

**Z U S E Z
23**

Programmierungsanleitung

ZUSE KG · BAD HERSFELD

Elektronische Rechanlagen



Programmierungsanleitung ZUSE ^Z₂₃

Ausgabe April 1964

V o r w o r t

Die ZUSE Z 23 stellt eine Weiterentwicklung der elektronischen Rechenanlage ZUSE Z 22 dar. Während in der letzteren noch grundsätzlich Röhren als Bauelemente verwendet werden, ist die ZUSE Z 23 volltransistorisiert. Sie ist eine im Dualsystem arbeitende Serienmaschine mit einer Wortlänge von 40 Binärstellen (bit).

Der interne analytische Code der ZUSE Z 22 wurde weitgehend beibehalten, nur an einigen Stellen wurden besondere Kombinationen geschaffen, um den Rechenablauf schneller zu gestalten. Das Prinzip, nur wenige logisch starke Einzelbefehle zu verdrahten, die jedoch in reicher Mannigfaltigkeit kombiniert werden können, wurde übernommen.

Für die Durchführung der Ein- und Ausgabe sowie der arithmetischen Operationen wird auch in der ZUSE Z 23 ein sogenanntes Grundprogramm gespeichert, das mit der Maschine geliefert wird und den bei der ZUSE Z 22 bewährten Freiburger Code verwendet. Grundsätzlich könnte jeder Benutzer auch einen eigenen externen Code wählen, müßte dann aber ein entsprechendes Grundprogramm selbst aufstellen.

Im Freiburger Code wird mit Zahlen im gleitenden Komma gearbeitet, und zwar mit einer Genauigkeit von 9 Dezimalstellen und einem Zahlenbereich von 10^{-39} bis 10^{+38} . Der Freiburger Code bedient sich einiger leicht einprägsamer Abkürzungen; so sind die Aufrufbefehle für die arithmetischen Programme durch die in der Mathematik gebräuchlichen Zeichen verschlüsselt (+, -, X, :). Er gestattet außerdem die Ein- und Ausgabe von Klartext (alphanumerischem Text) für Überschriften, Erklärungen usw..

Diese Programmieranleitung behandelt den Freiburger Code und geht nur so weit auf den internen Code ein, wie es für den Benutzer zum Umrechnen und Aufbauen von Adressen und Zählungen notwendig ist. Alle Einzelheiten des Internocodes werden in einer besonderen Ausarbeitung behandelt.

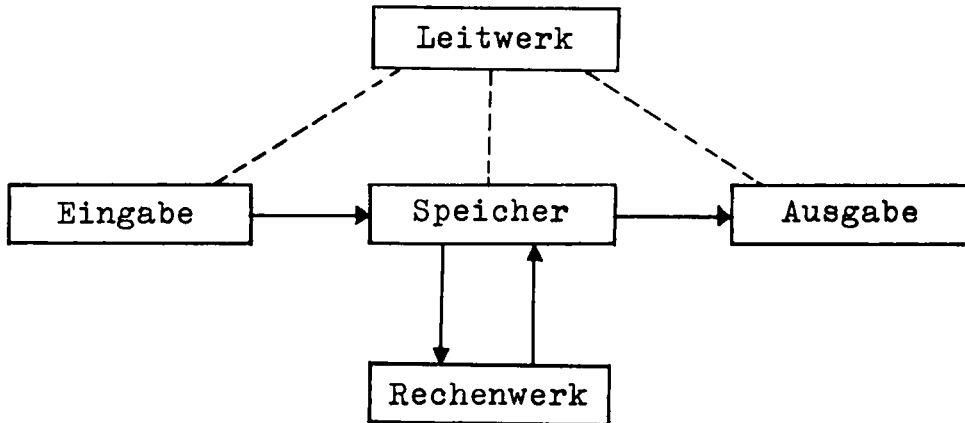
<u>Inhaltsübersicht</u>		Seite
1.	Einführung	5
1.1	Aufbauschema der Rechenanlage	5
1.2	Die Eingabe	5
1.3	Der Speicher	5
1.4	Das Leitwerk	6
1.5	Das Rechenwerk	6
1.6	Die Ausgabe	7
1.7	Das Grundprogramm	7
2.	Beschreibung der Wortarten	8
2.1	Befehle	8
2.2	Gleitkommazahlen	9
2.3	Strichzahlen	10
2.4	Klartext	10
3.	Erklärung der Befehle	11
3.1	Symbolik zur Beschreibung der Befehle	11
3.2	Sprung- und Rufbefehle	13
3.3	Lese- und Speicherbefehle	15
3.4	Blocktransferbefehle	17
3.5	Arithmetische Befehle	18
3.6	Druck- und Tabellierbefehle	21
3.6.1	Druckbefehle	21
3.6.2	Die Ausgabe von Befehlen (für Kontrollzwecke)	22
3.6.3	Die Ausgabe von Gleitkommazahlen	23
3.6.4	Die Ausgabe von Strichzahlen	24
3.6.5	Druckanordnung für Zahlen	24
3.6.6	Die Ausgabe von Klartext	25
3.6.7	Tabellierbefehle	26
3.7	Bedingte Befehle	27
3.8	Stopbefehle	29
3.9	Bandbefehle	30
4.	Programmierungsbeispiel	32
4.1	Vorbemerkung über die Programmierung	32
4.2	Symbolik für Flußdiagramme	32
4.3	Die Formulierung der Programmierungsaufgabe	33
4.4	Das Flußdiagramm zum Hauptprogramm	34
4.5	Das Flußdiagramm zum Unterprogramm	35
4.6	Das Unterprogramm	35
4.7	Das Hauptprogramm	37
5.	Herstellung und Verarbeitung des Programmstreifens	39
5.1	Die Lochstreifenherstellung	39
5.2	Die Lochstreifeneingabe	43
5.3	Das Bedienungspult der ZUSE Z 23	45
6.	Erklärung weiterer Befehle	48
6.1	Arithmetische Befehle für Strichzahlen	48
6.2	Befehle für Konstantenzuführung	52
6.3	Ein Beispiel zur Anwendung des Strichzahlrechnens	53
6.4	Befehle für logische Operationen	54

	Seite	
6.5	Befehle für Adressenänderungen	57
7.	Erneute Behandlung des Programmierungsbeispiels (abgeändert)	58
8.	Relatives und adressenloses Programmieren	62
8.1	Relativadressen	62
8.2	Ein Anwendungsbeispiel für Relativadressen	63
8.3	Symbolische Adressen	65
8.4	Ein Anwendungsbeispiel für symbolische Adressen	67
9.	Anhang:	69
	Befehlsliste für die ZUSE Z 23	70
	Liste der Schnellspeicher mit Sonderfunktionen	75
	Oktaltabelle	76
	Der Fernschreibcode	79
	Frontansicht des Bedienungspults	80
	Beschreibung des ZUSE-Formelcodes ("Kleiner Formelübersetzer")	81

1. Einführung

1.1 Aufbauschema der Rechenanlage

Die ZUSE Z 23 besteht im wesentlichen aus 5 Hauptteilen: Eingabe, Leitwerk, Rechenwerk, Speicher und Ausgabe.



Alle zu verarbeitenden Zahlenwerte sowie alle Programme, d.h. die im Befehlscode der Maschine in logischer Folge niedergeschriebenen Rechenanweisungen, werden der Maschine mittels Lochstreifen eingegeben. Die Lochstreifen werden auf einem gesonderten Programmierungstisch hergestellt.

1.2 Die Eingabe

Die Eingabevorrichtung, kurz Eingabe genannt, besteht aus einem photoelektrischen Abtaster für Lochstreifen, der bis zu 300 Zeichen/sec verarbeiten kann. Die eingegebenen Programme und Daten werden durch ein feststehendes Leseprogramm (vergleiche 1.7) in die interne Sprache der Maschine übersetzt und im Speicher aufbewahrt.

1.3 Der Speicher

Als Speicherorgane dienen eine Magnettrommel und ein Schnell-speicher (Kernspeicher).

Die Magnettrommel besitzt 8191 direkt adressierbare Speicherzellen, was einer Speicherkapazität von 8 191 "Worten" entspricht. Unter der Bezeichnung "Wort" versteht man eine Zahl, einen Befehl oder einen 7-stelligen alphanumerischen Text, sogenannten Klartext. Die Zugriffszeit zu einer beliebigen Trommelspeicherzelle beträgt im Mittel 5 msec.

Der Schnellspeicher besteht aus 256 Kernspeicherzellen, von denen jedoch einige für Sonderfunktionen vorgesehen sind (siehe Anhang). Kernspeicherzellen sind jederzeit ohne Zugriffszeit aufrufbar und können, ebenso wie die Trommelspeicherzellen, je ein Wort aufnehmen.

Zwischen Magnettrommel und Schnellspeicher ist ein direkter optimaler Blocktransfer in beiden Richtungen möglich, so daß z.B. ganze Programmteile oder auch Datengruppen in den Schnellspeicher gebracht werden können, wo sie dann ohne Zugriffszeiten ablaufen bzw. abgeholt werden können. Die Zeit des Blocktransfers beträgt für je 32 Worte 10 msec.

1.4 Das Leitwerk

Aufgabe des Leitwerks ist es, die bei der Eingabe in die Rechenanlage zunächst nur gespeicherten Befehle eines Programms vom Speicher abzuholen und zur Ausführung zu bringen. Zahlenwerte werden vom Speicher in das Rechenwerk befördert und Rechenergebnisse entweder wiederum gespeichert oder zur Ausgabe gebracht.

Das Leitwerk besteht aus drei Registern. Von diesen übernimmt eines die Aufnahme eines Befehls (Befehlsregister), ein zweites sorgt für die Entschlüsselung und Ausführung des Befehls (Steuerregister) und das dritte für den Aufruf des nächsten Befehls (Befehlszählregister). Bei der Ausführung eines Befehls wird die in der Adresse angegebene Speicherzelle angewählt, und gemäß dem Operationsteil werden entsprechende Stromwege in der Rechenanlage geöffnet bzw. geschlossen.

1.5 Das Rechenwerk

Das Rechenwerk ist ein Operationswerk, in dem ganze Zahlen (Strichzahlen) addiert und subtrahiert sowie logische Operationen (Konjunktion, Disjunktion) ausgeführt werden.

Das Operationswerk ist im allgemeinen dem Hauptakkumulator (Schnellspeicherzelle 4) vorgeschaltet, in dem sich vor der Operation der erste Operand und nach der Operation das Ergebnis befindet. Es kann jedoch unter Beibehaltung des Inhalts des Hauptakkumulators auch jeder anderen Kernspeicherzelle vorgeschaltet werden, wobei diese jetzt den ersten Operanden und das Ergebnis aufnimmt.

Die Multiplikation und Division von Strichzahlen, aber auch die Rechenoperationen mit sogenannten Gleitkommazahlen (siehe 2.2), die in der Maschine beim Einlesen der Daten aufgebaut werden, lassen sich letztlich auf Strichzahladditionen und -subtraktionen zurückführen. Für alle diese Operationen muß der erste Operand vorher

stets erst nach Schnellspeicherzelle 6 und der zweite Operand in den Akkumulator gebracht werden. Das Ergebnis steht anschließend sowohl im Akkumulator als auch in Schnellspeicherzelle 6 zur Verfügung (siehe 3.5 und 6.1).

1.6 Die Ausgabe

Die Ausgabevorrichtung, kurz Ausgabe genannt, besteht in ihrer einfachen Ausführung aus einem Fernschreiber, der mit 10 Zeichen/sec arbeitet und mit einem Streifenlocher ausgestattet ist. Sollen größere Mengen von Zahlen ausgegeben werden, so besteht die Möglichkeit, einen Streifenlocher mit 150 Zeichen/sec anzuschließen (Schnellocher). Die erhaltenen Streifen können dann später unabhängig von der Rechenanlage an lochstreifengesteuerten Fernschreibmaschinen ausgedruckt werden.

1.7 Das Grundprogramm

Da in der ZUSE Z 23 nur wenige Operationsanweisungen fest verdrahtet sind, werden die meisten in Form von Programmen eingegeben. Die detaillierten Anweisungen für die Ein- und Ausgabevorgänge sowie für die grundlegenden Rechenoperationen und Zahlenumformungen hat man in einem sogenannten Grundprogramm zusammengefaßt, das mit der Maschine geliefert wird. Es wird in den Trommelzellen 0 bis 1023 gespeichert und kann durch Sperrschalter vor versehentlichem Überschreiben geschützt werden. Als Bestandteile des Grundprogramms unterscheidet man:

das Leseprogramm, das die vom Lochstreifen abgetasteten Buchstaben und Ziffern in die interne Sprache der Maschine übersetzt (z.B. werden Dezimalzahlen als Dualzahlen in Gleitkommaform gespeichert);
die arithmetischen Programme für die Ausführung von Addition, Subtraktion, Multiplikation, Division und Wurzelziehen im gleitenden Komma;
das Druckprogramm, das die Rechenergebnisse in die Dezimalschreibweise zurückübersetzt und gespeicherten Text wieder ausgibt.

2. Beschreibung der Wortarten

2.1 Befehle

In der ZUSE Z 23 besitzt jedes zu verarbeitende Wort eine Wortlänge von 40 Binärstellen (bit). Man verwendet vier verschiedene Wortarten: Befehle, Gleitkommazahlen, Strichzahlen und Klartext. Bei der Beschreibung dieser Wortarten muß zwischen der Struktur vor und nach der Eingabe (Eingabeform bzw. maschineninterne Form) unterschieden werden. Vor der Eingabe besteht jeder Befehl aus zwei Bestandteilen:

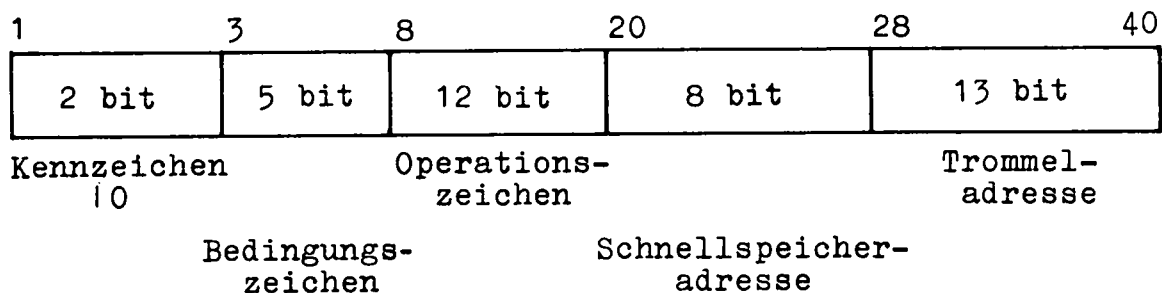
- 1) dem Operationsteil, dargestellt durch einen oder mehrere Buchstaben, die angeben, was durchgeführt werden soll und gegebenenfalls unter welchen Bedingungen;
- 2) dem Adressenteil, der durch ganze Zahlen dargestellt wird und angibt, welche Speicherzelle bei der im Operationsteil angegebenen Operation aufgerufen werden soll. Im Adressenteil können zwei Adressen auftreten, als erste eine Kernspeicheradresse und als zweite eine Trommeladresse; diese Adressen werden durch ein +Zeichen voneinander getrennt. In allen Einadreßbefehlen wird eine Adresse unter 256 vom Leseprogramm automatisch als Kernspeicheradresse erkannt und als solche in den zu speichernden Befehl eingesetzt. Wünscht man eine Trommelspeicheradresse unterhalb 256 anzuwählen, so wird sie durch Einfügen der Ziffer 0 und des + Zeichens zwischen dem Operationsteil und der Trommelspeicheradresse eindeutig als solche erkannt (z.B. Sprung auf die Trommelzelle 6: Eo+6).

Ausnahmen von diesem Aufbau eines Befehls aus Operations- und Adressenteil bilden die vercodeten Rufbefehle für die arithmetischen Operationen (+, -, X, : usw.). Auch die Ausführung dieser Operationen kann mit einer Bedingung verknüpft werden; jedoch muß hier die Reihenfolge

Bedingungszeichen - übriger Befehl

eingehalten werden, die auch sonst der Übersichtlichkeit wegen bevorzugt werden sollte.

Nach dem Einlesen läßt sich jeder Befehl im Innern der Maschine wie folgt zergliedern:



Den Bedingungs- und Operationszeichen eines Befehls sind in ihrer internen Form ganz bestimmte Binärstellen der Speicherzelle zugeordnet. Schnellspeicher- und Trommeladresse erscheinen als Dualzahlen acht- bzw. dreizehnstellig. Die beiden ersten Binärstellen enthalten das für Befehle (außer Stop) charakteristische Kennzeichen 10.

2.2 Gleitkommazahlen

Man hat weitgehende Freiheit in der Form, wie man Zahlen in die Maschine eingibt. Die Zahl +7683,850 kann z.B. auch als 7683,85 oder auch in halblogarithmischer Form als $7,68385/3$ oder $,768385/4$ usw. eingegeben werden. Positive Vorzeichen der Zahl und des Exponenten können weggelassen, negative müssen angegeben werden. Ist die Zahl bzw. die Mantisse kleiner als 1, so braucht man die Null vor dem Komma nicht mit einzugeben.

Die zu einer Zahl gehörenden Zeichen müssen bei der Eingabe ohne jeden Abstand unmittelbar aufeinanderfolgen, da ein Zwischenraum oder Wagenrücklauf oder Zeilentransport als Schlußzeichen für eine Zahl (für ein Wort überhaupt) aufgefaßt wird. Ein Zwischenraum innerhalb einer Zahl würde also die getrennten Teile für sich als Zahlen kennzeichnen.

Als größte Zahl kann $,1/39$ eingegeben werden (mit 9 Ziffern in der Mantisse kann die größte Zahl $,134217727/39$ lauten). Werden größere Zahlen verwendet, so schreibt die Maschine aus:
 BUEB Eo+285
 Hier bedeutet BUEB Bereichsüberschreitung, und in Eo+285 kennzeichnet 285 die Stelle des Leseprogramms, wo die Bereichsüberschreitung festgestellt wurde.

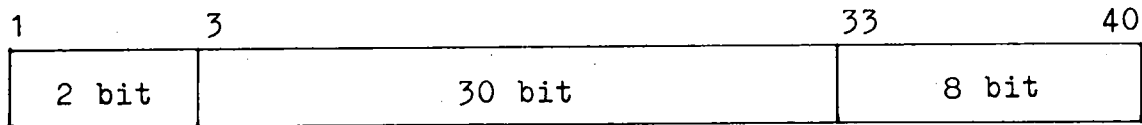
Die kleinste Zahl, die noch (von Null verschieden) in die Maschine eingegeben werden kann, ist $,2/-38$ (mit 9-stelliger Mantisse $,214748364/-38$).

Die Beschränkung der Eingabedaten auf den genannten Zahlenbereich sowie die höchstzulässige Stellenzahl 9 für die Mantisse sind durch die Anzahl der in der Maschine verfügbaren Binärstellen bedingt. Mit Ausnahme der sogenannten Strichzahlen (siehe 2.3) werden alle eingegebenen Zahlen in den Binärstellen als Gleitkommazahlen notiert. Zerlegt man eine Zahl x in ein Produkt aus Mantisse \hat{x} , deren Betrag zwischen 0,5 und 1 liegt, und einer Dualpotenz mit dem Exponenten \bar{x} , setzt also

$$\pm x = \pm \hat{x} \cdot 2^{\bar{x}}$$

so sind es die Zahlen \hat{x} und $\bar{x} + 128$, die in den Binärstellen als Dualzahlen erscheinen.

Die Aufteilung einer Gleitkommazahl auf die Dualstellen ist also folgende:



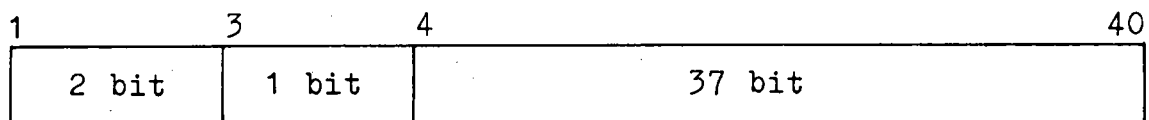
Vorzeichen	Mantisse \hat{x}	Exponent $\bar{x}+128$
00 positiv	$(0,5 \leq \hat{x} < 1)$	
$\overline{00}$ negativ		

Die fiktive Erhöhung des Dualexponenten um 128 wurde zur Vermeidung einer gesonderten Vorzeichenkennzeichnung vorgenommen. Mit 8 bit ist im Dualsystem für $\bar{x} + 128$ ein Bereich von 0 bis 255, d.h. für \bar{x} von -128 bis +127 möglich. Dem Zahlenbereich von 2^{-128} bis 2^{+127} entspricht im Dezimalsystem der Bereich von 10^{-39} bis 10^{+38} .

Negative Zahlen werden durch das Komplement der Mantisse dargestellt, wobei auch die beiden Kennzeichenstellen mit einbezogen werden.

2.3 Strichzahlen

Neben Gleitkommazahlen kann man für Zählzwecke und zum Adressenumrechnen noch eine andere Art von Zahlen eingeben und in der Maschine verarbeiten. Diese Zahlen werden bei der Ein- und Ausgabe mit einem Apostroph nach der letzten Ziffer gekennzeichnet und deshalb als "Strichzahlen" bezeichnet. Die Strichzahlen können nur ganze Zahlen sein und stehen in der Maschine von den niedrigsten Stellen ab aufbauend. Die größte verwendbare Strichzahl ist $2^{37}-1$, da die dritte Binärstelle zur Unterscheidung von einer Gleitkommazahl benötigt wird.



Vorzeichen-	Untersch.von	Strichzahl
stellen	Gleitkommazahl	(ganzzahlig)
00 bzw.	0 bzw.	

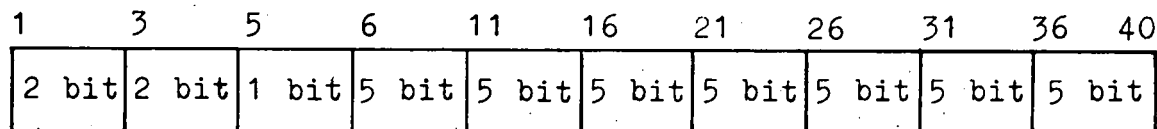
Das Rechnen mit Strichzahlen wird in 6.1 erklärt.

2.4 Klartext

Als weitere Wortart existiert in der Maschine der sogenannte Klartext. Er dient dazu, Überschriften und Erläuterungen zu den Ausgabewerten zu setzen, was vom Programm aus an den entsprechenden Stellen geschieht.

Zur Eingabe wird der Text am Anfang und am Ende mit einem ⑆ - Zeichen (Klingelzeichen) gekennzeichnet. Bei Verwendung eines Algol-Fernschreibers ist statt des ⑆ - Zeichens ein Semikolon zu setzen; beide Zeichen ergeben die gleiche Lochung, jedoch erscheint nur das zweite im Schriftbild.

