicheradresse, d.h. eine Kartenadresse K und eine Subadresse S. Die aus zwei Bytes bestehende Instruktion hat daher die Form

I X
K S

wobei I der entsprechende ICU-Befehl und X eine hexadezimale Ziffer ohne Bedeutung ist.

Mit den beiden Befehlen IEN und OEN kann die Eingabe von Daten in bzw. die Ausgabe von Daten aus dem Prozessor unterbunden werden, wenn der Wert der Variablen im Argument der Instruktion "0" ist. Beide Befehle benötigen daher ebenfalls eine Speicheradresse und ihre Instruktion hat auch die obige Form.

Befehl	Mnem	Cod	Wirkung
No Operation	NOP \emptyset	\emptyset	Wirkungslos
Load	LD	1	IS \rightarrow RR
Load Compl.	LDC	2	IS \rightarrow RR
And	AND	3	IS * IR \rightarrow RR
And Compl.	ANDC	4	IS * IR \rightarrow RR
Or	OR	5	IS + IR \rightarrow RR
Or Compl.	ORC	6	IS + IR \rightarrow RR
Exclusiv Nor	XNOR	7	"1" \rightarrow RR, falls IS = IR
Store	STO	8	IR \rightarrow S
Store Compl.	STOC	9	IR \rightarrow S
Input Enable	IEN	A	Eingabe ermöglicht, falls IS = "1"
Output Enable	OEN	B	Ausgabe ermöglicht, falls IS = "1"
Jump	JMP	C	Sprung auf die angegebene Programm- adresse
Return	RTN	D	Wirkungslos
Skip if Zero	SKZ	E	Überlesen des nächsten Befehls, falls IR = " \emptyset "
No Operation	NOPF	F	Sprung auf die Programmadresse 222

RR = Result Register, IR = Inhalt des Result Register
 S = Ein- bzw. Ausgabespeicher, der im Argument
 adressiert ist
 IS = Inhalt dieses Speichers

Lesebeispiel für den Befehl OR:
 Der Inhalt des Speichers S wird in den Prozessor ge-
 laden und dort mit dem Inhalt des Result Registers
 logisch ODER-verknüpft. Das Ergebnis dieser Operation
 wird im Result Register abgelegt.

Abb. 2: Befehlscode der ICU²⁾

Für einen Sprungbefehl ist die Angabe einer drei-
 ziffrigen hexadezimalen Adresse H₃H₂H₁, notwendig
 (s. 3.1). Die zugehörige Instruktion lautet also:

C H₃
 H₂H₁

Der RTN-Befehl ist für die Realisierung von Sub-
 routine-Strukturen gedacht. In der vorliegenden
 Steuerung ist diese Einrichtung nicht implemen-
 tiert, sodaß dieser Befehl ebenso wie der NOP \emptyset -
 Code keine Wirkung hat.

Der Befehl SKZ wird für die Programmierung von be-
 dingten Sprüngen benötigt. Er hat kein Argument,
 und wird daher in der Form

E X
 X X

verwendet.

Die durch den NOPF-Befehl gesetzte Flagge FF (s. Abb. 3)
 ist mit dem RESET-Eingang des Programmschrittzählers
 verbunden, sodaß die Instruktion

F X
 X X

den Rücksprung an den Anfang des Programms bewirkt.
 Es ist sinnvoll, sämtliche nicht benötigten Programm-
 speicherplätze mit dieser Instruktion zu belegen, da-
 mit im Störfall des Verlassens der programmierten
 Blöcke eine sofortige definierte Rückkehr in das Pro-
 gramm erzwungen wird.

Abb. 9 zeigt als Beispiel für ein Steuerungsprogramm den Block des Zustandes 2 aus Abb. 2. Die Startadresse des Blocks ist die hexadezimale Zahl 200. Mit der Instruktion ORC RR wird der Wert "1" in das Result Register geladen, welches hardwaremäßig mit dem Eingangskanal 00 verbunden ist, sodaß der Inhalt des Result Register hier für Ladebefehle zur Verfügung steht. Es folgt das Setzen der Ausgangsvariablen VPE, TPE und VFG durch Speichern des Wertes von RR bzw. seines Komplements in den entsprechenden Ausgangsregistern. Danach werden die Abfragen der Aussprungsbedingungen durchgeführt. Ist das Ergebnis negativ, so steht im Result Register der Wert "0", und durch den Befehl SKZ wird die zugehörige Sprunganweisung überlesen. Ist keine der Bedingungen erfüllt, so folgt am Schluß der Rücksprung an den Blockanfang.

