

Einführung in die Arbeitsweise
programmgesteuerter Rechenanlagen



ZUSE KG
BAD HERSFELD

Elektronische Rechenanlagen

I.	<u>Programmgesteuerte elektronische Rechenanlagen</u>	<u>Seite</u>
1.	Informationsdarstellung in programmgesteuerten elektronischen Rechenanlagen	1
1.1	Darstellung von Zahlen im binären Zahlensystem	1
1.2	Darstellung von Zahlen im binär-dezimalen Zahlensystem	2
1.3	Darstellung von Text und Befehlen	3
2.	Arbeitsweise einer programmgesteuerten elektronischen Rechenanlage	5
2.1	Das Speicherwerk	6
2.1.1	Das Register	6
2.1.2	Der Ferritkernspeicher	7
2.1.3	Der Magnettrommel- und Bandspeicher	8
2.2	Das Leitwerk	10
2.2.1	Der Aufbau eines Befehles	10
2.2.2	Programmablauf und Steuerung	12
2.3	Das Operationswerk	13
2.3.1	Verarbeitung von Informationen im Serienbetrieb	13
2.3.2	Verarbeitung von Informationen im Serienparallelbetrieb	13
2.3.3	Verarbeitung von Informationen im Parallelbetrieb	14
2.3.4	Ausbaustufen des Operationswerkes	14
2.4	Eingabe von Informationen	16
2.4.1	Eingabe über einen Pufferspeicher für 1 Zeichen	17
2.4.2	Eingabe über einen Pufferspeicher für mehrere Zeichen	18
2.4.3	Eingabe von Analogwerten	19
2.5	Ausgabe von Informationen	20
2.5.1	Ausgabe über einen Pufferspeicher für 1 Zeichen	21
2.5.2	Ausgabe über einen Pufferspeicher für mehrere Zeichen	21
2.5.3	Ausgabe von Analogwerten	22
2.6	Kontrolleinrichtungen von Rechenanlagen	22
2.6.1	Kontrollmöglichkeiten des Dezimalrechners	22
2.6.2	Kontrollmöglichkeiten des Binärrechners	23
2.6.3	Alarmauswertung der Kontrollstellen	23
2.7	Vorrangsteuerung einer Rechenanlage	24
2.8	Vergleich zwischen wissenschaftlichen und kommerziellen Anlagen	26
2.8.1	Rechenanlagen für wissenschaftliche Anwendungen	26
2.8.2	Rechenanlagen für kommerzielle Anwendungen	26

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				Binäre Wertigkeit
	2^3	2^2	2^1	2^0	
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 gleichzeitig 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 werden 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 Textzeichens 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 1 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 2.3 gebracht.

Das Beispiel sollte nur erläutern, wozu man Register verwenden kann. Man unterscheidet zwischen statistischen und dynamischen Registern. Statistische 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 Registerzugang 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 der 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 matrixförmig 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 "Fädung" der Kerne.

Die Fädungsarbeit eines Ferritkernspeichers ist beachtlich. Ein 1000-Wort-Speicher der ZUSE Z 31 enthält z.B. fast 50 000 Kerne von 2 mm ϕ . Durch das Loch eines jeden Kernes (1,3 mm ϕ) 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. 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 "1" 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 kann. Durch eine größere Anzahl von Spuren (einige Hundert) erhält man damit eine relativ hohe Speicherkapazität. Bei normalen Trommeln sind es etwa 3000 - 20 000 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 Nachschubspeicher benutzt, d.h. man transferiert 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:

$$\langle X \rangle + \langle 1000 \rangle \rightarrow X$$

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:

$$A \quad 1000 + 100$$

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 Befehls 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 seien 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 erkaufte. 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 = 7 Befehle,
die 2-Adressmaschine = 6 Befehle,
die 3-Adressmaschine = 5 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 Programmbe-fehl in das Befehlsregister gebracht, so wird der lineare Programmablauf unterbrochen. Der Programmbe-fehl 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 Sprungbefehl 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-, Serienparallel und Parallelverarbeitung von Information. 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. Serienmaschine 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 Serienparallelmaschine 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 Serien- bzw. Serienparallelmaschinen? 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:

<u>400</u> . 321	
400	1.Schritt = 400 . 1 (1.Partialprodukt)
8000	2.Schritt = 400 . 20 (2.Partialprodukt)
<u>120000</u>	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.