In den Binärstellen einer Speicherzelle lassen sich jeweils sieben aufeinanderfolgende Fernschreibzeichen unterbringen, da gemäß dem Fernschreibcode (siehe Anhang) jedes Fernschreibzeichen durch eine Kombination von fünf Binärstellen gekennzeichnet werden kann. Die interne Aufteilung eines Klartextwortes hat daher folgende Gestalt:



Kenn- Zusatz-
zeichen kennz.

01
(auch
Stop)

11
0 bzw. 1
↓ ↓

Ziffern- Buchstaben-
zeichen zeichen

7 Fernschreibzeichen

Das Zusatzzeichen 11 ist zur Unterscheidung von Stopbefehlen notwendig. An der fünften Binärstelle läßt sich erkennen, ob das erste Fernschreibzeichen ein Buchstaben- oder ein Ziffernzeichen sein soll. Unter den gespeicherten Fernschreibzeichen befinden sich auch die den Wechsel von Buchstaben- auf Ziffernzeichen bzw. von Ziffern- auf Buchstabenzeichen ankündigenden Zeichen für Zi bzw. Bu (siehe 5.1).

3. Erklärung der Befehle

3.1 Symbolik zur Beschreibung der Befehle

- a Der Akkumulator
- b Das Befehlsregister, das die einzelnen Befehle der Reihe nach aufnimmt und parallel an das Steuerregister weitergibt.
- r Das Steuerregister, von wo aus die einzelnen Stromtore in der Maschine geöffnet bzw. geschlossen werden, damit die den Befehlen entsprechenden Wege in der Maschine freigegeben werden.
- c Das Befehlszählregister, das zur Aufnahme der Adresse des Folgebefehls bestimmt ist.
- m, n Allgemeine Adressen
(Schnellspeicher- oder Trommeladressen).

- s Eine Schnellspeicheradresse.
- t Eine Trommeladresse.
- < m > Das Wort, das in der Speicherzelle steht, die mit der Adresse m aufgerufen werden kann; kurz: Inhalt von m.
- { x } Die Adresse der Speicherzelle, in der der Wert x steht.
- < m > p...q Inhalt der Dualstellen p...q der Speicherzelle m; d.h., hier ist nicht der gesamte Speicherinhalt von m gemeint, sondern nur der Teil, der in den bezeichneten Dualstellen steht. Hierbei ist eine Numerierung der Dualstellen innerhalb der Speicherzelle, von den Kennzeichenstellen angefangen, vorausgesetzt.
- < m > t Inhalt der Trommel-Adressenstellen der Speicherzelle m, also die Stellen 28 bis 40.
- < m > s Inhalt der Schnellspeicher-Adressenstellen der Speicherzelle m, also die Stellen 20 bis 27.
- ...→... Transportsymbol ("geht nach"); das bedeutet, daß der links angegebene Wert an die rechts angegebene Stelle transportiert wird.
- < m > → a heißt: Der Inhalt der Speicherzelle m wird in den Akkumulator gebracht; kurz: Inhalt von m geht nach Akku.
- ..⇒... "Ergibt"-Zeichen. Hier wird links vom Doppelpfeil angegeben, nach welchen Vorschriften aus bekannten Größen der rechts stehende Wert gebildet werden soll.
- k+1 ⇒ k Der um 1 erhöhte bisherige Index k "ergibt" für den nächsten Durchlauf den neuen Index k.
- x+Δ x ⇒ x Das bisherige Argument x, um die Schrittweite Δ x verändert, "ergibt" das neue Argument.
- k Eine ganze Zahl, aufbauend auf der untersten Dualstelle (Stelle 40).

3.2 Sprung- und Rufbefehle

A. Der Sprungbefehl, dargestellt durch das Operationszeichen E in Verbindung mit einer Adresse m: Em

Er hat zwei Wirkungen:

- 1) wird der in der angegebenen Zelle m stehende Befehl zur Ausführung ins Befehlsregister geholt,
- 2) wird der Abrufbefehl E(m+1) im Befehlszählregister bereitgestellt.

Befehl	Wirkung	
Em	$\langle m \rangle \rightarrow b,$	$E(m+1) \rightarrow c$

Durch den im Befehlszählregister stehenden Abrufbefehl wird während bzw. nach der Ausführung des aus der Speicherzelle m geholten Befehls der folgende Befehl aus der Speicherzelle m+1 ins Befehlsregister geholt. Da immer eine Erhöhung der Adresse um 1 erfolgt, ist gewährleistet, daß die Befehle in der Reihenfolge der Speicherzellen ausgeführt werden. Diese Folge wird unterbrochen, sobald aus einer Speicherzelle ein Sprungbefehl auf eine andere Stelle ins Befehlsregister geholt wird, der dann einen entsprechenden Ablauf von der anderen Stelle an auslöst.

Falls die Adresse des E-Befehls eine Trommeladresse ist, wird in den Trommel-Adressenstellen eine Eins (also 1') addiert; falls sie aber eine Schnellspeicheradresse ist, so wird beim Übergang vom Befehlsregister zum Befehlszählregister in den Schnellspeicher-Adressenstellen eine Eins (also $8192' = 2^{13}$) addiert. Speziell bewirken also die Befehle

Et	$\langle t \rangle \rightarrow b,$	$E(t+1) \rightarrow c,$
Es	$\langle s \rangle \rightarrow b,$	$E(s+1) \rightarrow c.$

Durch diese Organisation sind Programmabläufe auf der Trommel oder im Kernspeicher möglich.

B. Der Rufbefehl, dargestellt durch das Operationszeichen F in Verbindung mit einer Adresse m: Fm.

Er ist ebenfalls ein Sprungbefehl. Zusätzlich wird jetzt aber der E-Befehl, der den F-Befehl ins Befehlsregister gerufen hat, mit um 1 erhöhter Adresse (letzter Befehlszählregisterstand) in Schnellspeicherzelle 5 notiert. Der in 5 aufbewahrte Befehl ist also der Abrufbefehl (Rücksprungbefehl) für die nächste Speicherzelle hinter derjenigen, in der der F-Befehl stand.

Fm	$\langle m \rangle \rightarrow b, E(m+1) \rightarrow c,$	zuvor $\langle c \rangle = E(\{Fm\}+1) \rightarrow 5$
----	--	---

Diese F-Befehle dienen zur Einschaltung von Unterprogrammen an beliebiger Stelle des Programms. Der Aufrufbefehl hat

3.3 Lese- und Speicherbefehle

Diese Befehle dienen dazu, Zahlen oder andere Worte aus dem Speicher herauszulesen bzw. in gewisse Speicherzellen zur Aufbewahrung hineinzuschreiben.

Lesebefehl:

Bm < m > → a, < m > → m

Bringe Inhalt von m nach a:

Der Inhalt der angewählten Speicherzelle m wird in den Akkumulator gebracht; eine vorher im Akkumulator stehende Zahl wird vor der Übertragung gelöscht. Die in der Speicherzelle m stehende Zahl bleibt dort auch nach dem Ablesen erhalten. Gleichzeitig wird der nächste Befehl ins Befehlsregister geholt.

Speicherbefehl:

Um < a > → m, < a > → a

Umspeichern des Inhalts von a nach m:

Der gerade im Akkumulator stehende Wert wird in den angegebenen Speicher m umgespeichert unter Erhaltung des Wertes im Akkumulator. Gleichzeitig wird der nächste Befehl aus dem Programm in das Befehlsregister gebracht.

Ein kleines Beispiel soll die Anwendung dieser Befehle zeigen:

Es sollen die in den Speicherzellen 2000 bis 2004 stehenden Worte in die Speicherzellen 7053 bis 7057 umgespeichert werden (Es werden etwa beispielsweise die Anfangswerte für ein Programm in die dazu vorgesehenen Speicher gebracht).

Der Transport direkt von einer Trommelspeicherzelle in eine andere ist nicht möglich, deshalb muß die Umspeicherung in zwei Schritten erfolgen:

1. Bringe die Werte aus den alten Trommelzellen in den Akkumulator.
2. Speichere die Werte aus dem Akkumulator in die neuen Trommelzellen um.

Das Programm selbst, das die Umspeicherung vornimmt, soll etwa in den Speicherzellen 2050 ff stehen.

Zellen-Nr.	Inhalt	Erläuterung
2050	B2000	$\left. \begin{array}{l} \langle 2000 \rangle \rightarrow a \\ \langle a \rangle \rightarrow 7053 \end{array} \right\} \langle 2000 \rangle \rightarrow \langle 7053 \rangle$
2051	U7053	
2052	B2001	$\left. \begin{array}{l} \langle 2001 \rangle \rightarrow a \\ \langle a \rangle \rightarrow 7054 \end{array} \right\} \langle 2001 \rangle \rightarrow \langle 7054 \rangle$
2053	U7054	
2054	B2002	$\left. \begin{array}{l} \\ \end{array} \right\} \langle 2002 \rangle \rightarrow 7055$
2055	U7055	
2056	B2003	$\left. \begin{array}{l} \\ \end{array} \right\} \langle 2003 \rangle \rightarrow 7056$
2057	U7056	
2058	B2004	$\left. \begin{array}{l} \\ \end{array} \right\} \langle 2004 \rangle \rightarrow 7057$
2059	U7057	

Die Befehle Bm und Um mit nur einer Adresse wirken stets auf den Akkumulator a; m kann eine Schnellspeicheradresse ($m < 256$) oder eine Trommeladresse sein.

Die Befehlszeichen B und U können auch mit zwei Adressen auftreten, wobei die erste eine Schnellspeicher- und die zweite eine Trommeladresse ist. Mit der ersten Adresse wird die Schnellspeicherzelle bezeichnet, die in diesem Fall als Akkumulator dienen soll.

Bs+t $\langle t \rangle \rightarrow s,$ $\langle a \rangle \rightarrow a$

Es wird der Inhalt der Trommelzelle t in die angegebene Schnellspeicherzelle s gebracht; ein vorher in s stehender Wert wird zerstört. Der Inhalt des Akkumulators bleibt unberührt.

Us+t $\langle s \rangle \rightarrow t,$ $\langle s \rangle \rightarrow s,$ $\langle a \rangle \rightarrow a$

Umspeichern aus der Schnellspeicherzelle s in die Trommelzelle t. Der Inhalt von s bleibt erhalten. Auch hier bleibt der Inhalt des Akkumulators unverändert.

Wird dem Befehl U noch ein N zugesetzt, so wird nach der Umspeicherung aus der Schnellspeicherzelle diese auf Null gesetzt.

NUs+t $\langle s \rangle \rightarrow t,$ $0 \rightarrow s,$ $\langle a \rangle \rightarrow a$

Als Beispiel sei diesmal das Umspeichern von den Trommelzellen 2000 ff in die Schnellspeicherzellen 40 bis 43 durchgeführt.

Zellen-Nr.	Inhalt	Erläuterung
6000	B40+2000	<2000> → 40
1	B41+2001	<2001> → 41
2	B42+2002	<2002> → 42
3	B43+2003	<2003> → 43

Umgekehrt mögen die Ergebnisse eines Programms etwa in den Schnellspeicherzellen 60, 62, 64, 66 stehen und sollen auf die Trommel in die Zellen 2100 ff gebracht werden.

Zellen-Nr.	Inhalt	Erläuterung
2918	U60+2100	<60> → 2100
2919	U62+2101	<62> → 2101
2920	U64+2102	<64> → 2102
2921	U66+2103	<66> → 2103

3.4 Blocktransferbefehle

Als weitere Variante dieser beiden Befehlszeichen B und U gibt es die Befehle BVs+t und UVs+t bzw. NUVs+t für den Blocktransfer zwischen Trommel und Kernspeicher. Hier werden nicht nur einzelne Worte übertragen, sondern, durch einen solchen Befehl ausgelöst, eine ganze Reihe von Worten. Die Anzahl der zu übertragenden Worte muß vor dem Geben des Blocktransferbefehls als Strichzahl in Schnellspeicherzelle 13 eingegeben werden, und zwar muß man zur Übertragung von n+1 Worten n' nach 13 bringen. Diese Angabe in 13 geschieht am einfachsten mit einem Befehl, der erst später erklärt wird.

$$\left. \begin{array}{l} \text{CKB13+n} \\ \text{BV s+t} \end{array} \right\} \begin{array}{l} \text{Zuerst } n' \rightarrow 13; \text{ dann } \langle t+i \rangle \rightarrow (s+i) \\ \text{für } i=0,1,\dots,n \end{array}$$

Die n+1 Worte aus den Trommelzellen t bis t+n werden in die Schnellspeicherzellen s bis s+n gebracht. Entsprechend ist auch ein Blocktransfer vom Kernspeicher zur Trommel möglich:

$$\left. \begin{array}{l} \text{CKB13+n} \\ \text{UV s+t} \end{array} \right\} \begin{array}{l} \text{Zuerst } n' \rightarrow 13; \text{ dann } \langle s+i \rangle \rightarrow (t+i) \\ \text{für } i=0,1,\dots,n \end{array}$$

Die Inhalte der Schnellspeicherzellen bleiben hierbei erhalten. Wenn anstelle von UVs+t der Befehl NUVs+t gegeben wird, werden die Schnellspeicherzellen jeweils nach dem Umspeichern gelöscht.

Als Beispiel soll noch einmal die bei den einfachen B- und U-Befehlen aufgeführte Umspeicherung der Zelleninhalte 2000 bis 2004 in die Speicherzellen 7053 bis 7057, diesmal mit Blocktransferbefehlen, durchgeführt werden.

Das Programm stehe etwa wieder in den Zellen 2050 ff.

Zellen-Nr.	Inhalt	Erläuterung
2050	CKB13+4	4' → 13 (es sollen 5 Worte umgespeichert werden).
2051	BV25+2000	Übertragung der Inhalte der Trommelzellen 2000 ff in die Kernspeicherzellen 25 ff.
2052	CKB13+4	4' → 13 (es sollen 5 Worte umgespeichert werden).
2053	UV25+7053	Übertragung der Inhalte der Kernspeicherzellen 25 ff in die Trommelzellen 7053 ff.

Man sieht, daß das Programm mittels dieser Befehle wesentlich kürzer ist, d.h., daß es weniger Befehle und damit in der Maschine weniger Speicherraum benötigt. Außerdem läuft das Programm auch noch wesentlich schneller ab, da der Blocktransfer optimal, d.h. ohne Wartezeiten auf Trommelzellen, erfolgt. Weitere Blocktransferbefehle siehe unter 6.1 und 6.4.

3.5 Arithmetische Befehle

Für den Aufruf der arithmetischen Operationen im gleitenden Komma werden Zeichen verwendet, die den allgemein üblichen entsprechen. Für diese Operationen, die in der Maschine vom Grundprogramm ausgeführt werden, wird vorausgesetzt, daß der

1. Operand in Schnellspeicherzelle 6 und der
2. Operand im Akkumulator

steht, ehe die Operation aufgerufen wird. Das Ergebnis der Operationen steht stets in beiden Operandenregistern, also im Akkumulator und auch in Schnellspeicherzelle 6, zur Verfügung.

Bei den arithmetischen Unterprogrammen werden die Schnellspeicherzellen 2 bis 18 als Zwischenspeicher gebraucht; d.h. Werte, die in diesen Speicherzellen stehen, werden beim Aufruf der arithmetischen Operationen zerstört.

+ = Addition im gleitenden Komma

$$\langle 6 \rangle + \langle a \rangle \rightarrow a,6$$

Der Inhalt des Akkumulators wird zum Inhalt der Schnell-
speicherzelle 6 addiert.

- = Subtraktion im gleitenden Komma

$$\langle 6 \rangle - \langle a \rangle \rightarrow a,6$$

Der Inhalt des Akkumulators wird vom Inhalt der Schnell-
speicherzelle 6 subtrahiert.

X = Multiplikation im gleitenden Komma

$$\langle 6 \rangle \cdot \langle a \rangle \rightarrow a,6$$

Der Inhalt des Akkumulators wird mit dem Inhalt der Schnell-
speicherzelle 6 multipliziert.

: = Division im gleitenden Komma

$$\langle 6 \rangle : \langle a \rangle \rightarrow a,6$$

Der Inhalt des Akkumulators ist der Teiler, durch den der
Inhalt der Schnellspeicherzelle 6 dividiert wird.

M = Multiplikation mit -1 im gleitenden Komma

$$(-1) \cdot \langle a \rangle \rightarrow a,6$$

Der Inhalt des Akkumulators wird mit -1 multipliziert.

W = Quadratwurzelziehen im gleitenden Komma

$$\sqrt{\langle a \rangle} \rightarrow a,6$$

Der Inhalt des Akkumulators ist die Zahl, aus der die Quadrat-
wurzel gezogen wird.

Falls der Radikand negativ ist, also die Wurzel im Reellen
nicht gezogen werden kann, schreibt die Maschine auf einer
neuen Zeile aus

IM Em

und stoppt mit dem im Befehlsregister erscheinenden Sprung-
befehl auf Zelle 1039 (durch vorheriges Beschreiben dieser
Zelle mit einem entsprechenden Sprungbefehl und Drücken der
Starttaste kann man die Rechnung gegebenenfalls von einer
anderen als der ausgeschriebenen Adresse an weiterlaufen
lassen).

IM ist eine Abkürzung für "imaginär" und Em ist der in

Schnellspeicherzelle 5 aufbewahrte Aufrufbefehl für den auf das Unterprogramm W folgenden Befehl. Die Adresse (m-1) bezeichnet also die Programmstelle, an der ein aus dem Bereich der reellen Zahlen herausführendes Ergebnis entstanden ist.

Entsprechend schreibt die Maschine

BUEB Em

aus und stoppt mit einem Sprungbefehl auf Zelle 1039 im Befehlsregister, falls die Zahlen, die bei den arithmetischen Operationen entstehen, zu groß werden, insbesondere, wenn z.B. durch Null dividiert werden sollte.

BUEB ist eine Abkürzung für "Bereichsüberschreitung" und Em bezeichnet hier ebenfalls die Folgeadresse nach der Programmstelle, an der bei Ausführung des dort stehenden Befehls die Bereichsüberschreitung aufgetreten ist.

Ein weiterer Wurzelbefehl, der allerdings erst nach Einlesen eines Zusatzprogrammes zum Grundprogramm möglich wird, ist der Befehl W2 mit der Bedeutung

$$\sqrt{1 - \langle a \rangle^2} - a,6$$

Bei diesem Befehl werden Quadratbildung und Subtraktion in doppelter Stellenzahl ausgeführt, und das Wurzelziehen erfolgt aus dieser Zahl doppelter Länge, so daß kein Genauigkeitsverlust durch Abschneiden entsteht. Andernfalls würde z.B. der bei Quadrierung einer 9stelligen Zahl und anschließender Subtraktion entstehende 18stellige Radikand auf 9 Stellen abgerundet werden. Ein relativer Fehler von 10^{-10} würde dann nach dem Wurzelziehen einen solchen von 10^{-5} ergeben.

Auch hier wird IM und die auf den Befehl W2 folgende Stelle des Programms ausgeschrieben, wenn die Zahl im Akkumulator dem Betrag nach größer als 1 war.

An einem kleinen Beispiel soll die Anwendung der arithmetischen Befehle gezeigt werden:

Es soll der Ausdruck

$$\sqrt{\frac{p+q}{r-s}}$$

gebildet werden, wobei p, q, r, s gegebene Zahlen sind. Diese Zahlen seien in den Trommelspeicherzellen 3500 ff gespeichert. Das Programm selbst stehe in den Zellen 3600 ff.

Zellen-Nr.	Inhalt	Erläuterung
3500	78,653	p
3501	-79,831	q
3502	0,0364	r
3503	18,3869	s
3504	Zur Aufnahme von p+q bestimmt
3600	B6+3500	p → 6
3601	B3501	q → a
3602	+	p+q → a,6
3603	U3504	p+q → 3504
3604	B6+3502	r → 6
3605	B3503	s → a
3606	-	r-s → a,6
3607	B6+3504	p+q → 6, <a> = r-s → a
3608	:	$\frac{p+q}{r-s} \rightarrow a,6$
3609	W	$\sqrt{\frac{p+q}{r-s}} \rightarrow a,6$

3.6 Druck- und Tabellierbefehle

3.6.1 Druckbefehle

Die Ausgabe der errechneten Werte oder anderer Informationen erfolgt durch das Druckprogramm, das die internen dualen Worte in Dezimalzahlen, Buchstaben und Zeichen übersetzt und die entsprechenden Fernschreibzeichen an die Fernschreibmaschine bzw. auf den Schnellocher gibt. Auf dem Locher ausgestanzte Ergebnisse können später unabhängig von der Maschine auf einem Fernschreiber niedergeschrieben werden.

Das Druckprogramm, bei dem wiederum die Schnellspeicherzellen 2 bis 18 benutzt werden, wird durch den Befehl D aufgerufen.

D = Druckbefehl:

<a> → Fernschreibmaschine; <a> → a,6

Das gerade im Akkumulator stehende Wort wird übersetzt und auf dem Fernschreiber niedergeschrieben. Außerdem steht es nach der Ausgabe im Akkumulator und in Schnellspeicherzelle 6 weiter zur Verfügung.

D2 = Druckbefehl:

Wenn $\langle 1029 \rangle \geq 0$, $\langle a \rangle \rightarrow$ Fernschreibmaschine;

wenn $\langle 1029 \rangle < 0$, $\langle a \rangle \rightarrow$ Schnellocher;

$\langle a \rangle \rightarrow a,6$

Der Befehl D2 bewirkt ebenfalls die Übersetzung des im Akkumulator stehenden Wortes, jedoch erfolgt die Ausgabe in Abhängigkeit vom Inhalt der Zelle 1029 entweder auf den Fernschreiber oder auf den Schnellocher. Diese Möglichkeit ist geschaffen worden, damit man beim Ausprüfen von Programmen zunächst alle Ergebnisse sichtbar vom Fernschreiber erhalten kann, ohne die verschiedenen D2-Befehle abändern zu müssen, wenn bei der eigentlichen Rechnung die Ausgabe der Ergebnisse auf den schnelleren Locher erfolgen soll.

3.6.2 Die Ausgabe von Befehlen

Nur in ganz seltenen Fällen wird es erforderlich sein, gespeicherte Befehle ausdrucken zu lassen (z.B. für Kontrollzwecke). Der Operationsteil der Befehle wird in der internen Schreibweise ausgegeben; der Adressenteil erscheint stets in zwei Teilen, die durch ein +Zeichen getrennt sind, wobei der erste Teil die Schnellspeicheradresse, der zweite die Trommeladresse darstellt. Falls eine der beiden Adressen Null ist, wird an der entsprechenden Stelle nur 0 ausgegeben. Die vercodeten Rufbefehle +, -, X usw. werden in der internen Schreibweise als F-Befehle ausgegeben. Bei den Befehlen, die sich nur auf Schnellspeicher beziehen, kommt ein vom Leseprogramm zugesetztes K mit heraus (K=Kennzeichen, daß nur die Schnellspeicheradresse wirksam sein soll).