Vier Blöcke dieser Art mit den Anfangsadressen 000, 100, 200 und 300 ergeben das vollständige Programm für das Zustandsdiagramm in Abb. 2.

5. Die HASYLAB-Pumpstandsteuerung

Den Aufbau des HASYLAB-Turbomolekularpumpstandes zeigt Abb. 10. Die wesentlichen Bestandteile sind eine zwei-stufige Drehschiebervorvakuumpumpe, eine Turbopumpe (Saugvermögen 110 l/sec), ein UHV-Abzweigstück und ein elektropneumatisches Eckventil. Die UHV-Teile sind mit Heizungen ausgerüstet. Es gibt eine Vorvakuumdrukmeßstelle mit zwei einstellbaren Schwellen und ein Penning-Meßsystem mit einer Schwelle. Für die Belüftung wird ein Stromausfallfluter eingesetzt, der (auch bei Netzausfall) mit einer Zeitverzögerung von 5 Sekunden öffnet.

I/O-Deklaration

Variable	Kanal
VPE	10
TPE	11
VFG	12
PSE	10
D1	11
D2	12

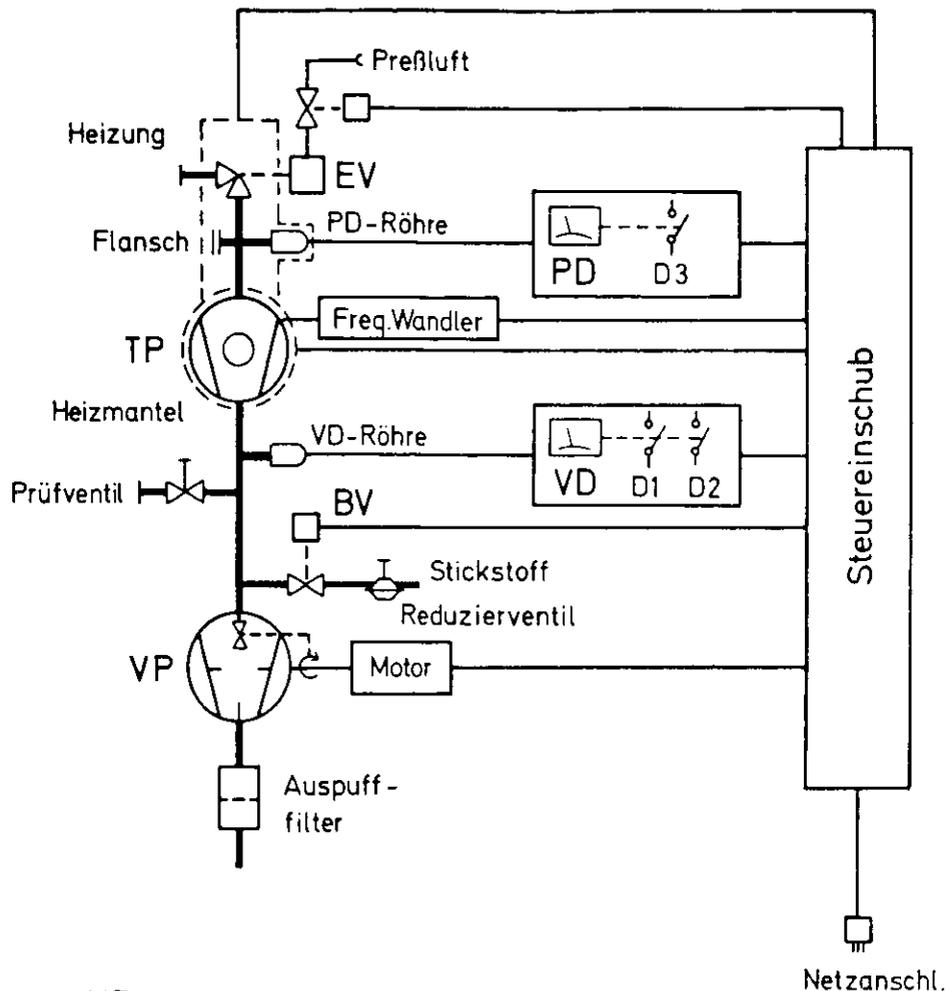
Startadressen

Zustand	Startadresse
0	000
1	100
2	200
3	300

Programm

Adresse	Mnemonic	Code	Bemerkung
000	ORC RR	6F	"1" → RR
		00	
000	STC VPE	8F	"1" → VPE
		10	
000	STC TPE	8F	"1" → TPE
		11	
000	STOC VFG	9F	"0" → VFG
		12	
000	LDC D1	2F	
		11	
00A	SKZ	EF	Falls D1 = "0", Sprung nach 100
		00	
00C	JMP 100	C1	
		00	
00E	LDC PSE	2F	
		10	
010	SKZ	EF	Falls PSE = "0", Sprung nach 000
		00	
012	JMP 000	C1	
		00	
014	LD D2	1F	
		12	
016	SKZ	EF	Falls D2 = "1", Sprung nach 300
		00	
018	JMP 300	C3	
		00	
01A	JMP 000	C2	Rücksprung nach 000
		00	

Abb. 9: Programm des Zustandes 2 aus Abb. 2



- VP Vorpumpe
- TP Turbopumpe
- VD Vorvakuum - Druckmeßgerät
- PD Penning - Druckmeßgerät
- EV Eckventil (elektro - pneumatisch, Viton - gedichtet)
- BV Belüftungsventil (elektro - magnetisch)
- D Druckschwelle

Abb. 10 HASYLAB - Pumpstand , Aufbau