$\langle Md \rangle$	$\langle P \rangle$	$\langle Mr \rangle$	durchgeführte Operationen
400	000	321	Ausgangszustand
400	400	320	$\langle P \rangle + \langle Md \rangle \rightarrow P$ (1. Partialprodukt)
400	040	032	Da die letzte Ziffer von Mr auf Null abgebaut ist, wird $\langle P \rangle$ und $\langle Mr \rangle$ gemeinsam nach rechts verschoben.
400	440	031	$\langle P \rangle + \langle Md \rangle \rightarrow P$ (2. Partialprodukt)
400	840	030	$\langle P \rangle + \langle Md \rangle \rightarrow P$
400	084	003	gemeinsame Rechtsverschiebung von P und Mr
400	484	002	$\langle P \rangle + \langle Md \rangle \rightarrow P$
400	884	001	$\langle P \rangle + \langle Md \rangle \rightarrow P$ (3. Partialprodukt)
400	1284	000	$\langle P \rangle + \langle Md \rangle \rightarrow P$
400	.128	400	gemeinsame Rechtsverschiebung von $\langle P \rangle$ und $\langle Mr \rangle$. 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. "Fünfvorteils" 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 je weils 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 "Gleitkomma-zusätze" 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 großem Aufwand erhöhte Rechenzeiten, da bestimmte Operationen durch Unterprogramme gelöst werden müssen. Die meisten Rechenanlagen verwenden aus wirtschaftlichen Gründen ein Serien- oder Serienparallel-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 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 Trapeze 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 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 Serien- oder Serienparallel-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 zu nächst 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:

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

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 Millisekunden 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 Datenverarbeitungs geschwindigkeit 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 errechneten 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 können 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 hervorgerufenen 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. Pseudotetraden dar. Tritt an den hierfür vorgesehenen Kontrollstellen der Rechenanlage eine der Pseudotetraden 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 ungerade Zahl ergänzt. Quer- und Längssummenkontrolle entdecken 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 Binärrechnern im Serienparallel-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 Alarmer 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 Stellung 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, 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. Ausgegeben 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 Anlage 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 auszugehenden 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.