Eine kleine Gegenüberstellung möge den Unterschied zwischen Ein- und Ausgabeform der Befehle klar herausstellen:

Eingabeform	Ausgabeform
B1408	NA0+1408
B24	NAK24+0
B0+24	NA0+24
B24+0	NA24+0
+	FO+454
-	FO+451
X	FO+362
:	FO+425
M	FO+419

{ $\langle 24 \rangle \rightarrow a$
 { $\langle 0 \rangle \rightarrow 24$ }

Eingabeform	Ausgabeform
D	FO+512
D2	FO+514
W	FO+908
W2	FO+910

und für die hier noch nicht behandelten internen Zeichen

L...	LLR...
Ø...	AS...
I...	UA...

3.6.3 Die Ausgabe von Gleitkommazahlen

In der Maschine gespeicherte Gleitkommazahlen können sowohl als Gleitkommazahlen (natürlich in der dezimalen Schreibweise) als auch in Normalform ausgegeben werden. Die Entscheidung hierüber wird in der sogenannten Druckanordnung festgelegt (siehe 3.6.5).

A) Ausgabe als Gleitkommazahl

Die Zahlen erscheinen immer mit einer Vorzeichenstelle, auch wenn das Vorzeichen selbst (bei positiven Zahlen) nicht geschrieben wird. Unmittelbar dahinter steht das Komma. Danach folgt die Mantisse mit der in der Druckanordnung angegebenen Stellenanzahl und, durch einen Schrägstrich von ihr getrennt, der stets zweistellige Exponent mit Vorzeichen. Bei 8-stelligen Mantissen wird z.B. gedruckt:

$$\begin{array}{l} -,31416384/+01 \\ ,12000361/-13 \\ 0 \end{array}$$

Eine exakte Null wird ohne Vorzeichen und Komma an die erste Mantissenstelle geschrieben.

B) Ausgabe in Normalform

Die Zahlen werden mit der in der Druckanordnung angegebenen Stellenanzahl vor und hinter dem Komma in normaler Schreibweise ausgegeben. Die letzte Stelle nach dem Komma wird vor der Ausgabe normal gerundet. Wie oben erscheint das positive Vorzeichen nicht, während das negative direkt vor der ersten gültigen Ziffer geschrieben wird. Nicht besetzte Stellen, vor dem Komma werden durch Leerstellen berücksichtigt, so daß die Kommata der einzelnen Zahlen einer Spalte richtig untereinander zu stehen kommen. Bei 4 Stellen vor und 3 Stellen nach dem Komma wird z.B. gedruckt:

-1234,567
 31,819
 -0,007
 0,000
 0

Für die exakte Null erscheint nur eine 0 unmittelbar vor der Kommastelle. Ist die Zahl kleiner als die letzte Einheit nach dem Komma, aber nicht Null, so erscheint mit Vorzeichen eine Null vor dem Komma, und entsprechend viele Nullen stehen nach dem Komma.

3.6.4 Die Ausgabe von Strichzahlen

Strichzahlen haben bei der Ausgabe wiederum eine Vorzeichenstelle, die bei positiven Zahlen mit einer Leerstelle direkt vor der ersten gültigen Ziffer, bei negativen Zahlen durch ein Minus-Zeichen besetzt wird. Nicht besetzte führende Stellen werden durch Leerstellen berücksichtigt. Unmittelbar nach der letzten Ziffer wird ein Apostroph zur Kennzeichnung ausgegeben. Vierstellig zu druckende Strichzahlen erscheinen somit in der Form:

-1234'
 6'
 142'
 1007'

3.6.5 Druckanordnung für Zahlen

Zur Aufnahme der Angaben, in welcher Form Gleitkommazahlen gedruckt, wie die Zahlen angeordnet und mit wieviel Stellen sie niedergeschrieben werden sollen, ferner, mit welcher Stellenanzahl Strichzahlen gedruckt werden sollen, sind die Trommelspeicherzellen 1030 bis 1036 vorgesehen. Sämtliche Angaben werden als Strichzahlen eingegeben (Bei der Bestimmung der jeweiligen Stellenanzahl werden die 2 Stellen für Vorzeichen und Komma bzw. Strich nicht mitgezählt).

- 1030 l' Stellenanzahl der Mantisse, mit der eine Gleitkommazahl ausgeschrieben werden soll (bei $\langle 1036 \rangle \geq 0$)
- 1031 α' Stellenanzahl vor (1031) und nach dem Komma (1032)
- 1032 β' einer Zahl in Normalform (bei $\langle 1036 \rangle < 0$)
- 1033 k' Anzahl der Spalten einer Tabelle (nach der letzten Spalte erfolgt automatisch Wagen- und Zeilentransport)
- 1034 z' Anzahl der Spaltenzwischenraumstellen
- 1035 n' Stellenanzahl einer Strichzahl
- 1036 +1' Durch die Angabe einer positiven bzw. negativen Zahl wird entschieden, ob Gleitkommazahlen in

Gleitkommaform (<1036> positiv) oder aber in Normalform (<1036> negativ) ausgegeben werden sollen.

Die Angaben sind gemäß 2.2 bzw. 2.3 gewissen Beschränkungen unterworfen. So darf z.B. die Stellenanzahl der Mantisse einer Gleitkommazahl höchstens 9 sein:

$$1 \leq l' \leq 9.$$

Entsprechend muß die Stellenanzahl vor und nach dem Komma bei Normalformausgabe so gewählt werden, daß die Summe höchstens 9 wird:

$$\alpha' + \beta' \leq 9.$$

Hier kann β auch Null sein, wenn nämlich keine Stellen nach dem Komma ausgeschrieben werden sollen.

Die Normaldarstellung der Zahlen sollte nur gewählt werden, wenn im voraus bekannt ist, wie groß die größte auszugebende Zahl sein wird.

Hat eine Zahl mehr als die angegebenen α' Stellen vor dem Komma, so gibt die Maschine die Zahlen automatisch im Gleitkomma mit der in 1030 notierten Stellenzahl aus. Deshalb sollten die Angaben für die Druckanordnung in den Zellen 1030 bis 1036 stets vollzählig gemacht werden.

Analog ist die Stellenzahl für Strichzahlen auf maximal 12 beschränkt;

$$1 \leq n' \leq 12.$$

Falls die auszugebenden Strichzahlen mehr als n' Stellen besitzen, so werden sie mit der maximal möglichen Stellenzahl, nämlich 12stellig, herausgeschrieben.

Die Anzahl der Spalten und Zwischenräume ist beliebig, nur ist zu beachten, daß auf einer Zeile des Fernschreibers maximal 104 Anschläge gemacht werden können und die Zahlen in Gleitkommadarstellung mit $l'+6$ Anschlägen, die Zahlen in Normalform mit $\alpha'+\beta'+2$ Anschlägen und die Strichzahlen mit $n'+2$ Anschlägen auf dem Papier erscheinen.

Es sei noch erwähnt, daß die Anzahl der Zahlenstellen und der Zwischenräume von Spalte zu Spalte variiert werden kann. Man hat dazu im Programm jeweils vor den Druckbefehlen die Angaben in den Zellen 1030 - 1036 entsprechend zu ändern.

3.6.6 Die Ausgabe von Klartext

Da Überschriften und sonstige Textangaben meist mehr als 7 Zeichen umfassen, also in mehrere aufeinanderfolgende Zellen

eingelezen werden, ist bei der Ausgabe von Klartext vorgesehen, daß man nur die erste Zelle des Klartextes durch einen zu programmierenden Befehl in den Akkumulator zu bringen braucht. Durch das Druckprogramm werden dann laufend die weiteren Zellen geholt und ausgegeben. Dieses geschieht so lange, bis aus einer Zelle ein Wort geholt wird, das kein Klartext mehr ist, z.B. die Zahl Null (die man zu diesem Zweck bei der Eingabe dorthin speichert).

Der Klartext wird bei der Ausgabe am Anfang und am Ende mit dem Kennzeichen \mathfrak{R} (kein Anschlag auf dem Fernschreiber, aber eine Lochung) oder, bei Algol-Fernschreibern, mit einem Semikolon versehen.

Mit Hilfe des Befehls F885 ist es jedoch möglich, den Inhalt des Akkumulators als Klartext ohne Klingelzeichen bzw. Semikolon auf den Fernschreiber ($\langle 2 \rangle \geq 0$) oder den Schnellocher ($\langle 2 \rangle < 0$) auszugeben. Nach der Ausgabe steht eine Null im Akkumulator.

Eine Spaltenzählung erfolgt beim Klartext nicht. Zwischenräume sowie Wagen- und Zeilentransport innerhalb des Textes werden berücksichtigt. Die Ausgabe des Klartextes ist allerdings nur möglich, wenn Klartext und Druckbefehl auf der Trommel stehen, also nicht, wenn sich der betreffende Programmteil im Kernspeicher befindet.

3.6.7 Tabellierbefehle

Für das Ausdrucken von Tabellen gibt es noch einige spezielle Tabellierbefehle:

F1000 Wagen- und Zeilentransport auf Fernschreiber; d.h., der Wagen der Fernschreibmaschine wird an den Anfang der nächsten Zeile gerückt.

F1002 Wagen- und Zeilentransport

bei $\langle 1029 \rangle \geq 0$ auf Fernschreiber

bei $\langle 1029 \rangle < 0$ auf Schnellocher

Der Befehl braucht nur gegeben zu werden, wenn eine Zeile nicht voll beschrieben werden soll oder wenn ein doppelter Zeilenabstand vorgesehen ist.

F800 Spaltensprung auf Fernschreiber; d.h. Weiterrücken um so viele Zeichenbreiten, daß der Wagen der Fernschreibmaschine am Anfang der nächsten Spalte steht.

F802 Spaltensprung

bei $\langle 1029 \rangle \geq 0$ auf Fernschreiber,

bei $\langle 1029 \rangle < 0$ auf Schnellocher.

Der im Akkumulator stehende Wert bleibt bei allen diesen Befehlen erhalten und wird gleichzeitig in Schnellspeicherzelle 6 notiert.

Mit den folgenden Befehlen kann man noch um beliebig viele Zeichenbreiten ein- oder weiterrücken und einen mehrfachen Zeilenvorschub bewirken, wiederum auch in Abhängigkeit von $\langle 1029 \rangle$ (z.B. zur Verteilung der ausgeschriebenen Ergebnisse auf einzelne Seiten DIN A4 mit entsprechenden freigelassenen oberen und unteren Rändern).

F840 Weiterrücken auf dem Fernschreiber um die im Akkumulator durch eine negative Strichzahl angegebene Anzahl von Zeichenbreiten.

F842 Weiterrücken

bei $\langle 1029 \rangle \geq 0$ auf Fernschreiber,

bei $\langle 1029 \rangle < 0$ auf Schnellocher.

Die Ausführung des Befehls F840 bzw. F842 wird bei der Spaltenzählung nicht berücksichtigt. Es ist daher in manchen Fällen erforderlich, durch weitere Tabellierbefehle zu vermeiden, daß die Spalten versetzt und die betreffenden Zeilen länger ausgeschrieben werden.

F850 Das Papier im Fernschreiber wird um so viele Zeilen weitertransportiert, wie die negative Strichzahl im Akkumulator angibt.

F852 Zeilenvorschub

bei $\langle 1029 \rangle \geq 0$ auf Fernschreiber,

bei $\langle 1029 \rangle < 0$ auf Schnellocher.

3.7 Bedingte Befehle

Alle Befehle der ZUSE Z 23 können mit Bedingungszeichen versehen werden, d.h., die Befehle sollen nur dann ausgeführt werden, wenn die abgefragte Bedingung erfüllt ist. Ist die Bedingung nicht erfüllt, so wird ohne Ausführung des Befehls auf die folgende Speicherzelle übergegangen.

Als Bedingungen können gestellt werden:

PP . . . Führe den Befehl . . . nur aus, wenn der Inhalt des Akkumulators positiv ist (Null mitgerechnet) ($\langle a \rangle \geq 0$).

QQ . . . Führe den Befehl . . . nur aus, wenn der Inhalt des Akkumulators negativ ist (Null ausgeschlossen) ($\langle a \rangle < 0$).

Die Kombination der beiden Zeichen ergibt die Nullabfrage:

PPQQ ... Führe den Befehl . . . nur aus, wenn im Akkumulator eine Null steht ($\langle a \rangle = 0$).

Als weitere Bedingungszeichen, die allerdings hauptsächlich im Grundprogramm benutzt werden (da die getesteten Schnellspeicherzellen beim Grundprogramm belegt werden), gibt es:

P . . . Führe den Befehl . . . nur aus, wenn der Inhalt der Schnellspeicherzelle 2 positiv ist ($\langle 2 \rangle \geq 0$).

Q . . . Führe den Befehl . . . nur aus, wenn der Inhalt der Schnellspeicherzelle 2 negativ ist ($\langle 2 \rangle < 0$).

Y . . . Führe den Befehl . . . nur aus, wenn die letzte Stelle von Schnellspeicherzelle 3 mit einer '1' besetzt ist ($\langle 3 \rangle_{40} = 1$).

Ferner verwendet man die Kombination:

PQQQ ... Führe den Befehl . . . nur aus, wenn die ersten beiden Stellen des Akkumulators nicht übereinstimmen (d.h.wenn im Akkumulator ein Befehl oder Klartext steht; $\langle a \rangle_1 \neq \langle a \rangle_2$).

In den obengenannten Fällen wird die Ausführung irgendeines Befehls vom Inhalt einer der Schnellspeicherzellen 2,3 oder 4 abhängig gemacht (es können auch 2 oder 3 Bedingungen in einem Befehl gestellt werden). Die übrigen Schnellspeicherzellen, von Adresse 6 an, können ebenfalls auf ihren positiven oder negativen Inhalt abgefragt werden, allerdings nur im Rahmen eines Doppeladrese-Sprungbefehls und nur zum Sprung auf eine Trommelzelle:

PEs+t Führe den Sprung auf die Trommelzelle t nur aus, wenn in der Schnellspeicherzelle s etwas Positives (einschließlich Null) steht ($\langle s \rangle \geq 0$). Also:

Wenn $\langle s \rangle \geq 0$, $\langle t \rangle \rightarrow b$, $E(t+1) \rightarrow c$; wenn $\langle s \rangle < 0$, nächster Befehl.

QEs+t Führe den Sprung auf die Trommelzelle t nur aus, wenn der Inhalt der Schnellspeicherzelle s negativ ist ($\langle s \rangle < 0$). Also:

Wenn $\langle s \rangle < 0$, $\langle t \rangle \rightarrow b$, $E(t+1) \rightarrow c$; wenn $\langle s \rangle \geq 0$, nächster Befehl.

Hier ist zu beachten, daß s die Adresse der auf ihren Inhalt zu testenden Schnellspeicherzelle ist, während t die Trommeladresse bezeichnet, die bei erfüllter Bedingung durch E ausgewählt werden soll. Zu einem Sprung auf eine Schnellspeicherzelle sind diese Befehle nicht verwendbar, da in diesem Fall ein K hinzugesetzt werden müßte und durch die so entstehenden Befehle PEKs+t, QEKs+t wiederum der Inhalt von 2 getestet würde. Dabei bliebe die Adresse t ebenso wie in dem

Sprungbefehl EKs+t wirkungslos; sie wird allerdings bei der Übernahme ins Befehlszählregister erhöht, was für bestimmte Zwecke wünschenswert sein kann: EKs+(t+1) → c.

Bedingungsschalter

Am Bedienungspult der ZUSE Z 23 (siehe Abb. im Anhang) sind 5 Tasten angebracht, die die einzelnen Bitstellen der Schnellspeicherzelle 14 repräsentieren (diese Zelle hat nur eine Wortlänge von 5 bit). Durch Betätigen der einzelnen Tasten, also durch additive Kombination der Stellenwerte 1', 2', 4', 8' oder 16', kann man diese Zelle auch während der Rechnung mit den Zahlen 1 - 31 füllen bzw. durch Herausnahme aller Tasten auf Null setzen. Man hat dadurch die Möglichkeit, den Rechenablauf an den Stellen, an denen das Programm auf Adresse 14 Bezug nimmt, von außen her zu beeinflussen. Zelle 14 kann vom Programm her nicht beschrieben werden.

Akkumulatortastatur

Daneben besteht noch die Möglichkeit, über die Akkumulatortastatur ein ganzes Wort in den Akkumulator zu bringen. Man kann z.B. während der Rechnung eine Zahl eintasten, die in den Akkumulator geholt wird, sobald im Programmablauf der folgende Befehl erscheint:

U4 Übernahme des in der Akkumulatortastatur
 stehenden Wortes in den Akkumulator.

3.8 Stopbefehle

Als Stopzeichen wirkt das Befehlszeichen Z, das jedem Befehl beigelegt werden kann; z.B. ZUs+t, ZB14, etc.. Befehle, die mit dem Stopzeichen versehen sind, bewirken einen Stop der Maschine. Gleichzeitig erscheint der betreffende Befehl im Befehlsregister. Er wird erst nach erneuter Betätigung der Starttaste ausgeführt; das Programm läuft anschließend weiter. In der Zwischenzeit kann man gegebenenfalls vom Bedienungspult aus eine Veränderung vornehmen und dadurch den weiteren Rechenablauf beeinflussen. Alle Z-Befehle, die keinen Arbeitsbefehl enthalten, bewirken beim erneuten Starten einen Sprung auf die angegebene Adresse (das E braucht also nicht geschrieben zu werden; Zm = ZEm).

Im allgemeinen wird am Ende eines Programms der Stopbefehl

Zo stehen: Stop der Maschine

Dieser Befehl kann auch unter Bedingungen gestellt werden; dann tritt erst ein Stop ein, wenn die angegebene Bedingung erfüllt ist. Z.B.:

PPZo Wenn der Akkumulatorinhalt positiv ist ($\langle a \rangle \geq 0$),
 soll ein Stop eintreten. Im anderen Fall wird
 auf die nächste Speicherzelle übergegangen.

QZs+t Wenn $\langle s \rangle < 0$, soll gestoppt und beim Starten
der Befehl Et ausgeführt werden.

Bedingter Stop

Eine Taste am Bedienungspult, die mit "Bed.Stop" (bedingter Stop) bezeichnet ist, bewirkt, daß die Maschine stehen bleibt (Taste eingedrückt), wenn eine Zahl ins Befehlsregister kommt. Man kann dies benutzen, um an verschiedenen Stellen einen Stop in das Programm einzubauen. Zur Unterscheidung der einzelnen Stellen verwendet man hierbei als Zahl jeweils die Adresse der nächsten aufzurufenden Speicherzelle (in die Zelle 4181 bringt man z.B. die Zahl 4182'). An diesen Stellen kann man entweder mit "Weiter" und "Start" fortfahren oder durch eine entsprechende Eintastung im Befehlsregister zu einem anderen Programmteil übergehen. Ist die Taste "Bed.Stop" nicht eingedrückt, werden diese so markierten Stellen überlaufen.

Die Verwendung der Taste "Bed.Stop" ist auch fürs Programmtesten von Bedeutung. Sobald nämlich das Programm Zahlen anstelle von Befehlen ins Befehlsregister bringt, bleibt dann die Maschine stehen.

3.9 Bandbefehle

Bisher wurden nur einzelne Befehle oder kleine Programmteile besprochen, die aus einem größeren Rahmen herausgerissen waren, und stets wurde vorausgesetzt, daß die Befehle und Zahlenwerte schon in den Speicherzellen standen.

Hier folgt nun die Behandlung der sogenannten Bandbefehle, mit denen der Maschine gewisse Angaben gemacht werden; nämlich erstens, wohin die einzelnen Befehle und Zahlen, die auf dem Lochstreifen stehen, gespeichert werden sollen, und zweitens, an welcher Stelle des Programms die Rechnung beginnen, also von welcher Speicherzelle ab beim Starten der Programmablauf e. folgen soll.

Diese Bandbefehle stehen nur auf dem Lochband und enthalten Anweisungen für das Leseprogramm, ohne aber selbst in Speicherzellen gebracht zu werden.

UmU Dieser Bandbefehl bewirkt, daß die folgenden Worte auf dem Lochstreifen in die Speicherzellen m , $m+1$, $m+2$, . . . usw. eingespeichert werden. Durch das Bandzeichen U wird der Befehl Um in Schnellspeicherzelle 19 notiert. Sobald das nach m zu speichernde Wort im Akkumulator aufgebaut ist, wird der Befehl ausgeführt, und die Adresse m des Befehls wird anschließend um 1 erhöht.

Z.B. soll in die Speicherzellen 1030 ff die Druckanordnung eingegeben werden:

6 Stellen der Mantisse bei Zahlen in Gleitkommaform

3 Stellen vor dem Komma	}	bei Zahlen in Normalform
4 Stellen nach dem Komma		
3 Spalten		
2 Zwischenräume zwischen den Spalten		
4 Stellen bei Strichzahlen		
Ausgabe in Normalform		

Auf dem Lochstreifen müssen dann folgende Angaben gemacht werden (vergleiche 3.6.5):

```

U1030U
 6'
 3'
 4'
 3'
 2'
 4'
-1'

```

Falls darauf wieder ein UmU folgt, so wird von der nun neu bestimmten Stelle m an das Weitere gespeichert.

EmE Dieser Bandbefehl, der am Ende des Lochstreifens gegeben wird, wenn das ganze Programm und alle Daten eingelesen sind, bewirkt, daß die Maschine mit dem Befehl in Speicherzelle m anfängt zu rechnen. Durch das Bandzeichen E wird der davorstehende Befehl ohne Speicherung sofort ausgeführt.

Häufig ist es zweckmäßig, den Startbefehl noch mit einem Stopzeichen Z zu versehen.

EZmE Dann stoppt die Maschine mit dem Sprungbefehl auf Zelle m im Befehlsregister. Man kann jetzt nochmals überprüfen, ob alle Lochstreifen eingegeben sowie die Schalter alle richtig gestellt sind, und danach mit der Starttaste das Programm beginnen.

Bei Befehlen mit Bandzeichen E hat Schnellspeicherzelle 26 die Funktion eines Hilfsakkumulators, da der Hauptakkumulator beim Einlesen ständig benötigt wird. Es ist also darauf zu achten, daß ein Befehl, der normalerweise auf den Inhalt des Akkumulators bezugnimmt, sich als Bandbefehl auf Schnellspeicherzelle 26 bezieht. Um die unterschiedliche Wirkung eines Befehls mit Bandzeichen E gegenüber der des entsprechenden gewöhnlichen Befehls herauszustellen, seien hier zwei Beispiele angeführt:

Bandbefehl			gewöhnlicher Befehl		
B19E	<19>	→	26	B19	<19> → a
U218E	<26>	→	218,	U218	<a> → 218,
	<26>	→	26		<a> → a

Eine weitere Bandbefehlsart wird im Kapitel 8 besprochen werden.

4. Programmierungsbeispiel

4.1 Vorbemerkung über die Programmierung

Die Programmierung eines Problems (Vorbereitung für die Rechenanlage) läßt sich in vier charakteristische Teilaufgaben zergliedern. Die erste Aufgabe, die vielfach die schwierigste und umfangreichste ist, besteht darin, das Problem überhaupt erst einmal mathematisch zu formulieren.