Die I/O-Deklaration (Abb. 11, 12) zeigt, daß für die Steuerung des Pumpstandes zwei Ein/Ausgabeeinheiten benötigt werden. Karte I übernimmt die Überwachung und Schaltung der Pumpstandskomponenten. Man findet unter den Eingangsvariablen die drei Druckschwellen (D* - D3), die Drehzahlüberwachung der Turbopumpe (TPN) und den Ventilstellungsgeber (EVO). Mit DSN werden die drei Phasen und der Drehsinn des Drehstromes kontrolliert, mit VPN ein möglicher Überstrom in der Vorpumpe gemeldet. Auf der Ausgabe-Seite sind die Kanäle zu finden, über die das Schalten der Aggregate erfolgt. Zusätzlich sind hier noch vier Zustandsbits untergebracht, welche eine Hexadezimalanzeige für die Zustandsmeldung treiben.

Auf der Karte II sind eingangsseitig die Schalter und Taster für die Bedienung des Pumpstandes zusammengefaßt, während die Ausgangskanäle die Leuchtdiodenanzeigen steuern.

Das Zustandsdiagramm (Abb. 13) umfaßt 10 Zustände. In jedem Zustand wird der Wert der Ausgangsvariablen von Karte I angegeben. Die Taste "Lampenprüfung" (TLP) führt alternativ in zwei Unterzustände, in denen die Ausgangskanäle von Karte II zusammengefaßt sind. Bei gedrückter Taste sollen alle Signallampen (A) leuchten, im anderen Fall (TLP) nur diejenigen, welche im Diagramm aufgeführt sind.

Einige Ausgangsvariablen sind mit einem Index versehen wie z.B. BVZ im Zustand Ø. Damit wird angedeutet, daß der Wert der Variablen nicht durch den Zustand, sondern durch eine direkte Verknüpfung mit einer Eingangsvariablen (in diesem Beispiel mit EVO) definiert ist. In diesen wenigen Fällen wird also vom Konzept der reinen Ablaufsteuerung abgewichen, und es werden Elemente der Verknüpfungssteuerung mitbenutzt. Der Grund liegt darin, daß auf diese Weise die Zahl der Zustände reduziert und damit der benötigte Programmspeicherplatz verringert werden kann. Die Struktur der Ablaufsteuerung wird dadurch nicht verwischt.

