EINFÜHRUNG IN DIE ARBEITSWEISE

der

programmgesteuerten Transistorrechenanlage

Z U S E Z 31

I Programmgesteuerte elektronische Rechenanlagen

I. PROGRAMMGESTEUERTE ELEKTRONISCHE RECHENANLAGEN

Die folgenden Ausarbeitungen sollen eine Einführung in die Arbeitsweise programmgesteuerter elektronischer Rechenanlagen geben. Im Kapitel I werden zunächst einige Grundlagen gebracht, die zum Verständnis der im Kapital II behandelten Rechenanlage ZUSE Z 31 erforderlich sind.

1. Informationsdarstellung in programmgesteuerten elektronischen Rechenanlagen

Eine programmgesteuerte elektronische Rechenanlage dient zur Verarbeitung von Daten (Zahlen und Text) nach einem bestimmten Programm (Folge von sog. Befehlen). Daten und Befehle sind Informationen, die in der Anlage in geeigneter Form „elektronisch" dargestellt werden. Aus technischen Gründen, auf die hier nicht näher eingegangen werden soll, werden die Informationen als Kombinationen von nur zwei Zustandswerten, den sog. Ja-Nein-Werten dargestellt. Man kann zum Beispiel definieren; daß

„Strom in einer Leitung" oder
„positiver Fluß in einem magnetischen Kreis" oder
„Ladung in einem Kondensator" oder
„ein geschlossener Schalter"

den Zustand „Ja" repräsentieren, die entgegengesetzten Begriffe jedoch den Zustand „Nein" bedeuten. In einem der genannten Elemente lassen sich also zwei verschiedene Informationen, eben „Ja" und „Nein" speichern. Man bezeichnet deshalb ein solches Element als eine „binäre Speicherzelle". (binär: aus zwei Einheiten bestehend.)

1.1 Darstellung von Zahlen im binären Zahlensystem

Zur Kennzeichnung der Zustände „Ja" und „Nein" werden zur Vereinfachung die Ziffern 1 und 0 geschrieben. Um Verwechslungen mit der 1 im dezimalen Zahlensystem zu vermeiden wird anstelle von „1" im folgenden „L" geschrieben.

Das binäre Zahlensystem besteht nur aus den Ziffern 0 und L und hat die Basis 2. Machen wir uns zunächst klar, was die Basis eines Zahlensystems bedeutet. Betrachten wir z.B. die Dezimalzahl 99.
Wir müßten eigentlich schreiben:

$$9 \cdot 10 + 9 \cdot 1 \quad \text{oder}$$
$$9 \cdot 10^1 + 9 \cdot 10^0 ,$$

lassen aber der Einfachheit halber die Zehnerpotenzen, d.h. in diesem Fall 10^1 und 10^0 weg und schreiben nur die sog. „Mantissen", hier 9 und 9, hin. Die Stellung der Ziffer bezogen auf das Komma der Zahl bestimmt definitionsgemäß ihren Wert eindeutig.

Das gleiche gilt für jedes Zahlensystem, also auch für das binäre. Die Zahl 99 in ausführlicher binärer Schreibweise würde dann lauten:

$$L \cdot 64 + L \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + L \cdot 2 + L \cdot 1 \quad \text{oder}$$
$$L \cdot 2^6 + L \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + L \cdot 2^1 + L \cdot 2^0 = 99$$

oder unter Weglassung der Zweierpotenzen analog der dezimalen Schreibweise:

L L 0 0 0 L L

Die Binärzahl ist 7-stellig, die dazugehörige Dezimalzahl jedoch nur 2-stellig. Um diese Zahl speichern zu können, sind also 7 Binärspeicherzellen oder 2 Dezimalspeicherzellen notwendig. Im Mittel gilt:

Die Anzahl der Binärziffern (bit) einer Binärzahl ist um den Faktor 3,3 größer als

die Anzahl der Dezimalziffern einer Dezimalzahl gleichen Wertes. Zur Binärdarstellung des Wertes einer Dezimalzahl sind also im Mittel 3,3 bits je Dezimalziffer erforderlich (bit = binary digit).

Das binäre Zahlensystem eignet sich sehr gut für das Rechnen in elektronischen Rechenanlagen. Da jedoch außerhalb der Anlage dezimale Zahlen verwendet werden, muß bei der Eingabe eine Umschlüsselung dezimal in binär, bei der Ausgabe binär in dezimal erfolgen. Diese Umschlüsselung erfordert Zeit und Speicherraum für die Befehle, so daß für Aufgaben, bei denen viele Zahlen nur wenigen Berechnungen unterworfen sind, das binäre Zahlensystem ungeeignet ist. Für diesen Zweck empfiehlt sich eine Abwandlung des binären Zahlensystems.

1.2 Darstellung von Zahlen im binär-dezimalen Zahlensystem

Es handelt sich hierbei gewissermaßen um eine Kombination des binären und dezimalen Zahlensystems. Es wird nämlich nicht mehr die gesamte Dezimalzahl in eine Binärzahl umgewandelt, sondern jede Dezimalziffer wird einzeln im Binärsystem, d.h. also mittels der Ziffern 0 und L, dargestellt.
Die Symbolvielfalt des dezimalen Zahlensystems ist 10, entsprechend den Ziffern 0 ... 9, die man z.B. folgendermaßen darstellen könnte:

Dezimalziffer	Binärzahl				
	2^3	2^2	2^1	2^0	Binäre Wertigkeit
0	0	0	0	0	
1	0	0	0	L	
2	0	0	L	0	
3	0	0	L	L	
4	0	L	0	0	
5	0	L	0	L	
6	0	L	L	0	
7	0	L	L	L	
8	L	0	0	0	
9	L	0	0	L	

Bei der aufgeführten Verschlüsselung der Dezimalziffern wurden diese genau in ihrer binären Wertigkeit dargestellt. Es lassen sich jedoch auch andere Zuordnungen aufstellen.
Wie aus der Tabelle hervorgeht, sind zur Darstellung einer Dezimalzahl mindestens 4 bits pro Dezimalziffer erforderlich, jedem bit ist eine sog. binäre Wertigkeit zugeordnet.
Die Zahl 99 würde also in dieser binär dezimalen Darstellung folgendermaßen aussehen:

L 0 0 L L 0 0 L
9 9

Die Umschlüsselung einer Dezimalzahl in dieses System bereitet keinerlei Schwierigkeiten und ist praktisch nur ein Zuordnungsproblem, das entweder durch Programme oder durch einen Umschlüsseler gelöst werden kann.
Da eine im binär-dezimalen Zahlensystem verschlüsselte Zahl pro Dezimalziffer 4 bits benötigt, eine im binären Zahlensystem verschlüsselte jedoch nur 3,3 bits im Mittel, hat also ein sog. „Dezimalrechner" mehr bits zu verarbeiten als ein sog. „Binärrechner", sofern beide in der Lage sein sollten, gleichgroße Zahlen zu verarbeiten.

Beispiel:

Zur Darstellung einer 10-stelligen Dezimalzahl benötigt
ein Dezimalrechner 40 bits, (mindestens)
ein Binärrechner 33 bits.

Die Verarbeitung jedes bits erfordert eine gewisse Zeit; demnach benötigt ein Binärrechner zur Verarbeitung einer gegebenen Zahl weniger Zeit als ein Dezimalrechner, sofern beide Rechner die Zahlen gleichartig verarbeiten, also z.B. im Serienbetrieb (s.a. 2.3).

Zusammenfassend kann gesagt werden:

Rechenanlagen, die eine rein binäre Darstellung der Zahlen benutzen, heißen Binärrechner. Sie eignen sich insbesondere für Aufgaben, bei denen viel gerechnet und wenig ein- und ausgegeben soll. Rechenanlagen, die eine binär-dezimale Darstellung der Zahlen benutzen, heißen Dezimalrechner. Sie sind besonders für Aufgaben geeignet, bei denen große Zahlenmengen durch die Anlage laufen und dabei nur wenigen Operationen unterworfen werden. Das ist überwiegend bei kommerziellen Anwendungen der Fall, deshalb ist der Dezimalrechner besonders gut für kommerzielle Zwecke geeignet.

1.3 Darstellung von Text und Befehlen

Zur Informationsdarstellung in einer Rechenanlage gibt es grundsätzlich zwei Möglichkeiten. Die erste besteht darin, daß man eine bestimmte Anzahl von bits zu einer Informationseinheit, das sog. „Wort", zusammenfaßt. Man definiert dann den Begriff der Wortlänge und bezeichnet damit beim Dezimalrechner die Anzahl der Dezimalziffern, beim Binärrechner die Anzahl der Binärziffern. Jedem Wort ist ein sog. Kennzeichen zugeordnet, das entscheidet, ob die im Wort enthaltenen bits von der Rechenanlage als Zahl, Text oder Befehl gedeutet werden sollen. Bei Zahlen unterscheidet das Kennzeichen außerdem noch zwischen positiven und negativen Zahlen.

Betrachten wir zunächst einen Dezimalrechner. Er benutzt zur Darstellung jeder Dezimalziffer 4 bits. Es lassen sich jedoch mit Hilfe von 4 bits 16 verschiedene Informationen darstellen ($2^4 = 16$). Hiervon werden zur Darstellung der 10 Ziffern 10 bit-Kombinationen ausgewählt und als sog. „gültige Ziffern" bezeichnet. Die restlichen 6 Kombinationen dürfen in der Anlage nicht auftreten und werden deshalb als „ungültige Ziffern" bezeichnet. Tritt eine der ungültigen Ziffern auf, so kann es sich nur um eine fehlerhafte Informationsverarbeitung handeln, die Anlage meldet mit Hilfe von entsprechenden Einrichtungen (Gültigkeitskontrollen) Alarm.

Der Dezimalrechner habe z.B. eine Wortlänge von (10 + 1) Dezimalstellen (10 Stellen für die Information, eine für das Kennzeichen). Unter der Voraussetzung, daß auch das Kennzeichen nur eine gültige Ziffer enthalten darf, können dadurch 10 verschiedene Wortarten dargestellt werden.
Aus mathematischen Gründen verwendet man in der Kennzeichenstelle eine 0 zur Darstellung einer positiven Zahl, eine 9 zur Darstellung einer negativen Zahl. Durch die restlichen 8 Ziffern kann man zwischen verschiedenen Textdarstellungen oder verschiedenen Befehlen unterscheiden.

Wie stellt man nun einen Text in der Anlage dar ? Jede Dezimalstelle des Wortes erlaubt lt. Voraussetzung nur die Darstellung von 10 verschiedenen Zeichen. Die Zeichenvielfalt des Textes ist jedoch größer; allein das Alphabet hat 29 Zeichen (mit Umlauten). Hinzu kommen noch die Satzzeichen und bestimmte Sonderzeichen. Man benutzt deshalb zur Darstellung eines Textzeichens 2 Dezimalziffern und erhält damit eine Zeichenvielfalt von $10^2 = 100$. Damit ließen sich sogar Groß- und Kleinbuchstaben getrennt darstellen.

Nur aufgrund des Textzeichens deutet die Rechenanlage die 10 Ziffern eines Textwortes als Text. Da für ein Textzeichen 2 Ziffern erforderlich sind, lassen sich in einem (10 + 1)-stelligen Wort 5 Textzeichen und das Kennzeichen unterbringen.

Die restlichen 7 Kombinationen des Kennzeichens kann man jetzt zur Kennzeichnung von Befehlen verschiedener Art verwenden. Die ZUSE Z 31 hat z.B. 6 verschiedene Befehlstypen.

Auch hier werden die 10 Dezimalziffern des Wortes nur aufgrund des Kennzeichens als Befehl gedeutet.

Mitunter ist es vorteilhaft, ein Textwort nicht durch sog. gültige Ziffern darzustellen. Das hat den Vorteil, einen beliebigen Externcode, d.h. einen Code, der außerhalb der Anlage zur Informationsdarstellung verwendet wird (z.B. auf dem Lochstreifen), unverändert in der Anlage verarbeiten zu können. Man spart sich damit die Umschlüsselung dieses Externcodes in den gültigen Zifferncode der Rechenanlage. Diese Möglichkeit benutzt die ZUSE Z 31. Ein besonderes Kennzeichen dient dann zur Unterscheidung eines solchen Textwortes. Zur Darstellung eines Text-

zeichens ist wie vorher der Platz von zwei Ziffern, also 8 bit, erforderlich. Allerdings ergeben die 8 bit nicht zwei gültige Ziffern, so daß beim Vorhandensein des entsprechenden Kennzeichens die sog. Gültigkeitskontrolle außer Betrieb gesetzt wird.

Die zweite Möglichkeit, Informationen in einem Dezimalrechner darzustellen, besteht in der Verwendung einer sog. variablen Wortlänge.

Es existiert hierbei kein Kennzeichen für das Wort, da jedes Zeichen in der Anlage aussagt, ob es sich um eine Ziffer, ein Befehlszeichen oder ein Textzeichen handelt.

Das setzt voraus, daß jedes Zeichen in der Anlage mit mehr als 4 bits dargestellt werden muß, Es werden im allgemeinen 6 bit verwendet, woraus sich $2^6 = 64$ Darstellungsmöglichkeiten ergeben. Da hier die erwähnte Gültigkeitskontrolle wegfällt, wird meist noch ein weiteres bit pro Zeichen zur sog. Quersummenkontrolle verwendet. Dieses bit ergänzt für jedes Zeichen die Anzahl der bits auf eine ungerade Zahl. Es wird nun laufend kontrolliert, ob die Anzahl der bits ungerade ist, im anderen Falle gibt die Anlage Alarm.

Wie schon erwähnt, gestattet die geschilderte Darstellung eine variable Wortlänge im Rechner. Die Länge einer Information (z.B. einer Zahl) wird durch sog. Markenbits bestimmt, die aber ein weiteres bit pro Zeichen erfordern. Somit sind also zur Darstellung eines Zeichens bei der geschilderten Verschlüsselung 8 bits erforderlich. Der Vorteil einer solchen Anlage besteht darin, daß z.B. bei der Zahlendarstellung jede Zahl nur soviel Dezimalspeicherzellen benötigt, wie die Dezimalzahl Stellen hat. Allerdings benötigt jede Dezimalziffer 8 bits im Gegensatz zur vorherigen Darstellung, wo pro Dezimalziffer nur 4 bits erforderlich waren. Der Vorteil der variablen Wortlänge wird dadurch aber weitgehend aufgehoben. Hinzu kommt noch, daß die Programmierung in fester Wortlänge sehr viel einfacher ist.

Betrachten wir nun einen Binärrechner. Hier wird immer mit fester Wortlänge gearbeitet. Es ist wieder ein Kennzeichen vorhanden, das entscheidet, wie die bits des Wortes gedeutet werden sollen, ob als Zahl, Text oder Befehl. Das Kennzeichen besteht im allgemeinen aus mehreren bits am Anfang des Wortes. Bei der ZUSE Z 23 sind es 2 bits, die folgende Bedeutung haben:

0 0	kennzeichnet eine positive Zahl,
L L	kennzeichnet eine negative Zahl,
L 0	kennzeichnet ein Befehlswort,
0 L	kennzeichnet ein Textwort.

Die Darstellung der Zahlen geschieht, wie unter 1.1 geschildert, im Binärsystem.

Ein Befehl besteht ebenfalls aus einer Binärzahl; vom Rechner werden lediglich die einzelnen Binärstellen als Operations- oder Adressenzeichen gedeutet (s. später).

Bei der Darstellung von Text wird das Wort in bit-Gruppen unterteilt; jede Gruppe enthält gemäß dem verwendeten Code eine bestimmte Anzahl von bits (bei der ZUSE Z 23 sind es 5 bit wegen des Fernschreibcodes).

Abschließend sei noch erwähnt, daß auch bei fester Wortlänge das Kennzeichen entfallen kann. Der Programmierer darf sich jedoch dann nicht irren, da die Maschine sonst z.B. ein Textwort als Befehl und umgekehrt deutet. Man kann also feststellen, daß das Kennzeichen des Wortes eine Sicherheit gegenüber Programmierungsfehlern darstellt; deshalb benutzen es alle ZUSE-Rechner.

2 Arbeitsweise einer programmgesteuerten elektronischen Rechenanlage

An Hand des Blockschaltbildes in Abb. 1 (am Ende des Kapitels I) werden zunächst die wichtigsten Elemente einer programmgesteuerten elektronischen Rechenanlage erklärt. In diesem Blockschaltbild sind zwei verschiedene Informationskreisläufe eingezeichnet.

Befehlskreislauf: In einer vom Programm bestimmten Reihenfolge werden die Befehle aus dem Befehlsspeicher nacheinander ins Leitwerk gebracht. Jeder Befehl steuert die entsprechende Operation und gelangt anschließend zurück in den Befehlsspeicher.

Datenkreislauf: Ein Befehl im Leitwerk bewirkt, daß die gewünschten Daten (Zahlen oder Text) aus dem Datenspeicher ins Operationswerk gebracht und dort entsprechend verarbeitet werden.
Das Ergebnis der Verarbeitung gelangt dann zurück in den Datenspeicher.

Befehls- und Datenspeicher können über die Ein- und Ausgabe gefüllt bzw. geleert werden.

Wie das Blockschaltbild zeigt, enthält eine programmgesteuerte elektronische Rechenanlage die folgenden 5 Hauptbestandteile:

das Speicherwerk,
das Leitwerk,
das Operationswerk,
die Eingabe,
die Ausgabe.

Die folgenden Abschnitte sollen nun den prinzipiellen Aufbau und die Wirkungsweise dieser Teile behandeln.

2.1 Das Speicherwerk

Um irgendwelche Informationen in der Anlage verarbeiten zu können, müssen diese auf Abruf verfügbar sein, d.h. gespeichert werden. Das geschieht, wie schon in Abschnitt 1 erläutert, in sog. binären Speicherzellen, also Elementen, die entweder den Zustand L oder den Zustand 0 speichern können. Man unterscheidet in einer Rechenanlage verschiedene Arten von Speichern, deren wichtigste im folgenden gebracht werden.

2.1.1 Das Register

Das Register dient dazu, Informationen während der Verarbeitung kurzzeitig zu speichern. Register haben die Eigenschaft, ein sog. „Durchschieben" der Informationen zu ermöglichen. Ein Beispiel soll das erläutern:

Es seien zwei 10-stellige Dezimalzahlen zu addieren. Beide Zahlen sollen sich in je einem Register befinden, das 10+ 1 Dezimalstellen aufnehmen kann. Die beiden Zahlen werden nun dem Operationswerk zugeführt und dort addiert. Das Ergebnis soll in ein 3. Register gebracht werden. In sehr vielen Anlagen, z.B. in Serien- bzw. Serie-Parallel-Rechnern, geschieht die Verarbeitung der Zahlen folgendermaßen:
Zuerst werden die beiden Ziffern mit dem niedrigsten Stellenwert aus dem 1. und 2. Register in das Operationswerk „geschoben" und addiert; dann die beiden nächst höheren Ziffern usw. Das Ergebnis wird in das 3. Register „geschoben". Abb. 2 (am Ende des Kapitels I) zeigt ein Blockschaltbild dieses Additionsvorganges. Die Verarbeitung der Zahlen erfolgte im Beispiel ziffernweise nacheinander (ziffernweiser Serienbetrieb). Weiteres über die Verarbeitung wird im Abschnitt 2.2.3 gebracht.
Das Beispiel sollte nur erläutern, wozu man Register verwenden kann. Man unterscheidet zwischen statischen und dynamischen Registern. Statische Register erlauben es, den gesamten Registerinhalt nach jedem der Zustände Z_0 ..... Z_{11} zu testen. Bei dynamischen Registern kann nur nach jedem der Zustände die Ziffer getestet werden, die sich gerade am Ausgang (rechts in Abb. 2) des Registers befindet.

Statische Register sind:	Flip-Flop-Register
	Relaisregister
Dynamische Register sind:	Verzögerungsleitungen
	(davon gibt es sehr viele Arten)

Beide Registerarten werden in Rechenanlagen verwendet, technische Einzelheiten würden hier zu weit führen.

In einer Rechenanlage gibt es im allgemeinen nur wenige Register, da diese einen relativ grossen technischen Aufwand erfordern. Für eine Speicherung größerer Datenmengen stehen andere Speicherarten zur Verfügung.

Jeder dieser Speicher besteht aus sog. Speicherzellen. Das sind Einrichtungen, von denen jede im allgemeinen eine Informationseinheit, also z.B. ein Wort aufnehmen kann. Jede Speicherzelle hat eine sog. Adresse. Ein 1000-Wortspeicher muß also 1000 Adressen haben, wenn die Rechenanlage in der Lage sein soll, jedes beliebige Wort dieses Speichers zu erreichen. Wir kommen hier zum Begriff der Zugriffszeit, der an einem Beispiel erläutert werden soll.
Es sollen aus dem Speicher nacheinander Zahlen in die oben angeführten Register gebracht werden. Die Zahlen sollen dort verarbeitet und zurück in den Speicher gebracht werden. Ein Befehl sei z.B.: Bringe den Inhalt der Speicherzelle 500 in das Register 1. Die Zeit, die vergeht, bis die erste Ziffer der Zahl (unterste Stelle) den Registereingang erreicht, nennt man Zugriffszeit. Es ist leicht einzusehen, daß die Zugriffszeit kurz sein muß, um eine hohe Datenverarbeitungsgeschwindigkeit zu erreichen. Jede Rechenanlage besitzt einen Speicher mit relativ kurzer Zugriffszeit, den sog. Arbeitsspeicher. Da er relativ aufwendig ist, kann er nicht beliebig groß gemacht werden. Sind größere Datenmengen zu speichern, verwendet man sog. Nachschubspeicher, die bei allerdings großer Zugriffszeit entsprechend preiswert sind.
Betrachten wir zunächst den Arbeitsspeicher. Ein Arbeitsspeicher mit sehr kleiner Zugriffszeit ist

2.1.2 Der Ferritkernspeicher

In diesem Speicher werden die Informationen L bzw. 0 in meist ringförmigen Ferritkernen gespeichert. Es sei definiert, daß ein positiver Fluß in diesem Ringkern ein L, ein negativer Fluß eine 0 darstellen soll. Ein positiver Fluß im Kern wird durch einen Strom I erzeugt, ein negativer Fluß durch den Strom -I (s. Abb. 3 am Ende des Kapitels I).
Die gezeigte Hysteresisschleife zeigt den Verlauf des magnetischen Flusses im Ferritkern in Abhängigkeit vom Magnetisierungsstrom I. Infolge der rechteckförmigen Hysteresisschleife des Ferritkernes bleibt z.B. ein durch den Strom I erzeugter Fluß ϕ auch bestehen, wenn der Strom I abgeschaltet wird. Damit hat also der Kern ein L gespeichert. Durch einen Strom von mindestens -I (oder größer) kann im Kern ein Fluß $-\phi$ erzeugt werden, der auch wieder nach Abschalten von -I bestehen bleibt. Damit hat also der Kern nunmehr eine 0 gespeichert.

Wie wird nun die gespeicherte Information wieder „gelesen", d.h. der Maschine zur Verarbeitung zugeführt ? Beim Lesen gibt man z.B. auf alle Speicherkerne eines Wortes (z.B. 44 bei einem Wort von 10+ 1 Dezimalen zu je 4 bit) einen Strom -I. Dieser Strom bewirkt, daß alle Kerne, die auf L gestanden haben ($+\phi$) auf 0 ($-\phi$) zurückkippen. Die schon in 0 befindlichen Kerne behalten ihre 0-Stellung. Durch jeden Kern ist nun ein sog. Lesedraht gefädelt, der dann eine Spannung abgibt, wenn z.B. ein Kern vom L- in den 0-Zustand zurückkippt.
Beim Lesen eines Wertes geben also alle in L befindlichen Kerne eine sog. Lesespannung ab, durch die dann ein Register über sog. Leseverstärker in den entsprechenden Zustand versetzt wird.
Beim Lesen des beschriebenen Ferritkernspeichers ist die Information im Speicher verlorengegangen, da alle Kerne auf 0 zurückgekippt sind. Die Information muß also erforderlichenfalls wieder zurückgeschrieben werden. Das Register, das beim Lesen die Informationen über die Leseverstärker aufgenommen hatte, hat also zwei Aufgaben.
Es gibt nach dem Lesen des Speichers die Informationen in das Operationswerk ab und veranlaßt gleichzeitig ein sog. Rückschreiben der Information in die Speicherzelle, wo sie vor dem Lesen gestanden hat.
Die geschilderten Vorgänge Lesen, -Speichern in ein Register-, Rückschreiben in den Speicher, nennt man einen Speicherzyklus. Er liegt beim Ferritkernspeicher in der Größenordnung von Mikrosekunden.