Hat man eine solche Formulierung gefunden, ist es vielfach notwendig, das günstigste Näherungsverfahren auszuwählen, mit dem man das Problem behandeln will. Wenn z.B. eine Differentialgleichung zu lösen ist, entsteht die Frage: Verwendet man das Runge-Kutta-Verfahren, das Adams'sche Interpolationsverfahren oder ein anderes? Die Wahl des Näherungsverfahrens richtet sich nach der Art des Problems.

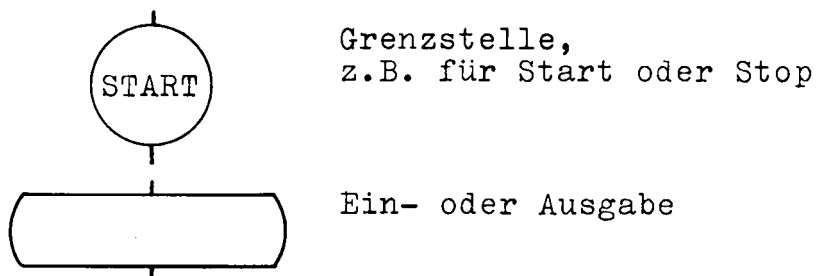
Als dritter Punkt käme dann das Zeichnen eines Flußdiagramms, das bildlich den Ablauf des Rechengvorgangs, den Zusammenhang der einzelnen Abschnitte der Rechnung, darstellt. Häufig ist es erforderlich, auch für die verwendeten Unterprogramme weitere Flußdiagramme anzufertigen.

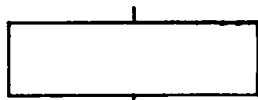
Zuletzt ist schließlich die Vercodung des im Flußdiagramm dargestellten Programmablaufs vorzunehmen, d.h. also das Programm im Formelcode der Rechenanlage aufzustellen. Nur diese letzte Teilaufgabe der Programmierung ist maschinengebunden, da nur hier die maschineneigene Programmierungssprache verwendet wird.

Als Beispiel soll bewußt eine nicht zu umfangreiche Aufgabe programmiert werden, bei der die ersten beiden Teilaufgaben schon erledigt sein sollen. Vorher soll jedoch eine kurze Beschreibung der Symbolik erfolgen, deren Kenntnis zur Anfertigung von Flußdiagrammen erforderlich ist.

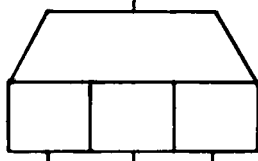
4.2 Symbolik für Flußdiagramme

Für die verschiedenen Teile eines Flußdiagramms und ihre Verbindung untereinander wird die Verwendung der folgenden, entsprechend zu beschriftenden Symbole empfohlen. Ferner soll die Konvention gelten, daß jedes Symbol-Kästchen nur einen Eingang besitzt und seitliche Ein- oder Ausgänge möglichst zu vermeiden sind.

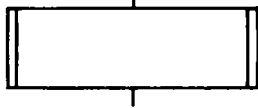




Unverzweigte (weil unbedingte) Operation



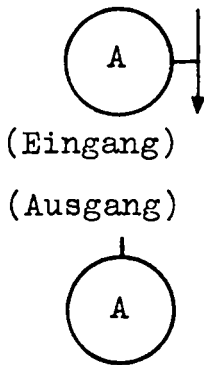
Verzweigte (weil bedingte) Operation.
Die Anzahl der benötigten Rechtecke (Ausgänge) entspricht der Anzahl der Entscheidungsmöglichkeiten.



Unter- oder Bibliotheksprogramm

Die Verbindung der Symbole durch Volllinien gibt die Folgerichtung der einzelnen Schritte an. Gegebenenfalls kann zur Verdeutlichung der Ablaufrichtung ein Pfeil verwendet werden. Bei der Zusammenführung zweier Linien wird die gemeinsame Richtung mit einem Pfeil versehen.

Die Vorzugsrichtungen aller Linien sind von oben nach unten und von links nach rechts.



Oft ist es aufgrund der räumlichen Anordnung nicht möglich, die Verbindungslinie zwischen zwei Symbolen (oder die Linien von mehreren Symbolen zu einer bestimmten Stelle) auszuziehen. In solchen Fällen verbindet man den betreffenden Ausgang (oder die betreffenden Ausgänge) mit einem Konnektor, der die gleiche Inschrift trägt, wie ein vor dem Eingang ebenfalls anzubringender Konnektor.

4.3 Formulierung der Programmierungsaufgabe

Es soll eine Funktion $F(x)$ tabelliert werden, die durch den Quotienten zweier Polynome 4. Grades dargestellt ist.

$$F(x) = \frac{a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0}{b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0} = \frac{P_4(a, x)}{P_4(b, x)}$$

Die Tabellierung soll für das Intervall von $x = 0$ bis $x = 3$ mit der Schrittweite $\Delta x = 0,125$ durchgeführt werden.

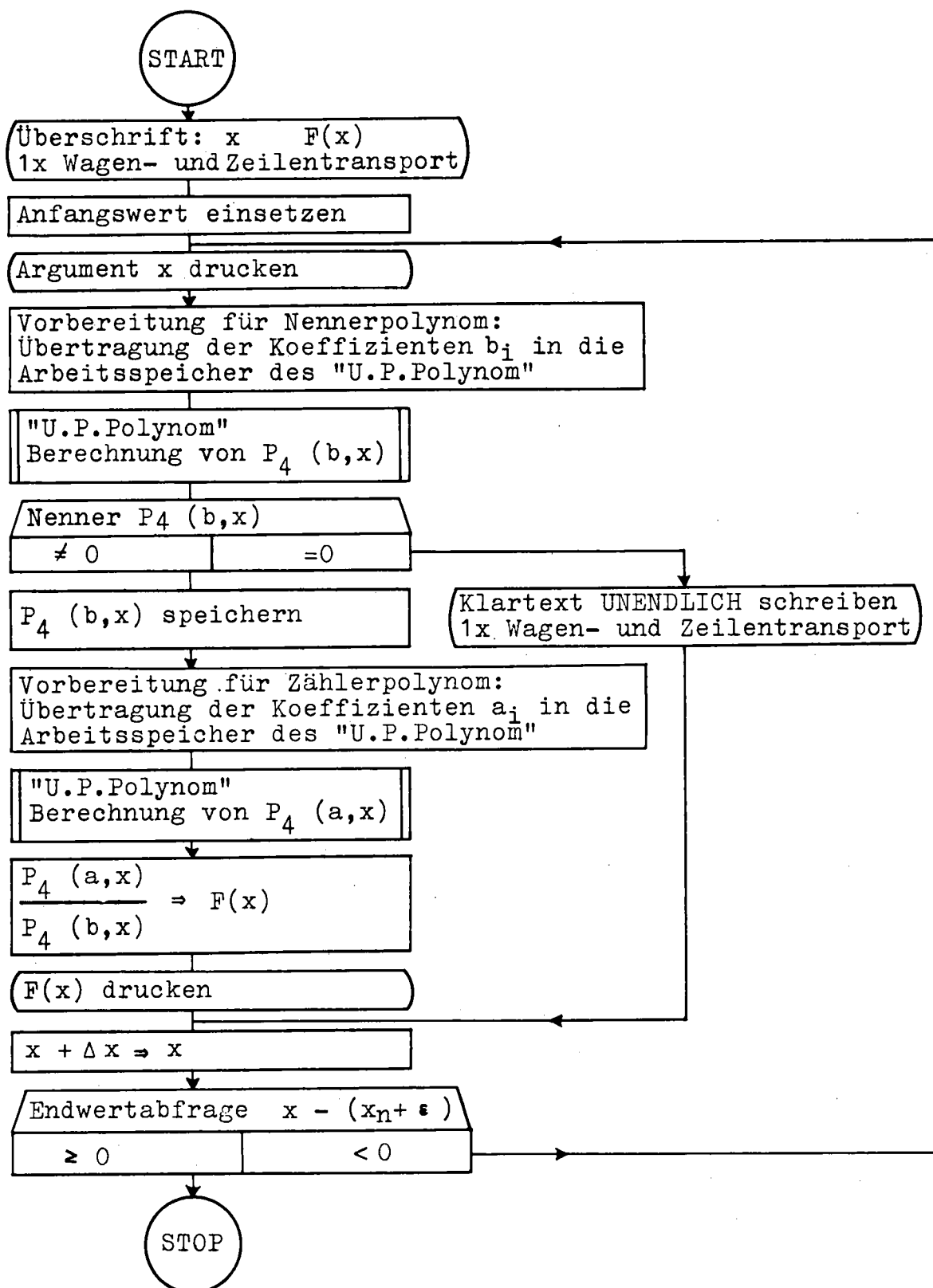
Niedergeschrieben werden sollen jeweils das Argument und der Funktionswert nebeneinander unter einer Überschrift in Klartext:

X	F(X)
---	------

Im Zähler und im Nenner der Funktion tritt je ein Polynom 4. Grades auf. Die beiden Polynome unterscheiden sich nur durch ihre Koeffizienten. Infolgedessen wird man hier ein

Unterprogramm Polynom aufstellen, für das vom Hauptprogramm nur die Parameter (hier also die Koeffizienten) an bestimmter Stelle bereitgestellt werden.

4.4 Das Flußdiagramm zum Hauptprogramm

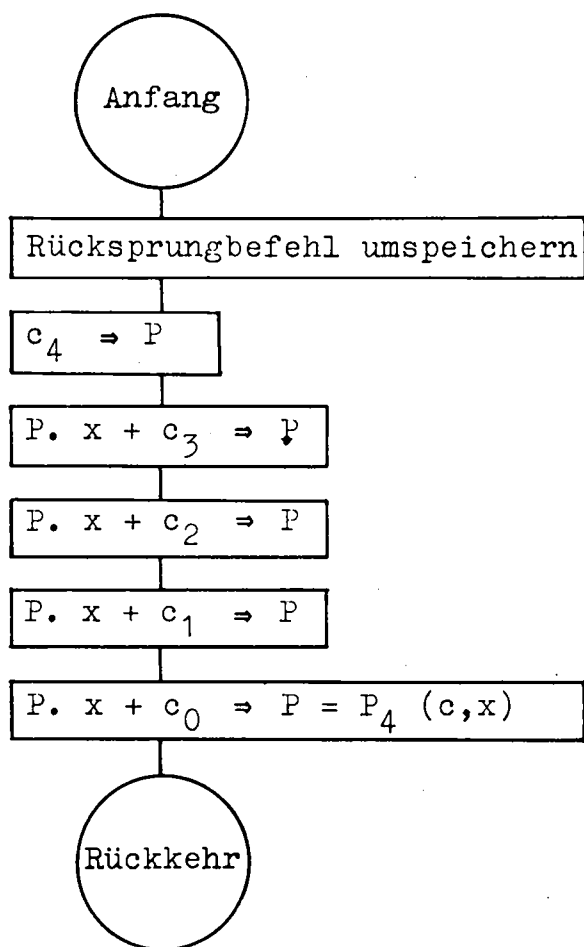


Hier mußte die Abfrage erfolgen, ob der Nenner Null ist, da sonst die Division Bereichsüberschreitung bringt und die Maschine stoppen läßt. Nach dem Ausschreiben von Klartext wird stets einmal Befehl für Wagen- und Zeilentransport gegeben, da Klartext bei der Spaltenzählung nicht mitgezählt wird und somit kein automatischer Übergang auf die nächste Zeile erfolgt (vergl. 3.5 "BUEB" und 3.6.6).

Für das "U.P. Polynom" wird ein gesondertes Flußdiagramm aufgestellt. Um mit möglichst wenig Operationen auszukommen, wird das Polynom nach dem Horner'schen Schema berechnet:

$$P_4(c, x) = c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0 = (((c_4 x + c_3) x + c_2) x + c_1) x + c_0$$

4.5 Das Flußdiagramm zum Unterprogramm



4.6 Das Unterprogramm

Als erstes soll das Unterprogramm Polynom aufgestellt werden. Und zwar soll es im Kernspeicher in den Zellen 200 ff untergebracht werden. In Kernspeicherzelle 190 soll das Argument x, in den Zellen 191 bis 195 sollen die Koeffizienten c_4, c_3, \dots, c_0 verfügbar sein, was vom Hauptprogramm erledigt wird.

Speicherliste für das Unterprogramm

Zellen-Nr.	Inhalt	Erläuterungen
190	$x_i \equiv x$	Das jeweilige Argument x_i wird vom Hauptprogramm geliefert.
1	c_4	Die c_i stehen hier stellvertretend für die entsprechenden a_i bzw. b_i , deren Zahlenwerte vom Hauptprogramm zugeführt werden.
2	c_3	
3	c_2	
4	c_1	
5	c_0	
200 ff		Unterprogramm

Unterprogramm

200	B5	Rücksprungbefehl umspeichern (in die letzte Zelle des Unterprogramms)
1	U220	
2	B191	
3	U6	$c_4 \Rightarrow P \rightarrow 6$
4	B190	$P \cdot x + c_3 \Rightarrow P \rightarrow a,6$
5	X	
6	B192	
7	+	
8	B190	$P \cdot x + c_2 \Rightarrow P \rightarrow a,6$
9	X	
210	B193	
1	+	
2	B190	$P \cdot x + c_1 \Rightarrow P \rightarrow a,6$
3	X	
4	B194	
5	+	
6	B190	$P \cdot x + c_0 \Rightarrow P = P_4(c, x) \rightarrow a,6$
7	X	
8	B195	
9	+	
220	Eo	Rücksprungbefehlszelle. Der Inhalt Eo wird nach Ausführung des in Zelle 201 stehenden Befehls U220 durch den Rücksprungbefehl ersetzt.

4.7 Das Hauptprogramm

Das Hauptprogramm soll auf der Trommel in den Speicherzellen ab 1800 stehen, während die Koeffizienten a_i ab 1790, die Koeffizienten b_i ab 1780 eingespeichert seien, die Überschrift ab 1750, der Text UNENDLICH ab 1770. Der Anfangswert x_0 , die Schrittweite Δx und der Endwert x_n sollen in den Zellen 1795 ff stehen.

Speicherliste für das Hauptprogramm

1750 ff	X	F(X)		Wieviel Zwischenräume im Klartext gegeben werden müssen, muß man an Hand der Form, in der ausgeschrieben werden soll, auszählen, damit die Bezeichnung über den entsprechenden Spalten steht.
	0	(siehe 3.6.6)		
1770 ff	UNENDLICH			Für je 7 Fernschreibzeichen muß eine Zelle freigehalten werden. Man braucht nicht genau abzuzählen, wieviele Speicherzellen tatsächlich belegt werden, nur muß man genügend Plätze vorsehen, damit kein unerwünschtes Überschreiben erfolgen kann.
	0	(siehe 3.6.6)		
1780	1		b_4	Die Koeffizienten für das Nennerpolynom
1	5,5		b_3	
2	3,25		b_2	
3	-6,375		b_1	
4	-3,375		b_0	
1790	1,65		a_4	Die Koeffizienten für das Zählerpolynom
1	3,135		a_3	
2	-10,824		a_2	
3	-17,127		a_1	
4	-2,97		a_0	
1795	0		x_0	
6	0,125		Δx	
7	3,01		$x_n + \epsilon$	Zur Abfrage, ob das Ende erreicht ist, verwendet man einen Wert, der etwas größer ist als das letzte x , für das die Berechnung noch vorgenommen werden soll.
1800 ff				Programm

Hauptprogramm

1800	B1750	}	Überschrift schreiben (ab Zelle 1750 gespeichert)
1	D		
2	F1000		1 x Wagen- und Zeilentransport
3	B190+1795		$x_0 \Rightarrow x \rightarrow 190$ (Anfangswert einsetzen)
4	B190	}	Argument x drucken
5	D		
6	CKB13+4	}	Vorbereitung für Nennerpolynom:
7	BV191+1780		Blocktransfer $b_i \rightarrow 191$ ff
8	F200		Aufruf des U.P.Polynom; das U.P. bringt $P_4(b,x)$ nach a und 6
9	PPQQE1825		Wenn $P_4(b,x) = 0$, Sprung auf Zelle ⁴ 1825
1810	U189		$P_4(b,x)$ speichern
1	CKB13+4	}	Vorbereitung für Zählerpolynom:
2	BV191+1790		Blocktransfer $a_i \rightarrow 191$ ff
3	F200		U.P.Polynom; $P_4(a,x) \rightarrow a,6$
4	B189	}	$\frac{P_4(a,x)}{P_4(b,x)} \Rightarrow F(x) \rightarrow a,6$
5	:		
6	D		F(x) drucken
7	B6+1796	}	$x + \Delta x \Rightarrow x \rightarrow 190$
8	B190		
9	+		
1820	U190		
1	B1797	}	Endwertabfrage $x - (x_n + \varepsilon)$;
2	-		wenn < 0 , Sprung nach 1804
3	QQE1804		
4	Zo		Stop
5	B1770	}	Klartext UNENDLICH schreiben
6	D		
7	F1000		1 x Wagen- und Zeilentransport
8	E1817		Sprung nach 1817

Weiter muß der Maschine noch eingegeben werden, in welcher Form die Ergebnisse niedergeschrieben werden sollen (Druckanordnung); diese Angaben gehören in die Speicherzellen 1030 ff (siehe 3.6.5). Es soll in Normalform herausgeschrieben werden mit 2 Stellen vor und 6 Stellen nach dem Komma.

Für das Argument x und den Funktionswert $F(x)$ sind je 1 Spalte vorgesehen, die durch 3 Zwischenräume voneinander getrennt sein sollen. Die Druckanordnung ist daher folgende:

1030	7'	Falls Zahlenwerte drei- oder mehrstellig werden, soll in Gleitkommadarstellung ausgeschrieben werden mit 7 Stellen Mantisse.
1	2'	Stellenzahl bei Normalform vor dem Komma
2	6'	Stellenzahl bei Normalform nach dem Komma
3	2'	2 Spalten
4	3'	3 Zwischenräume
5	4'	für Strichzahlen (hier unnötig)
6	-1'	Kennzeichen für Ausgabe in Normalform

Damit ist das Programm vollständig aufgestellt und muß nun in einen Lochstreifen gestanzt werden, der in die Maschine eingelesen werden kann.

5. Herstellung und Verarbeitung des Programmstreifens

5.1 Die Lochstreifenherstellung

Die Lochstreifenherstellung erfolgt auf dem Programmierungstisch, der aus einem Schaltkästchen, einem Fernschreiber mit angebautem Locher, einem Abtaster und einem Schrittstopgerät besteht.

Mit dem Schaltkästchen wird die ganze Anlage ein- (linker Knopf) bzw. ausgeschaltet (rechter Knopf). Auf dem Fernschreiber werden dann die Programme und Daten geschrieben, wobei ein Protokoll und ein zugehöriger Lochstreifen entstehen.

Zum Kopieren oder Ausbessern von Programmen benutzt man den Abtaster und das Schrittstopgerät. Der zu kopierende Lochstreifen wird in den Abtaster eingelegt und jeweils bis in die Nähe einer auszubessernden oder zu ändernden Stelle mit der vollen Kopiergeschwindigkeit durchlaufen gelassen. Die letzten Zeichen vor der Stelle werden mit Hilfe des Schrittstopgerätes einzeln kopiert, und anschließend werden die neuen Zeichen auf dem Fernschreiber eingetastet. Der ursprüngliche Lochstreifen wird danach im Abtaster um so viele Zeichen weitergelegt, daß wieder fortlaufend weiterkopiert werden kann.

Die Tastatur des Fernschreibers umfaßt:

1. Das Alphabet

Das Fernschreiber-Alphabet enthält nur die großen Druckbuchstaben. Die Tasten können nur betätigt werden, wenn vorher

einmal die Buchstabentaste (Bezeichnung A...) gegeben wurde.

2. Ziffern und Zeichen

Es sind die 10 Ziffern 0 bis 9, und die Zeichen

+ - : / , . ? ' { } = $\hat{\mu}$ bzw. beim Algol-Fernschreiber
+ - : / , . \times ' { } = ; [] ₁₀ vorhanden.

Diese Tasten können nur nach Drücken der Zifferntaste (Bezeichnung 1...) bedient werden.

3. Die Betriebszeichen für die Fernschreibmaschine

Buchstabentaste	(kurz Bu)
Zifferntaste	(kurz Zi)
Wagenrücklauf	Auf der Taste bezeichnet mit < (kurz Wr)
Zeilentransport	Auf der Taste bezeichnet mit = (kurz Zl)
Zwischenraum	Unbezeichnete große Taste unten in der Mitte (kurz Zw)
Leer	Auf der Taste bezeichnet mit # bzw. * (beim Algol-Fernschreiber mit \odot); # bewirkt im Anschluß an Buchstaben und * nach dem Schreiben von Ziffern (\odot in beiden Fällen) die Lochung des Transportloches. Im Gegensatz zu den übrigen Betriebszeichen, die eine spezielle Lochung ergeben, bewirkt "Leer" nur ein Weiterücken des Lochstreifens, jedoch keine Veränderung an der Fernschreibmaschine ("Zw" z.B. erzeugt eine Lochung und ein Weiter- rücken des Wagens).

Alle Betriebszeichen (Ausnahme siehe bei "Leer") können stets betätigt werden, gleichgültig, ob die Fernschreibmaschine gerade auf Buchstaben oder Ziffern eingestellt ist.

Eine Wiederholungstaste gestattet die fortlaufende Ausführung des zuletzt betätigten Zeichens; auf diese Weise kann man sich z.B. einen Leerstreifen beliebiger Länge herstellen. Weitere Drucktasten ermöglichen den schrittweisen Rücktransport und die Aus- bzw. Einschaltung des Lochers.

Bei der Lochstreifenherstellung ist zu beachten, daß die Betriebszeichen sowie einige Buchstaben- und Ziffernzeichen für das Leseprogramm eine besondere Bedeutung haben:

Im Operationsteil eines Befehls

- leer, Wr, Zw, Zl, sind wirkungslos
(nach X, M, W, D, sind Wr, Zw, oder Zl Schlußzeichen).
- Zi Schlußzeichen des Operationsteils; Übergang zum Einlesen des Adressenteils.
- Bu Irrungszeichen; der bisher eingelesene Operationsteil

des angefangenen Wortes wird gelöscht (Bu bewirkt eine Lochung über die ganze Breite des Lochstreifens; eine falsche Lochung kann daher auch mit "Bu" überschrieben werden).

- d) Die nicht benutzten Buchstaben J und T führen aus dem Leseprogramm heraus:
J auf die Zelle 1040,
T auf die Zelle 1041.

Im Adressenteil eines Befehls

- a) Zi, leer sind wirkungslos.
b) Wr, Zw oder Zl sind Schlußzeichen.
c) Bu führt auf den Teil des Leseprogramms, in dem Bandbefehle eingelesen werden.
Ein doppeltes Bu bedeutet Irrung; der bereits eingelesene Teil des angefangenen Wortes (Operationsteil und Adressenteil) wird gelöscht.
d) Die nicht benutzten Zeichen führen aus dem Leseprogramm, und zwar:

⊛	auf die Zelle 1042,
(" " " 1043,
)	" " " 1044,
?	" " " 1045,
.	" " " 1046,
=	" " " 1047,
[" " " 1049,
10	" " " 1050,
]	" " " 1051.