Projekt : HASYLAB - Pumpstand

Karten - Nr : I

Inputs

Mnem.	Name	Adresse hexad.	Stecker	Opto koppl.	Bemerkung
D1	Druckschwelle 1	10	16c	x	
D2	" 2	11	17a	x	
D3	" 3	12	17c	x	
TPN	Turbopumpe normal	13	18a	x	
EVO	Eckventil offen	14	18c	x	
DSN	Drehstromnetz normal	15	19a	x	
TO2	Timer off (nach 4Std)	17	/		
VPN	Vorpumpe normal	18	20a	x	
		19	20c		
		1A	21a		
		1B	21c		
		1C	22a		
		1D	22c		
/	Timer	1F	/		

Outputs

Mnem.	Name	Adresse hexad.	Stecker	Opto koppl.	Bemerkung
BVZ	Belüftungsventil zu	10	23c	x	
VPE	Vorpumpe ein	11	24a	x	
TPE	Turbopumpe ein	12	24c	x	
HVE	Hochvakuummeterein	13	25a	x	
EVA	Eckventil auf	14	25c	x	
HZE	Heizung ein	15	26a	x	
TS2	Timer Start	17	/		
ZB0	} Zustandsbits	18	27a		
ZB1		19	27c		
ZB2		1A	28a		
ZB3		1B	28c		
		1C	29a		
		1D	29c		
TS2	Timer Start	1F	/		

Abb. 11 : I/O Deklaration der Pumpstandsteuerung , Karte I

Projekt : HASYLAB - Pumpstand

Karten - Nr : II

Inputs

Mnem.	Name	Adresse hexad.	Stecker	Opto koppl.	Bemerkung
SPE	Schalter Pumpst. ein	20	16c	x	
THE	Taste Heizung ein	21	17a	x	
THA	" " aus	22	17c	x	
TVA	" Ventil auf	23	18a	x	
TVZ	" " zu	24	18c	x	
TLP	" Lampenprüfung	25	19a	x	
TO1	Timer off (n. 30 sec)	27	/		
TDE	Taste Druckwächter ein	28	20a	x	
		29	20c		
		2A	21a		
		2B	21c		
		2C	22a		
		2D	22c		
TO3	Timer off (n. 15 min)	2F	/		

Outputs

Mnem.	Name	Adresse hexad.	Stecker	Opto koppl.	Bemerkung
LHF	Lampe Heizungsfreigabe	20	23c	x	
LHE	" Heizung ein	21	24a	x	
LVF	" Ventilsfreigabe	22	24c	x	
LVA	" Ventil auf	23	25a	x	
LDF	" Druckw. freigabe	24	25c	x	
LDE	" Druckwächter ein	25	26a	x	
TS1	Timer Start	27	/		
LPE	Lampe Pumpst. ein	28	27a	x	
LAP	" Anpumpen	29	27c	x	
LPZ	" Pumpzeit	2A	28a	x	
LVS	" Vorpumpen-Störung	2B	28c	x	
LDS	" Drehstrom- "	2C	29a	x	
		2D	29c		
TS3	Timer Start	2F	/		

Abb. 12 : I/O Deklaration der Pumpstandsteuerung , Karte II

Die Pfeile geben die Richtung der erlaubten Übergänge an, wobei die aus der Eingangsvariablen gebildeten Boole'schen Ausdrücke die Übergangsbedingungen nennen. Dabei bedeutet das Zeichen "*" eine UND-Verknüpfung, das Zeichen "+" eine ODER-Verknüpfung und ein Querstrich über einer Variablen die Negation (der Wert der Variablen ist "0").

Mit Hilfe des Zustandsdiagramms lassen sich alle Funktionen des Pumpstandes erläutern: Nach dem Einschalten der Netzspannung wird die Steuerung durch den POWER-ON-RESET (POR) in den Zustand \emptyset versetzt, in welchem mit Ausnahme von BVZ allen Ausgangsvariablen der Wert "0" vorgeschrieben ist, d.h. alle Aggregate sind stromlos. Das Belüftungsventil ist nur dann offen (stromlos), wenn der Ventilstellungsgeber ein geschlossenes Eckventil (EVO) meldet, was bei störungsfreiem Betrieb der Fall ist.