Jede Speicherzelle eines Ferritkernspeichers enthält so viele Kerne, wie das Wort (bei fester Wortlänge) bits hat. Die Kerne werden in speziellen Rahmen matrizförmig angeordnet. Zu diesem Zweck wird im Rahmen ein Netz aus horizontalen und vertikalen Drähten gespannt. An jedem Kreuzungspunkt zweier Drähte befindet sich ein Ferritkern (die Drähte kreuzen sich im „Loch" des Ringkernes). Diese Drähte und noch zwei oder drei weitere durch den Kern geführte Drähte dienen zur Magnetisierung des Kernes bzw. zur Abnahme der durch das Kippen des Kernes induzierten Spannung. Man spricht bei der Herstellung eines solchen Rahmens von einer „Fädelung" der Kerne.

Die Fädelungsarbeit eines Ferritkernspeichers ist beachtlich. Ein 1000-Wort-Speicher der ZUSE Z 31 enthält z.B. fast 50 000 Kerne von 2 mm ø. Durch das Loch eines jeden Kernes (1,3 mm ø) müssen

5 Drähte gefädelt werden, um die oben geschilderten Vorgänge zu bewirken. Hinzu kommt noch eine Steuerelektronik zur Erzeugung der entsprechenden Stromimpulse.
Aufgrund dieses erheblichen Aufwandes kostet ein Wort von 44 bit ca. 50 - 100 DM. Die erforderliche Größe eines Ferritkernspeichers hängt vom Einsatz der Rechenanlage ab, wird aber die Speicherkapazität von einigen Tausend Worten kaum überschreiten.
Zur Speicherung größerer Datenmengen verwendet man, wie oben schon erwähnt, Speicher mit größerer Zugriffszeit. Hier bieten sich besonders Magnettrommel- und Magnetbandspeicher an, die wegen ihres gleichen Prinzips zusammen behandelt werden sollen.

2.1.3 Der Magnettrommel- und Bandspeicher

Die Wirkungsweise geht aus Abb. 4 hervor (am Ende des Kapitels I). Im Kern eines sog. Magnetkopfes wird die Information, die in Form von elektrischen Impulsen durch die Spule des Kopfes geschickt wird, in entsprechende magnetische Impulse umgewandelt. Durch einen Spalt des Kopfes wird der magnetische Fluß des Kernes unterbrochen. Dadurch wird in der Nähe des Spaltes ein sehr starkes Magnetfeld erzeugt. Dieses Feld ist in der Lage, ein in unmittelbarer Nähe befindliches magnetisches Material zu magnetisieren. Wird nun dieses Material bewegt, so können nacheinander magnetisierte Zonen auf dem Material erzeugt werden.
Wir definieren, daß jeder Impuls am Eingang des Magnetkopfes ein „L" darstellen soll, jeder nicht vorhandene Impuls eine 0. Bewegt man das magnetische Material mit konstanter Geschwindigkeit, so entsprechen die vorhandenen und nicht vorhandenen magnetisierten Zonen in ihrer örtlichen Folge auf dem magnetischen Material genau der Information, die die Impulsserie am Eingang des Kopfes in einer zeitlichen Folge enthielt. Diesen Vorgang nennt man „Schreiben" der Information.

Beim Lesen wird nun das magnetische Material in gleicher Weise (und gleicher Geschwindigkeit) am Kopf vorbeigeführt. Damit induzieren die magnetischen Zonen über den Kern des Kopfes in der Spule Spannungen, die in ihrer zeitlichen Folge genau der Impulsserie entsprechen, die vorher zum Schreiben des magnetischen Materials benutzt wurde.

Damit ist es aber gelungen, eine Information zu speichern (beim Schreiben) und sie wieder abzuholen (beim Lesen). Magnettrommel und Magnetband unterscheiden sich im Prinzip nur in der Form und Bewegung des magnetischen Materials.

Bei einer Trommel ist die Magnetschicht auf der Mantelfläche eines rotierenden Zylinders untergebracht. Durch die Rotation ist es möglich, die Information auf dem Umfang der Trommel zu speichern.
Jeder Magnetkopf erzeugt eine sog. (endlose) Spur, auf der eine größere Anzahl von Worten gespeichert werden können. Durch eine größere Anzahl von Spuren (einige Hundert) erhält man damit eine relativ hohe Speicherkapazität. Bei normalen Trommeln sind es etwa 3000 - 20000 Worte zu je etwa 40 bits. Die Kapazität der Trommel wird durch die begrenzte Oberfläche des Zylinders und durch die erreichbare Schreibdichte (Abstand der „Zonen") bestimmt.

Sog. Großraumtrommeln sind in Entwicklung; man versucht durch wesentliche Erhöhung der Schreibdichte und der Trommeloberfläche die Kapazität um etwa 1 - 2 Größenordnungen zu erhöhen.

Bei Magnetbändern werden im allgemeinen mehrere Spuren nebeneinander geschrieben, gebräuchlich sind 8 - 16 Spuren. Infolge der wesentlich höheren Oberfläche des Bandes und der aus technischen Gründen bedingten höheren möglichen Schreibdichte erreicht man pro 1000-Meter-Band eine Speicherkapazität von mehr als 1 Million Worten zu je 40 bit.

Hinsichtlich der Zugriffszeit bestehen zwischen Trommel und Magnetband erhebliche Unterschiede. Die mittlere Zugriffszeit einer Trommel ergibt sich aus der halben Trommelumdrehungszeit und liegt in der Größenordnung von Millisekunden, während sie beim Magnetband von der Durchlaufzeit des halben Bandes bestimmt wird und in der Größenordnung von Minuten liegt, also 3 - 5 Größenordnungen höher. Fast im umgekehrten Verhältnis stehen die Kosten beim Magnetbandgerät, um ein Wort von etwa 40 bit zu speichern. Bei der Trommel betragen die Kosten etwa 5 - 10 DM/Wort, beim Magnetband etwa 0,1 - 0,2 DM/Wort (einschl. der erforderlichen Elektronik).

Die Trommel wird sehr häufig auch als Arbeitsspeicher verwendet; sie verbindet den Vorteil einer relativ geringen Zugriffszeit mit dem eines relativ günstigen Preises. Das Magnetband wird nur als Nach-

schubspeicher benutzt, d.h. man tranferiert Blöcke von mehreren Hundert Worten auf einmal in den Arbeitsspeicher und hat somit nur einmal einen Zeitverlust infolge der großen Zugriffszeit des Magnetbandes.

Weitere Speicherarten sind in Entwicklung. Sie sollen die Forderung nach einer großen Kapazität bei geringer Zugriffszeit erfüllen und dabei noch preiswert sein. Einige dieser Speicher arbeiten schon; es sei hier nur der RAMAC-Plattenspeicher der IBM erwähnt.

2.2 Das Leitwerk

Die gesamte Rechenanlage wird durch das Leitwerk gesteuert. Es gibt sehr viele Möglichkeiten, eine Rechenanlage zu steuern. Es wird deshalb hier nur ein typisches Beispiel einer Steuerung gebracht, die bei sehr vielen Rechenanlagen verwendet wird. Obwohl die folgenden Ausführungen auf ZUSE-Anlagen abgestimmt sind, stellen sie doch das allgemeine Prinzip einer programmgesteuerten elektronischen Rechenanlage dar.

2.2.1 Der Aufbau eines Befehles

Ein Befehl ist eine Anweisung an die Rechenanlage, bestimmte Daten (Zahlen, Befehle oder Text) nach bestimmten Regeln zu verarbeiten. Grundsätzlich besteht ein Befehl aus dem Operationsteil und dem Adressenteil. Im Operationsteil wird angegeben, wie die im Adressenteil angegebenen Daten verarbeitet werden sollen. Man kann zwei Gruppen von Befehlen unterscheiden.

Zu der ersten Gruppe gehören alle Befehle, die sich auf die eigentliche Datenverarbeitung beziehen. Sie sollen deshalb Verarbeitungsbefehle genannt werden.

Zu der zweiten Gruppe gehören alle Befehle, die eine direkte Programmsteuerung bewirken. Sie sollen deshalb Programmbefehle genannt werden.

Wie schon oben erwähnt, gibt der Adressenteil des Befehles an, in welchen Speicherzellen sich die zu verarbeitenden Daten befinden („Adresse" der Daten). Im einfachsten Fall enthält der Adressenteil nur eine Adresse. Man spricht dann von einer sog.

1-Adressmaschine

Da mit den Daten operiert werden soll, ist jedoch im allgemeinen die Angabe von nur einer Adresse nicht ausreichend. Es soll ja z.B. ein Wort von einer Stelle zur anderen gebracht werden oder ein Wort mit einem anderen verknüpft werden.
Es kann also allgemein gesagt werden, daß in einer 1-Adressmaschine die zweite Adresse durch den Operationsteil des Befehles mitgegeben wird. Betrachten wir z.B. den ZUSE Z 31-Befehl

A 1000 (Verarbeitungsbefehl)

A	ist der Operationsteil des Befehles
1000	ist der Adressenteil des Befehles

Der Befehl bewirkt folgende Vorgänge in der ZUSE Z 31:
Es wird der Inhalt des X-Registers (die Zahl im X-Register) mit dem Inhalt der Speicherzelle Nr. 1000 (der Zahl in der Speicherzelle Nr. 1000) addiert. Das Ergebnis wird zurück in das X-Register gebracht. In der genormten Symbolschreibweise, die im folgenden immer verwendet wird, sieht das folgendermaßen aus:

$$(X) + (1000) \rightarrow X$$

In Worten: Bringe den Inhalt von X plus Inhalt von 1000 nach X

Die zweite Adresse, nämlich die Angabe, wo sich der zweite Operand befindet und wohin das Ergebnis soll, wird automatisch durch den Operationsteil A des Befehls gegeben.

Die 2-Adressmaschine

enthält im Adressenteil des Befehles die Angabe von 2 Speicherzellen. Ein Additionsbefehl der 2-Adressmaschine ZUSE Z 23 soll als Beispiel dienen:

A 1000 + 100

Durch den Befehl wird bewirkt, daß der Inhalt der Speicherzelle 1000 zum Inhalt der Speicherzelle 100 addiert wird; das Ergebnis wird in die Speicherzelle 100 transportiert.
In symbolischer Schreibweise:

$$\langle 1000 \rangle + \langle 100 \rangle \rightarrow 100$$

Die 3-Adressmaschine

enthält im Adressenteil des Befehles die Angabe von 3 Speicherzellen. Ein entsprechender Additionsbefehl könnte z.B. lauten

A + 1000 / 100 / 10

und in symbolischer Schreibweise bedeuten:

$$\langle 1000 \rangle + \langle 100 \rangle \rightarrow 10$$

Es ist wenig sinnvoll, Maschinen zu bauen, die mehr als 3 Adressen im Befehl enthalten. Anhand eines Beispiels sollen die 3 Maschinentypen verglichen werden. Es seinen zwei Zahlen, die sich in zwei beliebigen Speicherzellen befinden, zu addieren. Das Ergebnis soll in eine beliebige Speicherzelle gebracht werden.
Die 1-Adressmaschine braucht hierzu 3 Befehle:

1. Der erste Operand muß aus der beliebigen Speicherzelle in das Register gebracht werden, auf das sich der Additionsbefehl bezieht. (z.B. in das X-Register der ZUSE Z 31)
2. Der zweite Operand wird zu dem im X-Register befindlichen ersten Operanden addiert, das Ergebnis gelangt nach dem X-Register.
3. Das Ergebnis wird vom X-Register in die gewünschte Speicherzelle transportiert.

Die 2-Adressmaschine braucht nur 2 Befehle, da bei der Addition sofort die Speicherzellen beider Operanden angegeben werden können, der erste Befehl oben kann also wegfallen.

Die 3-Adressmaschine benötigt überhaupt nur einen Befehl.

Allgemein gilt also:

> Je mehr Adressen in einem Befehl gegeben werden können, umso weniger Befehle braucht eine Rechenanlage, um eine bestimmte Aufgabe zu lösen.

Dieser Vorteil der Mehradressmaschine gegenüber Einadressmaschinen wird jedoch mit einem beachtlichen technischen Aufwand erkauft. Außerdem wird bei sehr vielen Berechnungen die Möglichkeit, mehrere Adressen geben zu können, überhaupt nicht ausgenutzt. Soll z.B. eine Summe aus 6 in beliebigen Speicherzellen befindlichen Operanden gebildet und das Ergebnis in eine beliebige Speicherzelle gebracht werden, so benötigen die 3 geschilderten Maschinen, wie sich leicht überlegen läßt, folgende Anzahl von Befehlen:

Die 1-Adressmaschine	=	8 Befehle,
die 2-Adressmaschine	=	7 Befehle,
die 3-Adressmaschine	=	6 Befehle.

Es ist deshalb verständlich, daß aus wirtschaftlichen Gründen die Entwicklung eindeutig zu den 1- bzw. höchstens 2-Adressmaschinen tendiert.

2.2.2 Programmablauf und Steuerung

Wie bereits erwähnt, müssen die Befehle eines Programmes mit möglichst kurzer Zugriffszeit aus dem Speicher der Rechenanlage abgerufen werden können. Jeder Befehl ist im Speicher unter einer bestimmten Adresse erreichbar. Um die Befehle ausführen zu können, müssen sie nacheinander in das sog. Befehlsregister gebracht werden. Werden sie der Reihe nach aus dem Speicher abgerufen, so spricht man von einem linearen Programmablauf. Diesen linearen Programmablauf steuert das Befehlszählregister.

Im Befehlsregister werden die Befehle entschlüsselt, d.h. die entsprechenden Informationswege in der Rechenanlage geschaltet. Anschließend werden die Informationen in der vom Befehl angegebenen Weise verarbeitet; der Befehl wird ausgeführt.

Zu Beginn des Programmes wird die Anfangsadresse, also die Nr. der Speicherzelle, in der sich der erste Befehl des Programmes befindet, in das Befehlszählregister gebracht, das seinerseits die Übertragung des Befehles aus der angegebenen Befehlsspeicherzelle in das Befehlsregister bewirkt. Anschließend wird der Befehl ausgeführt. Während dieser Zeit wird die im Befehlszählregister befindliche Adresse um 1 erhöht, um das Aufrufen des nächsten Befehles im Speicher vorzubereiten. Das Befehlszählregister ruft nun den nächsten Befehl in das Befehlsregister, der Befehl wird ausgeführt, die Adresse des Befehlszählregisters erneut um 1 erhöht usw.

Beim linearen Programmablauf werden also die Befehle in linearer Folge, beginnend mit einer gegebenen Anfangsadresse, aus dem Befehlsspeicher nacheinander abgerufen.

Die lineare Befehlsfolge wird nur dann aufrechterhalten, wenn es sich bei den Befehlen um Verarbeitungsbefehle handelt. Wird jedoch ein Programmbefehl in das Befehlsregister gebracht, so wird der lineare Programmablauf unterbrochen. Der Programmbefehl bewirkt, daß nicht der nächste Befehl der Folge, sondern ein anderer Befehl als nächster ausgeführt werden soll. Es wird also aus dem linearen Programm „herausgesprungen", deshalb nennt man einen solchen Befehl auch Sprungbefehl. Seine Adresse gibt an, wo sich der nächste Befehl befindet. Dieser Befehl wird nun in das Befehlsregister gebracht und ausgeführt. Gleichzeitig wird die Adresse des Sprungbefehles um 1 erhöht und in das Befehlszählregister gebracht. Damit kann sich jetzt wieder ein vom Befehlszählregister aus gesteuerter linearer Programmablauf anschließen.

Ein Sprunbefehl gibt also die Anfangsadresse eines neuen linearen Programmablaufes an.

Sprungbefehle werden sehr oft „bedingt" gegeben, d.h. sie werden nur ausgeführt, wenn eine bestimmte Bedingung erfüllt ist, sonst werden sie „überlaufen", unterbrechen also nicht den linearen Programmablauf. Die sog. Bedingungen können sich auf den Zustand von bestimmten Registern, Speichern oder speziellen Schaltern am Bedienungspult beziehen. Durch die bedingten Befehle ist es der Rechenanlage möglich, logische Entscheidungen zu treffen, d.h. den weiteren Programmablauf vom Ergebnis einer vorhergehenden Berechnung abhängig zu machen. In ZUSE-Anlagen stehen eine sehr große Anzahl bedingter Befehle zur Verfügung, da alle Befehle (nicht nur die Sprungbefehle) bedingt gegeben werden können. Damit können außerordentlich befehlssparende Programme aufgebaut werden, die außerdem noch infolge des mit dem Befehl gekoppelten Tests (durch die Bedingung) sehr schnell sind.

2.3 Das Operationswerk

Im Operationswerk werden die vom Befehl angegebenen arithmetischen oder logischen Operationen durchgeführt. Es besteht aus sog. logischen Schaltkreisen, die so aufgebaut sind, daß sie die zu verarbeitenden Operanden nach den gewünschten Rechenregeln miteinander verknüpfen. Sollen z.B. zwei Zahlen in binärer Darstellung addiert werden, so muß das Operationswerk die Additionsregeln des binären Zahlensystems erfüllen; sollen zwei Zahlen in binär-dezimaler Darstellung addiert werden, so werden die bits der Ziffern nach binären Regeln addiert, die Ziffern aber nach dezimalen Regeln . Weiterhin wird der Aufbau eines Operationswerkes durch das System der Informationsverarbeitung in der Rechenanlage bestimmt. Man unterscheidet zwischen Serien-, Serieparallel- und Parallelverarbeitung von Informationen. Das verwendete System bestimmt bei gleicher Bitzeit die interne Datenverarbeitungsgeschwindigkeit der Anlage. Eine Bitzeit ist hierbei die Zeit, die zur Verarbeitung eines bits benötigt wird. Zur Verarbeitung eines Wortes ist die sog. Wortzeit erforderlich. Sie setzt sich aus der Operationszeit und der Schaltzeit zusammen. Die Schaltzeit ist z.B. zur Befehlsentschlüsselung und zum Öffnen der gewünschten Informationswege erforderlich und beträgt bei allen Systemen einige Bitzeiten. Die Operationszeit hingegen hängt von dem gewählten System ab und soll im folgenden näher behandelt werden.

2.3.1 Verarbeitung von Informationen im Serienbetrieb

Hierbei werden die bits eines Wortes nacheinander verarbeitet. Die Operationszeit einer sog. Seriemaschine ergibt sich somit aus folgender Gleichung:

$$t_o = n_w \cdot t_b$$

t_o = Operationszeit
n_w = Anzahl der bits pro Wort
t_b = Bitzeit

2.3.2 Verarbeitung von Informationen im Serienparallelbetrieb

Hierbei wird eine Gruppe von bits gleichzeitig verarbeitet, die einzelnen Bitgruppen des Wortes jedoch nacheinander. Die Operationszeit einer solchen Serieparallelmaschine ergibt sich aus folgender Gleichung:

$$t_o = \frac{n_w}{n_g} \cdot t_b$$

n_g = Anzahl der bits pro Gruppe.

Dieses System der Informationsverarbeitung findet insbes. bei Dezimalrechnern Anwendung. Es werden dabei die bits gleichzeitig verarbeitet, die eine Dezimalstelle darstellen.

2.3.3 Verarbeitung von Informationen im Parallelbetrieb

Hierbei werden alle bits eines Wortes gleichzeitig verarbeitet. Somit ist die Operationszeit gleich der Bitzeit. Ein Rechner in Parallelorganisation ist also weitaus schneller als die zwei geschilderten. Warum baut man überhaupt noch Serie- bzw. Serieparallelmaschinen ? Das ist lediglich eine Frage des technischen Aufwandes. Allgemein gilt:

Es sind soviele unabhängige Übertragungswege und Operationswerke erforderlich, wie bits gleichzeitig verarbeitet werden sollen.

Eine ausschließlich parallele Verarbeitung der Informationen findet praktisch nur in extrem schnellen und damit teuren Rechenanlagen Verwendung.

2.3.4 Ausbaustufen des Operationswerkes

Im einfachsten Fall besteht das Operationswerk aus einer reinen Zuordnungsschaltung. Sollen z.B. zwei Ziffern addiert werden, so bewirken die beiden Ziffern die Anwahl einer Speicheradresse, unter der das Ergebnis zu finden ist. Jedes Ergebnis, das die Addition zweier Ziffern haben kann, ist also in einer sog. Additionstabelle im Speicher aufbewahrt. Gleiches gilt für die Subtraktion und Multiplikation. Ein solches Operationswerk verwendet die IBM 1620. Die meisten Rechenanlagen verwenden jedoch ein System von Schaltkreisen, das zumindest die entsprechenden Additions- und Subtraktionsregeln erfüllt. Die Multiplikation und die Division werden meist in Form von sog. Unterprogrammen durchgeführt, die aus einer Vielzahl von Additions- bzw. Subtraktionsoperationen sowie Stellenverschiebungen bestehen. Das kostet zwar Zeit, hält aber den Aufwand für das Operationswerk relativ gering. Als Beispiel sei das Multiplikationsschema eines Dezimalrechners skizziert. Zur Vereinfachung soll angenommen werden, daß es sich um eine Rechenanlage handelt, die eine Wortlänge von nur 3 Dezimalen hat. Es seien die Zahlen 400 und 321 miteinander zu multiplizieren. 400 sei der Multiplikand, 321 der Multiplikator. Macht man diese Berechnung auf dem Papier, so kann man folgendermaßen verfahren:

400 · 321		
400	1.Schritt = 400 · 1	(1.Partialprodukt)
8000	2.Schritt = 400 · 20	(2.Partialprodukt)
120000	3.Schritt = 400 · 300	(3.Partialprodukt)
128400	4.Schritt = Addition der 3 Partialprodukte	

Es werden also die Ziffern des Multiplikators nacheinander, beginnend mit der niedrigsten Stelle des Multiplikators, verarbeitet. Die Rechenanlage löst nun per Programm die Multiplikation in ganz

ähnlicher Weise. Unter der Voraussetzung, daß die Rechenanlage nur addieren und verschieben kann, bildet sie die Partialprodukte durch wiederholte Addition des Multiplikanden; sie löst also den 3. Schritt in 3 Schritte auf, nämlich in eine dreimalige Addition von 400. Außerdem addiert sie sofort stellenrichtig das neue Partialprodukt auf die Summe der vorher gebildeten Partialprodukte. Im folgenden wird gezeigt, wie eine Rechenanlage mit Hilfe von 3 Registern die Multiplikation durchführen könnte. Der Multiplikand befindet sich im Md-Register, der Multiplikator im Mr-Register, das Produkt wird im P-Register gebildet. Bei jedem Additionsschritt wird der Multiplikator um 1 vermindert (abgebaut). Jedesmal, wenn die letzte Multiplikatorziffer auf Null abgebaut worden ist, werden P und Mr gemeinsam verkoppelt nach rechts verschoben. Dadurch wird das bisherige Ergebnis für die nächste Addition des Partialproduktes in die richtige Lage gebracht und die nächst höhere Multiplikatorziffer an die Teststelle des Mr-Registers gebracht.

⟨Md⟩	⟨P⟩	⟨Mr⟩	durchgeführte Operationen
400	000	321	Ausgangszustand
400	400	320	⟨P⟩ + ⟨Md⟩ → P (1. Partialprodukt)
400	040	032	Da die letzte Ziffer von Mr auf Null abgebaut ist, wird ⟨P⟩ und ⟨Mr⟩ gemeinsam nach rechts verschoben.
400	440	031	⟨P⟩ + ⟨Md⟩ → P (2. Partialprodukt)
400	840	030	⟨P⟩ + ⟨Md⟩ → P
400	084	003	gemeinsame Rechtsverschiebung von P und Mr
400	484	002	⟨P⟩ + ⟨Md⟩ → P
400	884	001	⟨P⟩ + ⟨Md⟩ → P (3. Partialprodukt)
400	1284	000	⟨P⟩ + ⟨Md⟩ → P
400	128	400	gemeinsame Rechtsverschiebung von ⟨P⟩ und ⟨Mr⟩. Da der Multiplikator völlig abgebaut ist (auf Null), ist die Multiplikation beendet. Das 6-stellige Ergebnis steht in P und Mr.

Zur Bildung des Produktes benötigt die Rechenanlage im vorliegenden Beispiel 9 Schritte, und zwar insgesamt 6 Additionen und 3 Verschiebungen. Unter der Voraussetzung, daß jeder Schritt eine Wortzeit benötigt, braucht die Rechenanlage für diese Multiplikation 9 Wortzeiten.

Bei höheren Multiplikatorziffern steigt die Anzahl der Additionen, so daß sich damit auch eine längere Multiplikationszeit ergibt. Man spricht deshalb häufig von einer mittleren Multiplikationszeit, die auf der Voraussetzung beruht, daß z.B. bei einem 10-stelligen Multiplikator alle Ziffern von 0 bis 9 einmal vorkommen.