Bei Zahlen

- a) Zi, leer sind wirkungslos.
b) Wr, Zw oder Zl sind Schlußzeichen.
c) Ein- oder mehrfaches Bu bedeutet Irrung; alle Zeichen der angefangenen Zahl werden gelöscht.

Bei Strichzahlen gilt das Apostroph als Schlußzeichen.

Zwischen zwei Worten können beliebig viele Betriebszeichen der Fernschreibmaschine gegeben werden.

Beim Lochen des Beispiels gibt man nun vor jedem Programmabschnitt bzw. jeder Zahlengruppe einen UmU-Bandbefehl, der angibt, wohin die folgenden Worte gespeichert werden sollen (auf der Trommel ab Zelle 1052 möglich). Der Programmtext wird auf dem Fernschreiber neben- oder auch untereinander geschrieben. Zwischen den einzelnen Abschnitten und am Anfang und Ende wird mehrfach "Leer" gegeben, um auf dem Lochstreifen Stellen zu erhalten, auf denen Beschriftungen angebracht werden können. Am Anfang ist ca. 15 cm Leerstreifen notwendig, damit man den Streifen in den Abtaster einlegen kann.

Bei größeren Programmen oder solchen, bei denen mit verschiedenen Eingabewerten gerechnet werden soll, erstellt man zweckmäßigerweise zwei getrennte Lochstreifen. Der erste Lochstreifen wird das Programm, die benötigten Unterprogramme, die unveränderlichen Daten (Konstanten, Klartext usw.), die Angaben zur Druckanordnung und den Stopbefehl enthalten. Auf den zweiten Lochstreifen wird man die veränderlichen Eingabewerte und den Startbefehl lochen.

Für unser Beispiel wären demnach auf dem ersten Streifen zusammenzufassen:

U200U	Unterprogramm
U1750U	Überschrift
U1770U	Klartext
U1800U	Programm
U1030U	Druckanordnung
Zo+1E	Stop

und auf dem zweiten Streifen:

U1780U	Koeffizienten des Nennerpolynoms
U1790U	Koeffizienten des Zählerpolynoms
U1795U	Anfangswert, Schrittweite, usw.
E1800E	Startbefehl

Im Protokoll erscheint dann folgendes Schriftbild (μ ist mit der Hand hinzugesetzt, da die Taste wohl eine Lochung, aber keinen Anschlag liefert):

(1.Streifen)

U200U

B5 U220 B191 U6 B190 X B192 + B190 X B193 + B190 X

B194 + B190 X B195 + Eo

U1750U

μ X F(X) μ 0

U1770U

μ UNENDLICH μ 0

U1800U

B1750 D F1000 B190+1795 B190 D CKB13+4 BV191+1780 F200

PPQQE1825 U189 CKB13+4 BV191+1790 F200 B189 : D B6+1796

B190 + U190 B1797 - QQE1804 Zo B1770 D F1000 E1817

U1030U

7' 2' 6' 2' 3' 4' -1'

Z0+1E

(2.Streifen)

U1780U

1 5,5 3,25 -6,375 -3,375

U1790U

1,65 3,135 -10,824 -17,127 -2,97

U1795U

0 0,125 3,01

E1800E

5.2 Die Lochstreifeneingabe

Die so hergestellten Lochstreifen können nun in die Maschine eingegeben werden. Hierzu legt man den jeweiligen Lochstreifen in den photoelektrischen Abtaster und startet das Leseprogramm, was auf zwei Arten erfolgen kann:

Start ab Trommelzelle 1

Eo+1 Am Bedienungspult (siehe Abb. im Anhang) ist der Befehl Eo+1 in der Tastenreihe des Befehlsregisters einzustellen, d.h., die Trommeladresse 1 ist einzudrücken (die übrigen Tasten sind herauszunehmen). Eine Taste für E wird nicht benötigt (Erklärung siehe 5.3 "Befehlsregister").

Durch Betätigen der Taste "Befehlsübernahme" (Bef.-Üb.) wird der eingestellte Befehl ins Befehlsregister übernommen, und die entsprechenden Stellen leuchten nun auf. Allerdings leuchtet nicht die dem Befehlsbit entsprechende erste Stelle "Start" auf, sondern die dem Zustand der Maschine entsprechende zweite Stelle "Stop". Drückt man jetzt die Taste "Start", so läuft das Leseprogramm an und beginnt den Lochstreifen einzulesen.

Beim Start des Leseprogramms mit Eo+1 wird ein Teil dieses Programms von der Trommel in den Schnellspeicher geholt und besetzt die Zellen bis 99, so daß während des Einlesens nur die Schnellspeicherzellen ab 100 belegt werden können.

Man kann jedoch immer so vorgehen: Alle Programmteile, sowohl die Trommel- als auch die Schnellspeicherprogrammteile, werden während des Einlesens auf Trommelzellen

gebracht. Nach dem Einlesen werden die Schnellspeicherprogrammteile während des Programmablaufs durch Transferbefehle von der Trommel in den Schnellspeicher übertragen und anschließend durch einen Sprungbefehl auf eine Schnellspeicherzelle in Tätigkeit gesetzt. Dabei können natürlich alle Schnellspeicher, auch die unterhalb 99, benutzt werden.

Das Leseprogramm, mit Eo+1 gestartet, braucht etwa 3 msec zum Entschlüsseln eines Zeichens. Für die Schlußverarbeitung wird zusätzlich noch einige Zeit benötigt.

Start ab Trommelzelle 6

Eo+6 Hierzu ist am Bedienungspult der Befehl Eo+6 in der Tastenreihe des Befehlsregisters einzustellen. Durch "Befehlsübernahme" gelangt der Befehl ins Befehlsregister, und "Start" läßt das Leseprogramm anlaufen.

Wenn das Leseprogramm mit Eo+6 gestartet wird, so werden nur die Schnellspeicherzelle bis 32 belegt, und die Entschlüsselung der Fernschreibzeichen wird auf der Trommel durchgeführt. Die Entschlüsselungszeit für ein Fernschreibzeichen beträgt hierbei 20 msec. Das Einlesen dauert also länger, hat aber den Vorteil, daß nur wenige Schnellspeicherzellen belegt werden.

Mit dem Befehl F522 hat man die Möglichkeit, während der Rechnung einzelnen Worte vom Lochstreifen in den Akkumulator zu bringen, ohne den Programmablauf unterbrechen zu müssen. Hierbei werden die Schnellspeicherzellen bis 32 benutzt, d.h. die gleichen Schnellspeicher, die nach einer Konvention auch bei den allgemeinen Unterprogrammen (Trigonometrische Funktionen, Logarithmus usw.) belegt werden. Da das Einlesen über Schnellspeicherzelle 15 erfolgt (jeweils ein Zeichen), kann man während des Rechnens auch ein einzelnes Zeichen in den Akkumulator lesen.

F522 1 Wort vom Abtaster in den Akkumulator lesen.

B15 1 Zeichen vom Abtaster in den Akkumulator lesen.

Das Ergebnis, das die Maschine bei unserem Programmierungsbeispiel ausschreibt, hat folgende Form:

X	F(X)
0	0,880000
0,125000	1,283050
0,250000	1,683744
0,375000	2,143883
0,500000	2,748900
0,625000	3,675136
0,750000	5,434000
0,875000	10,562842
1,000000	UNENDLICH

X	F(X)	(Die erneute Überschrift erscheint natürlich nicht im Lösungsprotokoll)
1,125000	-9,612453	
1,250000	-4,501304	
1,375000	-2,773282	
1,500000	-1,893375	
1,625000	-1,354222	
1,750000	-0,986480	
1,875000	-0,717424	
2,000000	-0,510593	
2,125000	-0,345659	
2,250000	-0,210373	
2,375000	-0,096904	
2,500000	-0,000000	
2,625000	0,083995	
2,750000	0,157711	
2,875000	0,223090	
3,000000	0,281600	

5.3 Das Bedienungspult der ZUSE Z 23

Am Bedienungspult sind, wie die Abbildung im Anhang zeigt, vier Tastenreihen vorhanden.

Befehlsregister Die 2. Reihe von oben ist eine Tastatur für das Befehlsregister. Für jede einzelne Binärstelle ist eine Taste vorgesehen. Einmal kann man hier einzelne Befehle eintasten, zum anderen zeigen Lampen in den Tasten den jeweiligen Stand des Befehlsregisters an.

Die ersten beiden, mit "Start" und "Stop" bezeichneten Tasten sind die Kennzeichenstellen. Bei nicht laufender Maschine leuchtet stets nur die zweite Lampe auf.

Die nächsten 5 Stellen bezeichnen die Bedingungszeichen PP, P, QQ, Q, Y.

Danach kommen die weiteren internen Befehlszeichen von C bis V. Das Zeichen E ist nicht gesondert vorhanden, da eine Kombination, die weder U noch A, S, F oder Z enthält, als E-Befehl definiert ist (Alle Befehle außer den Sprung-, Ruf- und Stopbefehlen müssen mindestens eins der Operationszeichen U, A, S enthalten; sie heißen Arbeitsbefehle).

Die nächsten 8 Stellen sind die Stellen der Schnell-speicheradressen, die sich aus den einzelnen Zweierpotenzen 1, 2, 4, 8, 16, 32, 64, 128 zusammensetzen.

Entsprechend sind die letzten 13 Stellen, die Trommeladressen, aufgebaut.

Befehlsübernahme Ein in die Tastatur des Befehlsregisters im Interncode eingetasteter Befehl wird nach Betätigung der

- Taste "Befehlsübernahme" (in der untersten Reihe) in das Befehlsregister übernommen, worauf dann die in die Tasten eingebauten Lämpchen aufleuchten.
- Weiter** Durch "Weiter" (in der untersten Reihe) wird der gerade im Befehlsregister stehende Befehl ausgeführt; danach stoppt die Maschine.
- Start** Mit "Start" (in der untersten Reihe) kann die Rechnung begonnen werden.
- Stop** Mit "Stop" (in der untersten Reihe) kann die laufende Maschine angehalten werden (Die 3 Tasten "Weiter", "Start" und "Stop" sind je zweimal in der untersten Reihe vorhanden).
- Akkumulator** In der 3.Reihe von oben ist für den Akkumulator, bis auf die ersten beiden Stellen, die gleiche Bezeichnung angebracht wie beim Befehlsregister. Die Lampen zeigen den jeweiligen Inhalt des Akkumulators an.
- Akku-übernahme** Auch über die Akkumulator-Tastatur kann 1 Wort im Interncode eingestellt und durch die Taste "Akku-übernahme" in den Akkumulator übernommen werden. Handelt es sich um einen Befehl, so muß auch das Befehlsbit eingetastet werden, da die Maschine sonst keine Möglichkeit hat, den Befehl als solchen zu erkennen. Die Einstellung der Akkumulator-tastatur kann auch während des Programms durch den Befehl U4 in den Akkumulator übertragen werden (vergl.3.7). Wenn alle Tasten herausgenommen sind, wird durch "Akkuübernahme" der Inhalt des Akkumulators gelöscht.
- Lautsprecher** In der untersten Reihe sind noch Schalter, mit denen Lautsprecher ein- und ausgeschaltet werden können.
- Worttransport** Bei jedem Worttransport in der Maschine wird ein Signal auf einen Lautsprecher gegeben, so daß man den Lauf der Maschine akustisch verfolgen kann. Mit einer Taste kann dieser Lautsprecher ausgeschaltet, mit einer zweiten können 2 Lautstärken eingestellt werden.
- Alarm** Daneben ist noch eine Taste mit der der "Alarm"-Lautsprecher ausgeschaltet werden kann. Dieser Lautsprecher gibt ein Signal (Dauerton), wenn beim Ablesen einer Speicherzelle die Quersummenkontrolle nicht stimmt (beim Beschreiben einer Zelle wird neben den 40 Informationsbit noch eine 41.Stelle mitgeschrieben, die die Zahl der Einsen in dem Wort auf eine gerade Zahl ergänzt). Beim Ablesen wird kontrolliert, ob in diesen 41 Stellen eine gerade

Anzahl von Einsen vorliegt. Wenn das nicht der Fall ist, stoppt die Maschine und gibt über den Lautsprecher das Dauerton-Signal. Außerdem leuchtet auch eine der mit "Alarm" bezeichneten Lampen auf, wodurch angezeigt wird, ob die Fehlablesung von der Trommel oder vom Schnellspeicher erfolgte. Die dritte Lampe läßt erkennen, ob eine Netzstörung vorlag. Für Prüfzwecke lassen sich die "Alarmer", die ja einen Stop der Maschine hervorrufen, ausschalten.

Nacht Mit der Taste "Nacht" (erste Taste links oben) kann man erreichen, daß sich die Maschine abstellt, sobald ein Stop eintritt. Man kann somit die Maschine abends weiterrechnen lassen, und sie schaltet sich von selbst ab, wenn die Rechnung beendet ist.

Trommel Maschine FS Abtaster In der oberen Reihe sind weiterhin Tasten zum Ein- und Ausschalten der Trommel, der Maschine sowie der Ein- und Ausgabegeräte angebracht.

FS Der Fernschreiber kann auch unabhängig von der Maschine eingeschaltet werden, so daß man mit ihm auch Programme und Zahlenwerte ablocken kann. Wird bei eingeschalteter Maschine die Taste "FS" nicht eingedrückt, so schaltet sich der Fernschreiber selbsttätig ab, wenn er länger als 30 sec nicht angesprochen wurde. Andernfalls wird die automatische Abschaltung unterbunden.

Abtaster Der Abtaster kann auf einen möglicherweise vorhandenen Drucker umgeschaltet werden ("Abtaster auf Drucker").

<2> Die mit <2> 1 bzw. <3> 40 bezeichneten Lampen
<3> 1 zeigen an, was sich in den entsprechend bezeichneten Stellen der Schnellspeicherzelle 2 bzw. 3 gerade befindet; Aufleuchten bedeutet eine 1, Erlöschen eine 0.
40

Schreibsperre Durch Drücken dieser Tasten kann man die jeweils angegebenen Gruppen von je 1 024 Speicherzellen der Trommel sperren; d.h., man kann diese Zellen wohl ablesen, aber man kann nichts in diese Zellen schreiben. Auf diese Weise kann man verhindern, daß Programmteile oder Zahlenwerte versehentlich überschrieben werden.

Löschung RO +1 Addw. Diese Tasten sind speziell für die Prüfung der Maschine vorgesehen.

Bedingungs- schalter Hier kann man zur Beeinflussung eines Programmablaufs 32 verschiedene Kombinationen der 5 Tasten einstellen, die vom Programm unter der Adresse 14

als Strichzahlen von 0 - 31' abgefragt werden können (vergl.3.7).

Adressen- Die restlichen Tasten der oberen Reihe sind zur stop Einstellung einer Adresse vorgesehen, an der die Maschine stoppen soll. Sobald die eingetastete Adresse irgendwo angesprochen wird, tritt ein Stop der Maschine ein. Diese Stopeinrichtung, die besonders beim Programmtest wertvolle Dienste leisten kann, wird erst nach Einschalten der Taste "Ein" wirksam.

Die Adressenangaben über der Tastatur des Adressenstops, des Befehlsregisters und des Akkumulators sind in voneinander farbig unterschiedene 3er-Gruppen eingeteilt. Diese Einteilung soll das Eintasten einer beliebigen in Dezimalschreibweise vorliegenden Adresse als Dualzahl erleichtern. Dazu liegt der Programmierungsanleitung eine sogenannte "Oktaltabelle" bei (siehe Anhang). Hierin ist jeder 16. Dezimalzahl ihre Darstellung im Oktalzahlensystem (Ziffern 0 bis 7) gegenübergestellt. Jede Ziffer dieser Oktalzahlen ist durch ihren Stellenwertfaktor (Potenz von $8 = 2^3$) einer dreistelligen Dualzahl der entsprechenden 3er-Gruppe äquivalent und gibt somit diejenige Zahl an, die in diese Gruppe einzutasten ist. Z.B. findet man für die Adresse 2512 die Oktalzahl 04720, die man in der betreffenden Tastenreihe wie folgt einzudrücken hat:

4096	2048	1024	512	256	128	64	32	16	8	4	2	1
	x			x	x	x		x				
o		4			7			2				o

Ist eine Adresse nicht unmittelbar in der Tabelle aufgeführt, tastet man zunächst die der nächstkleineren Dezimalzahl entsprechende Oktalzahl ein und ergänzt diese sodann in den 4 letzten Dualstellen um den Restbetrag (< 16).

6. Erklärung weiterer Befehle

6.1 Arithmetische Befehle für Strichzahlen

Da Strichzahlen einen ganz anderen internen Aufbau als Gleitkommazahlen haben, können die Befehle für die arithmetischen Operationen im gleitenden Komma nicht auf Strichzahlen angewandt werden. Wo Verwechslungen mit Gleitkommazahlen möglich sind, sollte eine Strichzahl in den Texterklärungen stets apostrophiert werden.

Addition und Subtraktion der Strichzahlen werden nicht durch Unterprogramme, sondern direkt durch in der Maschine verdrahtete Elementaroperationen ausgeführt.

hineinaddieren, ohne den Hauptakkumulatorinhalt zu zerstören:

As+t <s> ' + <t> ' → s, <a> → a

Der Inhalt der aufgerufenen Trommelzelle t wird zum Inhalt der Schnellspeicherzelle s addiert; das Ergebnis geht in Schnellspeicherzelle s.

Ss+t <s> ' - <t> ' → s, <a> → a

Der Inhalt der Trommelzelle t wird vom Inhalt der Schnellspeicherzelle s subtrahiert; das Ergebnis geht in Schnellspeicherzelle s.

USs+t <t> ' - <s> ' → s, <a> → a

Wiederum Subtraktion mit vertauschten Operanden.

Wenn den Doppeladreibefehlen ein N hinzugesetzt wird, bezieht sich jetzt das Nullsetzen auf die aufgerufene Schnellspeicherzelle s:

NSs+t - <t> ' → s, <a> → a

Hier wird der Inhalt der Trommelzelle t, als Strichzahl aufgefaßt, mit umgekehrtem Vorzeichen in die Schnellspeicherzelle s gebracht.

NUSs+t ≡ NASs+t ≡ Bs+t <t> ' → s, <a> → a

Alle diese Befehle können noch durch den Zusatz V ergänzt werden (Blocktransfer). Wie üblich muß vorher zur Übertragung von n+1 Worten die Zahl n' nach Schnellspeicherzelle 13 gebracht werden. Das i läuft dann von 0 bis n.

CKB13+n } <s+i> ' + <t+i> ' → s+i
AVs+t

CKB13+n } <s+i> ' - <t+i> ' → s+i
SVs+t

CKB13+n } - <t+i> ' → s+i
NSVs+t

CKB13+n } <t+i> ' - <s+i> ' → s+i
USVs+t

CKB13+n } <t+i> ' → s+i
NUSVs+t

Bei Blocktransferbefehlen mit s=0 wird nur die Trommeladresse erhöht:

CKB13+n } <a> ' + <t+i> ' → a
AVo+t

CKB13+n } <a> '- <t+i> ' → a
 SVo+t

In diesem Zusammenhang sei auch noch der folgende Befehl erwähnt:

CKB13+n } <a> → t+i
 UVo+t

Mit diesem Blocktransferbefehl kann man also ein im Akkumulator stehendes Muster auf eine Anzahl von Trommelzellen übertragen; jedoch auf höchstens 256, da der Schnellspeicher 13 nur 8 Binärstellen hat.

Für die neueren Maschinen des Typs ZUSE Z 23 gibt es noch sogenannte bedingte Blocktransferbefehle, nämlich PPθVs+t, QQθVs+t und HθVs+t mit θ = A, S, US, Ø, I oder IS. Die Operation θ wird so lange zwischen <s> und <t+v> ausgeführt (Ergebnis jeweils in a), bis Zähler 13 auf Null zurückgezählt hat oder bis

bei PP <a> < 0,

bei QQ <a> ≥ 0,

bei H <a> = 0 geworden ist (nur für s < 16 anwendbar).

Sollen z.B. die Trommelzellen 2000 bis 2099 auf den Inhalt 25 abgefragt werden, so programmiert man:

CB1 (damit am Anfang <a> ≠ 0 gesichert ist)

CKB6+25

CKB13+100

HSV6+2000

Da beim Rechnen mit Strichzahlen auch Multiplikationen oder Divisionen auftreten können, sind in das Grundprogramm Unterprogramme aufgenommen worden, die es erlauben, Strichzahlen miteinander zu multiplizieren oder durcheinander zu dividieren.

Hierbei wurde die gleiche Konvention wie bei Gleitkommaoperationen verwendet, nämlich daß der 1.Operand in Schnellspeicherzelle 6 und der 2.Operand im Akkumulator steht und daß das Ergebnis in den Akkumulator und in Schnellspeicherzelle 6 gebracht wird.

Die Unterprogramme für Strichzahl-Multiplikation und -Division benötigen als Zwischenspeicher die Schnellspeicherzellen 2 bis 6. Der Aufruf erfolgt durch vercodete Rufbefehle, die wie bei den Gleitkommaoperationen durch die üblichen arithmetischen Symbole dargestellt werden, denen ein Apostroph hinzugesetzt ist (die Unterscheidung durch das Apostroph ist für das Leseprogramm erforderlich).

X' = Multiplikation von Strichzahlen

<6> '. <a> ' → a,6

:' = Division von Strichzahlen

<6> ': <a> ' → a,6, Rest → 2

Falls bei der Multiplikation größere Zahlen als $2^{38}-1$ entstehen (eine sichere Entscheidung darüber, ob nicht etwa eine Gleitkommazahl vorliegt, ist der Maschine auch schon bei kleineren Zahlen als $2^{38}-1$ nicht mehr möglich; vergl. 2.3) oder wenn durch 0 dividiert werden soll, schreibt die Maschine zur Kenntlichmachung der Bereichsüberschreitung

BUEB Em

und stoppt (vergl. 3.5).

Durch den Befehl F1020 kann eine im Akkumulator stehende Strichzahl in eine Gleitkommazahl umgewandelt werden.

Bei den neueren Maschinen des Typs ZUSE Z 23 gibt es noch Sonderbefehle für die Multiplikation von positiven Strichzahlen mit Zehnerpotenzen:

LØKO	$\langle a \rangle \cdot 10$	$\rightarrow a$	
PQLØK1-n	$\langle a \rangle \cdot 10^n$	$\rightarrow a$	
RØKO	$\langle a \rangle \cdot 10^{-1}$	$\rightarrow a$	} Rest $\rightarrow 3$
PQRØK1-n	$\langle a \rangle \cdot 10^{-n}$	$\rightarrow a$	

6.2 Befehle zur Konstantenzuführung

Beim Aufbau von Adressen oder bei Zählungen braucht man häufig Strichzahlen, die addiert oder subtrahiert werden sollen. Man könnte die benötigten Strichzahlen in irgendwelche Speicherzellen bringen und Am- oder Sm-Befehle verwenden.

Durch die Möglichkeit der Konstantenzuführung aus dem Befehlsregister erübrigt sich jedoch eine gesonderte Speicherung der Strichzahlen.