Nach Betätigung des Schalters "Pumpstand ein" (SPE) schließt das Belüftungsventil, und die Vorpumpe startet (Zustand 1), falls keine Störung von der Drehstrom- und der Vorpumpenüberwachung gemeldet wird und das Eckventil geschlossen ist. Gleichzeitig wird der Timer T1 (30 sec) gestartet. Ist nach Ablauf dieser Zeitspanne die Vorvakuumdruckschwelle D1 (5 mbar) nicht unterschritten, schaltet der Pumpstand wieder ab und macht eine Störmeldung (Zustand 1 \emptyset). Im anderen Fall läuft die Turbopumpe an (Zustand 2), und ein zweiter Timer (T3 = 15 Min.) wird in Gang gesetzt, der die Hochlaufzeit der Pumpe überwacht, d.h. daß vor Ablauf dieser Zeit die Turbopumpe ihre Nennzahl erreicht und die Druckschwelle D2 (10^{-1} mbar) unterschritten sein muß. Dann wird im Zustand 3 das Hochvakuummeter eingeschaltet, andernfalls erfolgt ebenfalls eine Störmeldung. Zeigt das Penning einen Druck unter D3 (10^{-5} mbar) an, so wird die Freigabe für das Eckventil erteilt (Zustand 4).

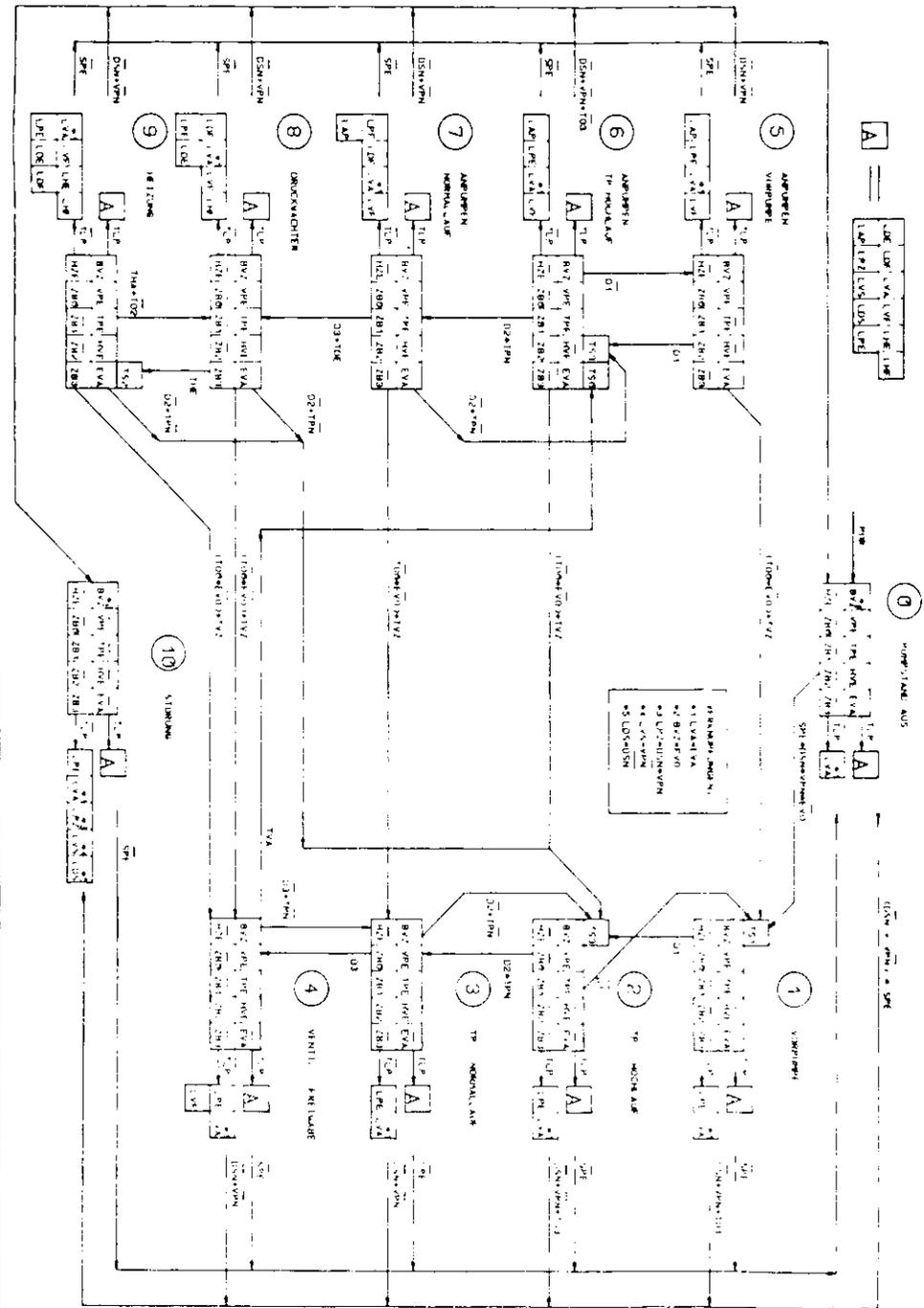


Abb. 13

UNIVERSITÄT DUISBURG
LEHRGEBIET ELEKTROTECHNIK

ZUSAMMENFASSUNG

TEB 790530

Nach Betätigung der Taste "Ventil auf" (TVA) öffnet das Eckventil. Gleichzeitig wird das Hochvakuummeter wieder abgeschaltet (Zustand 6), da mit einer Schockbelüftung zu rechnen ist. Der Timer T₁ (2 sec) berücksichtigt die Öffnungszeit des Ventils, indem für diese Zeitspanne die Abfrage nach der Rückmeldung des Ventilstellungsgebers überbrückt ist.

Ausgehend vom Zustand 6 begibt sich die Steuerung in Abhängigkeit von den Eingangsvariablen D1, D2, D3 und TPN in einen der Zustände 5 bis 8 in völliger Analogie zu dem Verhalten beim Übergang zwischen den Zuständen 1 bis 4, denn es ist ersichtlich, daß die Zustände n = 5, 6, 7, 8 mit Ausnahme des Ventilschalters EVA und der Anzeigelampen völlig identisch mit den Zuständen n-4 sind. Deshalb bewirkt die Betätigung der Taste "Ventil zu" (TVZ) bzw. das Fehlen der Rückmeldung des Ventilstellungsgebers zwei Sekunden nach Öffnen des Eckventils (TO \emptyset · EVO) einen Rücksprung in den korrespondierenden Zustand n-4.