Wie schon erwähnt, erfordert die Multiplikation per Programm nach diesem Prinzip viel Zeit. Man kann durch Anwendung des sog. „Fünfervorteils" die Zeit vom Programm her verkürzen, benötigt aber doch zur Bildung jedes Partialproduktes mehrere Additionen bzw. Subtraktionen. Es besteht jedoch die Möglichkeit, das Partialprodukt in einem Schritt zu bilden, wenn man eine sog. Vervielfacherschaltung einbaut, die sofort den n-fachen Multiplikanden auf die bisherige Summe der Partialprodukte addiert, wobei n jeweils die letzte Ziffer des Multiplikators ist.

Solche „festverdrahtete", also nicht durch Programm bewirkte Mehrfachadditionen bedingen einen höheren technischen Aufwand und werden meist als Zusätze eingebaut. Auch für die Division existieren solche Zusätze. Außerdem lassen sich auch noch die Befehlsfolgen, die zur Durchführung der Multiplikation erforderlich sind, fest verdrahten. Solche voll ausgebauten Operationswerke erhöhen dann wiederum den Aufwand und damit auch den Preis für die Rechenanlage.

Weiterhin ist es möglich, sog. festverdrahtete Einrichtungen für bestimmte Operationen im Operationswerk einzubauen. Als Beispiel seien „Gleitkommazusätze" erwähnt. Bei Gleitkommaoperationen werden ohne solche Zusätze die Mantissen und Exponenten der Gleitkommaoperanden getrennt verarbeitet. Damit erhöht sich im allgemeinen die Operationszeit gegenüber den entsprechenden Zeiten für die Festkommaoperationen. Der Einbau entsprechender Zusätze bringt

erhebliche Zeitersparnis und wird sich immer dann lohnen, wenn relativ viele Gleitkommaoperationen durchzuführen sind und sich damit eine erhebliche Verkürzung der gesamten Bearbeitungszeit ergibt.

Zusammenfassend kann gesagt werden:

Der Aufbau des Operationswerkes bestimmt die Rechengeschwindigkeit der Anlage. Operationswerke mit kleinem Aufwand erfordern gegenüber solchen mit großen Aufwand erhöhte Rechenzeiten, da bestimmte Operationen durch Unterprogramme gelöst werden müssen. Die meisten Rechenanlagen verwenden aus wirtschaftlichen Gründen ein Serie- oder Serieparallel-Operationswerk, verarbeiten also die bits oder Bitgruppen eines Wortes nacheinander.

2.4 Eingabe von Informationen

Man kann prinzipiell zwei Arten der Informationseingabe unterscheiden, und zwar die manuelle Eingabe und die programmgesteuerte Eingabe. Elektronischen Rechenanlagen gibt man manuell im allgemeinen nur relativ wenige Daten (Zahlen, Befehle, Text) ein, da hierbei die hohe mögliche Datenverarbeitungsgeschwindigkeit einer solchen Anlage nicht annähernd ausgenutzt würde, könnte doch die Anlage in der Zeit, die erforderlich ist, um eine Zahl von Hand einzutasten, bereits mehrere Tausend Rechenoperationen ausführen. Aus diesem Grunde wird die manuelle Eingabe im allgemeinen nur zur direkten Steuerung der Rechenanlage benutzt. Die Einrichtungen für die manuelle Eingabe sind meistens im sog. Bedienungspult der Rechenanlage zusammengefaßt, das im Kapitel II am Beispiel der ZUSE Z 31 näher behandelt wird.

Große Informationsmengen werden über sog. Lesegeräte in die Rechenanlage eingegeben. Diese Art der Eingabe wird vom Programm gesteuert, was im folgenden etwas näher behandelt werden soll. Als Beispiel soll ein Lochstreifenleser dienen. Auf dem Lochstreifen befinden sich die Informationen in Form von Lochkombinationen. Jede Lochkombination stellt ein Zeichen dar (Buchstabe, Ziffer oder Satzzeichen), wobei „ein Loch" einen Ja-Wert, „kein Loch" einen Nein-Wert bedeutet. Es ist also möglich, z.B. Ziffern auf dem Lochstreifen in der erwähnten binär-dezimalen Darstellung zu verschlüsseln. Die einzelnen Zeichen werden nacheinander auf dem Lochstreifen gelocht; allgemein beträgt der Abstand von Zeichen zu Zeichen 2,5 mm, so daß 400 Zeichen pro Meter Lochstreifen gelocht werden können. Die Lochstreifen werden entweder manuell erstellt oder entstehen automatisch (z.B. bei stanzenden Meßgeräten). Die Lochstreifen können nun sehr viel schneller als bei der manuellen Eingabe mit Hilfe der erwähnten Lesegeräte in die Anlage „eingelesen" werden. Wie arbeitet nun ein Leser ? Auf mechanischem oder fotoelektrischem Wege wird die Lochkombination eines Zeichens abgetastet. Die abgetastete Kombination von Ja-Nein-Werten könnte nun direkt von der Rechenanlage per Programm „abgeholt" werden. Das wäre jedoch sehr unwirtschaftlich, da die Rechenanlage immer wieder auf den für sie relativ langsamen Lochstreifenleser warten müßte, weil der Lochstreifenleser während der Zeit des Einlesens die richtige Information nur während einer relativ kurzen Zeit zur Verfügung stellt. Schnellste Lochstreifenleser bringen es auf eine Einlesegeschwindigkeit von etwa 1000 Zeichen/sec, während selbst „langsame" Elektronenrechner immerhin mehr als 10.000 Zeichen/sec verarbeiten können. Aus diesem Grunde speichert man zunächst die vom Lochstreifen abgelesene Information in einem sog. Pufferspeicher, der je nach Ausbau ein oder mehrere Zeichen aufnehmen kann. Der Leser arbeitet mit dem Puffer unabhängig zusammen. Durch einen Startimpuls wird Zeichen nach Zeichen vom Lochstreifen abgelesen und zwar solange, bis der Puffer „voll" ist. Eine Elektronik sorgt für die richtige Zeichenverteilung im Pufferspeicher, steuert den Lochstreifentransport usw. Ist der Pufferspeicher voll, so erzeugt die Elektronik ein sog. Freigabezeichen, das anzeigt, daß jetzt der Inhalt des Puffers von der Rechenanlage abgeholt werden kann. Die Rechenanlage selbst konnte während der Zeit, in der der Puffer vom Leser gefüllt wurde, z.B. früher gelesene Informationen verarbeiten. Werden neue Informationen benötigt, fragt die Rechenanlage per Programm das gewünschte Eingabegerät ab. Existiert dort ein Freigabezeichen, so wird der Inhalt des Pufferspeichers mit der hohen internen Übertragungsgeschwindigkeit der Rechenanlage in den Speicher der Rechenanlage übertragen und gleichzeitig ein neuer Startimpuls für die Elektronik des Lesegerätes gegeben, um ein erneutes Füllen des Puffers einzuleiten. Während des Einlesens der Informationen in den Pufferspeicher existiert für dieses Lesegerät kein Freigabezeichen, eine Abfrage

der Rechenanlage führt also nicht zum Erfolg. Die Rechenanlage muß jetzt entweder warten bis das Freigabezeichen kommt oder sie rechnet intern im Programm weiter, um später wiederum anzufragen, ob ein Freigabezeichen da ist. Der zweite Weg ist sinnvoll, wenn eine hohe Datenverarbeitungsgeschwindigkeit erreicht werden soll. Die Rechenanlage kann natürlich nur dann weiter rechnen, solange sie noch genügend zu verarbeitende Informationen in ihrem Speicher hat.

Es wird sich niemals ganz vermeiden lassen, daß die Rechenanlage ab und zu auf die Ein- (oder Ausgabe-) Geräte „warten" muß, es sei denn, die Programme erfordern komplizierte Berechnungen mit vielen Operationen. Anhand eines sog. Fluß- oder Strukturdiagrammes sollen die Vorgänge beim Einlesen nochmals im Prinzip gebracht werden. Das Flußdiagramm ist eine wertvolle Hilfe bei der Programmierung und wird deshalb im folgenden noch mehrmals verwendet werden. Operationen werden hierbei durch Rechtecke, sog. Ja-Nein-Entscheidungen durch ovale Umrandungen begrenzt. Durch Pfeile wird die Folge des Ablaufes gekennzeichnet. Ein Flußdiagramm ist nur dann richtig bzw. vollständig, wenn sich keine „Schleifen" ergeben, aus denen es keinen Ausweg gibt. Das Flußdiagramm in Abb. 5 am Ende des Kapitels I bedarf keiner weiteren Erklärung, da es nur eine Zusammenfassung der bisher gebrachten Grundlagen über die programmgesteuerte Eingabe enthält.

Zusammenfassend kann gesagt werden:

Bei der programmgesteuerten Informationseingabe werden die auf externen Informationsträgern (z.B. Lochkarte, Lochstreifen) befindlichen Zeichen durch ein sog. Lesegerät abgetastet. Die damit erzeugten bits eines Zeichens werden in sog. Pufferspeichern gespeichert. Pufferspeicher und Lesegerät werden von einer Elektronik gesteuert, die von der Rechenanlage gestartet, den Pufferspeicher je nach Ausbau mit einem oder mehreren Zeichen füllt. Ist der Pufferspeicher gefüllt, kann sein Inhalt von der Rechenanlage per Programm abgeholt werden.

Die erforderliche Kapazität des Pufferspeichers hängt einmal vom Programm der Rechenanlage und zum anderen von der Geschwindigkeit des Lesegerätes ab.

2.4.1 Eingabe über einen Pufferspeicher für 1 Zeichen

Eine Pufferung beim Einlesen ist dann erforderlich, wenn während der Zeit, in der das Lesegerät arbeitet, in der Rechenanlage gerechnet werden soll. Die zur Verfügung stehende Rechenzeit darf aber, bezogen auf die Operationsgeschwindigkeit der Anlage, nicht zu kurz sein. Ein Beispiel soll das verdeutlichen. Ein Lesegerät sei in der Lage, 1000 Zeichen pro sec abzutasten, benötigt also pro Zeichen 1 ms. Mittelschnelle Rechenanlagen können in dieser Zeit aber nur wenige Elementaroperationen durchführen. Elementaroperationen sind z.B. Additionen, Subtraktionen, Vergleiche, Sprungbefehle, Worttransporte von einem zum anderen Speicher usw. Die Abfrage eines Lesegerätes durch die Rechenanlage benötigt jedoch ebenfalls einige Elementaroperationen, so daß zum Rechnen praktisch keine Zeit mehr bleibt. Man muß also versuchen, die Anzahl der Abfragen zu verringern, um mehr zusammenhängende Rechenzeit zu erhalten. Das bedeutet aber, daß der Pufferspeicher in der Lage sein muß, mehrere Zeichen aufzunehmen. Somit kann also gesagt werden:

Die Eingabe über einen sog. Einzeichenpuffer ist nur dann sinnvoll, wenn entweder ein relativ langsames Lesegerät verwendet wird oder während des Einlesens nicht gerechnet werden soll.

2.4.2 Eingabe über einen Pufferspeicher für mehrere Zeichen

Ein Mehrzeichenpuffer vermeidet die oben erwähnten Nachteile; ist aber relativ kostspielig. Ebenso kostspielig ist die erforderliche Steuerelektronik. Diese stellt nämlich praktisch eine kleine programmgesteuerte Einheit dar, deren Programm „festverdrahtet" ist. Dieses Programm bewirkt die Steuerung des Lesegerätes sowie die richtige Verteilung der Zeichen auf den Pufferspeicher. Die Verteilung muß so geschehen, daß die Rechenanlage die Zeichen im Puffer möglichst schnell und in einer für sie günstigen Ordnung abholt. Betrachten wir ein Lochstreifenlesegerät mit einem Puffer, der 10-stellige Zahlen aufnehmen kann.

Auf dem Lochstreifen wird die Zahl beginnend mit der höchsten Ziffer abgetastet und auch so im Pufferspeicher abgesetzt. Eine Serie- oder Serieparallel-Rechenanlage verarbeitet aber die Zahl beginnend mit der niedrigsten Ziffer. Soll nur eine Elementaroperation nötig sein, um die Zahl aus dem Pufferspeicher bei der Abfrage abzuholen, dann muß die Rechenanlage die Zahl im Puffer so angeboten bekommen, daß die niedrigste Ziffer zuerst erscheint. Diese Umschaltung im Pufferspeicher muß z.B. die erwähnte Elektronik besorgen.

Für bestimmte Eingabegeräte (z.B. Lochkartengeräte) ist infolge des Einleseprinzips sowieso ein Mehrzeichenpuffer notwendig, wenn die Rechenanlage während des Einlesens einer Lochkarte rechnen soll. Betrachten wir zunächst einmal als Beispiel die Informationsdarstellung auf einer 80-spaltigen Lochkarte. In jeder der 80 Spalten kann ein Zeichen dargestellt werden. Jede Spalte enthält 12 mögliche Plätze, an denen ein Loch oder kein Loch sein kann. Diese 12 möglichen Plätze nennt man Zeilen. Grundsätzlich läßt sich in jeder Spalte ein Zeichen von der Vielfalt $2^{12} = 4096$ unterbringen, d.h. es sind 4096 Lochkombinationen pro Spalte möglich. In jeder Spalte könnte z.B. eine beliebige Zahl von 0 bis 4095 gelocht sein. Von dieser Möglichkeit wird in Sonderfällen Gebrauch gemacht, man erhält dann die sog. dual verschlüsselte Lochkarte. Im allgemeinen wird jedoch eine andere Verschlüsselung gewählt, da zur Darstellung des Alphabets, der Satzzeichen und Ziffern nur wesentlich weniger Kombinationen notwendig sind. Man kommt dann mit höchstens 3 Löchern pro Spalte aus. Damit ergibt sich eine Zeichenvielfalt von 299 entsprechend der Gleichung:

$$\text{Zeichenvielfalt} = \sum_{i=0}^{3} \binom{n}{i} \quad \text{mit } n = 12 \text{ (Anzahl der Zeilen).}$$

Diese Zeichenvielfalt wird bei der Lochkarte niemals ausgenutzt, sie wurde nur gewählt, um die Kombinationen verwenden zu können, die eine einfache Umschlüsselung in den Code der Rechenanlage (Maschinencode) gestatten.

Wie wird nun die Lochkarte vom Leser abgetastet ? Die erste Möglichkeit besteht darin, Spalte nach Spalte, also zeichenweise abzutasten. Die Karte wird also in 80 Schritten (entsprechend der 80 Spalten) abgetastet. Diese Methode würde das Einlesen über einen Einzeichenpuffer erlauben. Wegen der 80 notwendigen Schritte ergeben sich beim Lesegerät technische Schwierigkeiten, die dazu führen, daß die Einlesegeschwindigkeit begrenzt ist oder die Abtastgeschwindigkeit beim Abtasten einer Lochkarte nicht konstant ist. (Die Karte wird nach einem Kartenstart mit wachsender Geschwindigkeit durch die Abfühleinrichtung bewegt).

Schnelle Lesegeräte verwenden eine zeilenweise Abtastung. Es sind also pro Karte nur 12 Abtastungen erforderlich. Bei jeder Abtastung werden somit 80 Spalten gleichzeitig auf das Vorhandensein von Löchern getestet. Das bedeutet, daß vor dem Abtasten der letzten Zeile einer Karte noch keines der 80 Zeichen feststeht. Erst nach dem vollendeten Abtasten einer Karte ist also eine Umschlüsselung der Zeichen vom Lochkartencode in den Maschinencode möglich.

Es müssen also auf alle Fälle 80 Zeichen im 12-bit-Lochkartencode gespeichert werden. Dazu ist ein $12 \cdot 80 = 960$-bit-Speicher erforderlich, der in Verbindung mit der Steuerelektronik einen beachtlichen Aufwand darstellt. Man gewinnt allerdings damit eine große zusammenhängende freie Rechenzeit für die Anlage beim Einlesen einer Karte. Selbst bei einem sehr schnellen Lochkartenleser mit der Einlesegeschwindigkeit von max. 48 000 Karten/h ist die Einlesezeit für eine Karte immer noch 75 msec. Diese Zeit steht fast voll als Rechenzeit zur Verfügung, da das Abholen des Puffers einschließlich einiger vorbereitender Programmschritte nur wenige Millesekunden dauert.

Bei manchen Anlagen wird der Informationsinhalt einer Karte nicht in einem besonderen Pufferspeicher gespeichert, sondern sofort zeilenweise in einen dafür bestimmten Teil des Arbeitsspeichers der Anlage eingelesen. Es werden also praktisch die Löcher der Karte als magnetisierte Kerne im Speicher abgebildet. Bei der Umschlüsselung in den Maschinencode werden dann die „Spalten" des internen Speichers nacheinander verarbeitet. Diese Methode hat den Nachteil, daß nahezu während der gesamten Einlesezeit der Karte die Rechenanlage blockiert ist. Nur während der relativ kurzen Transportzeit zwischen zwei Lochkarten kann die Rechenanlage operieren. Diese Zeit wird u.U. gerade ausreichen, um den Lochkartencode in den Maschinencode umzusetzen und die Zeichen in die entsprechenden Speicherzellen zu transportieren.

Zusammenfassend kann gesagt werden:

Die Eingabe über Mehrzeichenpuffer erhöht die effektive Datenverarbeitungsgeschwindigkeit der Rechenanlage, da die Rechenanlage während des größten Teils der Einlesezeit rechnen kann.

2.4.3 Eingabe von Analogwerten

Unter Analogwerten versteht man Informationen, die nicht in digitaler (ziffernmäßiger) Form vorliegen. Analogwerte sind z.B. Ströme von Meßgeräten, die irgendwelche physikalischen Zustände messen. Sehr oft sollen eine Vielzahl solcher Meßwerte in die Rechenanlage schnell eingelesen werden, um nach einer entsprechenden Verarbeitung neue Werte für die Steuerung des Systems zu erhalten, von dem die physikalischen Werte stammen (Regelungs- und Steuerungsaufgaben). Um die Analogwerte in die Rechenanlage eingeben zu können, müssen sie zuvor mit Hilfe von sog. Analog-Digital-Wandlern in digitale Werte umgewandelt werden. Diese Wandler arbeiten meist unabhängig von der Rechenanlage mit den zugehörigen Meßgeräten für die physikalischen Zustände zusammen. Jedem Wandler ist ein Pufferspeicher zugeordnet, der in bestimmten Zeitabständen gefüllt wird, d.h. jeweils den letzten Stand der Meßwertabfrage anzeigt. Die Einstell- bzw. Abfragehäufigkeit wird aufgrund der maximal möglichen Änderungsgeschwindigkeit der Meßwerte gewählt. Zur Erfassung der wichtigsten Zustandsgrößen eines zu steuernden Systems (z.B. einer chemischen Anlage) ist im allgemeinen eine Vielzahl von verschiedenen Meßwerten erforderlich. Die Meßwerte stehen auf ebensovielen Pufferspeichern für eine Zeit zwischen zwei durch die Analog-Digital-Wandler gesteuerten Einstellungen zur Verfügung und können von der Rechenanlage abgefragt werden. Ein Beispiel soll das verdeutlichen:

Bei einem System von Meßstellen sollen 100 verschiedene Meßwerte anfallen, die in Abständen von einer Minute zur Einstellung von 100 Pufferspeichern über 100 Analog-Digital-Wandler benutzt werden. Die Meßwerte erfordern bei der Verarbeitung eine Genauigkeit von 4 Stellen, also müssen mindestens 100 Pufferspeicher mit einer Speicherkapazität von je 4 Dezimalstellen vorhanden sein. Die Rechenanlage hat somit eine Minute Zeit, um den Inhalt der 100 Pufferspeicher zu übernehmen. Im Prinzip geschieht das folgendermaßen:
Jeder Analog-Digital-Wandler gibt nach vollendeter Einstellung des zugehörigen Pufferspeichers ein Freigabezeichen. Die Rechenanlage fragt zyklisch die 100 entsprechenden Leitungen ab und übernimmt beim Vorhandensein eines Freigabezeichens den Inhalt des zur „Freigabeleitung" gehörigen Pufferspeichers. Gleichzeitig wird das Freigabezeichen für diesen Pufferspeicher gelöscht und kann erst wieder nach erneuter Einstellung des Pufferspeichers durch den Analog-Digital-Wandler erscheinen. Leitungen ohne Freigabezeichen werden bei der zyklischen Abfrage überlaufen. Wie oft muß diese zyklische Abfrage erfolgen ? Sie muß im Beispiel mindestens einmal in der Minute erfolgen, um wirklich alle Meßwerte zu erfassen. Die Abfragegeschwindigkeit und Meßwertübernahmegeschwindigkeit der Rechenanlage muß so groß sein, daß die für diese Aufgabe benötigte Zeit wesentlich kürzer als eine Minute ist, damit die Meßdaten in der Rechenanlage auch noch verarbeitet werden können. Als Beispiel sei angeführt, daß die ZUSE Z 31 die Abfrage der 100 Meßstellen einschließlich der Übertragung der Digitalwerte in den Arbeitsspeicher der Rechenanlage in weniger als einer Sekunde durchführt, also noch genügend Zeit zur Verarbeitung der Meßwerte hat.

2.5 Ausgabe der Informationen

Bei der Eingabe werden die auf dem externen Informationsträger befindlichen Daten über entsprechende Anschlußgeräte in den Code der Rechenanlage umgewandelt und in diese eingegeben. Bei der Ausgabe wird genau der umgekehrte Weg beschritten. Es ist ebenso wie bei der Eingabe eine Pufferung, aber diesmal eine Pufferung der von der Rechenanlage gelieferten Informationen erforderlich. Die Größe des Pufferspeichers hängt von dem verwendeten Ausgabegerät ab und soll im folgenden näher betrachtet werden.

2.5.1 Ausgabe über einen Pufferspeicher für 1 Zeichen

Die Ausgabe über einen sog. Einzeichenpuffer ist nur möglich, wenn das Ausgabegerät die von der Rechenanlage gelieferten Zeichen auch zeichenweise auf dem externen Informationsträger speichern kann. Das ist z.B. bei folgenden Ausgabegeräten möglich:

Ausgabeschreibmaschine:
Es wird Zeichen nach Zeichen angeschlagen.

Lochstreifenstanzer:
Es wird Zeichen nach Zeichen gelocht.

Lochkartenstanzer:
Nur bei spaltenweiser Lochung der Zeichen (s.a. Eingabe) wird Zeichen nach Zeichen gelocht.

Genau wie bei der Eingabe ergibt sich bei zeichenweiser Ausgabe eine relativ kurze zusammenhängende Zeit, in der die Rechenanlage bei der Ausgabe rechnen kann, sofern die Ausgabegeräte dauernd arbeiten, also nicht auf die Rechenanlage warten sollen. Bei der Ausgabe liegen aber die Verhältnisse etwas günstiger, da die erreichbaren Ausgabegeschwindigkeiten der angeführten Geräte fast um eine Größenordnung niedriger liegen als bei den entsprechenden **Eingabegeräten.**

2.5.2 Ausgabe über einen Pufferspeicher für mehrere Zeichen

Bestimmte Ausgabegeräte benötigen einen Mehrzeichenpuffer, um überhaupt arbeiten zu können. Zwei wichtige Beispiele seien genannt:

Lochkartenstanzer für zeilenweises Lochen:
Für diesen kann ein 80-bit-Pufferspeicher (für 80-spaltige Lochkarten) verwendet werden. Hierbei ist allerdings erforderlich, daß der Inhalt einer gesamten Lochkarte vorher im Arbeitsspeicher der Rechenanlage so gespeichert wird, daß die Ausgabe Zeile für Zeile ermöglicht wird. In diesem Fall ergibt sich bei der Ausgabe eine wiederum nur relativ kurze zusammenhängende Zeit, in der die Rechenanlage Programme rechnen kann, die nicht mit der Ausgabe zusammenhängen. Alle diese Nachteile werden durch einen Ausgabepufferspeicher vermieden, der den gesamten Inhalt einer Lochkarte aufnehmen kann (s.a. Eingabe).

Ebenfalls einen Mehrzeichenpuffer benötigt:

Der Zeilen- oder Schnelldrucker
Hier wird praktisch eine gesamte Zeile auf einmal ausgedruckt, also müssen auch alle auf der Zeile auszudruckenden Zeichen auf einmal zur Verfügung stehen. Je nach der möglichen Anzahl der Anschläge pro Zeile (etwa 100) ist ein entsprechender Pufferspeicher erforderlich.

Die Zusammenarbeit der Ausgabegeräte mit der Rechenanlage erfolgt ähnlich wie bei der Eingabe. Pufferspeicher und Ausgabegerät arbeiten unabhängig von der Maschine zusammen. Ist der Puffer von der Rechenanlage gefüllt worden, so wird ein Startsignal an die Steuerelektronik gegeben, das eine Übertragung des Pufferinhaltes auf den externen Informationsträger einleitet. Die Rechenanlage kann den Pufferspeicher nur dann füllen, wenn der Inhalt des Pufferspeichers schon auf den externen Informationsträger gebracht wurde. Nur dann findet die Rechenanlage bei der Abfrage ein Freigabezeichen vor. Das Freigabezeichen beeinflußt also genau wie bei der Eingabe den Programmablauf der Rechenanlage. Das Flußdiagramm für die programmgesteuerte Ausgabe ist im Prinzip genauso aufgebaut wie das entsprechende Flußdiagramm für die Eingabe im Abschnitt I 2.4.