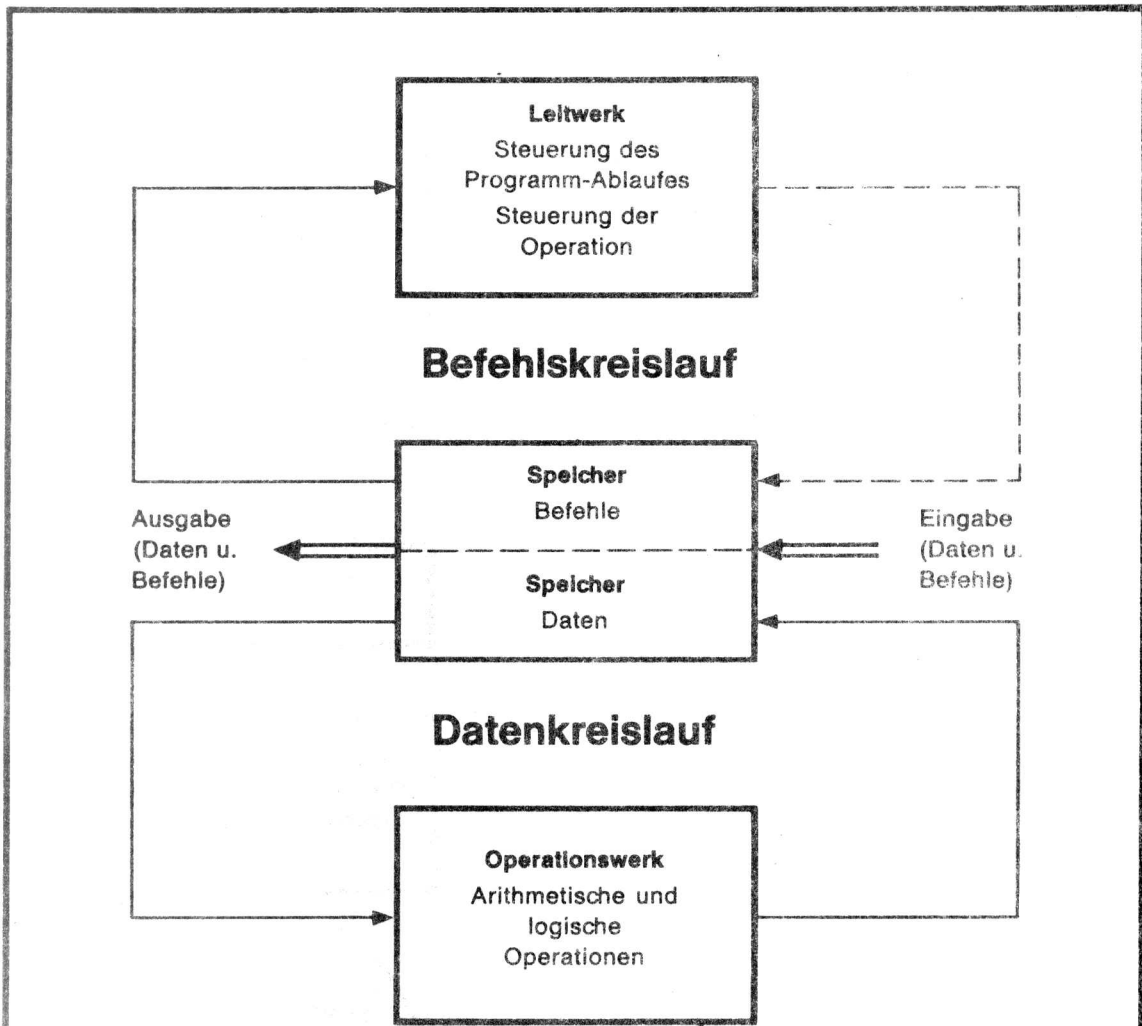


Abb. 1: Blockschaltbild einer programmgesteuerten Rechenanlage

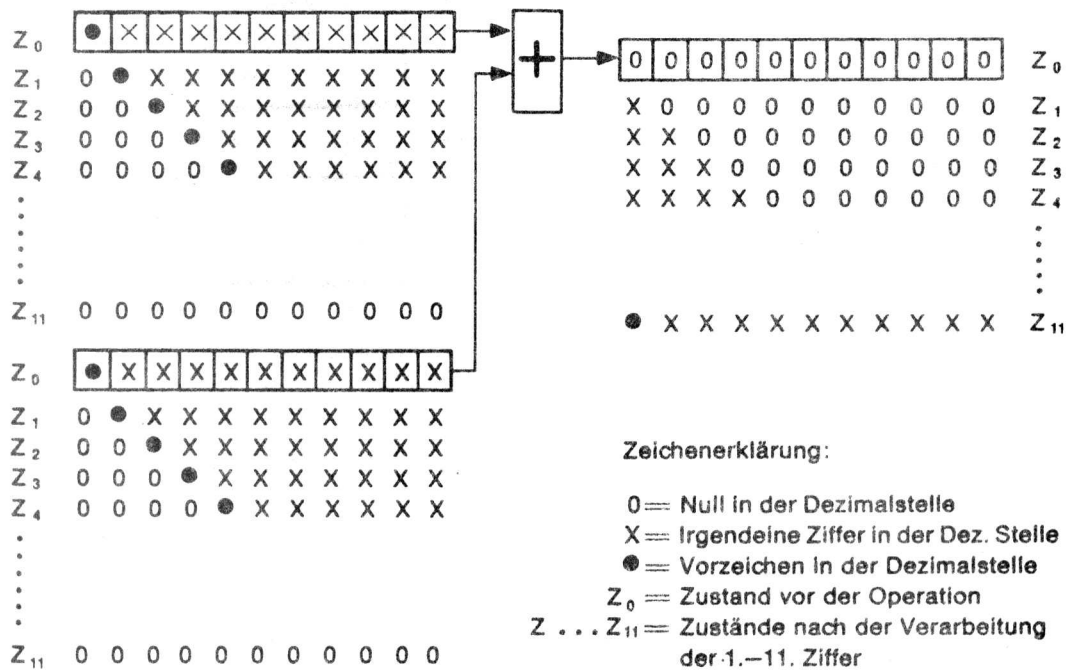


Abb. 2 Addition zweier Zahlen mit Hilfe von Schieberegistern

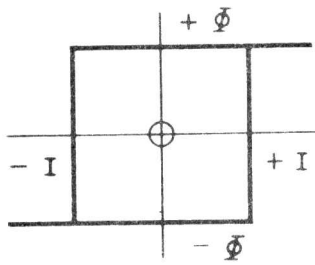


Abb. 3 Hysteresisschleife eines Ferritkerns