In den Zuständen 5 bis 7 leuchtet eine gelbe Lampe mit der Aufschrift "Anpumpen". Damit wird angezeigt, daß für eine begrenzte Zeit des Anpumpens eines Rezipienten (z.B. eines Strahlrohres) die Ventilstellung nicht vom Vakuumdruck überwacht wird. Diese Phase, die vom Operateur kontrolliert werden sollte, wird im Zustand 7 automatisch durch Unterschreiten der Druckschwelle D3 oder manuell (die Lampe "Druckwächterfreigabe" leuchtet) durch Betätigung der Taste "Druckwächter ein" (TDE) beendet.

In der Betriebsart "Druckwächter" (Zustände 8 und 9) wird beim Überschreiten der Druckschwelle D2 oder beim Unterschreiten der Nenndrehzahl der Turbopumpe das Eckventil geschlossen, (Übergang in den Zustand 2), weil davon auszugehen ist, daß nach Beendigung der Anpumpphase ein plötzlicher Druckanstieg aus einem Leckauflbruch resultiert.

Im Zustand 8, also bei geöffnetem Eckventil und automatischer Überwachung, wird die Freigabe für die Heizung erteilt. Mit der Taste "Heizung ein" (THE) wird gleichzeitig der Timer T2 (256 Min.) gestartet (Zustand 9), der die Heizperiode automatisch beendet, wenn nicht vorher die Taste "Heizung aus" (THA) betätigt wurde (Übergang in den Zustand 8).

Eine Störmeldung in den Zuständen 0 bis 9 oder eine Zeitüberschreitung in den Zuständen 1, 2 oder 6 führt zum sofortigen Abschalten aller Aggregate mit Ausnahme des Belüftungsventils, welches solange zugehalten wird, bis der Ventilstellungsgeber das Schließen des Eckventils gemeldet hat (Zustand 10). Auf diese Weise wird verhindert, daß bei Störungen im Pumpstand ein Rezipient unbeabsichtigt geflutet wird. Die Störmeldung wird quittiert durch Abschalten des Pumpstandes (SPE).