2.5.3 Ausgabe von Analogwerten

Eine Ausgabe von Analogwerten kann erforderlich sein, um z.B. irgendwelche von der Rechenanlage digital errechnete Werte sofort in Kurvenform darstellen zu können. Weiterhin wird die Analogausgabe benutzt, um z.B. aufgrund von errechneten Werten Stellgrößen für ein Regelsystem zu erhalten. Dieser Fall ist für die Automatisierung von Produktionsabläufen besonders wichtig und soll hier etwas näher betrachtet werden. Es sei nochmals auf das bei der Eingabe von Analogwerten betrachtete Beispiel verwiesen und angenommen, daß die 100 Meßwerte die physikalischen Zustände während eines

chemischen Prozesses in einer entsprechenden Anlage darstellen. Die Werte könnten z.B. Drücke, Temperaturen oder Durchflußmengen an verschiedenen Stellen der chemischen Anlage darstellen. Der chemische Prozeß soll nun in bestimmter Art und Weise ablaufen. Die Meßwerte kontrollieren den Ablauf, etwaige Abweichungen vom Sollwert müssen durch Veränderung der Stellglieder (z.B. Ventile, Heizeinrichtungen usw.) ausgeglichen werden. Mit Hilfe der eingegebenen Analogwerte stellt die Rechenanlage durch Vergleich mit gespeicherten Sollwerten die Abweichungen fest, berechnet die erforderlichen Stellgrößen und gibt sie in Form von Digitalwerten auf entsprechende Pufferspeicher aus. Jede Änderung des Pufferspeichers muß nun eine Veränderung der Stellglieder des chemischen Systems bewirken. Die Stellgrößen dafür sind meistens elektrische Werte (z.B. Strom in einer Heizeinrichtung, Strom eines Reglermotors usw.) Diese Analogwerte müssen also über einen sog. Digital-Analog-Wandler aus den Digitalwerten erzeugt werden, die die Rechenanlage berechnet hat. Die Einstellung der Stellglieder erfolgt meist relativ langsam, bezogen auf die Rechengeschwindigkeit der Anlage. Die Rechenanlage kann also den Pufferspeicher erst dann wieder erneut füllen, wenn die vorangegangene Einstellung der Stellglieder vollendet ist. Ein Freigabezeichen regelt auch hier wieder die Ausgabe der Werte auf den Pufferspeicher des zugehörigen Digital-Analog-Wandlers.

2.6 Kontrolleinrichtungen von Rechenanlagen

Eine programmgesteuerte elektronische Rechenanlage enthält eine sehr große Anzahl von elektrischen Bauelementen (Transistoren, Dioden, Widerstände, Kondensatoren usw.). Selbst kleine Anlagen bringen es auf mehr als 10.000 solcher Elemente. Es ist nicht zu vermeiden, daß ab und zu ein Element ausfällt und ersetzt werden muß. Wichtig ist es jedoch, daß die durch das Ausfallen eines Elementes hervorgerufene Fehler bei der Informationsverarbeitung sofort erkannt werden. Zu diesem Zweck besitzt jede Rechenanlage ein mehr oder weniger umfangreiches Kontrollsystem, das aus sog. Kontrollstellen besteht, an denen die Information auf ihre Richtigkeit geprüft wird. Es gibt verschiedene Methoden zur Prüfung, die im folgenden betrachtet werden.

2.6.1 Kontrollmöglichkeiten des Dezimalrechners

Wie schon unter I 1.3 gebracht, sind zur Darstellung einer Ziffer im Dezimalrechner mindestens 4 bit notwendig, mit denen sich 16 verschiedene Kombinationen bilden lassen. Zur Darstellung der Ziffern 0 ... 9 sind aber nur 10 Kombinationen erforderlich. Diese Tatsache kann man zur sog.

Gültigkeitskontrolle

benutzen. Die restlichen 6 Kombinationen stellen ungültige Ziffern oder sog. Pseudotetraten dar. Tritt an den hierfür vorgesehenen Kontrollstellen der Rechenanlage eine der Pseudotetraten auf, so meldet die Kontrollstelle „Alarm", dessen Auswertung unter I 2.6.3 behandelt wird. Eine weitere Fehlererkennungsmöglichkeit bietet die sog.

Quersummenkontrolle

Hierbei wird zur Darstellung einer Ziffer noch ein bit mehr verwendet. Dieses bit ergänzt die Anzahl der L-Zustände einer Ziffer auf eine ungerade Zahl. Die Kontrolleinrichtung zählt dann die Anzahl der L-Zustände einer Ziffer und gibt Alarm, falls sich eine gerade Zahl ergibt. Eine weitere Kontrollmöglichkeit bietet die sog.

Längssummenkontrolle

Hierbei wird im einfachsten Falle jedes Wort um ein bit erweitert, das wiederum die Anzahl der L-Zustände eines Wortes auf eine ungerde Zahl ergänzt. Quer- und Längssummenkontrolle entdekken einen Fehler nicht, wenn eine gerade Anzahl von L-Werten durch einen Fehler in der Anlage hinzukommt oder verschwindet, oder wenn die gleiche Anzahl von L-Werten hinzukommt und verschwindet. Der letzte Fehler ist aus technischen Gründen sehr unwahrscheinlich. Bei der Längssummenkontrolle kann die Sicherheit bei der Fehlererkennung erhöht werden, wenn mehr als ein bit pro Wort zur Kontrolle hinzugesetzt wird. Mit zwei bit wirkt die Kontrolle folgendermaßen: Ein Zähler zählt die L-Zustände eines Wortes, indem er zählt:

0 - 1 - 2 - 3 - 0 - 1 - 2 - 3 usw.

Am Ende des Wortes bleibt der Zähler auf einem Wert von 0 ... 3 stehen, dieser Wert muß mit dem Wert der zwei zusätzlichen bits übereinstimmen, die dem Wort zur Kontrolle angehängt wurden. Diese Art der Kontrolle nennt man eine Längssummenkontrolle „modulo 4". Prinzipiell ließe sich diese Möglichkeit auch auf die Quersummenkontrolle anwenden, wird aber aus wirtschaftlichen Gründen kaum verwendet werden, da sich dann pro Ziffer noch ein zusätzliches bit ergibt.

2.6.2 Kontrollmöglichkeiten des Binärrechners

Die Pseudotetradenkontrolle ist wegen der geschlossenen Binärdarstellung der gesamten Zahl nicht möglich. Bei Binärrechner im Serieparallel-Betrieb könnte man eine Quersummenkontrolle jeder Bitgruppe (s. 2.3.3) durchführen. Allgemein üblich bei Binärrechnern ist jedoch die Längssummenkontrolle im Rechner und evtl. eine Quersummenkontrolle bei der Ein- und Ausgabe, wo ja im allgemeinen eine binär-dezimale Zeichendarstellung verwendet wird (s.a. I 1.1).

2.6.3 Alarmauswertung der Kontrollstellen

Je nach Aufbau der Rechenanlage werden die Alarme der Kontrollstellen einzeln oder zusammengefaßt auf dem Bedienungspult angezeigt. Die Einzelanzeige hat den Vorteil, daß der Fehler in der Rechenanlage schneller gefunden wird, da die alarmanzeigende Kontrollstelle schon auf den fehlerhaften Maschinenkomplex hinweist.

Was soll nun bei einem Alarm geschehen ?

Im einfachsten Falle stoppt die Rechenanlage. Da der Inhalt des Befehlsregisters im allgemeinen angezeigt wird, kann der Bediener entsprechende Maßnahmen ergreifen, also z.B. den Teil des Programms, bei dem der Fehler auftrat, noch einmal von der Anlage rechnen lassen. Oftmals werden nämlich Fehler durch äußere Störungen hervorgerufen (starke Netzstörungen), so daß in der Rechenanlage in diesem Falle kein Defekt vorliegt. Erst wenn eine zweite Rechnung oder ein spezielles Fehlerprogramm, das die Anlage unter besonders harten Bedingungen prüft, nicht fehlerfrei abläuft, muß eine systematische Fehlersuche einsetzen. Größere Anlagen leiten diese Fehlerprogramme automatisch ein und stoppen die Anlage erst dann, wenn wirklich ein Defekt der Rechenanlage vorliegt. Bei Störungen von außen merkt dann der Bediener der Rechenanlage überhaupt nichts von dem Fehler, da dieser automatisch korrigiert wird.

Allgemein kann jedoch gesagt werden:

Es ist wichtig, daß eine fehlerbehaftete Informationsverarbeitung in der Rechenanlage möglichst bald angezeigt wird, damit entsprechende Maßnahmen ergriffen werden können.

2.7 Vorrangsteuerung einer Rechenanlage

Die Datenverarbeitungsgeschwindigkeit einer Rechenanlage hängt insbesondere bei kommerziellen Aufgaben in starkem Maße von der erreichbaren Ein- und Ausgabegeschwindigkeit der Daten ab. Die Geschwindigkeit der Ein- und Ausgabegeräte liegen im allgemeinen mehrere Größenordnungen niedriger als die interne Verarbeitungsgeschwindigkeit der Rechenanlage. Es muß also angestrebt werden, daß die Ein- und Ausgabegeräte dauernd arbeiten. Die Geräte müssen deshalb von der Rechenanlage optimal versorgt werden. Außerdem muß die Rechenanlage ohne Störung des Daten-Ein- und Ausgabeflusses die geforderten Operationen mit den Daten „nebenbei" erledigen. Die Ein- und Ausgabegeräte selbst haben unterschiedliche Geschwindigkeiten, so daß meistens das langsamste Gerät die effektive Datenverarbeitungsgeschwindigkeit begrenzt. Dieses Gerät muß demnach vor allen anderen von der Rechenanlage versorgt werden, sobald es in der Lage ist, mit der Rechenanlage zu verkehren, d.h. sobald das Gerät ein Freigabezeichen gibt.

Weiterhin müssen auftretende Fehler und Eingriffe vom Bedienungspult her mit besonderem Vorrang behandelt werden, so daß man mehrere sog. Vorrangstufen unterscheiden kann, die im folgenden näher betrachtet werden sollen. Zur ersten Vorrangstufe gehören die Eingriffe in das Programm der Rechenanlage durch auftretende Fehler bei der Datenverarbeitung oder durch die Stel-

lung bestimmter Schalter des Bedienungspultes. Es seien zunächst die Möglichkeiten beim Auftreten von Fehlern behandelt. Wie schon unter 2.6.3 erwähnt, kann die Rechenanlage stoppen oder automatisch ein entsprechendes Testprogramm aufrufen. Um das zu erreichen, wird nach der Ausführung jedes Befehls geprüft, ob ein Fehler aufgetreten ist. Der Fehler hat also eine absolute Vorrangigkeit. Gleichberechtigt in dieser Hinsicht sind gewisse Schalter des Bedienungspultes, z.B. die Stoptaste, der Adressenstop und einige vom Programm abfragbare Blockierungsschalter. Bei der Behandlung des Bedienungspultes der ZUSE Z 31 wird darauf noch näher eingegangen werden.

Erst wenn die Forderungen der Einrichtungen der ersten Vorrangstufe erfüllt sind bzw. keine Forderungen vorliegen (z.B. keine Programmunterbrechung wegen eines angezeigten Fehlers), können die Forderungen der zweiten Vorrangstufe erfüllt werden. Die zweite Vorrangstufe bezieht sich auf die Ein- und Ausgabegeräte, die unter sich wieder unterschiedliche Vorrangigkeit haben. Ein Beispiel soll das erläutern. Eine Rechenanlage habe eine Lochkarten-Ein- und Ausgabe sowie einen Zeilendrucker. Der Zeilendrucker habe so viele Informationen auszudrucken, daß er dauernd mit seiner Höchstgeschwindigkeit arbeiten muß, um die gestellte Aufgabe zu erfüllen. Erst wenn der Zeilendrucker mit seiner Information versorgt ist, kann der Lochkartenstanzer und erst dann der Lochkartenleser versorgt werden. Das Programm sei so beschaffen, daß es in der Zeit, in der die 3 Geräte unabhängig von der Rechenanlage arbeiten, durchgeführt werden kann. Der Lochkartenleser soll so schnell arbeiten, daß dauernd genügend Daten für das Programm zur Verfügung stehen.

Im einfachsten Fall werden bei der Vorrangprogrammierung im Programm sog. Abfragezyklen eingebaut, die in bestimmten Abständen die 3 Anschlußgeräte abfragen, ob sie in der Lage sind, mit der Rechenanlage zu verkehren. Zur Steuerung dient wiederum das schon erwähnte Freigabezeichen, das besagt, daß der Puffer des Lochkartenlesers voll, die Puffer des Lochkartenstanzers und des Zeilendruckers leer sind. Im Abfragezyklus wird also in der Reihenfolge Zeilendrucker, Lochkartenstanzer und Lochkartenleser abgefragt. Die Zeit zwischen Abfragezyklen muß kleiner als die Taktzeit des schnellsten Anschlußgerätes sein. Die Taktzeit des Lochkartenlesers ist z.B. die Einlesezeit einer Lochkarte in den Pufferspeicher zuzüglich der Kartentransportzeit. Überschreitet die Zeit zwischen zwei Abfragezyklen die Taktzeit eines Gerätes, so kann dieses Gerät nicht mehr optimal arbeiten.

Bei der beschriebenen Vorrangsteuerung bestimmt immer noch das Programm der Rechenanlage, wann es durch Einrichtungen der zweiten Vorrangstufe unterbrochen werden darf. Eine Unterbrechung ist eben nur dann möglich, wenn das Programm einen Abfragezyklus durchführt. Es wird an den Stellen des Programmes sein, die hierfür günstig sind, wo also z.B. eine logische zusammenhängende Operationsfolge abgeschlossen ist. Es wird nicht immer möglich sein, die oben erwähnten Zeiten zwischen den Abfragezyklen einzuhalten, wodurch sich gewisse Wartezeiten der Anschlußgeräte ergeben können. Diese Zeiten lassen sich jedoch im Vergleich zur wirklichen Arbeitszeit der Anschlußgeräte im allgemeinen so gering halten, daß sich nur eine relativ kleine Verlängerung der gesamten Datenverarbeitungszeit ergibt. Der Vorteil einer solchen Vorrangsteuerung ist ein nur geringer zusätzlicher Aufwand, so daß sich dieses System für kleinere und mittlere Anlagen anbietet.

Großanlagen benutzen hingegen oft eine sog. direkte Vorrangsteuerung der Rechenanlage durch die Anschlußgeräte. Sobald ein Anschlußgerät frei ist, also die Ein- oder Ausgabe des zugehörigen Pufferspeichers beendet ist, veranlaßt es die Unterbrechung des laufenden Rechenprogramms der Anlage, wobei wiederum die Geräte untereinander verschiedene Vorrangstellungen haben. Das erfordert in der Rechenanlage erheblichen technischen Aufwand, da z.B. alle Zwischenergebnisse des Rechenprogrammes weggespeichert werden müssen, die Unterbrechungsstelle des Programmes gekennzeichnet werden muß usw. Das muß mit entsprechender Geschwindigkeit geschehen, die im allgemeinen nur größere Rechenanlagen besitzen.

2.8 Vergleich zwischen wissenschaftlichen und kommerziellen Anlagen

Wissenschaftliche und kommerzielle Aufgaben unterscheiden sich im Prinzip folgendermaßen:

Bei wissenschaftlichen Aufgaben werden im allgemeinen nur relativ wenige Daten eingegeben, aber diese Daten umfangreichen und meist komplizierten Berechnungen unterworfen. Ausgege-

ben werden ebenfalls nur relativ wenige Daten.

Bei kommerziellen Anlagen werden im allgemeinen sehr viele Daten eingegeben und nur wenige, meist sehr einfache Operationen mit diesen Daten unternommen. Ausgegeben werden ebenfalls sehr viele Daten.

Aus wirtschaftlichen Gründen werden deshalb verschiedene Rechenanlagen für wissenschaftliche und kommerzielle Anwendungen gebaut. Man spricht direkt von wissenschaftlichen oder kommerziellen Rechenanlagen. Die charakteristischen Eigenschaften der beiden Rechenanlagen sollen im folgenden kurz betrachtet werden.

2.8.1 Rechenanlagen für wissenschaftliche Anwendungen

Wissenschaftliche Programme enthalten im allgemeinen eine große Anzahl von Befehlen, die während des gesamten Programmes dauernd benötigt werden und deshalb in der Anlage gespeichert werden müssen. Außerdem entstehen in vielen Fällen eine große Anzahl von Zwischenergebnissen. Oft ist es auch notwendig, viele Tabellenwerte zu speichern.
Aus diesen Gründen benötigt eine wissenschaftliche Rechenanlage einen relativ großen Arbeitsspeicher.

Die umfangreichen Programme bedingen eine relativ hohe interne Datenverarbeitungsgeschwindigkeit. Da die Operationsgeschwindigkeit von Binärrechnern bei gleicher Bitzahl größer ist als die von Dezimalrechnern (s.a. I 1.2), sind wissenschaftliche Anlagen meistens Binärrechner.

Der Befehlscode einer wissenschaftlichen Anlage hat einen wesentlichen Einfluß auf die Länge der Programme. Sog. flexible Codes sparen Befehle, da durch einen Befehl mehrere Operationen gleichzeitig bewirkt werden können. Somit wird Speicherraum für die Befehle sowie Verarbeitungszeit gespart.

Für die Ein- und Ausgabe hingegen ist nur wenig Aufwand erforderlich. Im allgemeinen genügen relativ langsame Geräte, um die wenigen Daten ein- und auszugeben.

2.8.2 Rechenanlagen für kommerzielle Anwendungen

Die Programme bestehen im allgemeinen aus relativ wenigen Befehlen, da die zwar großen Datenmengen gruppenweise nach den gleichen Gesichtspunkten verarbeitet werden (z.B. Lohnabrechnung, Materialabrechnung, Bankbuchungsprogramm usw.).
Es sind also nur relativ kleine Programme in der Anlage zu speichern. Im allgemeinen gibt es auch nur wenige Zwischenergebnisse bei der Berechnung.

Zur Speicherung von Daten als Grundlage für die Berechnung ist allerdings meistens eine außerordentlich große Speicherkapazität erforderlich wie z.B. bei der Speicherung des Lagerbestandes, der Angaben für die einzelnen Bankkonten usw. Diese Daten werden in den schon erwähnten Nachschubspeichern gespeichert.

Kommerzielle Anlagen benötigen also einen relativ kleinen Arbeitsspeicher, jedoch sehr oft außerordentlich große Nachschubspeicher.

Die interne Operationsgeschwindigkeit braucht jedoch nicht sehr hoch zu sein. Allerdings muß sie genügen, um eine optimale Versorgung der Anschlußgeräte bei der Vorrangsteuerung zu gewährleisten.

Der Befehlscode einer rein kommerziellen Anlage ist meist recht einfach, weshalb sich wissenschaftliche Aufgaben auch nur sehr schwierig (sehr lange Programme) mit einer solchen Anlagen lösen lassen.

Das Hauptgewicht bei einer kommerziellen Anlage liegt bei den Ein- und Ausgabegeräten. Hier sollte die Anlage flexibel sein, d.h. den Anschluß von Ein- und Ausgabegeräten verschiedener Art und Geschwindigkeit erlauben. Das gleiche gilt für die Nachschubspeicher.

Eine kommerzielle Anlage sollte also im sog. Baukastensystem aufgebaut sein, um die Anlage den verschiedenen Aufgaben optimal anpassen zu können. Wegen der großen ein- und auszugebenden Datenmengen ist ein Dezimalrechner besonders zur Lösung kommerzieller Aufgaben geeignet.

Die scharfe Trennung in wissenschaftliche und kommerzielle Anlagen hat gewisse Nachteile, da in vielen Fällen auf der Anlage wissenschaftliche und kommerzielle Aufgaben gerechnet werden müssen, wobei das eine Gebiet meistens nur nebenbei gemacht werden soll, dann aber wegen der erwähnten Eigenart der Anlage zu einem schwierigen Problem wird.

Aus diesem Grunde bieten Anlagen, die zumindestens in einigen Punkten die Forderungen für beide Aufgaben erfüllen, erhebliche wirtschaftliche Vorteile. Als wichtigster Punkt sei hier der Befehlscode sowie die Möglichkeit eines Speicherausbaues der Anlage erwähnt.

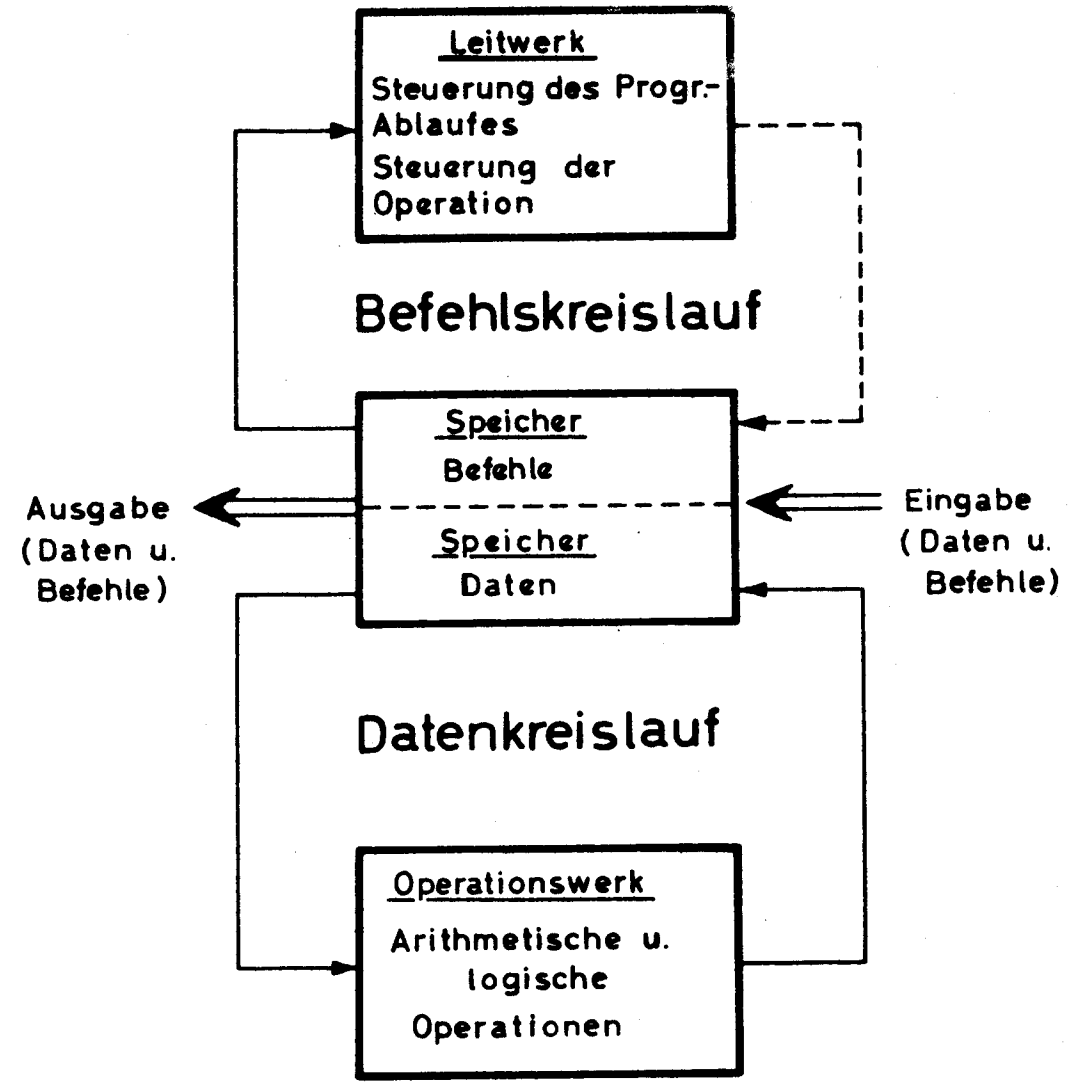


Abb.1: Blockschaltbild einer programmgesteuerten Rechenanlage.