Das Befehlszeichen C bewirkt, daß die in den Adressenstellen des Befehls stehende Zahl nicht zur Anwahl einer Speicherzelle benutzt, sondern dem Addierwerk als Strichzahl zugeführt wird. Welche Operation mit der ankommenden Strichzahl ausgeführt wird, geben die übrigen Operationszeichen A, S, US an.

CAn	$\langle a \rangle '+ n'$	$\rightarrow a$
	(dagegen An: $\langle a \rangle '+ \langle n \rangle'$	$\rightarrow a$)
CSn	$\langle a \rangle '- n'$	$\rightarrow a$
CUSn	$n' - \langle a \rangle'$	$\rightarrow a$
CBn	n'	$\rightarrow a$
CNSn	$-n'$	$\rightarrow a$

Mit den C-Befehlen lassen sich Strichzahlen bis

$$n' = 2^{22}-1 = 4194303'$$

dem Hauptakkumulator zuführen. Denn es dürfen hier nur die

Bitstellen bis einschließlich V verwendet werden.

Da eine Strichzahl intern vom 40.bit her dual aufgebaut ist und die Maschine bei einem in der Form $s+t$ geschriebenen Ausdruck die vor dem +Zeichen stehende Zahl den Schnellspeicheradressenstellen zuordnet, also als ganzes Vielfaches von $2^{13}=8192$ erkennt, läßt sich jede Strichzahl in die Summe $n'=(8192 s+t)$ zerlegen (mit $s \leq 511$ und $t \leq 8191$). In manchen Fällen ist eine solche Darstellung vorteilhafter; man schreibt dann bei Befehlen auch $C \theta s+t$ anstelle von $C \theta n$. Hierbei steht θ stellvertretend für A, S, US, B oder NS. Das s ist in diesem Fall keine Schnellspeicheradresse, sondern der Befehl bezieht sich auf den Akkumulatorinhalt.

6.3 Ein Beispiel zur Anwendung des Strichzahlrechnens

Mit einem kleinen Programm sollen die Inhalte der Speicherzellen 2001 bis 2056 ausgeschrieben werden. Das Programm soll in Zelle 6000 anfangen.

6000	CNS56	}	-56'	-	35		
1	U35						
2	B2001	}	<2001>	-	a	} Zelleninhalt	
3	D		<a>	-	FS		ausdrucken
4	B35	}	<35>	' + 1'	-	a	
5	CA1						
6	PPZo		<a> = 0 ?	Schon 56 Worte ausgedruckt?			
				Wenn ja, Stop.			
7	U35		<a>	-	35		
8	B6002	}	<6002>	-	a	} Mit diesen 3 Befehlen wird die Adresse des in Speicherzelle 6002 stehenden Befehls um 1 erhöht. Beim 1.Rücksprung auf die Speicherzelle 6002 findet die Maschine dort den Befehl B2002 beim 2.Rücksprung B2003, beim 3.Rücksprung B2004 usw.	
9	CA1		<a>	' + 1'	-		a
10	U6002		<a>	-	6002		
11	E6002						

Für Zählungen ist es häufig notwendig, Inhalte von Schnellspeicherzellen zu vergrößern oder zu verkleinern. Das kann man natürlich tun, indem man den Inhalt der Schnellspeicherzelle in den Akkumulator bringt, mit CA- bzw. CS-Befehlen die Veränderung vornimmt und das Ergebnis wieder wegspeichert, wie es im obigen Beispiel in den Zellen 6004 bis 6007 durchgeführt worden ist. Dabei wird allerdings der ursprüngliche Inhalt des Akkumulators zerstört.

Durch Einfügen des Buchstabens K in die C-Befehle läßt sich aber eine solche Änderung in einem beliebigen Schnellspeicher ausführen, ohne den Inhalt des Hauptakkumulators zu berühren.

Hierbei bezeichnet die Schnellspeicheradresse den Speicher, in dem gezählt werden soll, und die Trommeladresse die Änderung. Der übrige Operationsteil (A, S, US, B, NS) gibt an, welche Operation ausgeführt werden soll.

CKAs+t	$\langle s \rangle ' + t' \rightarrow s,$	$a \rightarrow a$
CKSs+t	$\langle s \rangle ' - t' \rightarrow s,$	$a \rightarrow a$
CKUSs+t	$t' - \langle s \rangle ' \rightarrow s,$	$a \rightarrow a$
CKBs+t	$t' \rightarrow s,$	$a \rightarrow a$
CKNSs+t	$- t' \rightarrow s,$	$a \rightarrow a$

Mit Hilfe dieser Befehle würde das obige Beispiel zum Ausdrucken der Zelleninhalte 2001 ff lauten:

6000	CKNS35+56	-56 → 35
1	B2001	} Zelleninhalte drucken
2	D	
3	CKA35+1	$\langle 35 \rangle ' + 1' \rightarrow 35$
4	PZ35+0	Wenn $\langle 35 \rangle \geq 0$, Stop
5	B6001	} Verändern der Adresse des Befehls in Zelle 6001
6	CA1	
7	U6001	
8	E6001	Rücksprung auf 6001

6.4 Befehle für logische Operationen

Außer den arithmetischen Operationen kann die Maschine auch logische Operationen ausführen, nämlich "Und" (Konjunktion) und "Oder" (Disjunktion).

Die Konjunktion zweier Speicherinhalte wird in der Maschine binärstellenweise durchgeführt und deshalb als Intersektion bezeichnet. Das Ergebnis besitzt nur in den Stellen Einsen, die bei beiden eingehenden Worten Einsen waren. Die Inhalte der aufgerufenen Speicher bleiben dabei erhalten.

Enthält der Befehl nur eine Adresse, so erfolgt die Intersektion zwischen dem Inhalt der aufgerufenen Speicherzelle m und dem des Akkumulators.

Im $\langle a \rangle \wedge \langle m \rangle \rightarrow a, \quad \langle m \rangle \rightarrow m$

Bei Intersektionsbefehlen mit zwei Adressen werden die beiden Speicherinhalte zur Intersektion gebracht. Auch hier geht das Ergebnis in den Akkumulator.

Is+t $\langle s \rangle \wedge \langle t \rangle \rightarrow a, \quad \langle s \rangle \rightarrow s, \quad \langle t \rangle \rightarrow t$

Die Intersektion kann auch mit dem negativ gemachten Inhalt

der aufgerufenen Speicherzelle durchgeführt werden.

ISm $\langle a \rangle \wedge (-\langle m \rangle) \rightarrow a, \quad \langle m \rangle \rightarrow m$

ISs+t $\langle s \rangle \wedge (-\langle t \rangle) \rightarrow a, \quad \langle s \rangle \rightarrow s, \quad \langle t \rangle \rightarrow t$

Man kann auch Intersektionen mit Konstanten, die durch C-Befehle aus dem Befehlsregister zugeführt werden, durchführen.

CI_n $\langle a \rangle \wedge n' \rightarrow a$

CI_{s+t} $\langle a \rangle \wedge (s+t)' \rightarrow a$

CI_{S_n} $\langle a \rangle \wedge (-n') \rightarrow a$

CI_{S_{s+t}} $\langle a \rangle \wedge -(s+t)' \rightarrow a$

Diese Intersektionen werden oft zum Herausschneiden bestimmter Teile von Speicherinhalten benutzt, die man anschließend weiter verwenden möchte (z.B. zum Heraustrennen der Trommeladresse oder des Operationsteils eines Befehls).

CI8191 $\langle a \rangle \wedge 8191' \rightarrow a$

Nach Ausführung dieses Befehls steht im Akkumulator nur noch die Trommeladresse des ursprünglich im Akkumulator enthaltenen Befehls.

CI255+o $\langle a \rangle \wedge (255+0)' \rightarrow a$

Nach Ausführung dieses Befehls steht nur noch die Schnellspeicheradresse des ursprünglich im Akkumulator enthaltenen Befehls im Akkumulator.

CI_{S1+o} oder CI_{S8192} $\langle a \rangle \wedge (\underbrace{|\dots|}_{27} \underbrace{0\dots0}_{13})' \rightarrow a$

Der Operationsteil und die Schnellspeicheradresse des im Akkumulator stehenden Befehls bleiben im Akkumulator, während die Trommeladresse gelöscht wird.

CI_{S256+o} $\langle a \rangle \wedge (\underbrace{|\dots|}_{19} \underbrace{0\dots0}_{21})' \rightarrow a$

Der Operationsteil des Befehls wird herausgetrennt, d.h. Schnellspeicher- und Trommeladresse werden auf Null gesetzt.

Wenn s nicht als Konstante wirken, sondern zur Anwahl einer Schnellspeicherzelle dienen soll, so muß ein CK-Befehl verwendet werden:

CKI_{s+t} $\langle s \rangle \wedge t' \rightarrow a, \quad \langle s \rangle \rightarrow s$

CKI_{S_{s+t}} $\langle s \rangle \wedge (-t') \rightarrow a, \quad \langle s \rangle \rightarrow s$

Als Beispiel soll die Trennung einer Gleitkommazahl in

6.5 Befehle für Adressenänderungen

Die Länge eines Programms läßt sich oft erheblich abkürzen, wenn man einzelne Programmteile so einrichtet, daß sie mehrfach durchlaufen werden können. Dies ist z.B. dann möglich, wenn für das mehrmalige Durchlaufen eines Programmteils lediglich bei einigen Befehlen Adressenänderungen nötig sind. Mit Hilfe der sogenannten G-Befehle kann man nämlich erreichen, daß die gewünschten Adressen erst während des Programmablaufs beim jeweiligen Aufruf der betreffenden Befehle stets neu aufgebaut werden. Handelt es sich um den Aufbau einer Trommeladresse, so setzt man vor den Operationsteil des Befehls, der hier allgemein mit θ bezeichnet werden soll, ein G; soll eine Schnellspeicheradresse aufgebaut werden, schreibt man GK θ . Der Adressenteil wird in einer allgemeinen Form geschrieben, aus der dann durch die G-Befehlszeichen nach bestimmten Vorschriften eine neue Adresse gebildet wird. Ein G-Befehl bewirkt also, daß zunächst eine Trommel- bzw. Schnellspeicheradresse aufgebaut wird und anschließend dann der Befehl θ mit der gebildeten Adresse ausgeführt wird.

G θ t $\langle t \rangle_{28\dots 40} \Rightarrow \hat{t}$, dann $\theta \hat{t}$
 G θ s+0 $\langle s \rangle_{28\dots 40} \Rightarrow \hat{t}$, dann $\theta \hat{t}$
 G θ s+t $(\langle s \rangle + t)_{28\dots 40} \Rightarrow \hat{t}$, dann $\theta \hat{t}$; $\langle s \rangle \rightarrow s$

Im ersten Befehl wird die neue Trommeladresse \hat{t} aus der Trommelzelle t, im zweiten aus der Schnellspeicherzelle s geholt. Und zwar werden hierfür jeweils die in den Stellen 28 bis 40 des Inhalts von t bzw. s befindlichen Zahlen verwendet. Im dritten Befehl wird die neue Adresse aus den 13 letzten Stellen der Summe $\langle s \rangle + t$ gebildet.

GK θ t nicht möglich, da K den Weg zur Trommel sperrt
 GK θ s+0 $\langle s \rangle_{33\dots 40} \Rightarrow \hat{s}$, dann $\theta \hat{s}$
 GK θ s+t $(\langle s \rangle + t)_{33\dots 40} \Rightarrow \hat{s}$, dann $\theta \hat{s}$; $\langle s \rangle \rightarrow s$

Beim Aufbau neuer Schnellspeicheradressen werden also nach Anwendung der gleichen Rechenvorschriften sinngemäß nur die in den 8 Stellen 33 bis 40 stehenden Zahlen verwendet.

Der Inhalt der aufgerufenen Speicherzelle s bleibt bei den Befehlen G θ s+t und GK θ s+t erhalten. Man sagt: Das Indexregister bleibt unverändert. Setzt man diesen beiden Befehlen jedoch das Zeichen C voran, so gelangt die Summe $\langle s \rangle + t$ in das Indexregister.

CG θ s+t $(\langle s \rangle + t)_{28\dots 40} \Rightarrow \hat{t}$, dann $\theta \hat{t}$; $\langle s \rangle + t \rightarrow s$
 CGK θ s+t $(\langle s \rangle + t)_{33\dots 40} \Rightarrow \hat{s}$; dann $\theta \hat{s}$; $\langle s \rangle + t \rightarrow s$

Das Zeichen C hat hier also eine andere Funktion als beim Rechnen mit Strichzahlen (siehe 6.2).

7. Erneute Behandlung des Programmierungsbeispiels (abgeändert)

Es soll hier nochmals das schon in Kapitel 4 gegebene Beispiel durchgeführt werden, jedoch mit einigen Abänderungen. Das Zählerpolynom soll diesmal 6. Grades und das Nennerpolynom 8. Grades sein.

Es soll also die Funktion

$$F(x) = \frac{P_6(x)}{P_8(x)}$$

tabelliert werden von $x_0 = 0$ bis $x_n = 3$ mit einer Schrittweite $\Delta x = 0,125$.

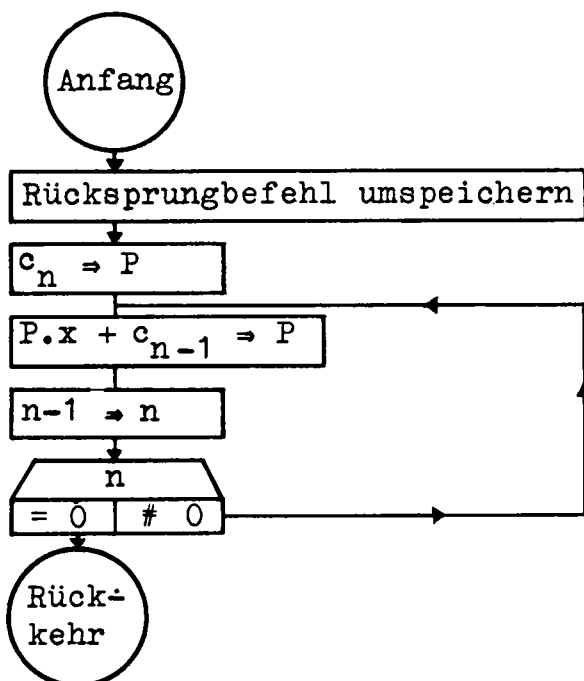
Da das Argument stets nur 3 Stellen nach dem Komma hat, sollen hier auch nur diese Stellen herausgeschrieben werden, während das Ergebnis mit 6 Stellen nach dem Komma erscheinen möge (Änderung der Druckanordnung).

Außerdem sollen (als Beispiel zur Anwendung des Bedingungsalters) die Werte des Zähler- und Nennerpolynoms zusätzlich neben Argument und Funktionswert ausgeschrieben werden, wenn der Bedingungsalters ausgeschaltet ist ($\langle 14 \rangle = 0$). Wenn dagegen eine der Tasten des Bedingungsalters eingedrückt ist, sollen nur Argument und Funktionswert ausgeschrieben werden.

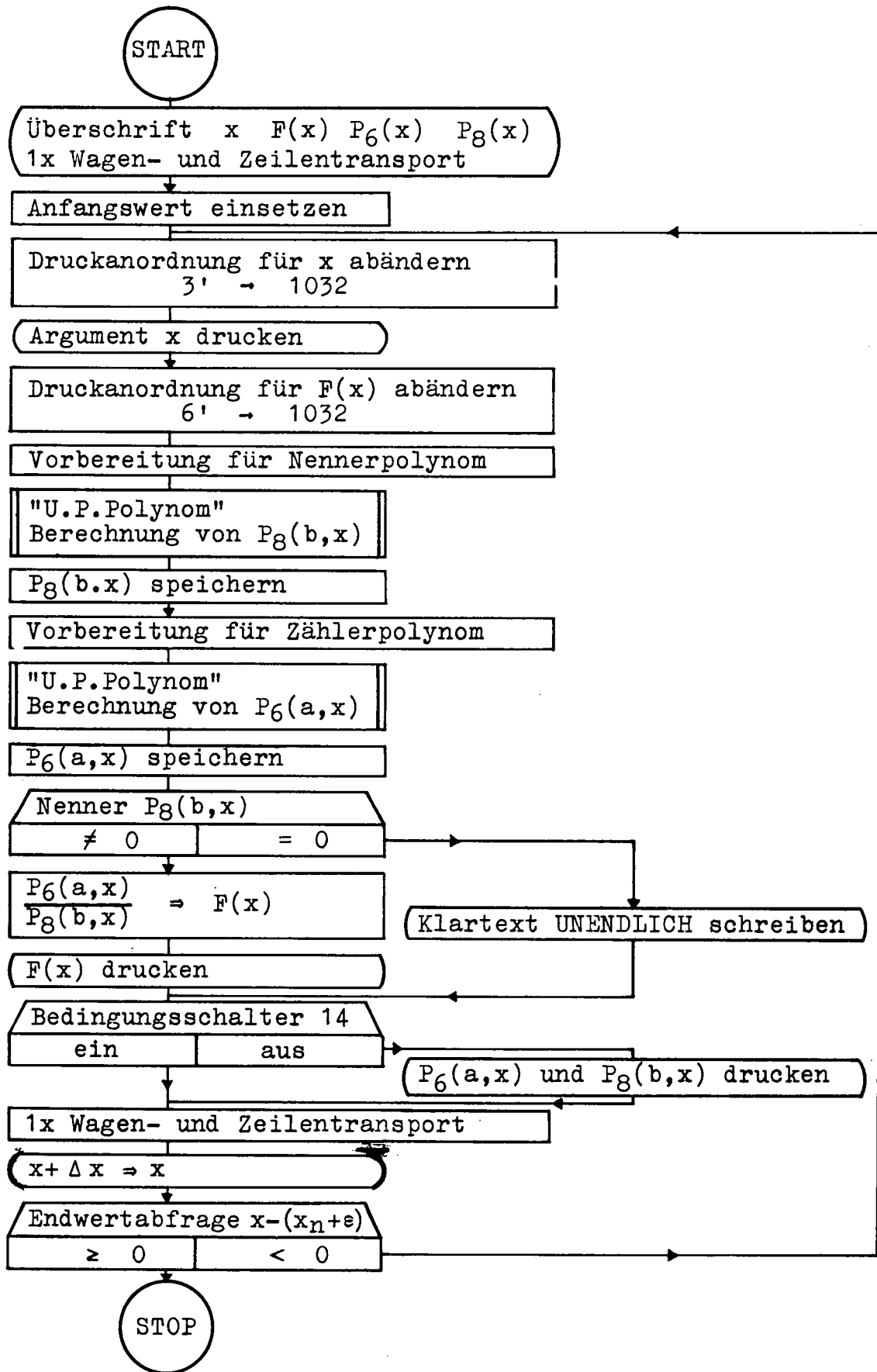
Das Unterprogramm Polynom soll so gestaltet sein, daß es für jeden beliebigen Grad des Polynoms verwendbar ist. An zwei Stellen enthält es die soeben eingeführten G-Befehle.

Da diese neue Programmierungsaufgabe nur eine Abänderung des im 4. Kapitel behandelten Beispiels ist, vergleiche man die einander entsprechenden Teilabschnitte der Ausführungen.

Flußdiagramm zum Unterprogramm



Flußdiagramm zum Hauptprogramm



Speicherliste für das Unterprogramm

Zellen-Nr.	Inhalt	Erläuterungen
30	$x_1 = x$	Das jeweilige Argument wird vom Hauptprogramm geliefert.
31	$\{c_n\}$	Adresse des höchsten Koeffizienten (Trommeladresse). Das c_n steht hier stellvertretend für die höchsten Koeffizienten a_n bzw. b_n , deren Zahlenwerte jeweils vom Hauptprogramm nach $t = \{c_n\}$ gebracht werden.
32	$-n'$	Grad des jeweils zu berechnenden Polynoms als negative Strichzahl. Auch diese Angabe wird vom Hauptprogramm gemacht.
200 ff		Unterprogramm

Unterprogramm

Zellen-Nr.	Inhalt	Erläuterungen
200	B5	} Rücksprungbefehl umspeichern in die letzte Zelle des Unterprogramms
1	U212	
2	GB31+0	} $c_n \Rightarrow P \rightarrow 6$
3	U6	
4	B30	
5	X	} $P \cdot x + c_{n-1} \Rightarrow P \rightarrow a, 6$
6	CGB31+1	
7	+	
8	CKA32+1	
9	B32	} Wenn $-n < 0$, dann Sprung nach 204
10	QQE204	
11	B6	$P_n(c, x) \rightarrow a$
12	E0	Rücksprungbefehlszelle. Der Inhalt E0 wird nach Ausführung des in Zelle 201 stehenden Befehls U212 durch den Rücksprungbefehl ersetzt.

Speicherliste für das Hauptprogramm

6000 ff	Programm
6080 ff	Anfangs- und Endwert, Schrittweite
6100 ff	Überschrift in Klartext: x $F(x)$ $P_6(x)$ $P_8(x)$
6120 ff	Klartext: UNENDLICH
6150 ff	Koeffizienten des Zählerpolynoms $a_6, a_5 \dots a_0$
6160 ff	Koeffizienten des Nennerpolynoms $b_8, b_7 \dots b_0$

Hauptprogramm

6000	B6100	}	Überschrift schreiben
1	D		
2	F1000	}	Wagen- und Zeilentransport
3	B30+6080		$x_0 \Rightarrow x \rightarrow 30$ (Anfangswert einsetzen)
4	CB3	}	$3' \rightarrow 1032$ (Änderung der Druckanordnung für x)
5	U1032		
6	B30	}	Argument x drucken
7	D		
8	CB6	}	$6' \rightarrow 1032$ (Änderung der Druckanordnung für F (x))
9	U1032		
10	CKB31+6160	}	Vorbereitung für Nennerpolynom
1	CKNS32+8		
2	F200		U.P.Polynom; $P_8(b,x) \Rightarrow a,6$
3	U35		$P_8(b,x)$ speichern
4	CKB31+6150	}	Vorbereitung für Zählerpolynom
5	CKNS32+6		
6	F200		U.P.Polynom; $P_6(a,x) \Rightarrow a,6$
7	U36		$P_6(a,x)$ speichern
8	B35	}	Wenn $P_8(b,x) = 0$, Sprung nach 6037
9	PPQQE6037		
20	:	}	$\frac{P_6(a,x)}{P_8(b,x)} \Rightarrow F(x)$ drucken
1	D		
2	NS14	}	Wenn Bedingungsschalter eingeschaltet, Sprung nach 6028
3	QQE6028		
4	B36	}	$P_6(a,x)$ drucken
5	D		
6	B35	}	$P_8(b,x)$ drucken
7	D		
8	F1000		Wagen- und Zeilentransport
9	B6+6082		$\Delta x \Rightarrow 6$
30	B30	}	$x + \Delta x \Rightarrow x \rightarrow 30$
1	+		
2	U30		
3	B6081	}	Endwertabfrage $x - (x_n + \varepsilon)$; wenn < 0 , Sprung nach 6004
4	-		
5	QQE6004		

Hauptprogramm

6	Zo	Stop
7	B6120	} Klartext UNENDLICH schreiben
8	D	
9	E6022	Sprung nach 6022

8. Relatives und adressenloses Programmieren

8.1 Relativadressen

Beim Aufstellen von Programmen will man sich häufig nicht auf bestimmte Anfangsadressen festlegen. Das trifft besonders für Unterprogramme zu, da diese ja für möglichst viele verschiedene Programme verwendbar sein sollen.