Das Ausschalten des Pumpstandes kann in allen Zuständen erfolgen. Es bewirkt einen direkten Sprung in den Zustand 0 mit verzögerter Belüftung, wobei auch hier die Anweisung zum Fluten erst bei geschlossenem Eckventil erteilt wird.

In Abb. 14 ist als Beispiel die Programmierung des Zustandes 1 gezeigt. Die Startadresse ist die hexadezimale Zahl 100. Die ersten beiden Instruktionen laden eine logische "1" in das Result Register und aktivieren die Ausgabe der ICU. Die nächsten beiden Instruktionen erzeugen einen Rechteckimpuls im Ausgangskanal 27, womit der Timer T2 gestartet wird (s. I/O-Deklaration Abb. 12).

Adresse	Mnemonic	Code	Adresse	Mnemonic	Code	Adresse	Mnemonic	Code
100	ORC RR	6F	100	STO LPR	2F	160	SKZ	EF
1		00	1		2A	1		00
2	GEN RR	BF	2	STO LVS	2F	2	JHP 010	00
3		00	3		2B	3		10
4	STO TSA	2F	4	STO LDS	2F	4	LD 3A	1F
5		27	5		2C	5		10
6	STOC TSA	3F	6	STO LPE	2F	6	SKZ	EF
7		24	7		2B	7		00
8	ORC PR	6F	8	LDC TLP	2F	8	JHP 200	C2
9		00	9		25	9		00
A	GEN RR	BF	A	GEN RR	BF	A	JHP 108	CA
B		00	B		00	B		08
C	STO BYZ	2F	C	STO LPE	2F	C		
D		10	D		2B	D		
E	STO VPE	2F	E	STOC LHF	3F	E		
F		1A	F		20	F		
110	STOC TPE	3F	110	STOC LHF	3F			
1		12	1		21			
2	STOC HVE	3F	2	STOC LVE	3F			
3		13	3		22			
4	STOC EVA	3F	4	STOC LDF	3F			
5		14	5		24			
6	STOC HRE	3F	6	STOC LDE	3F			
7		15	7		25			
8	STO ZBZ	2F	8	STOC LAP	3F			
9		18	9		29			
A	STOC ZB1	3F	A	STOC LPR	3F			
B		19	B		2A			
C	STOC ZB2	3F	C	STOC LVS	3F			
D		1A	D		2B			
E	STOC ZB3	3F	E	STOC LDS	3F			
F		1B	F		2C			
120	GEN TLP	BF	150	LD EVO	2F			
1		25	1		1			
2	STO LHF	2F	2	STO LVA	2F			
3		20	3		23			
4	STO LHE	2F	4	LDC B30	2F			
5		21	5		15			
6	STO LVE	2F	6	ORC VPA	6F			
7		22	7		1D			
8	STO LVA	2F	8	ORC TLP	6F			
9		23	9		29			
A	STO LDF	3F	A	SKZ	EF			
B		24	B		20			
C	STO LDE	3F	C	JHP ZBZ	00			
D		25	D		10			
E	STO LAP	2F	E	LDC LPE	2F			
F		20	F		20			

Die Adresse 108 ist die Rücksprungadresse für den Zustandszyklus. Daher werden hier zur Erhöhung der Störsicherheit noch einmal die ersten beiden Instruktionen wiederholt. Es folgen die zehn Speicherbefehle, welche die vorgeschriebenen Werte der Ausgangsvariablen von Karte I setzen.

In der Adresse 120 beginnt die Verzweigung in die beiden Unterzustände. Ist die Taste "Lampenprüfung" gedrückt (TLP), so wird die Ausgabe der ICU weiterhin ermöglicht, und die nächsten elf Instruktionen schreiben eine "1" in sämtliche Ausgangskanäle für die Lampen. Im anderen Fall ist die Ausgabe der ICU unterdrückt, und die elf folgenden Befehle werden zwar ausgeführt, bleiben aber ohne Wirkung. Dann ist aber TLP = "1", und in der Adresse 13A wird die Ausgabe wieder freigegeben zur Ausführung der nächsten zehn Speicherbefehle.

In den beiden Instruktionen 150 und 152 folgt die Steuerung der Anzeigelampe "Ventil auf" (LVA) in Abhängigkeit vom Ventilstellungsgeber, indem einfach der Wert von EVO nach LVA abgespeichert wird.