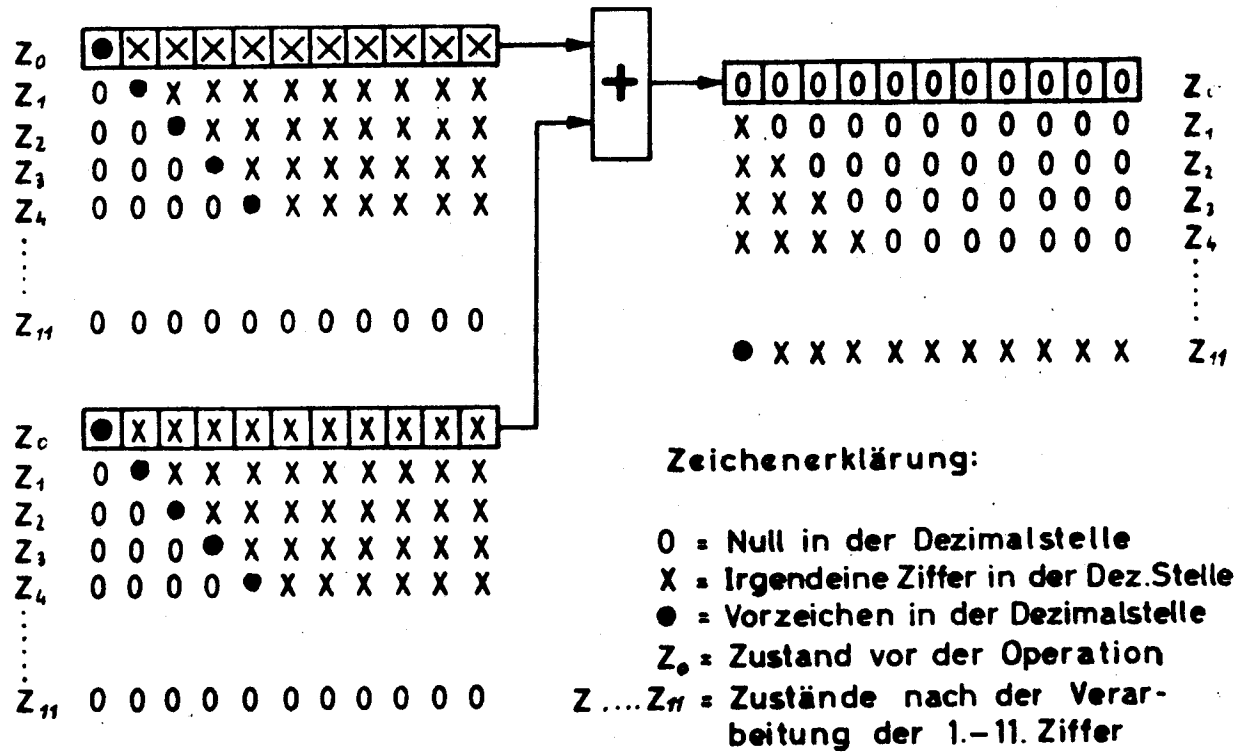


Abb.2: Addition zweier Zahlen mit Hilfe von Schieberegistern.

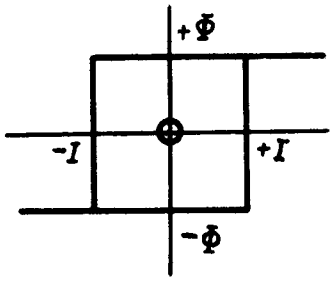


Abb.3: Hysteresisschleife eines Ferritkerns.

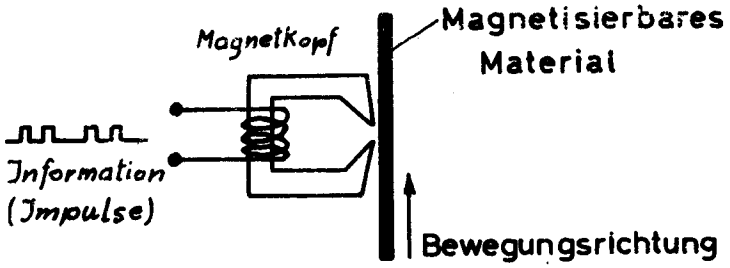


Abb.4 Prinzip des Magnetspeichers

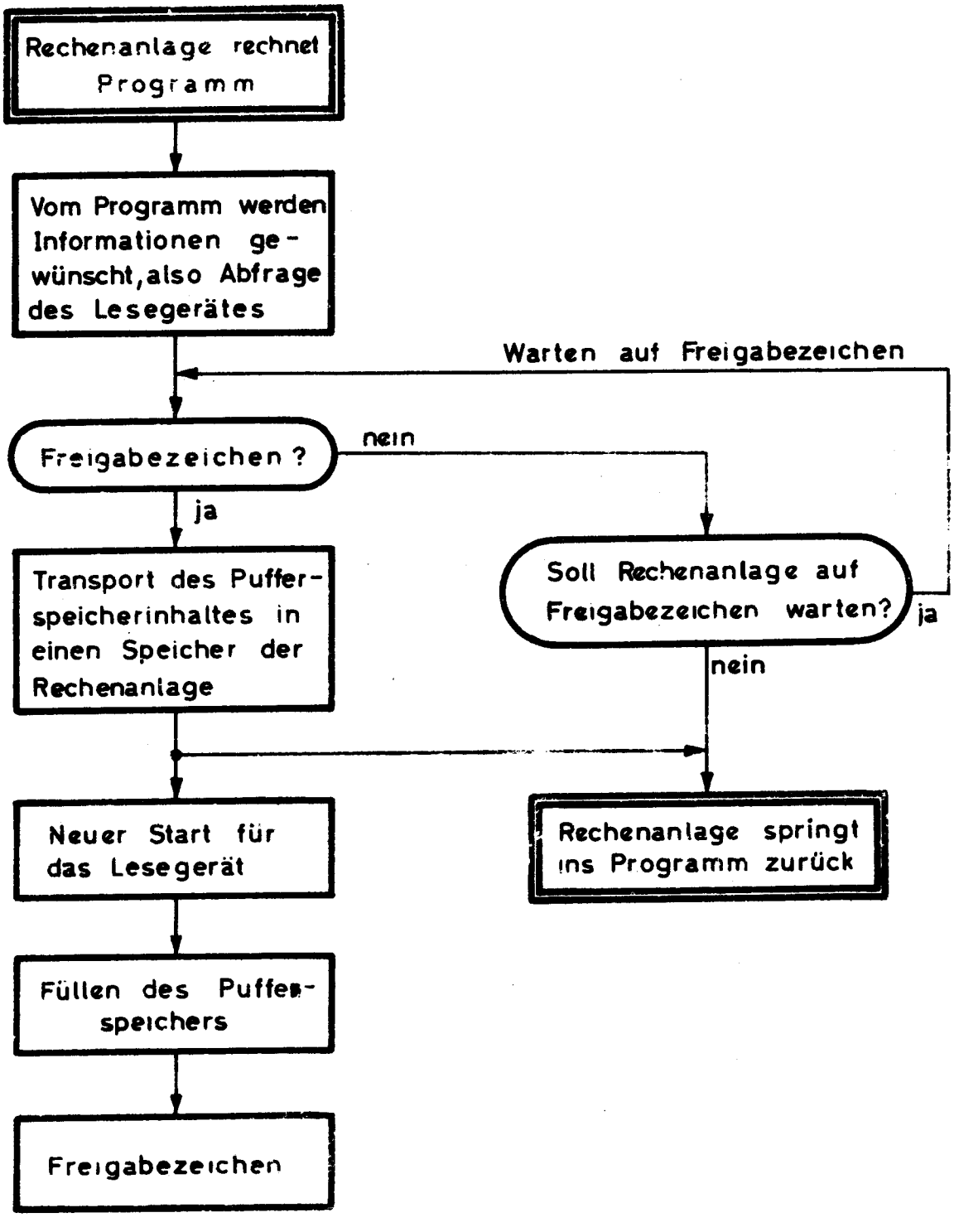


Abb 5. Strukturdiagramm für die programmgesteuerte Eingabe

EINFÜHRUNG IN DIE ARBEITSWEISE

der

programmgesteuerten Transistorrechenanlage

Z U S E Z 31

II Aufbau und Arbeitsweise der programmgesteuerten Transistorrechenanlage ZUSE Z 31

II. Aufbau und Arbeitsweise der programmgesteuerten elektronischen Rechenanlage ZUSE Z 31

II. AUFBAU UND ARBEITSWEISE DER PROGRAMMGESTEUERTEN ELEKTRONISCHEN RECHENANLAGE Z U S E Z 31

Die ZUSE Z 31 ist die Zentraleinheit eines Systems von Datenverarbeitungsgeräten. Die Ausbaufähigkeit dieses Systems geht aus der Druckschrift

ZUSE Z 31 - System einer Datenverarbeitungsanlage

hervor. Während dort der Aufbau und die Arbeitsweise der Zentraleinheit nur in großen Zügen geschildert wurden, soll im folgenden etwas näher auf die Zentraleinheit und ihre Anschlußgeräte eingegangen werden. Befehle der ZUSE Z 31 werden nur erklärt, soweit es zum Verständnis notwendig ist. Genaue Auskunft über die möglichen Befehle und ihre Bedeutung gibt eine spezielle Ausarbeitung über den Interncode der ZUSE Z 31.

1. Informationsdarstellung in der Zentraleinheit ZUSE Z 31

Die Informationseinheit der ZUSE Z 31 ist das Wort, das aus 44 bits besteht. Die Information wird im Wort binär-dezimal verschlüsselt und zwar in 11 Gruppen zu je 4 bits. In jeder Gruppe kann eine Dezimalziffer dargestellt werden. Die sich somit ergebenden 11 Dezimalstellen werden mit D 1 ... D 11 bezeichnet, wobei D 1 die niedrigste Wertigkeit (z.B. bei Zahlen) und D 11 die höchste Wertigkeit hat. D 11 dient als Kennzeichenstelle und gibt an, ob die Dezimalen D 10 ... D 1 als Zahlen-, Befehls- oder Textwort in der Maschine gedeutet werden sollen.

Als binär-dezimale Verschlüsselung jeder Dezimale wird der sogenannte 3-Excess-Code verwendet. 3-Excess-Code deshalb, da die binäre Wertigkeit der 4 bits einer Gruppe um 3 höher als der definierte Wert der entsprechenden Dezimalziffer ist. Nachstehende Tabelle zeigt die Verschlüsselung der Ziffern 0 ... 9 in diesem Code. 6 dieser 16 Kombinationsmöglichkeiten sind ungültige Ziffern (Pseudotetraden) und werden zur Kontrolle ausgenutzt (s. a. I 2.6.1).

Dezimalziffer in der ZUSE Z 31		Binärzahl 2^3	2^2	2^1	2^0	Binärzahl in dezimaler Schreibweise (binäre Wertigkeit)
Ungültige Ziffern (Pseudotetraden)		0	0	0	0	0
		0	0	0	L	1
		0	0	L	0	2
Gültige Dezimalziffern	0	0	0	L	L	3
	1	0	L	0	0	4
	2	0	L	0	L	5
	3	0	L	L	0	6
	4	0	L	L	L	7
	5	L	0	0	0	8
	6	L	0	0	L	9
	7	L	0	L	0	10
	8	L	0	L	L	11
	9	L	L	0	0	12
Ungültige Ziffern (Pseudotetraden)		L	L	0	L	13
		L	L	L	0	14
		L	L	L	L	15

1.1 Darstellung von Zahlen in der ZUSE Z 31

Jedes Wort, das in der Kennzeichenstelle D 11 eine 0 oder 9 hat, wird von der Anlage als Zahl gedeutet, wobei

0 eine positive Zahl
und 9 eine negative Zahl kennzeichnet.

Negative Zahlen werden im sogenannten 10er Komplement dargestellt, woraus sich automatisch die 9 in D 11 ergibt. Die Rechenregeln mit so dargestellten Zahlen können in der entsprechenden Literatur nachgeschlagen werden.

1.1.1 Darstellung von Festkommazahlen

Bei der Festkommadarstellung wird für eine bestimmte Aufgabe das Komma an einer festen Stelle der Zahl definiert. Das Komma wird nicht wirklich im Wort gesetzt, sondern der Programmierer hat das Programm so aufgestellt, als sei das Komma eben an dieser Stelle der Zahl. Unter der Annahme, daß das Komma hinter D1 ist, es sich also um ganze Zahlen handelt, ergibt sich in Festkommadarstellung für die Zahl z der Bereich

$$-10^{10} < z < +10^{10}.$$

Für kommerzielle Aufgaben wird fast immer mit Festkommazahlen gerechnet werden, da hier die Größenordnung der zu verarbeitenden Zahlen und der erwarteten Ergebnisse von vornherein festliegt, also die Stellung des Kommas festgelegt werden kann. Anders ist es jedoch bei wissenschaftlichen Aufgaben. Man benutzt hier beim Rechnen die

1.1.2 Darstellung von Gleitkommazahlen.

Es wird eine sogenannte halblogarithmische Zahlendarstellung verwendet. Jede Zahl besteht aus dem Vorzeichen, der 8-stelligen Mantisse und dem 2-stelligen Exponenten. Die Mantisse wird in den Stellen D10 ... D3, der Exponent in den Stellen D2 und D1 des Wortes dargestellt. D11 dient wie bisher als Vorzeichenstelle.
Damit ist folgender Zahlenbereich möglich:

$$\pm 0{,}00\,000\,001 \cdot 10^{-50} \quad \ldots\ldots\ldots\ldots\ldots \quad \pm 0{,}99\,999\,999 \cdot 10^{+49}$$

Um das Vorzeichen des Exponenten nicht gesondert darstellen zu müssen, wird folgende Konvention bei der Exponentendarstellung getroffen:

Die Zahlen des Exponenten

00 ..... 99 in der ZUSE Z 31 (dargestellt in D2 und D1)

stellen die wirklichen Exponentenwerte

-50 ..... +49 in der Rechnung dar.

Es wird also z.B. der Exponent 50 der ZUSE Z 31 in der Rechnung als Exponent 00 gedeutet.

Die Zahl $0{,}88\,888\,888 \cdot 10^{10}$ würde also z.B. in der ZUSE Z 31 folgendermaßen dargestellt:

D11	D10 . . . . . D3	D2 D1
0	8 8 8 8 8 8 8 8	6 0
Vorzeichen	Mantisse	Exponent

44 bit

Bei der Gleitkommadarstellung ergibt sich also ein absoluter Zahlenbereich, der über 100 Zehnerpotenzen geht, während er sich bei Festkommadarstellung nur über 10 Zehnerpotenzen erstreckt. Hinsichtlich der Vor- und Nachteile der beiden Darstellungsarten wird auf entsprechende Literatur verwiesen.

1.2 Darstellung von Befehlen in der ZUSE Z 31

Jedes Wort, das in der Kennzeichenstelle D11 eine 1, 2, 3, 6, 7 oder 8 hat, wird von der Rechenanlage als Befehl gedeutet. Es handelt sich hierbei um 6 verschiedene Befehle, auf die später noch eingegangen wird. Die Stellen D10 ..... D1 eines solchen Befehlswortes müssen gültige Ziffern sein. Nachstehend wird der Aufbau eines Befehlswortes gezeigt:

D11	D10 D9	D8 D7 D6 D5	D4 D3 D2 D1
Befehlskennzeichen	Bedingungszeichen	Operationszeichen	Adresse

Sowohl die Bedingungs- als auch die Befehlszeichen sind in den entsprechenden Dezimalen so gewählt, daß die Dezimalen weitgehend miteinander kombinierbar sind. Es können also in einem Befehl 2 Bedingungen und 4 Operationszeichen gegeben werden. Damit ergibt sich eine große Anzahl von sinnvollen Befehlen, die eine Aufstellung von befehlssparenden und trotzdem schnellen Programmen erlauben.

1.3 Darstellung von Text in der ZUSE Z 31.

Jedes Wort, das in der Kennzeichenstelle D11 eine 4 oder 5 hat, wird von der Rechenanlage als Text gedeutet.

Ist das Kennzeichen 5, so müssen die Dezimalen D10 ..... D1 gültige Ziffern sein. Jeweils 2 Ziffern stellen dann ein Textzeichen dar, so daß sich in einem Wort 5 Textzeichen darstellen lassen. Mit dem so dargestellten Textwort können auch arithmetische Operationen durchgeführt werden, da jedes Textwort eine gültige Zahl darstellt. Es können also z.B. bei Vergleichsoperationen Additionen und Subtraktionen ausgeführt werden, wodurch sich ganz erhebliche Vorteile ergeben.

Die Verschlüsselung der Textzeichen (einschließlich der Ziffern im Text) zeigt nachstehende Tabelle:

00 ... 09	Ziffern 0 ... 9,
10 ... 19	Satz- und Funktionszeichen,
20 ... 29	a ..... j,
30 ... 39	k ..... t,
40 ... 49	u ..... z sowie 4 weitere Funktionszeichen.

Ein weiterer Vorteil dieser Textverschlüsselung besteht darin, daß bei der Lochstreifen-Ein- und Ausgabe keine Umschlüsselungen notwendig sind, da hier die externe und interne Verschlüsselung bis auf das Quersummenbit (s.a. I 2.6.1) gleich sind.

Ist das Kennzeichen 4, so können in den 40 zu den Dezimalstellen D10 ... D1 gehörigen bits Informationen in einem beliebigen Code dargestellt werden. Es können also ungültige Ziffern vorkommen.
Verwendet man 8 bits zur Darstellung eines Textzeichens, so können auch hier 5 Textzeichen pro Wort dargestellt werden. Spezielle logische Befehle erlauben eine Verarbeitung dieser Wörter in der Rechenanlage. Es wird sich hierbei im wesentlichen um Transport- oder Vergleichsbefehle handeln. Wegen der möglichen ungültigen Ziffern bei dieser Textdarstellung wird die Gültigkeitskontrolle durch die 4 in der Kennzeichenstelle automatisch abgeschaltet.

2 Aufbau der ZUSE Z 31

Die Zentraleinheit ZUSE Z 31 läßt sich logisch in 4 Hauptteile gliedern:

1. Das Speicherwerk
2. Die Rechenregister
3. Das Leitwerk
4. Das Operationswerk

Anhand des Blockschaltbildes am Ende dieses Kapitels folgt zunächst eine kurze Beschreibung dieser Teile der Rechenanlage.

2.1 Das Speicherwerk

Die ZUSE Z 31 besitzt zwei Arbeitsspeicher mit geringer Zugriffszeit. Es besteht weiterhin die Möglichkeit, zwei verschiedene Arten von Nachschubspeichern, und zwar Magnettrommeln und Magnetbänder anzuschließen. In diesem Abschnitt sollen nur die beiden Arbeitsspeicher und einige Hilfsspeicher der Zentraleinheit behandelt werden.

Es sind dies:

2.1.1 Der Programmspeicher
2.1.2 Der Schnellspeicher

Bei den Arbeitsspeichern müssen wir strukturell zwei Typen unterscheiden:

a.) den sog. Festspeicher
b.) den variablen Speicher.

Beim Festspeicher enthält jedes Wort eine bestimmte konstante (feste) Information, die bei Bedarf von der Rechenanlage beliebig oft abgerufen werden kann. Eine Veränderung dieser Information im Festspeicher ist nur auf technischem Wege möglich (z.B. Verdrahtungsänderung, Kernwechsel).

Im Gegensatz hierzu kann der variable Speicher nach dem Abruf von Informationen mit beliebig verschiedenen neuen Informationen im Rahmen des Programms versorgt werden. Der variable Speicher ermöglicht überhaupt erst die Funktion einer programmgesteuerten Rechenanlage.

2.1.1 Der Programmspeicher

In den verschiedenen Programmen einer Rechenanlage werden im allgemeinen gleiche Programmteile mehrmals benötigt, entweder in Form der schon erwähnten Unterprogramme oder in Form von allgemeinen Hauptprogrammteilen. Bei kleineren und mittleren Rechenanlagen, deren Operationswerke aus wirtschaftlichen Gründen sehr einfach aufgebaut sind, ergeben sich für diese Zwecke mehrere hundert Befehle. Würde man diese Befehle, die ja mit einer möglichst kurzen Zugriffszeit erreichbar sein müssen, im variablen Arbeitsspeicher unterbringen, müßte dieser entsprechend größer sein. Da sich die Befehle dieser Programme (z.B. Ein- und Ausgabeprogramme, arithmetische Programme usw.) nicht ändern, bedient man sich als Programmspeicher eines Festspeichers, der bei gleicher Kapazität nur etwa 20 - 25 % des Aufwandes eines variablen Speichers erfordert. Die Programmteile im Festspeicher werden dann durch die Befehle des Hauptprogrammes, die sich ihrerseits im variablen Speicher befinden, abgerufen. Es bleibt damit die Flexibilität der Anlage erhalten, da ja die Befehle des variablen Speichers bei Bedarf geändert werden können.

Weiterhin ist es möglich, konstante Werte im Festspeicher unterzubringen, z.B. Umschlüsselungstabellen, trigonometrische und logarithmische Tabellen usw. Auch hierdurch kann wieder Speicherkapazität im variablen Speicher gespart werden.

Der Programmspeicher der ZUSE Z 31 kann sowohl Befehle als auch Konstanten speichern und hat eine Kapazität von 2600 Worten. Er besteht aus 26 Kernreihen zu je 44 bits. Die Verdrahtung jeder Kernreihe bietet die Möglichkeit, hundert Worte zu speichern. Zu diesem Zwecke existieren 100 sogenannte Programmleitungen. Jede Programmleitung ist so durch alle Kernreihen gefädelt, daß ein Strom durch eine der Programmleitungen alle 26 Kernreihen in einer gewünschten Weise magnetisiert. Soll ein Kern auf L stehen, ist die Programmleitung durch den Kern gefädelt, soll er auf 0 bleiben, geht sie daran vorbei. Es sei nochmals betont, daß die Information des Programmspeichers durch die Lage der 100 Programmleitungen bestimmt wird.

Wie sieht nun ein Zyklus dieses Speichers aus ?
Im Anfang werden alle Kerne gelöscht, d.h. auf 0 gesetzt. Durch die Dezimalen D2 und D1 eines Befehlswortes kann eine der 100 Programmleitungen ausgewählt werden. Ein Strom in dieser Leitung aktiviert die bisher nur durch die Leitungsführung vorbestimmten Kerne aller 26 Kernreihen. Durch die Dezimalen D4 und D3 wird anschließend eine der 26 Kernreihen ausgewählt, d.h., sie erhält einen Strom, der in der Lage ist, alle Kerne dieser Reihe auf 0 zurückzukippen. Nur die aktivierten Kerne können zurückkippen (die anderen sind ja schon auf 0) und beim Kippen ein Lesesignal abgeben. Die vorhandenen oder nicht vorhandenen Lesesignale aller 44 Kerne der Reihe (des Wortes) werden benutzt, um ein Register von 44 bits (11 Dezimalen) über 44 parallele Leitungen zu „setzen", d.h. die Information des gelesenen Wortes in einem Register zu speichern.

Befindet sich im angewählten Wort des Programmspeichers ein Befehl, so wird das Befehlsregister (s. II 4.1) gesetzt, befindet sich im angewählten Wort eine Konstante, so wird das Konstantenregister gesetzt. Nach dem Setzen des entsprechenden Registers kann dann die weitere Verarbeitung des Wortes erfolgen.

Die 26 Kernreihen des Programmspeichers befinden sich auf 2 auswechselbaren Blöcken, die die Kerne mit der entsprechenden Verdrahtung für je 1300 Worte enthalten.

Der eine Block dient zur Aufnahme der sogenannten Grundprogramme (Unterprogramme) und evtl. erforderlicher Konstanten, der zweite Block kann konstante Teile des Hauptprogrammes sowie ebenfalls konstante Zahlen- und Textworte aufnehmen. Die Austauschbarkeit der Blöcke ermöglicht es, Programme in wenigen Sekunden gegeneinander auszutauschen. Es könnte also z.B. das Programm für die Materialabrechnung nach der Durchführung gegen das der Lohnabrechnung ausgewechselt werden. Infolge des befehlssparenden Interncodes der ZUSE Z 31 werden jedoch meistens die 2 600 Programmspeicherzellen zur Lösung aller Aufgaben eines Betriebes ausreichen.

2.1.2 Der Schnellspeicher

Der Schnellspeicher ist ein variabler Ferritkernspeicher, der durch eine Vielzahl von Einheiten zu je 200 Worten von einer Anfangskapazität von 200 Worten bis zu einer Maximalkapazität von 9 000 Worten ausgebaut werden kann. Die Adresse eines Wortes wird durch die Dezimalen D4 ... D1 des Befehlswortes ausgewählt.

Der Schnellspeicher gibt entsprechend der Serienorganisation der ZUSE Z 31 seine Information von 44 bits in Serie ab, ist also in Bezug auf das Lesen ein Serienspeicher. Das Speichern eines Wortes erfolgt parallel aus dem im Blockschaltbild eingezeichneten Konstantenregister.