Man möchte solche Programme möglichst so einrichten, daß man sie an einer beliebigen Stelle der Trommel speichern kann (relatives Programmieren). Diese Möglichkeit ist durch die Verwendung von Relativadressen gegeben. Man stellt die Programme zunächst so auf, als ob sie ab Trommelzelle 0 gespeichert werden sollen und hängt an alle Adressen, die sich auf eine solche Trommelzelle "relativ zum Anfang" beziehen (Relativadressen), das Bandbefehlszeichen An. Beim Einlesen wird dann zu der jeweiligen Relativadresse die im Speicher n stehende tatsächliche Anfangsadresse addiert und der Befehl mit der fertigen Adresse (Absolutadresse) weggespeichert.

... An <n> zum eingelesenen Befehl ... addieren

Als Speicher n für die tatsächliche Anfangsadresse kann Speicherzelle 26, jede Schnellspeicherzelle ab 100 oder eine beliebige Trommelzelle verwendet werden. Sehr häufig wird die Schnellspeicherzelle 26 dafür benutzt, da sie sich besonders leicht mit der entsprechenden Anfangsadresse füllen läßt, wie sogleich zu erkennen ist.

Das Einlesen eines relativadressierten Programms kann auf zwei Arten erfolgen:

1) Dem Programm wird der Vorstreifen

UmU
Zo+1E

vorausgeschickt, und das eigentliche Programm wird durch den Bandbefehl

CKI19+8191E

eingeleitet.

Durch UmU wird die Stelle m festgelegt, von der ab das Programm gespeichert werden soll; das U-Bandbefehlszeichen bewirkt, daß der Befehl Um nach Schnellspeicherzelle 19 gebracht wird (vergleiche 3.9). Durch Zo+1E stoppt die Maschine mit Sprung auf die Startadresse des Leseprogramms zum Einlesen des folgenden

Programmstreifens. Der Bandbefehl CKI19+8191E bewirkt dann, daß der Trommeladressenteil des Inhalts von 19, also die Anfangsadresse m, in die Schnellspeicherzelle 26 gebracht wird.

2) Dem Programm wird der Vorstreifen

CBmE
Zo+1E

vorausgeschickt, und das eigentliche Programm wird durch den Bandbefehl

UoA26U

eingeleitet.

Durch CBmE wird m in die Schnellspeicherzelle 26 gebracht. Durch Zo+1E stoppt die Maschine mit Sprung auf die Startadresse des Leseprogramms zum Einlesen des Programmstreifens. Der Bandbefehl UoA26U bewirkt dann, daß die in 26 angegebene Adresse m zu Uo addiert und der so entstehende Befehl Um infolge des U-Bandbefehlszeichens nach Schnellspeicherzelle 19 gebracht wird.

Diese beiden Möglichkeiten für das Einlesen eines relativ-adressierten Programms waren speziell für die Verwendung der Schnellspeicherzelle 26 als Anfangsadressenspeicher formuliert, was in den Programmen durch die angehängten Bandbefehle A26 zum Ausdruck kommt.

Will man durch die Bandbefehle An auf eine andere Zelle n bezugnehmen, so muß man im ersten Fall an den Befehl CKI19+8191E des Programmstreifens noch den Umspeicherbandbefehl UnE (d.h. <26> → n) anhängen. Auch im zweiten Fall ist eine sinnvolle Umformung für beliebige Anfangsadressenspeicher möglich.

8.2 Ein Anwendungsbeispiel für Relativadressen

Es soll ein Unterprogramm für das Skalarprodukt zweier n-dimensionaler Vektoren aufgestellt werden:

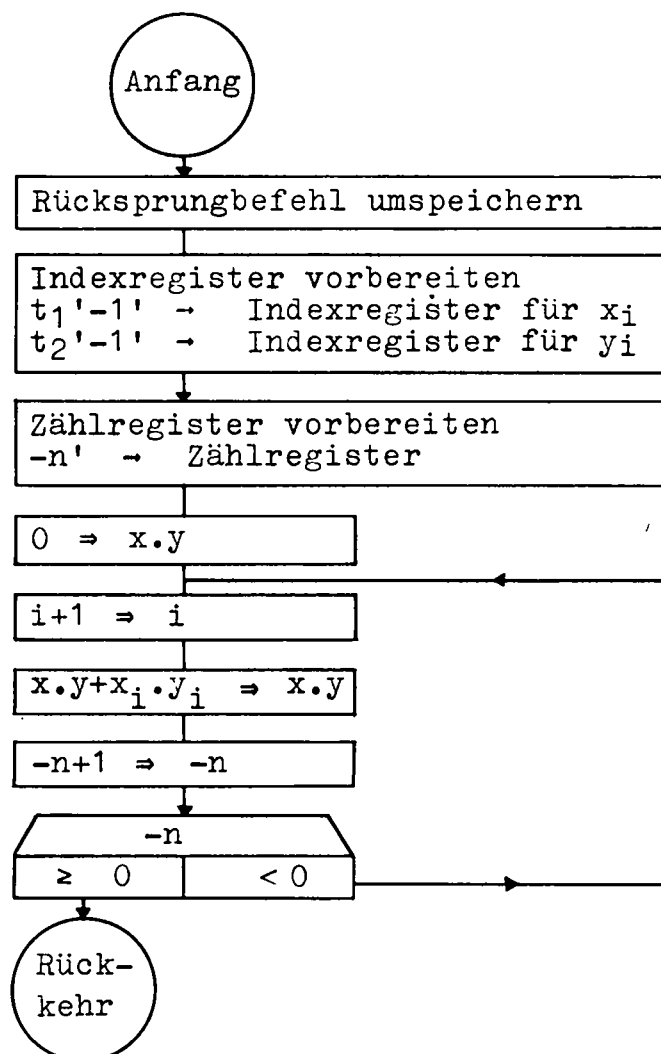
$$x \cdot y = \sum_{i=1}^n x_i y_i$$

Wenn sich das Unterprogramm die jeweils benötigten Komponenten x_i , y_i durch G-Befehle holen soll, müssen vorher die Adressen der ersten Komponenten in je einem Indexregister und die Dimension in einem Zählregister notiert werden. Diese Angaben können im Hauptprogramm zwischen der Stelle t des Sprungs in das Unterprogramm und der des Rücksprungs gemacht werden (der Rücksprungbefehl, der normalerweise nach t+1 zielt, muß dann natürlich entsprechend abgeändert werden). Die Speicherliste für das Hauptprogramm sieht somit wie folgt aus:

t	Fm	Aufruf des Unterprogramms Skalarprodukt
t+1	t ₁ '	Adresse von x ₁
t+2	t ₂ '	Adresse von y ₁

t+3 n' Dimension der Vektoren (evtl. auch -n')
t+4 Fortsetzung des Hauptprogramms

Flußdiagramm für das Unterprogramm Skalarprodukt



Speicherliste für das Unterprogramm

Relative Zellen-Nr.	Inhalt	Erläuterungen
0 ff		Unterprogramm
Absolute Zellen-Nr.		
20	{x _i }	Indexregister für x _i
21	{y _i }	Indexregister für y _i
22	-n'	Zählregister
23	x.y	Zwischenspeicher und Speicher für das Skalarprodukt

Unterprogramm

Relative Zellen-Nr.	Inhalt	Erläuterungen
0	B5	Rücksprungbefehl (mit um 3 erhöhter Adresse) umspeichern in Relativzelle 21
1	CA3	
2	U21A26	
3	GB5+0	$t_1' \rightarrow a$
4	CS1	$t_1' - 1' = \{x_1\} - 1 \rightarrow 20$
5	U20	
6	GB5+1	$t_2' \rightarrow a$
7	CS1	$t_2' - 1' = \{y_1\} - 1 \rightarrow 21$
8	U21	
9	GNS5+2	$-n' \rightarrow 22$
10	U22	
1	CKB23+0	$0 \Rightarrow x.y \rightarrow 23$
2	CGB20+1	$\{x_i\} + 1 \Rightarrow \{x_i\} \rightarrow 20, x_i \rightarrow a,6$
3	U6	
4	CGB21+1	$\{y_i\} + 1 \Rightarrow \{y_i\} \rightarrow 21, y_i \rightarrow a$
5	X	
6	B23	$x.y + x_i.y_i \Rightarrow x.y \rightarrow 23$
7	+	
8	U23	
9	CKA22+1	$(-n+1)' \Rightarrow -n' \rightarrow 22$
20	QE22+12A26	Wenn $-n < 0$, Sprung auf Relativzelle 12
21	Eo	Rücksprungbefehlszelle

8.3 Symbolische Adressen

Beim Aufstellen von Programmen möchte man vielfach nicht gleich zu Beginn fortlaufend nummerierte Adressen verwenden, da jede nachträgliche Veränderung Adressenverschiebungen hervorruft. Wenn man "adressenlos" programmiert, kann man die Numerierung der Speicherplätze einfach weglassen. Anstelle der Trommeladressen setzt man Namen in die Befehle ein (symbolische Adressen) und bezeichnet diejenigen Programmstellen, auf die diese Befehle bezugnehmen, mit dem gleichen Namen. Ein zusätzliches Leseprogramm (Adressierprogramm) ermöglicht es der ZUSE Z 23, bei der Eingabe des Programmfelds jedem Namenssymbol eine bestimmte Adresse zuzuordnen und diese Adresse in die betreffenden Befehle einzusetzen. Es ist nunmehr gleichgültig, ob man an irgendwelchen Programmstellen nachträglich noch etwas einfügt oder wegläßt, da hierdurch beim Einlesen in die Maschine lediglich eine andere Zuordnung Name-Adresse vorgenommen wird.

Die beim adressenlosen Programmieren verwendeten Namen können aus bis zu 5 Fernschreibzeichen bestehen und werden stets in runde Klammern eingeschlossen. Aus praktischen Gründen verwendet man häufig Abkürzungssymbole, die mit dem Programm in irgendeinem Zusammenhang stehen. Für einen Sprungbefehl auf das U.P.Sinus würde man z.B. schreiben F(SIN). Es können aber auch Befehle der Form B6(A) gegeben werden, worin 6(A) als 6.Zelle nach der mit (A) bezeichneten Stelle gelesen wird.

Das (-Zeichen vor jedem Namen führt beim Einlesen vom Lese- auf das Adressierprogramm (Sprung auf Zelle 1043, in der der Sprungbefehl zum Adressierprogramm notiert ist). Das)-Zeichen hat die umgekehrte Wirkung, indem es das Ende des Symbols und den Rücksprung ins Leseprogramm veranlaßt (Sprung auf Zelle 1044, in der der Rücksprungbefehl zum Leseprogramm notiert ist).

Jeder zur Kennzeichnung einer Programmstelle gesetzte Name (vor dem Befehl stehend) wird mit der Adresse der nächstniedrigen noch freien Trommelspeicherzelle versehen und in eine "Adreßbuch" genannte Speicherliste eingetragen. Bei jedem auf den Operationsteil eines Befehls folgenden Namen wird zunächst festgestellt, ob dieser Name bereits im Adreßbuch steht. Wenn ja, wird die betreffende Adresse dem zu speichernden Befehl hinzugesetzt; andernfalls wird der noch unfertige Befehl vorläufig weggespeichert und der Name, mit der Adresse der Speicherzelle versehen, in eine "Vormerkbuch" genannte weitere Speicherliste eingetragen. Beim Einlesen eines vor einem Befehl stehenden Namens wird jeweils festgestellt, ob dieser bereits im Vormerkbuch notiert ist. Ist dies der Fall, so kann jetzt mindestens ein unfertiger Befehl vollständig adressiert und der betreffende Name im Vormerkbuch gestrichen werden.

Adreßbuch und Vormerkbuch können von der Maschine ausgedruckt werden, was beim Ausprüfen eines Programms von Vorteil ist. Es erscheinen dann jeweils Name und Adresse nebeneinander. Wenn mindestens eine Taste des Bedingungsschalters 14 eingedrückt ist ($\langle 14 \rangle > 0$), wird während des Einlesens jeder ins Adreßbuch eingelesene Name mit der zugeordneten Adresse ausgeschrieben. Durch ein alleinstehendes)-Zeichen läßt sich anschließend auch das Vormerkbuch ausdrucken. Ist keine Taste des Bedingungsschalters eingedrückt, wird während des Einlesens nichts ausgeschrieben. Erscheint dann am Schluß das alleinstehende)-Zeichen, so werden jetzt Adreßbuch und Vormerkbuch ausgedruckt. Das Vormerkbuch kann nur dann noch Eintragungen enthalten, wenn in Befehlen enthaltene Namen nicht auch zur Kennzeichnung einer benutzten Speicherzelle verwendet oder wenn zwei einander entsprechende Namen versehentlich verschieden geschrieben wurden. Durch den Befehl V(NAM) können alle von der Stelle NAM ab im Adreßbuch stehenden Namen wieder gelöscht werden. Steht dieser Name nicht im Adreßbuch, stoppt die Maschine und löscht nach dem Start sämtliche Eintragungen des Adreßbuchs und des Vormerkbuchs.

) und $\langle 14 \rangle > 0$ Nach dem Einlesen Vormerkbuch ausschreiben (das Adreßbuch wird während des Einlesens ausgeschrieben).

) und <14> = 0 Nach dem Einlesen Adreßbuch und Vor-
merkbuch ausschreiben

V(NAM) Stop der Maschine, wenn NAM nicht im
Adreßbuch steht. Nach Start Löschung
des Adreß- und des Vormerkbuchs.

Die Eingabe eines symbolisch adressierten Programms erfolgt so,
daß zunächst das benötigte Adressierprogramm mit dem Vorstreifen

UmU
n' (Länge des Adreß- und Vormerkbuchs)
ZO+1E

eingelezen wird. Anschließend folgt das Programm mit dem Vor-
streifen

UmU
ZO+1E

wobei gegebenenfalls der Bedingungsschalter 14 verwendet werden
kann.

8.4 Ein Anwendungsbeispiel für symbolische Adressen

Gegeben seien für $1 \leq i \leq n$ die Zahlenwerte von l_i , r_i und $\sin \alpha_i$.

Es sollen die Ausdrücke $p_i \equiv \frac{l_i}{r_i} \cdot \sin \alpha_i$ berechnet werden.

Die Größen l_i seien ab (L), die r_i ab (R), die α_i ab (ALF)
und der Zahlenwert n' an der Stelle (N) gespeichert. Das
U.P.Sinus stehe ab (SIN).

Der Programmstreifen zur Berechnung der p_i sieht dann wie folgt
aus:

(L)	l_1		F(SIN)
	...		GB35+(L)
	l_n		X
(R)	r_1		GB35+(R)
	...		:
	r_n		GU35+(P)
(ALF)	α_1		CKA35+1
	...		B35
	α_n		S(N)
(N)	n'		QQE1(ANF)
(ANF)	CKB35+0	(P)	ZO
	GB35+(ALF)		0
			...
			ZE(ANF)E

A N H A N G

Befehlsliste für die ZUSE Z 23

Liste der Schnellspeicher mit Sonderfunktionen

Oktaltabelle

Der Fernschreibcode

Frontansicht des Bedienungspults

Beschreibung des ZUSE-Formelcodes ("Kleiner Formelübersetzer")

Befehlsliste für die ZUSE Z 23

Befehl	Wirkung	
	1) Sprung- und Rufbefehle	
Et	$\langle t \rangle \rightarrow b,$	$E(t+1) \rightarrow c$
Es	$\langle s \rangle \rightarrow b,$	$E(s+1) \rightarrow c$
PEs+t	Wenn $\langle s \rangle \geq 0,$ $\langle t \rangle \rightarrow b,$ $E(t+1) \rightarrow c;$ wenn $\langle s \rangle < 0,$ wird nächster Befehl geholt	
QEs+t	Wenn $\langle s \rangle < 0,$ $\langle t \rangle \rightarrow b,$ $E(t+1) \rightarrow c;$ wenn $\langle s \rangle \geq 0,$ wird nächster Befehl geholt (Mit P, Q Test für alle $s \geq 6$ möglich)	
Fm	$\langle m \rangle \rightarrow b,$	$\langle c \rangle \rightarrow 5, E(m+1) \rightarrow c$
	2) Lese- und Speicherbefehle	
Bm	$\langle m \rangle \rightarrow a,$	$\langle m \rangle \rightarrow m$
Bs+t	$\langle t \rangle \rightarrow s,$	$\langle a \rangle \rightarrow a$
Um	$\langle a \rangle \rightarrow m,$	$\langle a \rangle \rightarrow a$
Us+t	$\langle s \rangle \rightarrow t,$	$\langle s \rangle \rightarrow s, \langle a \rangle \rightarrow a$
NUM	$\langle a \rangle \rightarrow m,$	$o \rightarrow a$
NUs+t	$\langle s \rangle \rightarrow t,$	$o \rightarrow s, \langle a \rangle \rightarrow a$
	3) Blocktransferbefehle Vor jedem Befehl steht noch CKB13+n ($n' \rightarrow 13$). Das i läuft dann von 0 bis n. (x)	
BVs+t	$\langle t+i \rangle \rightarrow s+i$	
UVs+t	$\langle s+i \rangle \rightarrow t+i$	
NUVs+t	$\langle s+i \rangle \rightarrow t+i,$ danach stets $0 \rightarrow s+i$	
AVs+t	$\langle s+i \rangle' + \langle t+i \rangle' \rightarrow s+i$	(x)
SVs+t	$\langle s+i \rangle' - \langle t+i \rangle' \rightarrow s+i$	(x)
NSVs+t	$-\langle t+i \rangle' \rightarrow s+i$	
USVs+t	$\langle t+i \rangle' - \langle s+i \rangle' \rightarrow s+i$	
IVs+t	Unbrauchbar, denn nur $\langle s+n \rangle \wedge \langle t+n \rangle \rightarrow a$ bleibt	
ØVs+t	$\langle s+i \rangle \vee \langle t+i \rangle \rightarrow \langle s+i \rangle$	(x)
...0+t	Für $s=0$ ist $s+i$ durch a zu ersetzen (x)	
Befehl	Wirkung	Benutzte Schnell- speicher
	4) Arithmetische Befehle	
+	$\langle 6 \rangle + \langle a \rangle \rightarrow a, 6$	} 2-6, 11-13, 16, 17
-	$\langle 6 \rangle - \langle a \rangle \rightarrow a, 6$	
X	$\langle 6 \rangle \cdot \langle a \rangle \rightarrow a, 6$	
:	$\langle 6 \rangle : \langle a \rangle \rightarrow a, 6$	
W	$\sqrt{\langle a \rangle} \rightarrow a, 6$	} 2-6, 11-13, 16-18
W2	$\sqrt{1 - \langle a \rangle^2} \rightarrow a, 6$	
M	$(-1) \cdot \langle a \rangle \rightarrow a, 6$	4-6

Befehl	Wirkung	Benutzte Schnell- speicher
	5) Druck- und Tabellierbefehle	
D D2	$\langle a \rangle \rightarrow$ FS (Fernschreiber); $\langle a \rangle \rightarrow a,6$ $\langle a \rangle \rightarrow$ FS bei $\langle 1029 \rangle \geq 0$, $\langle a \rangle \rightarrow$ SL (Schnellocher) bei $\langle 1029 \rangle < 0$; $\langle a \rangle \rightarrow a,6$	2-6,11-13 16-18
F885	$\langle a \rangle \rightarrow$ FS bei $\langle 2 \rangle \geq 0$, $\langle a \rangle \rightarrow$ SL bei $\langle 2 \rangle < 0$; $0 \rightarrow a$	2-5
	(Klartext ohne Klingelzeichen bzw. Semikolon)	
F1000	Wagen- und Zeilentransport auf FS; $\langle a \rangle \rightarrow a,6$	
F1002	W.u.Z. auf FS bei $\langle 1029 \rangle \geq 0$, auf SL bei $\langle 1029 \rangle < 0$; $\langle a \rangle \rightarrow a,6$	2-6
F800 F802	Spaltensprung auf FS; Sp.auf FS bei $\langle 1029 \rangle \geq 0$, auf SL bei $\langle 1029 \rangle < 0$; $\langle a \rangle \rightarrow a,6$	2-6,11
F840 F842	Weiterrücken auf FS um $-\langle a \rangle$ Zeichen W.auf FS bei $\langle 1029 \rangle \geq 0$, auf SL bei $\langle 1029 \rangle < 0$	2-5,11
F850 F852	Zeilenvorschub auf FS um $-\langle a \rangle$ Zeilen Z.auf FS bei $\langle 1029 \rangle \geq 0$, auf SL bei $\langle 1029 \rangle < 0$	2-5,11

Druckanordnung

U1030U Die folgenden Angaben werden ab $t=1030$ gespeichert

l' Stellenanzahl der Mantisse bei Gleitkommazahlen

a' Stellenanzahl vor dem Komma } bei Normalform

B' Stellenanzahl nach dem Komma }

k' Anzahl der Spalten

z' Anzahl der Zwischenraumstellen

n' Stellenanzahl der Strichzahlen

+1' bzw. -1' Ausgabe in Gleitkomma- bzw. Normalform

Befehl	Wirkung
	6) Bedingte Befehle
PP...	$\langle a \rangle \geq 0$
QQ...	$\langle a \rangle < 0$
PPQQ...	$\langle a \rangle = 0$
P...	$\langle 2 \rangle \geq 0$
Q...	$\langle 2 \rangle < 0$
Y...	$\langle 3 \rangle_{40} = 1$
PQQQ	$\langle a \rangle_1 \neq \langle a \rangle_2$