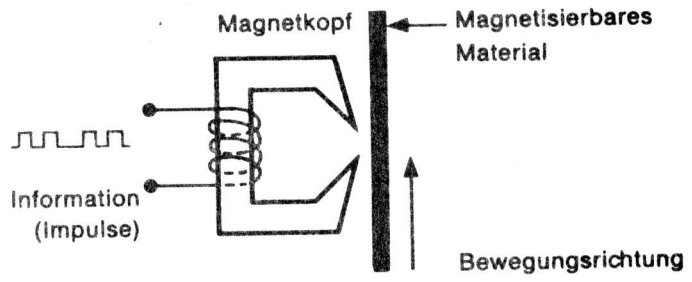


Abb. 4 Prinzip des Magnetspeichers

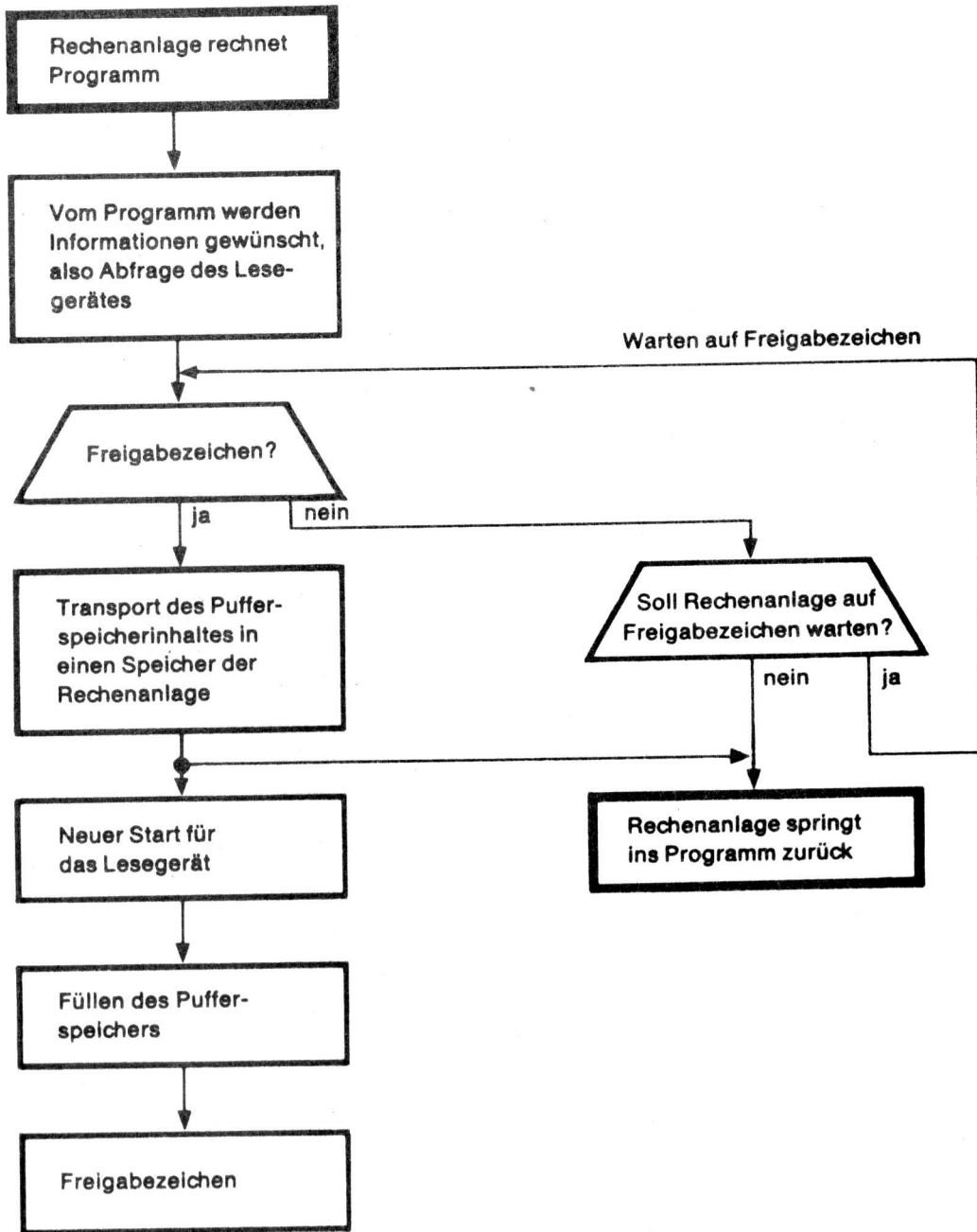


Abb. 5 Strukturdiagramm für die programmgesteuerte Eingabe

ZUSE

ZUSE KG · BAD HERSFELD
Datenverarbeitungsanlagen
Telefon 841 · Telex 0493329