Dadurch ist es möglich, in einer Wortzeit den Speicher zu lesen und die gelesene oder eine andere (berechnete) Information zurückzuschreiben. Jeder Kernreihe eines Wortes sind zwei weitere Kerne zugeordnet, die eine Längssummenkontrolle modulo 4 ermöglichen. Beim Schreiben werden die L-Werte der 44 bits von einem Zähler modulo 4 gezählt (s. I 2.6.1) und die beiden Kontrollkerne entsprechend gesetzt; beim Lesen erfolgt wiederum eine gleichartige Zählung der L-Werte. Entspricht der Zählerstand nicht der Stellung der beiden Kontroll-bits, so wird Alarm gegeben.

2.1.3 Hilfsspeicher

Die Hilfsspeicher dienen überwiegend zur Steuerung des Programmablaufes in Verbindung mit den Bedingungszeichen in den Dezimalen D10 und D9 des Befehlswortes.

Die beiden Zählregister ZR1 und ZR2 können je eine Dezimalziffer aufnehmen. Durch bestimmte Befehle können diese Register gesetzt, durch andere, sogenannte Zählbefehle um +1 oder -1 verändert werden.

Weiterhin hat die ZUSE Z 31 6 einstellige Binärspeicher B1 ... B6, die durch entsprechende Befehle auf L oder 0 gesetzt und genau wie die Zählregister durch Bedingungsspeicher abgefragt werden können. Die Bedingungsspeicher sind im Blockschaltbild nicht eingezeichnet, um dieses nicht zu überlasten.

Sowohl die Zählregister als auch alle 6 Bedingungsspeicher gemeinsam sind adressiert; die darin enthaltene Information kann damit als Operand in einer Rechnung verwendet werden.

2.2 Die Rechenregister

Die ZUSE Z 31 besitzt zwei unabhängige Rechenregister, und zwar das X- und das Y-Register. In jedem Register läßt sich ein Wort speichern. Jedes Register kann seinen Inhalt als Operand für eine Rechnung abgeben, kann aber auch das Ergebnis der Rechnung aufnehmen. Beide Register besitzen sogenannte Verschiebeeinrichtungen, und zwar kann in jeder Wortzeit der Inhalt des Registers entweder um eine Dezimalstelle nach links oder rechts verschoben werden; das entspricht einer Multiplikation des Registerinhaltes mit 10 bzw. 0,1. Abb. 6 am Ende von Kapitel II zeigt das Blockschaltbild des X-Registers. Es soll daran gezeigt werden, wie das Register Informationen aufnehmen und abgeben kann. Außerdem soll die Wirkungsweise der Links- und Rechtsverschiebung erklärt werden.

Jedes der mit X_0 - X_{12} bezeichneten Kästchen stellt eine Dezimalspeicherstelle dar, besteht also auf Grund des ZUSE Z 31-Codes aus 4 Binärspeichern (Flip-Flops). In den Stellen X_1 ... X_{11} befinden sich im Ruhezustand die Dezimalen D1 ... D11 eines Wortes. In X_{12} wird nochmals die

Kennzeichenstelle D 11 gespeichert, um das Kennzeichen bei sogenannten Überläufen von X_{10} nach X_{11} zu erhalten. Die ZUSE Z 31 arbeitet also in den Rechenregistern mit 12-stelligen Worten (10-stellige Zahlen mit 2 Vorzeichen). Die Stelle X_0 wird bei der Linksverschiebung benötigt.

Betrachten wir zunächst einen einfachen Kreislauf der Information im Register, die sogenannte Regeneration des Registers. In der im Kapitel I gebrauchten symbolischen Schreibweise wäre das:

$\langle x \rangle \longrightarrow x$ (Inhalt von X wird nach X gebracht)

Durch einen „Tetradenschiebeimpuls" (Folge von 4 Impulsen wegen der 4 Flip-Flops) wird die Information um eine Stelle nach rechts verschoben (in Abb. 6 nach unten). Steht der Ausgangsschalter auf „keine Verschiebung" und der Eingangsschalter auf der Ausgangsleitung des X-Registers, so wandert bei jedem Tetradenschiebeimpuls der Inhalt von X_1 nach X_{12}, da der Eingang des X-Registers mit dem Ausgang verbunden ist. Durch die angegebenen 12 Tetradenimpulse wird also die Information einmal „im Kreise" geschoben und steht am Ende genau wie vor dem Kreisen im Register.

Steht jedoch der Ausgangsschalter auf „Rechtsverschiebung", so sind nur noch 11 Dezimalspeicherstellen im Kreis; somit wird durch 12 Tetradenimpulse die Information im Register insgesamt um 1 Dezimalstelle nach rechts verschoben. Um zu verhindern, daß in der 12. Tetradenzeit der Inhalt von X_2 erneut nach X_{12} gelangt, wird der Ausgangsschalter wieder auf „keine Verschiebung" geschaltet, also der Inhalt von X_1 nach X_{12} gebracht. In X1 steht, wie sich leicht überlegen läßt, zu diesem Zeitpunkt genau der Inhalt von X_{12} vor der Verschiebung, also das Kennzeichen des Wortes. Damit ist genau die Rechtsverschiebung erfüllt, der Inhalt von X_1 ist nämlich nach X_0 gewandert, wo er bei einer folgenden Rechnung nicht mehr berücksichtigt wird. Der Inhalt von X_{12} bis X_2 wurde um eine Stelle nach rechts verschoben und die Stelle X_{12} mit der Kopie des Inhaltes von X_{12} vor der Verschiebung aufgefüllt.

Steht der Schalter auf „Linksverschiebung", so sind 13 Dezimalspeicherstellen im Kreis; somit wird durch 12 Tetradenimpulse die Information im Register um insgesamt eine Stelle nach links verschoben. Da die Stelle X_0 immer eine nicht verwertbare Information enthält, (bei der Rechtsverschiebung wurde z.B. die nicht benötigte letzte Stelle des Wortes nach X_0 geschoben), wird bei der Linksverschiebung in der 1. Tetradenzeit der Ausgangsschalter von der Stelle X_0 weggeschaltet und auf den „Nullgenerator" gelegt, wodurch dann nach vollendeter Linksverschiebung wie gewünscht eine Null in X_1 steht.

Soll das X-Register eine Information vom Operationswerk empfangen, so wird der Eingangsschalter auf den Ausgang des Operationswerkes geschaltet. Da der Regenerationsweg dann unterbrochen ist, wird nur die Information vom Operationswerk übernommen. Am Ausgang des X-Registers kann die unverschobene oder die rechts- bzw. linksverschobene Information des X-Registers für das Operations- oder Speicherwerk abgenommen werden. Es ist also z.B. möglich, in einem Befehl den Inhalt des X-Registers zu verschieben und den verschobenen Inhalt als Operanden einer anderen Operation zu verwenden.

Wie schon erwähnt, ist das Y-Register genau wie das X-Register aufgebaut. Durch besondere Befehle ist es möglich, beide Register verkoppelt zu verschieben, was bei der in II 5.2 behandelten Multiplikation verwendet wird.

2.3 Das Leitwerk

Das Leitwerk dient zur Steuerung des Programmablaufes und regelt die richtige Ausführung der Befehle. Es besteht aus folgenden Teilen:

Im Befehlsregister B kann ein Befehl von 11 Dezimalstellen gespeichert werden. Das Wort kann entweder auf parallelem Wege aus dem Programmspeicher oder durch Serienübertragung aus dem Schnellspeicher, Rückkehradressenspeicher bzw. einem anderen adressierten Speicher kommen.
Die Stellen B_{11} ... B_6 des Befehlsregisters enthalten die Dezimalen D 11 - D 6 des Befehlswortes und wirken über Entschlüsseler direkt auf die Steuerschalter der Rechenanlage. Die Stellen B_5 ... B_1 werden auf parallelem Wege erst nach dem Steuerregister R übertragen und dort entschlüsselt. Das Steuer-

register R besitzt ein Zählwerk zur automatischen Adressenerhöhung beim Blocktransfer aus dem Nachschubspeicher.

Das Befehlszählregister C steuert den linearen Programmablauf und enthält zu diesem Zwecke ein Zählwerk zur automatischen Erhöhung der Befehlsadresse. Außerdem wird vom C-Register aus die Auswahl der Programmspeicherworte über entsprechende Entschlüsseler getroffen.

Der Rückkehradressenspeicher RAS dient zur Aufnahme der Adresse des Befehles, bei dem das Programm beim Ruf des Rückkehradressenspeichers fortfahren soll. Außerdem enthält die höchste Stelle von RAS die Angabe, ob sich diese Adresse auf den Programmspeicher oder den Schnellspeicher bezieht.

Die Wirkungsweise des Leitwerkes wird unter II 4 näher beschrieben.

2.4 Das Operationswerk

Das Operationswerk besteht aus zwei Teilen, dem Addierwerk mit logischen Operationswerk und dem ± 1 - Addierwerk. Beide Werke können unabhängig voneinander arbeiten. Das Operationswerk erhält seine Informationen über 3 Eingangsleitungen. Jede Leitung darf nur die Information eines Speichers oder Registers (durch Befehl gesteuert) aufnehmen, da sonst eine unerwünschte Überlagerung der Informationen erfolgt.

Die Eingangsleitung 1 kann Informationen aus dem X-, Y- oder B-Register aufnehmen. Der Weg aus dem B-Register ist vorgesehen, um eine Adressenmodifikation des Befehles zu ermöglichen.

Die Eingangsleitung 2 dient zur Aufnahme von Informationen aus adressierten Speichern oder Registern sowie der Stellen $B_4 \ldots B_1$ des Befehlsregisters (Adressenmodifikation). Alle Register der ZUSE Z 31 sind adressiert, ebenso alle Ein- und Ausgabegeräte sowie die Nachschubspeicher. Lediglich das Befehlsregister ist nicht adressiert, da ein Abholen des Befehles sofort die Steuerung in der Anlage unterbrechen würde.

Die Eingangsleitung 3 dient zur Aufnahme von Informationen aus Speichern und Registern, bei denen eine Zählung möglich ist. (s. Blockschaltbild).

Die Ausgänge der beiden Teile des Operationswerkes können unabhängig voneinander auf Speicher oder Register geschaltet werden. Die zahlreichen Möglichkeiten sind aus dem Blockschaltbild ersichtlich. Es ist zu beachten, daß beide Ausgänge nicht gleichzeitig auf einen Speicher oder ein Register geschaltet werden dürfen, da sonst eine unerwünschte Informationsüberlagerung erfolgt (disjunktive Verknüpfung).

Das Operationswerk verarbeitet die Informationen im Serienbetrieb, also bit nach bit. Jeweils 4 bits einer Tetrade (Ziffer) werden im Operationswerk zwischengespeichert, um den Dezimalübertrag bilden zu können.

In den Abschnitten II 4 und II 5 werden einige Befehle behandelt und in diesem Zusammenhang die Wege genau angegeben, auf denen die Informationen transportiert werden müssen, damit der Befehl entsprechend ausgeführt wird. Zunächst ist jedoch noch einiges zu sagen über den

3. Arbeitstakt der Zentraleinheit ZUSE Z 31

Die ZUSE Z 31 ist eine sogenannte Synchronmaschine. In einer elektronischen Zeitschaltung („Uhr" der Anlage) wird die Zeit in sogenannte Wortzeiten unterteilt.

Eine Wortzeit der ZUSE Z 31 dauert 420 Mikrosekunden.

Die Wortzeit wiederum ist in 16 Tetradenzeiten (T 1 - T 16) unterteilt, jede Tetradenzeit besteht aus 4 Bitzeiten. Damit besteht also eine Wortzeit der ZUSE Z 31 aus 64 Bitzeiten. Eine Bitzeit dauert demnach etwa 6,6 Mikrosekunden.

3.1 Die Schaltzeit

Die Schaltzeit besteht aus den Tetradenzeiten T 1 und T 2. In dieser Zeit können u.a. folgende Vorgänge stattfinden:

Lesen des Programmspeichers und Setzen des Befehls- oder Konstantenregisters,

Entschlüsselung des Befehls im Befehlsregister und Schalten der entsprechenden Informationswege,

Abfragen von im Befehl gegebenen Bedingungen, Kontrollstellen und Schaltern des Bedienungspultes.

Zu einer bestimmten Bitzeit in der Schaltzeit kann die Zentraleinheit gestoppt werden (Stop vor Befehlsausführung).

3.2 Die Operationszeit

Die Tetradenzeiten T 3 - T 16 gehören zur Operationszeit.

In den 13 Tetradenzeiten T 3 ... T 15 werden die vom Befehl angegebenen Operationen durchgeführt. Hierbei werden die einzelnen bits des Wortes nacheinander, also im Serienbetrieb verarbeitet. Es sind deshalb so viele Tetradenzeiten erforderlich, wie das Wort Dezimalen hat. Im Abschnitt II 2 wurde bereits erwähnt, daß in den Rechenregistern mit 12-stelligen Zahlen gerechnet wird. Zur Verarbeitung der 12-stelligen Zahlen sind also 12 Tetradenzeiten erforderlich. Eine weitere Tetradenzeit ist im Operationswerk zur Zwischenspeicherung jeder Dezimalziffer erforderlich, um den Dezimalübertrag bilden zu können.

In der letzten Tetradenzeit T 16 wird erforderlichenfalls der Inhalt des Konstantenregisters in die angewählte Schnellspeicherzelle parallel zurückgeschrieben. Außerdem finden weitere Schaltvorgänge statt. In der Tetradenzeit T 16 kann außerdem die Maschine gestoppt werden, wenn eine der Kontrollstellen Alarm gegeben hat (Stop nach Befehlsausführung).

4. Programmablauf und Steuerung

Wie schon erwähnt, können die Befehle der Programme im Programmspeicher und Schnellspeicher stehen. In diesem Abschnitt sollen nun verschiedene Möglichkeiten des Programmablaufes aus dem Programm- und Schnellspeicher behandelt werden. Alle Befehle der ZUSE Z 31 werden durch ein oder mehrere Symbole gekennzeichnet. Hinzu kommt meistens noch die Adresse, wenn es sich nicht um einen adressenlosen Befehl handelt. Die Adresse ist stets eine 4-stellige Dezimalzahl (in D 4 ... D 1). Auch die Befehlssymbole werden in der Anlage in Form von gültigen Ziffern der Befehlsdezimalen D 5 ... D 11 dargestellt. In der Abhandlung „Interncode der ZUSE Z 31" wird die Wirkung der zugehörigen Befehle eingehend erklärt.

4.1 Programmablauf aus dem Programmspeicher

Der Programmablauf aus dem Programmspeicher wird durch einen Sprungbefehl (Symbol P in der Befehlsdezimale D 5) eingeleitet. Die Adresse des P-Befehles gibt an, in welcher Zelle des Programmspeichers der Befehl steht. Unter der Annahme, daß sich der P-Befehl bereits im Befehlsregister B befindet, geschieht nun folgendes (vgl. Blockschaltbild):

In der Operationszeit wandert der P-Befehl (mit Adresse) in Serie aus den Stellen $B_5 \ldots B_1$ des B-Registers in das C-Register. Dort wird die Adresse entschlüsselt und die entsprechende Programmleitung (s. II 2.1.1) aktiviert.

In der folgenden Schaltzeit wird der Inhalt der gewählten Programmspeicherzelle parallel ins B-Register gebracht. Damit steht also ein neuer Befehl im B-Register. Ist dieser Befehl ein Verarbeitungsbefehl, so wird während der Ausführung dieses Befehles in der Operationszeit die Adresse des im Befehlszählregister stehenden Befehls um 1 erhöht. Das geschieht in einem Kreislauf des C-Registers über das zugehörige +1 - Addierwerk. Nach Ausführung des Verarbeitungsbefehles wird der Befehl aus der nächsten Programmspeicherzelle geholt, gesteuert wiederum durch das C-Register. Weitere

Verarbeitungsbefehle setzen diesen linearen Ablauf aus dem Programmspeicher fort. Erst ein neuer ins B-Register gelangter Sprungbefehl unterbricht den linearen Programmablauf, „springt" nach einer anderen Speicherzelle und setzt gleichzeitig eine neue Adresse in das C-Register, wodurch sich ein neuer linearer Programmablauf mit eben dieser Anfangsadresse anbahnt.

4.2 Programmablauf aus dem Schnellspeicher

Der Programmablauf aus dem Schnellspeicher wird ebenfalls durch einen Sprungbefehl (Symbol E in der Befehlsdezimale D5) eingeleitet. Der E-Befehl mit der zugehörigen Adresse soll sich wiederum im B-Register befinden. Folgende Vorgänge finden statt:

In der Schaltzeit wird der E-Befehl parallel aus den Stellen $B_5 \ldots B_1$ des Befehlsregisters in die Stellen $R_5 \ldots R_1$ des Steuerregisters R übertragen. Durch die Entschlüsseler vom R-Register gesteuert, wird die durch die Adresse des E-Befehls angegebene Schnellspeicherzelle angewählt und der Befehl aus dieser Zelle serienmäßig in der folgenden Operationszeit ins B-Register gebracht. Gleichzeitig wird der E-Befehl aus den Stellen $B_5 \ldots B_1$ des B-Registers in die Stellen $C_5 \ldots C_1$ des C-Registers geschoben. Dadurch wird ein linearer Programmablauf aus dem Schnellspeicher vorbereitet. Ist der durch den E-Befehl gerufene Befehl ein Verarbeitungsbefehl, so beginnt der lineare Programmablauf aus dem Schnellspeicher. Während der Ausführung des Verarbeitungsbefehls wird die Adresse des E-Befehles im C-Register auf die gleiche Weise wie beim linearen Ablauf aus dem Programmspeicher erhöht. In der folgenden Schaltzeit wird der E-Befehl mit der um 1 erhöhten Adresse über $B_5 \ldots B_1$ parallel nach $R_5 \ldots R_1$ gebracht und bewirkt in der folgenden Operationszeit das Holen eines neuen Befehls aus der entsprechenden Schnellspeicherzelle. Beim linearen Programmablauf arbeitet die ZUSE Z 31 also im 2-Taktbetrieb. In der 1. Taktzeit (Wortzeit) wird der Befehl aus dem Schnellspeicher ins B-Register geholt, in der 2. Taktzeit (Wortzeit) wird der geholte Verarbeitungsbefehl ausgeführt und die Adresse des C-Registers um 1 erhöht.

4.3 Programmablauf aus dem Programm- und Schnellspeicher

Ein linearer Programmablauf aus dem Programmspeicher geht dann in einen Programmablauf aus dem Schnellspeicher über, sobald ein E-Befehl aus dem Programmspeicher ins B-Register gelangt. Entsprechend kann ein Übergang vom Schnellspeicherablauf zum Programmspeicherablauf erfolgen. Programme aus dem Programmspeicher laufen doppelt so schnell ab wie gleichartige aus dem Schnellspeicher (wegen des 1-Taktbetriebes des Programmspeichers).

Sehr oft ist es notwendig, beim Sprungbefehl die sogenannte Rückkehradresse zu notieren. Die Rückkehradresse ist die Adresse, bei der das Programm fortgefahren wäre, wenn der Sprungbefehl nicht ausgeführt wäre (z.B. Überlaufen des Befehles wegen nicht erfüllter Bedingung). Gibt man zu einem E- oder P-Befehl noch das Symbol F (in D6 des Befehles), so wird automatisch der Inhalt des C-Registers über das +1-Addierwerk in den Rückkehradressenspeicher RAS gebracht. Damit steht dann in RAS ein E- oder P-Befehl (je nachdem, ob vor dem Sprungbefehl aus dem Schnell- oder Programmspeicher gerechnet wurde). Die Adresse des Befehls wurde bei der Serienübertragung von C nach RAS um 1 erhöht, stellt also die Rückkehradresse dar. Durch einen späteren E-Befehl mit der Adresse von RAS (0005) kommt der Befehl aus RAS als Sprungbefehl ins B-Register und setzt somit das an dieser Stelle unterbrochene Programm fort.

Zu jedem Sprungbefehl können zwei Bedingungen in den Dezimalen D10 und D9 gegeben werden, wodurch fast 100 verschiedene bedingte Sprungbefehle möglich sind, die eine sehr flexible Programmgestaltung erlauben.

5. Einige Verarbeitungsbefehle am Beispiel eines Unterprogrammes

Das Unterprogramm für die Multiplikation zweier 10-stelliger Dezimalzahlen im Festkomma eignet sich besonders zur Demonstration einiger Verarbeitungsbefehle der ZUSE Z 31, da es einmal relativ kurz ist und zum anderen aus fast jeder typischen Befehlsgruppe mindestens einen Befehl enthält. Das Programm wurde aus Gründen der Anschau-

lichkeit noch etwas vereinfacht. Es wurden z.B. der sogenannte „Fünfervorteil", die Abrundung des Ergebnisses usw. weggelassen.

Zunächst ist über die Schreibweise der Befehle einiges zu sagen. Wie schon erwähnt, besteht ein Befehlswort der ZUSE Z 31 aus den 11 Dezimalen D11 ... D1. Im allgemeinen werden nicht alle Dezimalen „besetzt" sein, d.h. eine von Null verschiedene Ziffer enthalten. Null in einer der Dezimalen D10 ... D5 bedeutet keine Operation oder keine Bedingung. Ein besonderes Leseprogramm setzt beim Einlesen des Programmes (z.B. über Lochstreifen) die erforderlichen Nullen in den entsprechenden Dezimalen D10 ... D5 hinzu. Beim Schreiben des Befehls, der dann in gleicher Anordnung in den Lochstreifen gestanzt wird, müssen folgende Voraussetzungen beachtet werden:

Die Bedingungszeichen (D10 und D9) müssen in Klammern gesetzt werden und am Anfang des Befehles stehen.

Die Adresse muß zusammenhängend geschrieben werden.

Handelt es sich um einen sogenannten Normalbefehl (eine 1 in D11), so soll das nicht besonders vermerkt werden.

Im übrigen können die Befehlssymbole bei Kombinationen in beliebiger Reihenfolge stehen.

5.1 Wirkungsweise der Befehle anhand des Blockschaltbildes

Es werden zunächst die Wirkungen der zum Multiplikationsprogramm benötigten Befehlssymbole erklärt. Mehrere Befehlssymbole in einem Befehl bewirken gleichzeitig die angegebenen Funktionen, sofern die Kombination nicht eine ausdrücklich angegebene Sonderbedeutung hat.

Der Additionsbefehl An

Das Symbol A dieses Befehls wird durch eine 1 in D8, die Adresse in den Dezimalen D4 ... D1 des Befehlswortes dargestellt. Er gehört zur Gruppe der arithmetischen Befehle und hat, in symbolischer Schreibweise dargestellt, folgende Wirkung:

$$\langle x \rangle + \langle n \rangle \rightarrow x$$

n kann hierbei eine beliebige Adresse sein. Unter der Annahme, daß es sich um die Schnellspeicherzelle 1000 handelt, finden in der ZUSE Z 31 folgende Vorgänge statt (s. Blockschaltbild):

Der Inhalt des Rechenregisters X wandert über die dick ausgezogene Leitung zum Eingang 1 des Addierwerkes. Der Inhalt der Schnellspeicherzelle 1000 wandert über die entsprechende Leitung zum Eingang 2. Wie immer in der Anlage werden bei Serienübertragung die Wörter beginnend mit der niedrigsten Stelle (D1) transportiert. (Man addiert ja auch zwei Zahlen „von hinten nach vorn", um die Überträge berücksichtigen zu können).

Im Operationswerk werden die beiden Zahlen aus X und 1000 stellenrichtig addiert, das Ergebnis wird in Serie zurück nach X gebracht (über die dick ausgezogene Ausgangsleitung des Addierwerkes bis zum Eingang des Rechenregisters X). Damit ist der erste Operand (Inhalt des X-Registers vor der Addition) verlorengegangen. Der zweite Operand (Inhalt von 1000) wird aber gleichzeitig, wie man sagt, „regeneriert" entsprechend der symbolischen Schreibweise:

$$\langle n \rangle \rightarrow n$$

Das geschieht folgendermaßen:
Der Ausgang des Schnellspeichers, der bei dieser Operation auf die zum Eingang 2 des Operationswerkes gehörigen Leitung geschaltet wurde, wird „angezapft", so daß der Inhalt der Schnellspeicherzelle 1000 außerdem noch in das Konstantenregister gelangt. Diese Verbindung ist im Blockschaltbild nicht eingezeichnet. In der Tetradenzeit T16 wird dann der Inhalt des Konstantenregisters parallel in die Schnellspeicherzelle 1000 zurückgeschrieben. Im folgenden werden die Operationen nicht mehr so ausführlich geschildert. Die zu den Eingängen 1 - 3 des Operationswerkes gehörigen Leitungen werden mit Eingangsleitung 1 - 3 bezeichnet. Außerdem wird noch besonders erwähnt, daß der Inhalt eines bei Operation benutzten aber nicht veränderten Registers oder Speichers immer regeneriert wird.