Abschließend werden die Aussprungbedingungen für eine Störmeldung (Zustand 10, Startadresse 200), für eine Pumpstandabschaltung (Zustand 1, Startadresse 010) und für den Übergang in den Zustand 2 (Startadresse 200) abgefragt. Die letzte Instruktion des Blocks ist der Sprungbefehl in die Zyklusstartadresse 108.

Der Block hat eine Länge von 56 Instruktionen. Die Reaktionszeit der Steuerung beträgt in diesem Zustand also maximal 224 µsec. Das gesamte Programm umfaßt rund 620 Instruktionen und füllt damit etwa 60% des vorhandenen Programmspeicherplatzes.

Abb. 14 : Programm für den Zustand 1 der HASYLAE - Pumpstandsteuerung

6. Zusammenfassung

Es wird eine binäre speicherprogrammierte Ablaufsteuerung vorgestellt. Der elektronische Aufbau ist modular und besteht aus einer zentralen Prozessoreinheit und aus einer oder mehreren Ein/Ausgabeeinheiten.

Die zentrale Prozessoreinheit arbeitet mit einem Ein-Bit-Prozessor (16 Befehle) und einem EPROM (2k x 8 bit) als Programmspeicher. Die Ausführungszeit für eine Instruktion beträgt 4 μ sec.

Die Ein/Ausgabeeinheit verfügt über je 12 Eingabekanaläle (+ 15 V-Level bei direktem Anschluß oder 1 mA Strom bei Verwendung von Optokopplern) und 12 Ausgangskanälen (Belastung 20 mA oder 13 mA bei direktem bzw. Optokoppleranschluß). Jede Karte ist zusätzlich mit zwei Timern (Zeitstufen wählbar zwischen 1 Sek. und 15 Min.) ausgerüstet, die wahlweise zu einem Timer mit maximal 256 Minuten zusammengefaßt werden können.

Die Steuerung kann für alle binären Probleme eingesetzt werden, deren Programmaufwand 1024 Instruktionen nicht überschreitet. Die individuelle Problemanpassung erfolgt softwaremäßig. Daraus ergeben sich folgende Vorteile:

- a) Geringer Aufwand für die Hardware
- b) Schnelle Lösung des Steuerungsproblems
- c) Hohe Flexibilität
- d) Gute Dokumentierbarkeit

Als Beispiel für eine Anwendung dieses Konzeptes wird die Steuerung des HASYLAB-Pumpstandes beschrieben.

Alle Rechte an der hier beschriebenen Steuerung liegen beim II. Institut für Experimentalphysik der Universität Hamburg.

Verzeichnis der Abbildungen

- Abb. 1a: Zuordnung der Ausgangsvariablen zu den möglichen Wertekombinationen der Eingangsvariablen.
- Abb. 1b: Reduktion der Tabelle aus Abb. 1a.
- Abb. 2 : Zustandsdiagramm einer Ablaufsteuerung
- Abb. 3 : Blockschaltbild der zentralen Prozessoreinheit.
- Abb. 4a: Schaltbild der zentralen Prozessoreinheit.
- Abb. 4b: Schaltbild der Diagnosekarte.
- Abb. 5 : Blockschaltbild der Ein/Ausgabeeinheit.
- Abb. 6 : Schaltbild der Ein/Ausgabeeinheit.
- Abb. 7 : Flußdiagramm eines Zustandsblocks.
- Abb. 8 : Befehlscode der ICU.
- Abb. 9 : Programm des Zustandes 2 aus Abb. 2.
- Abb. 10 : HASYLAB-Pumpstand, Aufbau.
- Abb. 11 : I/O-Deklaration der Pumpstandsteuerung, Karte I.
- Abb. 12 : I/O-Deklaration der Pumpstandsteuerung, Karte II.
- Abb. 13 : Zustandsdiagramm der HASYLAB-Pumpstandsteuerung.
- Abb. 14 : Programm für den Zustand 1 der HASYLAB-Pumpstandsteuerung.

Literatur

- 1) Steuerungstechnik. Begriffe, Normen, Darstellung.
Siemens AG, 1978
- 2) MC 14500B Industrial Control Unit Handbook
Motorola Inc., 1977