Der Bringbefehl Bn

Symbol B, dargestellt durch eine 8 in D8, Adresse in D4 ... D1. Er gehört zur Gruppe der Transportbefehle. Symbolische Schreibweise:

$$\langle n \rangle \longrightarrow X$$

Der Inhalt der gewählten Adresse (diese kann eine Schnellspeicherzelle oder ein adressiertes Register sein) wandert über die Eingangsleitung 2 durch das Addierwerk und gelangt in das Rechenregister X. Am Eingang 1 des Addierwerkes wird „Nullgenerator" (nicht eingezeichnet) angelegt, so daß die Information über den Eingang 2 das Operationswerk unverändert passiert.

Der Transferbefehl Tn

Symbol T, dargestellt durch eine 9 in D8, Adresse in D4 ... D1. Er gehört ebenfalls zur Gruppe der Transportbefehle. Symbolische Schreibweise:

$$\langle X \rangle \longrightarrow n$$

Der Inhalt von X gelangt über die Eingangsleitung 1 wiederum unverändert durch das Operationswerk entweder direkt in das durch n adressierte Register oder in das Konstantenregister, wenn n eine Schnellspeicherzelle ist.
In der Tetradenzeit T16 wird wiederum der Inhalt von K parallel in die angewählte Schnellspeicherzelle n gesetzt. Durch Zusätze kann die Funktion des Transferbefehles abgewandelt werden.

Der Zusatz Y (9 in D5) bewirkt, daß nicht der Inhalt des X-Registers sondern der Inhalt des Y-Registers nach n gebracht wird.

Der Zusatz M (9 in D7) bewirkt, daß der negative Inhalt des X- oder Y-Registers nach n gebracht wird. Für das Multiplikationsprogramm werden folgende Befehle gebraucht:

TMn	Funktion:	$-\langle X \rangle \longrightarrow n$
TMYn	Funktion:	$-\langle Y \rangle \longrightarrow n$

Der Rechtsverschiebungsbefehl R

Symbol R, dargestellt durch eine 1 in D7, bewirkt eine Rechtsverschiebung des X-Registers um eine Stelle (Multiplikation mit 0,1):

$$\langle X \rangle \cdot 10^{-1} \longrightarrow X$$

Die Information des X-Registers läuft hierbei über das Verschiebewerk, die Eingangsleitung 1 und das Addierwerk zurück ins X-Register.

Der verkoppelte Rechtsverschiebungsbefehl RR

Symbol RR, dargestellt durch eine 9 in D7, bewirkt eine Rechtsverschiebung des X- und Y-Registers, wobei gleichzeitig die Ziffer in der Stelle X_1 des X-Registers in die Stelle Y_{11} (2. Vorzeichenstelle) des Y-Registers geschoben wird:

$$\langle X \rangle \cdot 10^{-1} \longrightarrow X \qquad \langle X \rangle_1 \longrightarrow Y_{11}$$
$$\langle Y \rangle \cdot 10^{-1} \longrightarrow Y$$

Die Verschiebung des X-Registers erfolgt wie bei R; der Inhalt des Y-Registers wird über das zugehörige Verschiebewerk, die Eingangsleitung 3 und das ± 1-Addierwerk zurück ins Y-Register geschoben; der Weg für $\langle X \rangle_1 \longrightarrow Y_{11}$ ist nicht eingezeichnet.

Der Sonderbefehl TQYn

Die Symbole T, Y und Q (8 in D6) zusammen in einem Befehl ergeben eine Sonderbedeutung, die nicht mit den 3 Einzelfunktionen der Symbole übereinstimmt.
Der Befehl TQY bewirkt, daß der Inhalt des Y-Registers mit dem Vorzeichen des X-Registers nach der angegebenen Adresse n gebracht wird, also

$$\langle Y \rangle_{10\text{-}1} \longrightarrow n_{10\text{-}1}$$
$$\langle X \rangle_{12} \longrightarrow n_{11}$$

Die Information nimmt den gleichen Weg wie bei Tn, jedoch mit der Ausnahme, daß zur Tetradenzeit T 13 die Information nicht aus dem Y- sondern aus dem X-Register der Sammelleitung 1 zugeführt wird (Zuführung des Vorzeichens vom X-Register).

Die Stellbefehle I1 und Ji

Symbol I1, dargestellt durch eine 5 in D5.
Funktion: $n_1 \longrightarrow ZR1$

Die letzte Stelle der Adresse n wird über einen nicht eingezeichneten Weg in das Zählregister ZR1 gebracht.

Symbol Ji, dargestellt durch eine 3 in D6.
Funktion: $L \longrightarrow Bi$

Es wird der durch eine der Ziffern i = 1 ... 6 in D10 gekennzeichnete Bedingungsspeicher in L gesetzt.
Die 6 Bedingungsspeicher sind im Blockschaltbild nicht eingezeichnet.

Die Zählbefehle V1 und VY

Symbol V1, dargestellt durch eine 1 in D6.
Funktion: $\langle ZR1 \rangle + 1' \longrightarrow ZR1$

Die Bezeichnung +1' besagt, daß in der letzten Stelle eines Registers oder Speichers eine 1 addiert wird. Die +1-Addition erfolgt beim Kreislauf des ZR1-Registers über das zugeordnete ±1-Addierwerk, das bei den V1 auf „+" geschaltet ist.

Symbol VY, dargestellt durch eine 7 in D5.
Funktion: $\langle Y \rangle + 1' \longrightarrow Y$

Die Information nimmt vom Ausgang des Y-Registers ihren Weg über die Eingangsleitung 3, über das auf „+1" geschaltete ±1-Addierwerk und kommt zurück zum Eingang des Y-Registers.

Die Befehle V1 und VY bewirken also in einer Wortzeit ein einmaliges Addieren einer 1 im ZR1- bzw. Y-Register (Vorzählen).

Der Wiederholungsbefehl W

Symbol W, dargestellt durch eine 3 in D11 (eines der 6 möglichen Befehlskennzeichen). Alle bisher behandelten Befehle hatten in D11 eine 1 (beim Einlesen der Befehle automatisch durch das Befehlsleseprogramm gesetzt).

Der mit W verbundene Befehl wird solange ausgeführt, bis seine hier notwendige Bedingung nicht mehr erfüllt ist. Demnach muß der Befehl auch das Register oder den Speicher verändern, das bzw. der die Teststelle für die gegebene Bedingung enthält. Ist dann die Bedingung nicht mehr erfüllt, wird der nächste Befehl des Programmes aufgerufen. (Der nächste P- oder E-Befehl steht im C-Register). Da der Wiederholungsbefehl immer mit einer Bedingung gegeben wird, muß das Symbol W beim Schreiben des Befehles mit in die Klammer für die Bedingungssymbole gesetzt werden.

Die Bedingungen NE, PO, 1Z, YZ , Ji.

Jeder bedingte Befehl wird nur ausgeführt, wenn die gegebene Bedingung erfüllt ist. Für das Multiplikationsprogramm werden 5 der möglichen ca. 100 Bedingungen benötigt.

Symbol	Stellung im Wort	Befehlsausführung, wenn
NE	1 in D9	$\langle X \rangle < 0$
PO	2 in D9	$\langle X \rangle > 0$
1Z	8 in D10	$\langle ZR1 \rangle \neq 0$
YZ	7 in D10	$\langle Y \rangle_1 \neq 0$
Ji	i in D10	Bi in L
	i = 1, 2 oder 3	

Damit wurden alle zum Multiplikationsprogramm im Festkomma benötigten Befehle bzw. Bedingungen behandelt. Es handelt sich nur um einen kleinen Teil der möglichen Befehle. Die Symbole sind in nach-

stehender Tabelle zusammengefaßt; alle freien Stellen der Tabelle enthalten weitere Symbole, die im „Interncode der ZUSE Z 31" ausführlich gebracht werden.

	K.-Z.	Bedingungen		Operation			
	D11	D10	D9	D8	D7	D6	D5
0	pos. Zahl						
1	Normal-befehl	J1	NE	A	R	V1	E
2		J2	PO				P
3	W	J3			RR	Ji	
4	Text						
5	Text						I1
6							
7		YZ					VY
8		1Z		B		Q	
9	negative Zahl			T	M	F	Y

5.2 Das Multiplikationsprogramm

Das Strukturdiagramm der Multiplikation im Festkomma zeigt **Abb.**7 am Ende dieses Kapitels. Der Multiplikand (Md) muß in der Schnellspeicherzelle 1000, der Multiplikator (Mr) im X-Register stehen. Am Ende steht dann das 20-stellige Produkt im X- und Y-Register (obere Hälfte in X, untere in Y) und kann weiter im Programm verarbeitet werden.

Das Multiplikationsprogramm befindet sich in den Programmspeicherzellen 2000 - 2010 und wird vom Hauptprogramm durch den Befehl PF 2000 gestartet. Durch das F wird die Folgeadresse des Hauptprogramms in RAS notiert. Der Befehl E0005 am Ende des Multiplikationsprogrammes holt den Inhalt von RAS zurück ins B-Register und sorgt somit für die Fortführung des Hauptprogrammes.

Die einzelnen Entscheidungs- oder Funktionsteile des Struktur-Diagramms wurden zur besseren Erklärung numeriert.

In 1) wird entschieden, ob der Multiplikator Mr positiv oder negativ ist. (Md sei positiv)

Mr negativ: Es wird der Betrag von Mr im Y-Register gebildet (s.2.) und zur Kennzeichnung für später B1 in L gesetzt. Anschliessend Fortsetzung bei 3)

Mr positiv: Sofort Fortsetzung bei 3)

In 3) wird das X-Register gelöscht (dort sollen die Partialprodukte summiert werden) und eine 1 nach ZR1 gebracht.

In 4) wird getestet, ob die letzte Stelle des Multiplikators (in Y_1) Null ist.

$\langle Y \rangle_1 \neq 0$: 5) Wiederholte Addition des Multiplikanden zum Inhalt des X-Registers. Gleichzeitig erfolgt bei jedem durchgeführten Additionsschritt eine Verminderung der letzten Stelle von Y um 1. Die Schleife 4) und 5) wird also sooft durchlaufen, wie die letzte Multiplikatorziffer in Y_1 angibt. Erst wenn $\langle Y \rangle_1 = 0$, erfolgt Anschluß an 6).

$\langle Y \rangle_1 = 0$: Sofort Anschluß an 6).

6) Hier wird entschieden, ob noch eine Multiplikatorziffer behandelt werden soll. ZR1 wurde bei 3) auf 1 voreingestellt, in 7) wird bei jedem Durchlauf 1 dazugezählt, also wird 6) neunmal durchlaufen. Es sind also 9 Durchläufe von 4) nach 6) und einer gleich am Anfang unmittelbar nach 3) mög-

lich, was genau der Behandlung des 10-stelligen Multiplikators entspricht.

⟨ZR 1⟩ ≠ 0: Anschluß an 7), damit verkoppelte Rechtsverschiebung von X und Y sowie ⟨ZR 1⟩ + 1' → ZR 1.

⟨ZR 1⟩ = 0: Anschluß an 8), damit 10. verkoppelte Rechtsverschiebung, da 7) schon neunmal durchlaufen wurde.

In 9) wird anschließend entschieden, ob der Multiplikator negativ war.

B 1 in L (Mr negativ): Das Vorzeichen des gesamten Produktes wird umgekehrt, danach Anschluß an 11).

B 1 ≠ L: Sofort Anschluß an 11).

11) Abschließende Rechtsverschiebung von Y, um die untere Häfte des Produktes auf die richtigen Stellen des Y-Registers zu bringen. Gleichzeitig Setzen des Vorzeichens von X in die Vorzeichenstellen von Y, da das Vorzeichen von X das wirkliche Vorzeichen des gesamten Produktes darstellt.

12) Durch den Befehl E 0005 wird der Rückkehradressenspeicher aufgerufen, in dem sich die Folgeadresse des Hauptprogrammes befindet.

Es folgen nun die 10 Befehle des Multiplikationsprogrammes:

Nr. der Programmspeicherzelle	Befehl	Nr. im Strukturdiagramm	Anzahl der Befehlsausführungen
2000	(NE) J 1 TM 0006	1) 2)	1
2001	(PO) T 0006	1) 2)	1
2002	B 0000	3)	1
2003	I 1 0001	3)	1
2004	(WYZ) A ZY 1000	4) 5)	55
2005	(1 Z) RR V JP 2004	6) 7)	9
2006	RR	8)	1
2007	(J 1) TM 0004	9) 10)	1
2008	(J 1) TMY 0006	9) 10)	1
2009	TQY R 0006	11)	1
2010	E 0005	12)	1

Verwendete Registeradressen:

0004	=	X-Register
0005	=	Rückkehradressenspeicher RAS
0006	=	Y-Register

Man erkennt schon an diesem kurzen Programm die Kombinationsfähigkeit der Symbole der ZUSE Z 31-Befehlscode, wodurch sowohl Befehle als auch Zeit gespart werden, da mehrere Operationen in einem Befehl gegeben und in einer Wortzeit ausgeführt werden können. Unter der Voraussetzung, daß im 10-stelligen Multiplikator jede der Ziffern 0 ... 9 einmal vorhanden ist, ergibt sich dann die Anzahl der Befehlsausführungen. (Hierin ist auch jedes Überlaufen des Befehls enthalten).

Es ergeben sich somit 73 Wortzeiten zu je 420 Mikrosekunden. Demnach dauert dieses Multiplikationsprogramm im Mittel 31 Millisekunden. Bei der ZUSE Z 31 wird, wie schon erwähnt, ein abgewandeltes, eleganteres Multiplikationsprogramm verwendet, das im Mittel nur etwa 28 Millisekunden dauert.

6. Anschluß der Ein- und Ausgabegeräte sowie der Nachschubspeicher an die Zentraleinheit.

Der Informationsaustausch zwischen der Zentraleinheit und allen Anschlußgeräten findet über sogenannte Sonderadressen statt. Der Adressenteil eines Befehlswortes der ZUSE Z 31 besteht, wie schon erwähnt, aus 4 Dezimalstellen, erlaubt also eine Darstellung von 10 000 verschiedenen Adressen. Steht im Befehl das Symbol P,

so bezieht sich die Adressenangabe auf den Programmspeicher, im anderen Falle auf adressierte Register, Schnellspeicherzellen oder Sonderadressen, und zwar nach folgender Aufteilung:

0000 ... 0010	Interne Sonderadressen (z.B. Register),
0011 ... 0999	Externe Sonderadressen (Anschlußgeräte),
1000 ... 9999	Schnellspeicheradressen.

Damit stehen für die Anschlußgeräte ca. 1000 Sonderadressen zur Verfügung.

Der Anschluß aller Geräte erfolgt nach einem einheitlichen System, das im folgenden behandelt wird.

Jedes Anschlußgerät hat einen Ein- oder Mehrzeichenpuffer (s.a. I 2.4 und I 2.5). Durch die schon erläuterten Freigabezeichen wird der Informationsaustausch zwischen der Zentraleinheit und den Anschlußgeräten gesteuert.

Ein Informationsaustausch zwischen der Zentraleinheit und einem Anschlußgerät kann nur dann stattfinden, wenn die der entsprechenden Sonderadresse zugeordnete Freigabeleitung ein Freigabezeichen an die Zentraleinheit gibt.

Bei nicht vorhandenem Freigabezeichen wartet entweder die Zentraleinheit auf das Anschlußgerät oder sie rechnet weiter im Hauptprogramm und fragt nach einer bestimmten Zeit erneut das Anschlußgerät ab (je nach Abfragebefehl).

Die Sonderadressen bestimmen die Art der Informationsübertragung zwischen der Zentraleinheit und den Anschlußgeräten. Hat z.B. ein Anschlußgerät nur einen Einzeichenpuffer, so wird bei der Anwahl der entsprechenden Sonderadresse auch nur ein Zeichen übertragen. Entsprechendes gilt für die Übertragung von Worten und Wortblöcken. Für gleichartige Informationsübertragungen gibt es dann nur ein Lese- bzw. Ausgabeprogramm. Diese Programme stehen im Programmspeicher und können durch entsprechende Sprungbefehle gestartet werden. Durch sogenannte Adressenmodifikationsbefehle ist es dann möglich, eines der Anschlußgeräte auszuwählen. Anhand eines Beispiels soll das näher erläutert werden.

Es seien zwei Lochstreifenleser an die ZUSE Z 31 angeschlossen. Der 1.Leser ist durch die Sonderadresse 0011, der 2.Leser durch 0013 erreichbar. Sie besitzen beide einen Einzeichenpuffer. Das Leseprogramm, das hier nur aus wenigen Befehlen besteht, liest also die Informationen gleichartig ein. Nur ein Befehl des Leseprogrammes enthält die Adresse des Lesers. Es könnte z.B. der Bringbefehl Bn sein. Sobald n den Wert 0011 oder 0013 hat, löst der B-Befehl folgende Vorgänge in der Maschine aus:

$$\langle n \rangle \longrightarrow X_{1-2}$$

$$0 \longrightarrow X_{3-12}$$

Der Inhalt des Einzeichenpuffers wird in die letzten beiden Dezimalstellen des X-Registers übertragen. Die Maschine „weiß" also, daß sie nur 2 Dezimalen (entsprechend der 8 bits des Lochstreifenzeichens) übernehmen und die restlichen Dezimalen des Registers mit Nullen auffüllen muß. Die Adresse 0011 oder 0013 hatte ihr auch angegeben, daß im Einzeichenpuffer gültige Ziffern waren; also braucht die Gültigkeitskontrolle nicht abgeschaltet zu werden. Bei den Adressen 0012 und 0014 jedoch hätte die Zentraleinheit zwar auch die 8 bits in die Stellen X1-2 übernommen, aber sogenannte binäre Nullen nach X_{3-10} und das Textkennzeichen 4 nach X_{11-12} gebracht, um die Gültigkeitskontrolle abzuschalten.

Die Sonderadressen bestimmen also nicht nur die Wahl des Anschlußgerätes, sondern auch die Art der Informationsübertragung.

Im Beispiel ist also ein Leseprogramm für 4 Sonderadressen vorhanden. Wie wird nun die entsprechende Sonderadresse in das Leseprogramm eingesetzt?

Kehren wir zum B-Befehl zurück. Um das Einsetzen der Adresse zu ermöglichen, wird der sogenannte Adressensubstitutionsbefehl (Symbol G, dargestellt durch eine 2 in D11) verwendet. Damit wird der B-Befehl zum GB-Befehl. Als Adresse des Befehles steht im Leseprogramm eine bestimmte Schnellspeicheradresse, in die der Programmierer die jeweils gewünschte Sonderadresse vor Anwahl des Leseprogrammes bringt. Die Adresse des GB-Befehles im Leseprogramm sei z.B. 1500. Der Befehl GB 1500 bewirkt dann folgendes:

1. Wortzeit: $\langle 1500_{4-1} \rangle \longrightarrow B_{4-1}$ Löschung von G

In der Schnellspeicherzelle 1500 soll z.B. die vorher eingespeicherte Adresse 0011 stehen. In der 1.Wortzeit

wurde zunächst in die Adressenstellen B_{4-1} des Befehlsregisters die Adresse 0011 gebracht. Im Befehlsregister steht also dann der gewünschte Befehl

B 0011,

der in der 2. Wortzeit ausgeführt wird.

Alle anderen Anschlußgeräte sind entsprechend angeschlossen (Lochstreifenstanzer, Lochkartenleser- und Stanzer, Schreibmaschinen, Drucker usw.). Bei der geschilderten Organisation fand ein Informationsaustausch zwischen einem Register der Zentraleinheit und dem Puffer des entsprechenden Anschlußgerätes statt. Es ist jedoch auch ein sogenannter Blocktransfer direkt zwischen dem Schnellspeicher der Zentraleinheit und dem Puffer möglich. Durch 2 oder 3 Befehle kann erreicht werden, daß Blöcke bis zu mehreren hundert Worten direkt vom Anschlußgerät in den Schnellspeicher oder umgekehrt transportiert werden. Diese Organisation wird beim Anschluß der Nachschubspeicher (Magnettrommel, Magnetband) und bei den Lochkarteneinheiten verwendet. Als Beispiel sei das Magnetbandgerät angeführt.

Auf dem Magnetband sollen sich die Informationen in Blöcken variabler Länge befinden. Jeder Block hat eine Adresse (Blockadresse). Eine entsprechende Anschlußelektronik des Magnetbandes ist in der Lage, bei Angabe des Blockes die Stelle des Bandes zu finden.

Durch den Vorbefehl (Symbol VB, dargestellt durch eine 6 in D11) wird mit dessen Adresse n die gewünschte Blockadresse angegeben. In einem 2. Befehl (entweder WB oder WT, je nachdem, ob das Magnetband beschrieben oder gelesen werden soll) wird in den Dezimalen D1 und D2 die Adresse des Magnetbandgerätes gegeben, in den Dezimalen D3 und D4 die Anfangsadresse des Hunderter-Blocks vom Schnellspeicher, der die Informationen abgeben oder aufnehmen soll. Damit startet das Magnetband und sucht die Blockadresse. Während dieser Zeit wartet die ZUSE Z 31, da kein Freigabezeichen vorhanden ist. Hat die Elektronik den Block gefunden, wird z.B. beim Lesen der gesamte Block über zwei Zwischenpuffer (Kapazität je ein Wort) vom Magnetband in den Schnellspeicher der Zentraleinheit übertragen. Soll nicht der gesamte Block übertragen werden, kann das durch einen weiteren Befehl vor dem Vorbefehl angegeben werden. Das Warten der Zentraleinheit beim Suchen der Blockadresse kann vermieden werden, wenn nach dem Vorbefehl nicht sofort der Transportbefehl sondern ein spezieller Suchbefehl gegeben wird.

Bei mehreren Magnetbandgeräten ist es möglich, allen Geräten Suchbefehle zu geben. Die Zentraleinheit rechnet während des Suchens der Magnetband-Blockadresse im Hauptprogramm weiter. Im Hauptprogramm sind Abfragezyklen eingebaut, in denen alle Geräte von Zeit zu Zeit abgefragt werden. Bei vorhandenem Freigabezeichen findet dann ein Blocktransfer statt. Anschließend kann ein neuer Suchbefehl gegeben werden. Während des erneuten Suchens der Blockadresse ist die Zentraleinheit wiederum in der Lage, mit den übernommenen Werten weiterzurechnen.

Diese Organisation in Verbindung mit der unter I 2.7 geschilderten Vorrangsteuerung der Anschlußgeräte ergibt bei entsprechender Programmierung eine sehr hohe effektive Datenverarbeitungsgeschwindigkeit der gesamten Anlage.

Eine ausführliche Behandlung der einzelnen Anschlußgeräte der ZUSE Z 31 würde über den Rahmen dieser Abhandlung hinausgehen. Es wird auf die Beschreibungen und Programmierungsanleitungen der entsprechenden Geräte verwiesen.

7. Steuerung der Zentraleinheit vom Bedienungspult

Durch die Tasten und Schalter des Bedienungspultes kann das Programm der ZUSE Z 31 beeinflußt werden. Eine große Anzahl von Leuchtanzeigen gibt Auskunft über den Stand von Speichern in der Anlage sowie über den Zustand der Kontrollstellen. Über eine Zehnertastatur können Informationen direkt in die Maschine eingegeben werden.

Der Aufbau des Bedienungspultes geht aus Abb. 8 hervor. (Am Ende von Kapitel II).

Es folgt nun eine Beschreibung der Tastenfunktionen sowie der Leuchtanzeigen.

Die Starttaste dient zum Starten eines Programmes.
Leuchtet die Anzeige „Stop vor Befehlsausführung", so beginnt das Programm mit dem Befehl, der sich gerade im Befehlsregister befindet. Der Inhalt des Befehlsregisters wird durch eine 11-stellige Ziffernanzeigevorrichtung (Befehlsregisteranzeige) angezeigt.

Leuchtet die Anzeige „Stop nach Befehlsausführung", so beginnt das Programm nicht mit dem angezeigten, sondern mit dem nächsten Befehl des Programmes, dessen Adresse im Befehlszählregister steht. Zur Kontrolle kann diese Adresse angezeigt werden.

Unter gewissen Alarm- oder Blockierungsbedingungen bleibt ein Drücken der Starttaste unwirksam (s.u.). Nach einer erfolgreichen Betätigung der Starttaste verlöscht die entsprechende Stopanzeige, es wird dann die Anzeige „Programmablauf" gebracht.

Die Weitertaste unterscheidet sich von der Starttaste dadurch, daß die Rechenanlage nur für eine Wortzeit gestartet wird, dann aber stoppt. Es ist also ein schrittweises Arbeiten möglich, das bei Programm- und Maschinenprüfungen sehr vorteilhaft ist.

Die Stoptaste beendet den Programmablauf und bringt die Anzeige „Stop vor Befehlsauführung".

Der Adressenwahlschalter hat drei Funktionen.
Ist die rastende Taste Adressenstop gedrückt, so stoppt die Anlage, sobald ein Befehl ins Befehlsregister gelangt, dessen Adresse gleich der auf dem Adressenwahlschalter eingestellten Adresse ist. Es leuchtet dann sowohl die Anzeige „Adressenstop" als auch „Stop vor Befehlsausführung".
Die zweite Funktion des Adressenwahlschalters erfolgt in Verbindung mit einer der beiden Programmtasten.
Wird die Taste P-Programm gedrückt, so wird ein P-Befehl mit der auf dem Adressenwahlschalter eingestellten Adresse ins Befehlsregister gebracht.
Die Taste E-Programm bringt entsprechend einen E-Befehl ins Befehlsregister. Die beiden Tasten sind nur bei gestoppter Anlage wirksam. Nach erfolgreicher Betätigung einer der Tasten steht die Anlage unbedingt auf „Stop vor Befehlsausführung", so daß durch ein folgendes Drücken der Starttaste der E- oder P-Befehl ausgeführt werden kann.
Die dritte Funktion des Adressenwahlschalters erfolgt in Verbindung mit der Übernahmetaste auf der Zehnertastatur.
Die erste Betätigung der Übernahmetaste bewirkt, daß der Inhalt des Befehlsregisters in eine auf dem Adressenwahlschalter eingestellten Schnellspeicherzelle gelangt. Gleichzeitig gelangt der Inhalt der Schnellspeicherzelle ins Befehlsregister und wird angezeigt. Ist auf dem Adressenwahlschalter eine Registeradresse eingestellt, so wird der Befehlsregisterinhalt im Konstantenregister aufbewahrt; der Inhalt des angewählten Registers bleibt unverändert und gelangt außerdem zur Anzeige ins Befehlsregister.

Die zweite Betätigung der Übernahmetaste bringt den ursprünglichen Inhalt des Befehlsregisters zurück ins Befehlsregister, sofern die Stellung des Adressenwahlschalters nicht verändert wurde. Gleichzeitig gelangt der Befehlsregisterinhalt (z.B. der Inhalt der vorher aufgerufenen Schnellspeicherzelle) zurück in die auf dem Adressenwahlschalter eingestellten Schnellspeicheradresse. Wird jedoch zwischen der ersten und zweiten Betätigung der Übernahmetaste eine der Tasten Start, Stop, Weiter, P-Programm, E-Programm, Zehnertastatur oder Irrung gedrückt, so bewirkt ein zweites Betätigen der Übernahmetaste die oben geschilderten Funktionen der ersten Betätigung.

Mit Hilfe der Übernahmetaste ist es also u.a. möglich, den Inhalt jedes Registers oder jeder Schnellspeicherzelle im Befehlsregister anzuzeigen sowie den Inhalt des Befehlsregisters in jede beliebige Schnellspeicherzelle zu bringen.

Mit Hilfe der Zehnertastatur können Ziffern direkt ins Befehlsregister eingegeben werden. Es ist also möglich, durch anschliessende Betätigung der Übernahmetaste eine Zahl in jede gewünschte Schnellspeicherzelle zu bringen.

Beim Eintasten von Ziffern ins Befehlsregister geschieht folgendes:
Das Drücken einer Ziffer bewirkt, daß der Inhalt des Befehlsregisters um eine Stelle nach links verschoben wird. In der freigewordenen unteren Stelle erscheint dann die gedrückte Ziffer. Somit kann das Befehlsregister ziffernweise aufgefüllt werden; dabei ist durch die Befehlsregisteranzeige sofort eine optische Kontrolle möglich.

Die Betätigung der Irrungstaste verschiebt den Inhalt des Befehlsregisters um eine Stelle nach rechts, schiebt also die zuletzt eingetastete Ziffer wieder hinaus.

Die Taste „B-Löschen" setzt das gesamte Befehlsregister auf Null.

Die 6 Bedingungsspeicher können durch 12 entsprechende Tasten gesetzt oder gelöscht werden. Befindet sich ein Bedingungsspeicher in L, so leuchtet die zugehörige Anzeige auf.

Durch das Drücken einer der 6 rastenden Blockierungstasten kann eine Änderung des zugehörigen Bedingungsspeichers verhindert werden. Kommt von der Anlage im Programm ein Änderungsbefehl an einen blockierten Bedingungsspeicher, so stoppt die Anlage bei gleichzeitiger Anzeige „Bedingungsspeicher blockiert" und „Stop vor Befehlsausführung". Ein erneutes Starten ist erst nach Lösen der entsprechenden Blockierungstaste möglich.

Die Alarmanzeige besteht aus 18 Einzelanzeigen. Dadurch ist es möglich, einen auftretenden Maschinenfehler sofort zu lokalisieren und ihn somit leichter zu finden. Insgesamt 13 Anzeigen beziehen sich auf die erwähnten Gültigkeitskontrollen, eine Anzeige auf die Längssummenkontrolle modulo 4 im Schnellspeicher, 2 Anzeigen auf Quer- und Längssummenkontrollen und 2 Anzeigen auf die Überwachung der Stromversorgung. Die Bezeichnung der Alarme ist eindeutig, so daß die Teststellen sofort im Blockschaltbild gefunden werden können. Die Alarme der Ein- und Ausgabegeräte sowie der Nachschubspeicher sind zu je einer Anzeige zusammengefaßt, werden aber einzeln nochmals am entsprechenden Anschlußgerät angezeigt.
Zu jeder Alarmanzeige auf dem Bedienungspult gehört eine rastende Alarmfreigabetaste. Ist sie nicht gedrückt, stoppt die Anlage je nach Alarmfall vor oder nach der Befehlsausführung. Ein erneutes Starten ist erst möglich, wenn die entsprechende Freigabetaste gedrückt wird. Das Leuchten der Alarmanzeige bleibt aber dann bestehen. Ist eine Alarmfreigabetaste gedrückt, so stoppt die Anlage nicht bei dem entsprechenden Alarm; es leuchtet lediglich die Alarmanzeige.

Durch die Alarmlöschtaste werden alle Alarmanzeigen gelöscht, unabhängig davon, ob die Freigabetaste gedrückt ist. Nach der Löschung ist sofort ein erneuter Start der Anlage möglich, falls sie sich durch einen Alarm gestoppt hatte.

Die Tasten Maschine Ein und Maschine Aus dienen zur Ein- bzw. Ausschaltung der Stromversorgung der Anlage.

Ist die rastende Taste Nacht gedrückt, so wird die Stromversorgung der Anlage automatisch unterbrochen, falls der Programmablauf durch irgendeinen Stop unterbrochen wird. Damit ist es möglich, längere Programme ohne Aufsicht rechnen zu lassen und nach Beendigung des Programmes die Anlage automatisch abzuschalten. Auch bei Störungen, die die Anlage gefährden könnten, wird diese vom Netz abgeschaltet.

8. Technischer Aufbau der ZUSE Z 31.

Die Zentraleinheit besteht aus dem sogenannten Rechenschrank und dem Bedienungstisch mit dem zugehörigen Bedienungspult.

Der Rechenschrank enthält die logischen Schaltkreise, Register und Speicher der Rechenanlage. Außerdem sind in diesem die elektronisch stabilisierten Netzgeräte zur Erzeugung der benötigten Spannungen untergebracht. Die Schaltkreise und Speicher sind unter Verwendung von Transistoren, Dioden, Ferritkernen in Verbindung mit Widerständen, Kondensatoren und Spulen aufgebaut. Die einzelnen Schaltelemente sind auf sogenannten gedruckten Platten untergebracht. Die Platten haben etwa die Größe 180 x 86 mm bzw. 180 x 330 mm und werden in entsprechende Fassungen gesteckt, die auf 4 Schwenkrahmen angebracht sind.

Der Bedienungstisch enthält die Anschlußschaltungen für das Stromnetz (Sicherungen, Schaltschütze usw.).

Das Bedienungspult enthält die Steuerschaltungen für die geschilderten Tasten und Leuchtanzeigen.

Die Anschlußelektronik jedes Anschlußgerätes (Ein- und Ausgabegeräte, Nachschubspeicher) ist in einem besonderen Schrank bzw. Tisch untergebracht. Teilweise dient ein Schrank oder Tisch zum Anschluß mehrerer gleichartiger Geräte.

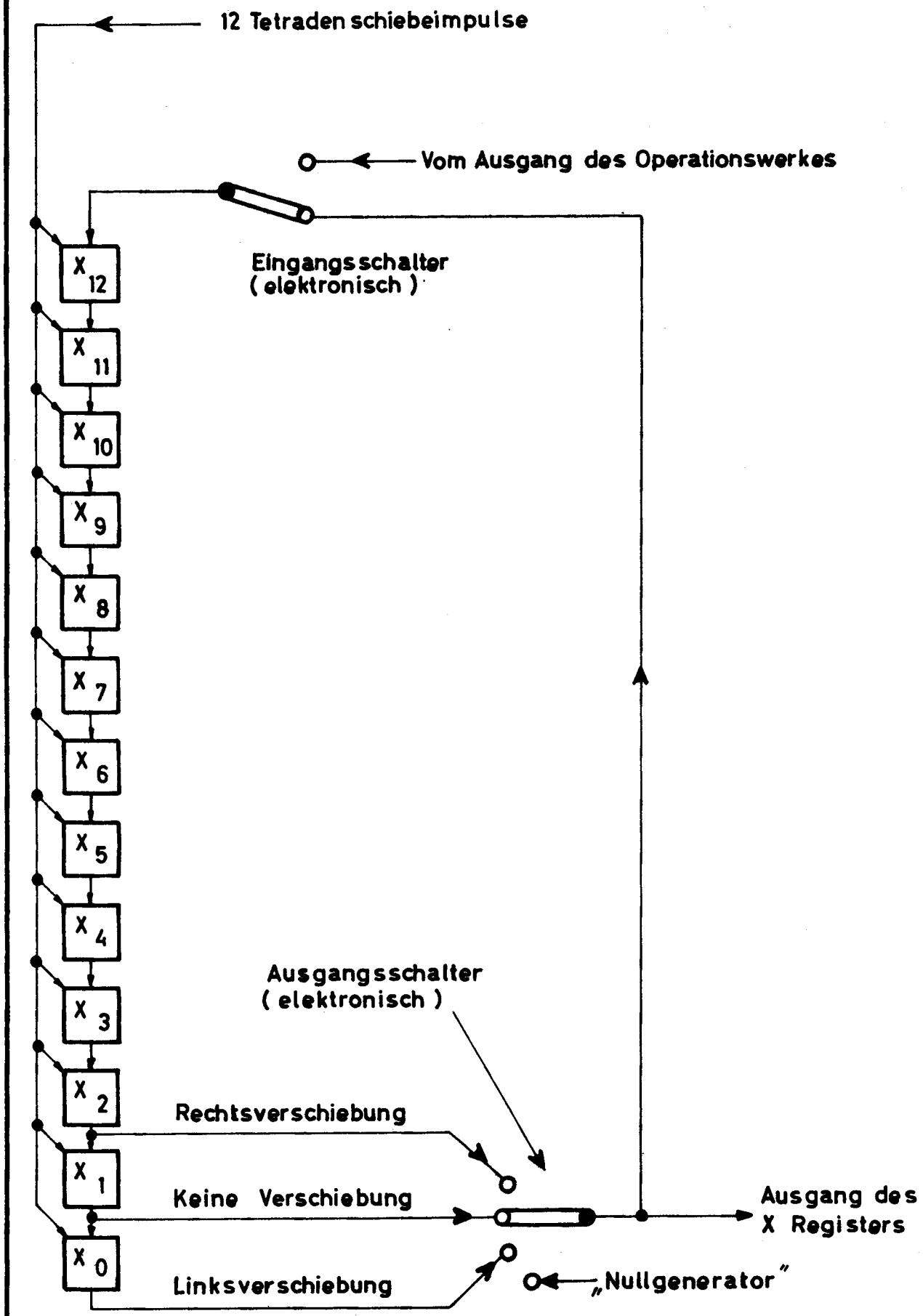


Abb. 6 Blockschaltbild des Rechenregisters X

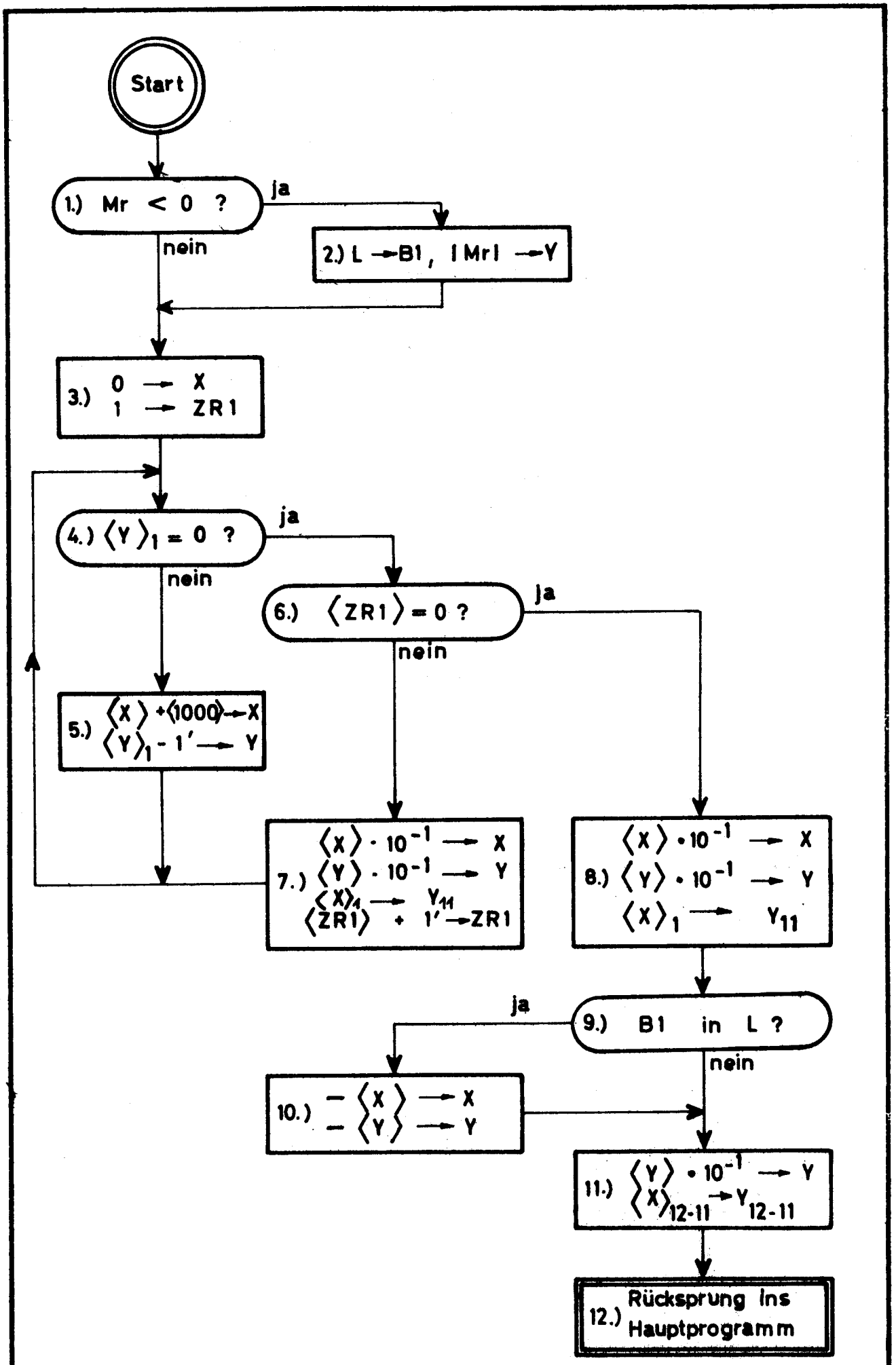


Abb.7 Strukturdiagramm der Festkommamultiplikation

Anzeige des Befehlsregisters

0	1	2	3	4	5

Lautsprecher aus	B1 blockiert	B1 in L	B1 in 0
Lautsprecher Leise	B2 blockiert	B2 in L	B2 in 0
Lautsprecher Laut	B3 blockiert	B3 in L	B3 in 0
Nacht	B4 blockiert	B4 in L	B4 in 0
Netzalarm Anschlußgeräte	B5 blockiert	B5 in L	B5 in 0
Alarm RAS	B6 blockiert	B6 in L	B6 in 0
Alarm ZR1	Alarm E1 Addierw.	Alarm E Schnellsp.	Alarm Mult.-Werk
Alarm ZR2	Alarm E2 Addierw.	L Σ Alarm A Schnellsp.	Alarm Ein-Ausgabe
Alarm C-Register	Alarm E X-Register	Alarm E Op.-Werk	Alarm Nachsch.-sp.
Alarm B-Register	Alarm E Y-Register	Alarm E ±1 Addierw.	Alarm Stromvers
			Maschine Ein
			Maschine Aus

Zeichenerklärung:

2 = Anzeigen bezw. Schriftfelder

= Tasten

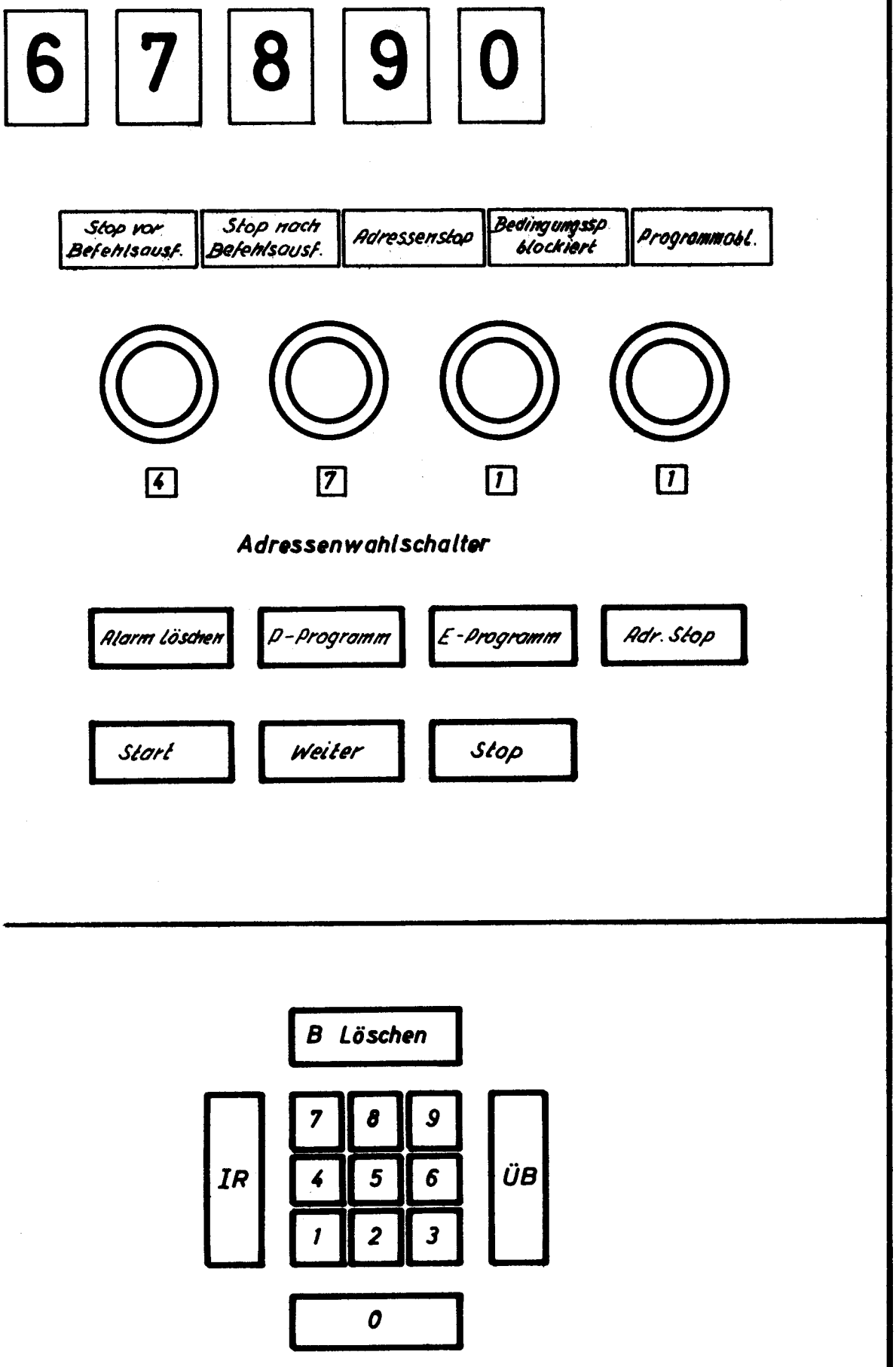
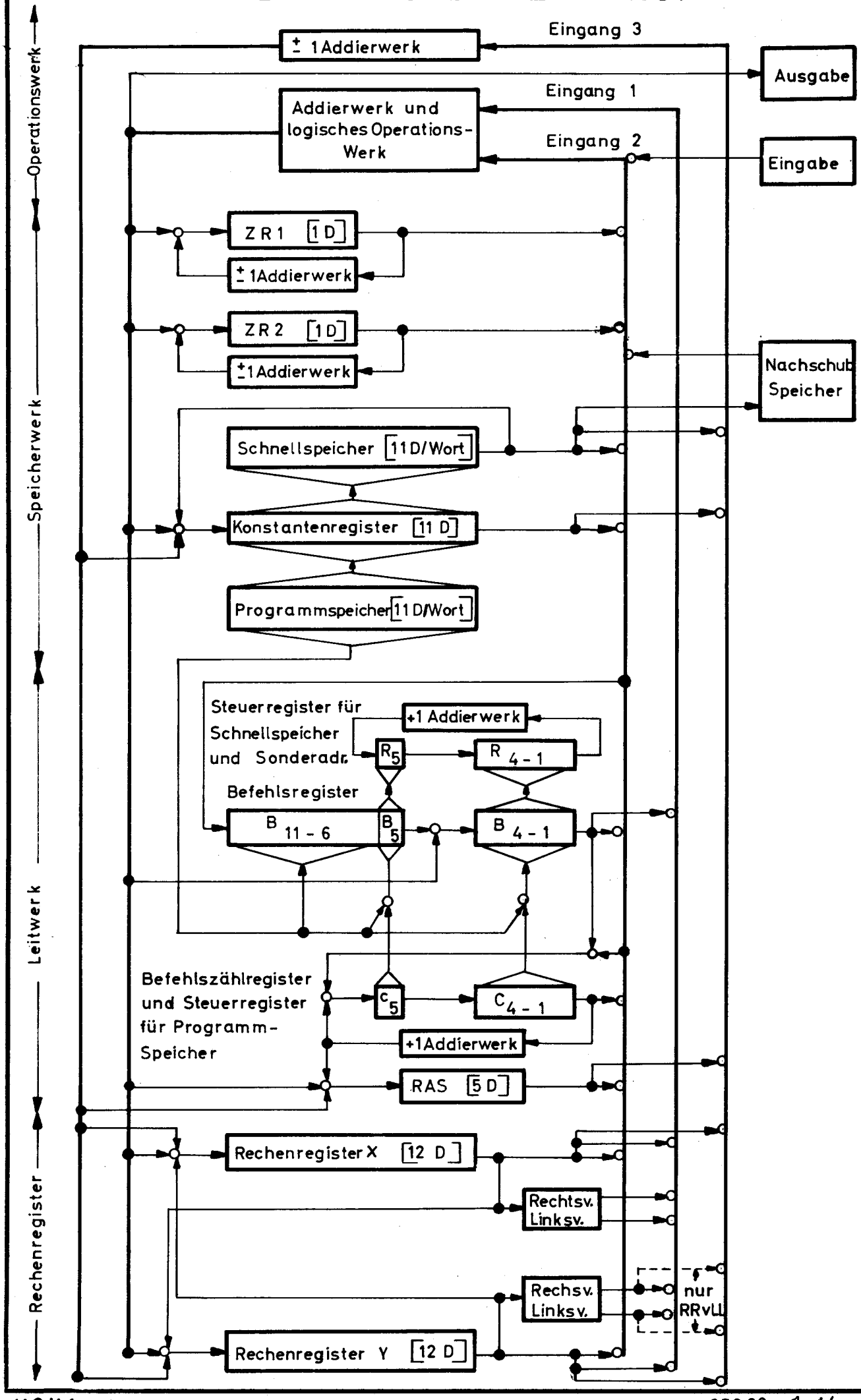


Abb. 8 Bedienungspult mit Zehnertastatur

Blockschaltbild Zuse Z31



U 3/4.1

25.9.63 A. Adam