Befehl	Wirkung	
	7) Stopbefehle	
Zm PPZm PZs+t	Stop der Maschine. Nach erneutem Starten Sprung auf Zelle m. Stop nur, wenn $\langle a \rangle \geq 0$ Stop, wenn $\langle s \rangle \geq 0$. Nach erneutem Starten Sprung auf Zelle t.	
	8) Bandbefehle	
UmU ...E ...An	Die folgenden Angaben ab Zelle m speichern. Den Befehl ... sofort ausführen (s=26 tritt hierbei an die Stelle von a). Zum Befehl ... die Zahl n' addieren (Relativadresse +n' = Absolutadresse).	
Befehl	Wirkung	Benutzte Schnell- speicher
	9) Arithmetische Befehle für Strichzahlen	
Am As+t Sm Ss+t USm USs+t NSm NSs+t X' :' F1020	$\langle a \rangle' + \langle m \rangle' \rightarrow a$ $\langle s \rangle' + \langle t \rangle' \rightarrow s, \langle a \rangle \rightarrow a$ $\langle a \rangle' - \langle m \rangle' \rightarrow a$ $\langle s \rangle' - \langle t \rangle' \rightarrow s, \langle a \rangle \rightarrow a$ $\langle m \rangle' - \langle a \rangle' \rightarrow a$ $\langle t \rangle' - \langle s \rangle' \rightarrow s, a \rightarrow a$ $-\langle m \rangle' \rightarrow a$ $-\langle t \rangle' \rightarrow s, a \rightarrow a$ $\langle 6 \rangle' \cdot \langle a \rangle' \rightarrow a,6$ $\langle 6 \rangle' : \langle a \rangle' \rightarrow a,6: \text{Rest} \rightarrow 2$ $\langle a \rangle' \rightarrow a,6$ als Gleitkommazahl	2 - 6 2 - 6 2-6,12,13, 16
	10) Befehle für Konstantenzuführung	
CAn CSn CUSn CNA CNSn CKAs+t CKSs+t CKUSs+t CKBs+t CKNSs+t	$\langle a \rangle' + n' \rightarrow a$ $\langle a \rangle' - n' \rightarrow a$ $n' - \langle a \rangle' \rightarrow a$ $n' \rightarrow a$ $-n' \rightarrow a$ $\langle s \rangle' + t' \rightarrow s, \langle a \rangle \rightarrow a$ $\langle s \rangle' - t' \rightarrow s, \langle a \rangle \rightarrow a$ $t' - \langle s \rangle' \rightarrow s, \langle a \rangle \rightarrow a$ $t' \rightarrow s, \langle a \rangle \rightarrow a$ $-t' \rightarrow s, \langle a \rangle \rightarrow a$	

Befehl	Wirkung	
	11) Befehle für logische Operationen	
Im	$\langle a \rangle^{\langle m \rangle} \rightarrow a$	
Is+t	$\langle s \rangle^{\langle t \rangle} \rightarrow a, \langle s \rangle \rightarrow s$	
ISm	$\langle a \rangle^{\neg \langle m \rangle} \rightarrow a$	
ISS+t	$\langle s \rangle^{\neg \langle t \rangle} \rightarrow a, \langle s \rangle \rightarrow s$	
CIn	$\langle a \rangle^{\langle n \rangle'} \rightarrow a$	
CIs+t	$\langle a \rangle^{\langle s+t \rangle'} \rightarrow a$	
CISn	$\langle a \rangle^{\neg \langle n \rangle'} \rightarrow a$	
CISs+t	$\langle a \rangle^{\neg \langle s+t \rangle'} \rightarrow a$	
CKIs+t	$\langle s \rangle^{\langle t \rangle'} \rightarrow a, \langle s \rangle \rightarrow s$	
CKISS+t	$\langle s \rangle^{\neg \langle t \rangle'} \rightarrow a, \langle s \rangle \rightarrow s$	
$\emptyset m$	$\langle a \rangle^{\vee \langle m \rangle} \rightarrow a$	
$\emptyset s+t$	$\langle s \rangle^{\vee \langle t \rangle} \rightarrow s, \langle a \rangle \rightarrow a$	
$\emptyset n$	$\langle a \rangle^{\vee \langle n \rangle'} \rightarrow a$	
$\emptyset s+t$	$\langle a \rangle^{\vee \langle s+t \rangle'} \rightarrow a$	
CK $\emptyset s+t$	$\langle s \rangle^{\vee \langle t \rangle'} \rightarrow s, \langle a \rangle \rightarrow a$	
	12) Befehle für Adressenänderungen	
G θt	$\langle t \rangle_{28...40} \Rightarrow \hat{t}, \text{ dann } \theta \hat{t} \quad (\theta = \text{Operations- teil})$	
G $\theta s+0$	$\langle s \rangle_{28...40} \Rightarrow \hat{t}, \text{ dann } \theta \hat{t}$	
G $\theta s+t$	$(\langle s \rangle + t)_{28...40} \Rightarrow \hat{t}, \text{ dann } \theta \hat{t}; \langle s \rangle \rightarrow s$	
GK θt	nicht möglich, da K den Weg zur Trommel sperrt	
GK $\theta s+0$	$\langle s \rangle_{33...40} \Rightarrow \hat{s}, \text{ dann } \theta \hat{s}$	
GK $\theta s+t$	$(\langle s \rangle + t)_{33...40} \Rightarrow \hat{s}, \text{ dann } \theta \hat{s}; \langle s \rangle \rightarrow s$	
CG $\theta s+t$	$(\langle s \rangle + t)_{28...40} \Rightarrow \hat{t}, \text{ dann } \theta \hat{t}; \langle s \rangle + t \rightarrow s$	
CGK $\theta s+t$	$(\langle s \rangle + t)_{33...40} \Rightarrow \hat{s}, \text{ dann } \theta \hat{s}; \langle s \rangle + t \rightarrow s$	
Befehl	Wirkung	Benutzte Schnell- speicher
	13) Aufruf des Leseprogramms	
Eo+1	Leseprogramm mit Übertragung der Entschlüsselungstabelle in die Kernspeicher (Entschlüsselungszeit für ein Zeichen 3 msec)	2 - 99
Eo+6	Leseprogramm ohne Übertragung der Entschlüsselungstabelle in die Kernspeicher (Entschlüsselungszeit für ein Zeichen 20 msec)	2 - 32
F522	1 Wort vom Lochstreifen ablesen und in den Akkumulator bringen	2 - 32

Befehl	Wirkung
	14) Zusätzliche Befehle für die ZUSE Z 23V
	a) Bedingter Blocktransfer
	Vor jedem Befehl steht noch CKB13+n (n' → 13). Das v läuft dann von 0 bis n.
PPθVs+t QQθVs+t H Vs+t mit θ = A, S, US, Ø, I oder IS	Die Operation θ wird so lange zwischen ⟨s⟩ und ⟨t+v⟩ ausgeführt (Ergebnis jeweils in a), bis Zähler 13 auf Null zurückgezählt hat oder bis bei PP ⟨a⟩ < 0, bei QQ ⟨a⟩ ≥ 0, bei H ⟨a⟩ = 0 geworden ist (s < 16).
	b) Schnelle Multiplikation und Division
LØKO PQLØK1-n RØKO PQRØK1-n	$\left. \begin{array}{l} \langle a \rangle \cdot 10 \rightarrow a \\ \langle a \rangle \cdot 10^n \rightarrow a \\ \langle a \rangle \cdot 10^{-1} \rightarrow a \\ \langle a \rangle \cdot 10^{-n} \rightarrow a \end{array} \right\} \text{Rest} \rightarrow 3$
	c) Schnelles Zahlenlesen
CKGLA15+t oder CKGLA10+t	Vor.: In s=t, t+1, ... t+31 muß für jedes Fernschreibzeichen ein dem beabsichtigten Lesevorgang entsprechender Befehl stehen. Zunächst $\langle 15 \rangle + t \Rightarrow \hat{s}$; dann $\langle \hat{s} \rangle \rightarrow b$ (bei Befehlen) bzw. $\langle a \rangle \cdot 10 + \langle \hat{s} \rangle \rightarrow a$ (bei Zahlen). Unter- brechung des Einlesens nur bei Sprungbefehlen.

Liste der Schnellspeicher mit Sonderfunktionen

Adresse	Wirkung beim Lesen	Wirkung beim Schreiben
0	liefert 0	
1	liefert 2^{39}	Schnellocher (150 Zeichen/sec)
2	Test P, Q bei beliebigen Befehlen	
3	Verlängerung des Akkus; Test Y auf letzte Stelle	
4	Akkumulator; Test PP, QQ, PPQQ, PQQQ	Übernahme der Handeinstellung
5	Rückkehradressenspeicher	
6	Ohne Besonderheit; wird bei arithmetischen Operationen als Speicher für den ersten Operanden benutzt	
7	} für Sonderzwecke vorgesehen (Zusatzgeräte)	
8		
9		
10		
11	Ohne Besonderheit	
12	≡ 5	
13	Zähler für Serienschiebung und Blocktransfer (nur 8 Binärstellen!)	
14	Bedingungsschalter (nur 5 Binärstellen!)	
15	Ferranti-Abtaster (bis 300 Zeichen/sec)	Fernschreiber (10 Zeichen/sec)
16	} Ohne Besonderheit. Bei UmU-Bandbefehlen wird der Befehl Um in s=19 notiert. Bei E-Bandbefehlen wird s=26 als Hilfsakkumulator verwendet.	
.		
.		
.		
255		

Für Trommelsprungbefehle sind testbar (PEs+t, QEs+t) die Kernspeicher 2, 6, 11 und 16 bis 255.

Oktaltabelle

0'	0000	704'	01300	1408'	02600	2112'	04100	2800'	05360
16'	00020	720'	01320	1424'	02620	2128'	04120	2816'	05400
32'	00040	736'	01340	1440'	02640	2144'	04140	2832'	05420
48'	00060	752'	01360	1456'	02660	2160'	04160	2848'	05440
64'	00100	768'	01400	1472'	02700	2176'	04200	2864'	05460
80'	00120	784'	01420	1488'	02720	2192'	04220	2880'	05500
96'	00140							2896'	05520
		800'	01440	1504'	02740	2208'	04240		
112'	00160	816'	01460	1520'	02760	2224'	04260	2912'	05540
128'	00200	832'	01500	1536'	03000	2240'	04300	2928'	05560
144'	00220	848'	01520	1552'	03020	2256'	04320	2944'	05600
160'	00240	864'	01540	1568'	03040	2272'	04340	2960'	05620
176'	00260	880'	01560	1584'	03060	2288'	04360	2976'	05640
192'	00300	896'	01600					2992'	05660
				1600'	03100	2304'	04400		
208'	00320	912'	01620	1616'	03120	2320'	04420	3008'	05700
224'	00340	928'	01640	1632'	03140	2336'	04440	3024'	05720
240'	00360	944'	01660	1648'	03160	2352'	04460	3040'	05740
256'	00400	960'	01700	1664'	03200	2368'	04500	3056'	05760
272'	00420	976'	01720	1680'	03220	2384'	04520	3072'	06000
288'	00440	992'	01740	1696'	03240			3088'	06020
						2400'	04540		
304'	00460	1008'	01760	1712'	03260	2416'	04560	3104'	06040
320'	00500	1024'	02000	1728'	03300	2432'	04600	3120'	06060
336'	00520	1040'	02020	1744'	03320	2448'	04620	3136'	06100
352'	00540	1056'	02040	1760'	03340	2464'	04640	3152'	06120
368'	00560	1072'	02060	1776'	03360	2480'	04660	3168'	06140
384'	00600	1088'	02100	1792'	03400	2496'	04700	3184'	06160
400'	00620	1104'	02120	1808'	03420			3200'	06200
416'	00640	1120'	02140	1824'	03440	2512'	04720	3216'	06220
432'	00660	1136'	02160	1840'	03460	2528'	04740	3232'	06240
448'	00700	1152'	02200	1856'	03500	2544'	04760	3248'	06260
464'	00720	1168'	02220	1872'	03520	2560'	05000	3264'	06300
480'	00740	1184'	02240	1888'	03540	2576'	05020	3280'	06320
496'	00760					2592'	05040	3296'	06340
		1200'	02260	1904'	03560				
512'	01000	1216'	02300	1920'	03600	2608'	05060	3312'	06360
528'	01020	1232'	02320	1936'	03620	2624'	05100	3328'	06400
544'	01040	1248'	02340	1952'	03640	2640'	05120	3344'	06420
560'	01060	1264'	02360	1968'	03660	2656'	05140	3360'	06440
576'	01100	1280'	02400	1984'	03700	2672'	05160	3376'	06460
592'	01120	1296'	02420			2688'	05200	3392'	06500
				2000'	03720				
608'	01140	1312'	02440	2016'	03740	2704'	05220	3408'	06520
624'	01160	1328'	02460	2032'	03760	2720'	05240	3424'	06540
640'	01200	1344'	02500	2048'	04000	2736'	05260	3440'	06560
656'	01220	1360'	02520	2064'	04020	2752'	05300	3456'	06600
672'	01240	1376'	02540	2080'	04040	2768'	05320	3472'	06620
688'	01260	1392'	02560	2096'	04060	2784'	05340	3488'	06640

3504'	06660	4203'	10160	4912'	11460	5600'	12740	6304'	14240
3520'	06700	4224'	10200	4928'	11500	5616'	12760	6320'	14260
3536'	06720	4240'	10220	2944'	11520	5632'	13000	6336'	14300
3552'	06740	4256'	10240	4060'	11540	5648'	13020	6352'	14320
3568'	06760	4272'	10260	4976'	11560	5664'	13040	6368'	14340
3584'	07000	4288'	10300	4992'	11600	5680'	13060	6384'	14360
						5696'	13100		
3600'	07020	4304'	10320					6400'	14400
3616'	07040	4320'	10340	5008'	11620	5712'	13120	6416'	14420
3632'	07060	4336'	10360	5024'	11640	5728'	13140	6432'	14440
3648'	07100	4352'	10400	5040'	11660	5744'	13160	6448'	14460
3664'	07120	4368'	10420	5056'	11700	5760'	13200	6464'	14500
3680'	07140	4384'	10440	5072'	11720	5776'	13220	6480'	14520
3696'	07160			5088'	11740	5792'	13240	6496'	14540
		4400'	10460						
3712'	07200	4416'	10500	5104'	11760	5808'	13260	6512'	14560
3728'	07220	4432'	10520	5120'	12000	5824'	13300	6528'	14600
3744'	07240	4448'	10540	5136'	12020	5840'	13320	6544'	14620
3760'	07260	4464'	10560	5152'	12040	5856'	13340	6560'	14640
3776'	07300	4480'	10600	5168'	12060	5872'	13360	6576'	14660
3792'	07320	4496'	10620	5184'	12100	5888'	13400	6592'	14700
3808'	07340	4512'	10640	5200'	12120	5904'	13420	6608'	14720
3824'	07360	4528'	10660	5216'	12140	5920'	13440	6624'	14740
3840'	07400	4544'	10700	5232'	12160	5936'	13460	6640'	14760
3856'	07420	4560'	10720	5248'	12200	5952'	13500	6656'	15000
3872'	07440	4576'	10740	5264'	12220	5968'	13520	6672'	15020
3888'	07460	4592'	10760	5280'	12240	5984'	13540	6688'	15040
				5296'	12260				
						6000'	13560	6704'	15060
3904'	07500	4608'	11000			6016'	13600	6720'	15100
3920'	07520	4624'	11020	5312'	12300	6032'	13620	6736'	15120
3936'	07540	4640'	11040	5328'	12320	6048'	13640	6752'	15140
3952'	07560	4656'	11060	5344'	12340	6064'	13660	6768'	15160
3968'	07600	4672'	11100	5360'	12360	6080'	13700	6784'	15200
3984'	07620	4688'	11120	5376'	12400	6096'	13720		
				5392'	12420				
4000'	07640	4704'	11140					6800'	15220
4016'	07660	4720'	11160	5408'	12440	6112'	13740	6816'	15240
4032'	07700	4736'	11200	5424'	12460	6128'	13760	6832'	15260
4048'	07720	4752'	11220	5440'	12500	6144'	14000	6848'	15300
4064'	07740	4768'	11240	5456'	12520	6160'	14020	6864'	15320
4080'	07760	4784'	11260	5472'	12540	6176'	14040	6880'	15340
4096'	10000			5488'	12560	6192'	14060	6896'	15360
		4800'	11300						
4112'	10020	4816'	11320	5504'	12600	6208'	14100	6912'	15400
4128'	10040	4832'	11340	5520'	12620	6224'	14120	6928'	15420
4144'	10060	4848'	11360	5536'	12640	6240'	14140	6944'	15440
4160'	10100	4864'	11400	5552'	12660	6256'	14160	6960'	15460
4176'	10120	4880'	11420	5568'	12700	6272'	14200	6976'	15500
4192'	10140	4896'	11440	5584'	12720	6288'	14220	6992'	15520

7008'	15540	7312'	16220	7600'	16660	7904'	17340
7024'	15560	7328'	16240	7616'	16700	7920'	17360
7040'	15600	7344'	16260	7632'	16720	7936'	17400
7056'	15620	7360'	16300	7648'	16740	7952'	17420
7072'	15640	7376'	16320	7664'	16760	7968'	17440
7088'	15660	7392'	16340	7680'	17000	7984'	17460
				7696'	17020		
7104'	15700	7408'	16360			8000'	17500
7120'	15720	7424'	16400	7712'	17040	8016'	17520
7136'	15740	7440'	16420	7728'	17060	8032'	17540
7152'	15760	7456'	16440	7744'	17100	8048'	17560
7168'	16000	7472'	16460	7760'	17120	8064'	17600
7184'	16020	7488'	16500	7776'	17140	8080'	17620
				7792'	17160	8096'	17640
7200'	16040						
7216'	16060	7504'	16520	7808'	17200	8112'	17660
7232'	16100	7520'	16540	7824'	17220	8128'	17700
7248'	16120	7536'	16560	7840'	17240	8144'	17720
7264'	16140	7552'	16600	7856'	17260	8160'	17740
7280'	16160	7568'	16620	7872'	17300	8176'	17760
7296'	16200	7584'	16640	7888'	17320		

Der Fernschreibcode

Zeitlicher Ablauf Löcher - ●	Bandwert dual Bei der Eingabe	Bei der Ausgabe	Fernschreibziffer oder - Zeichen	Fernschreib- Buchstabe
○○○●○○	0	$0 \cdot 2^{34}$	*	#
○○○●○●	1	$1 \cdot 2^{34}$	3	E
○○○●●○	2	$2 \cdot 2^{34}$	<i>Zeilentransport</i>	
○○○●●●	3	$3 \cdot 2^{34}$	—	A
○○●●○○	4	$4 \cdot 2^{34}$	<i>Zwischenraum</i>	
○○●●○●	5	$5 \cdot 2^{34}$	9	S
○○●●●○	6	$6 \cdot 2^{34}$	8	I
○○●●●●	7	$7 \cdot 2^{34}$	7	U
○●●●○○	8	$8 \cdot 2^{34}$	<i>Wagentransport</i>	
○●●●○●	9	$9 \cdot 2^{34}$	⊗	D
○●●●●○	10	$10 \cdot 2^{34}$	4	R
○●●●●●	11	$11 \cdot 2^{34}$	⊕ oder ;	J
○●●●○○	12	$12 \cdot 2^{34}$	9	N
○●●●○●	13	$13 \cdot 2^{34}$	⊔	F
○●●●●○	14	$14 \cdot 2^{34}$:	C
○●●●●●	15	$15 \cdot 2^{34}$	(K
●○○●○○	16	$16 \cdot 2^{34}$	5	T
●○○●○●	17	$17 \cdot 2^{34}$	+	Z
●○○●●○	18	$18 \cdot 2^{34}$)	L
●○○●●●	19	$19 \cdot 2^{34}$	2	W
●○●●○○	20	$20 \cdot 2^{34}$	10	H
●○●●○●	21	$21 \cdot 2^{34}$	6	Y
●○●●●○	22	$22 \cdot 2^{34}$	0	P
●○●●●●	23	$23 \cdot 2^{34}$	1	Q
●●○●○○	24	$24 \cdot 2^{34}$	9	O
●●○●○●	25	$25 \cdot 2^{34}$	2 oder X	B
●●○●●○	26	$26 \cdot 2^{34}$]	G
●●○●●●	27	$27 \cdot 2^{34}$	<i>Zifferntaste</i>	
●●●●○○	28	$28 \cdot 2^{34}$.	M
●●●●○●	29	$29 \cdot 2^{34}$	/	X
●●●●●○	30	$30 \cdot 2^{34}$	—	V
●●●●●●	31	$31 \cdot 2^{34}$	<i>Buchstabentaste</i>	
Transportloch				

Beschreibung des ZUSE-Formelcodes ("Kleiner Formelübersetzer")

Der Freiburger Code für die Rechenanlagen ZUSE Z 22 und ZUSE Z 23 gilt allgemein als einer der anschaulichsten Maschinencodes. Trotzdem wird häufig nach einem Code gefragt, der mit den direkten mathematischen Formeln arbeitet.

Für die ZUSE Z 23 wurde deshalb der ZUSE-Formelcode ("Kleiner Formelübersetzer") geschaffen, der diesen Wünschen weitgehend entgegenkommt. Der ZUSE-Formelcode ist nur für den Benutzer der ZUSE Z 23 gedacht, folglich nicht so umfangreich und allgemein wie ALGOL, der ja für verschiedene Typen von Rechenanlagen verwendbar sein soll.

Zur Bezeichnung von Funktionen, Sprüngen und Schleifen wurden Abkürzungen aus der deutschen Sprache eingeführt, um auch dem Ingenieur und Techniker eine leicht verständliche und erlernbare Programmierungssprache zu geben.

Der Formelübersetzer verarbeitet einfache Formeln ohne Klammerausdrücke, errechnet Funktionswerte, kann Entscheidungen treffen, Schleifen bilden u.a. Variablen können in einfacher Form oder indiziert auftreten.

Bevor die einzelnen Möglichkeiten erklärt werden, sollen zwei kleine Beispiele gegeben werden, wie ein Programm in dieser Formelsprache aussieht.

Beispiel 1 Bildung von $z = \sqrt{a^2 + b^2}$,

wobei verschiedene Wertepaare a und b eingelesen werden sollen.

=	}	Vorbereitung (Bedeutung siehe unter "Ablochvorschrift")
1		
×	}	1) = Marke zur Kennzeichnung; Einlesen der Größen a und b
1) LIES A		
LIES B	}	Drucken der Größen a und b
DRU A		
DRU B	}	Bildung von $a^2 = c$
C = A.A		
D = B.B	}	Bildung von $b^2 = d$
D = D+C		
D = D+C	}	Bildung von $d = d+c \equiv a^2 + b^2$
Z = WURZ D		
DRU Z	}	Ziehen der Quadratwurzel
DRU Z		
ZLN 1	}	Drucken von Z
SPR 1		
SPR 1	}	1mal Zeilenvorschub
START 1		
START 1	}	Springen auf die mit der Marke 1) bezeichnete Stelle
START 1		
START 1	}	Starten an der mit der Marke 1) bezeichneten Stelle
START 1		

3,78	a
5,91	b
7,843	a
9,586	b
.	
.	
.	

Z0+1E Stop vom Lochstreifen her

Beispiel 2 Für $0 \leq x \leq 4$ sollen mit der Schrittweite
0,125 die Werte der Funktion

$$F(x) = \frac{\sin x}{e^x} \cdot x$$

errechnet werden.

=	}	Vorbereitung
1		
X	}	Eröffnung der Schleife (0=Anfangswert, 4=Endwert; Schrittweite 0,125)
1) FUER X = 0:0,125:4		
A = SIN X	}	Berechnung von F(x)
B = EXP X		
C = A/B		
F = C * X		
DRU X	}	Drucken von x und F(x)
DRU F		
ZLN 1		1 mal Zeilenvorschub
WDH X		Ende der Schleife; Rücksprung, falls der Endwert noch nicht erreicht war
STOP		Stop vom Programm her
START 1		Starten an der mit der Marke 1) bezeichneten Stelle

Erklärung der einzelnen Möglichkeiten des ZUSE-Formel-Codes

1. Nicht indizierte Variable

- A) Gleitkomma-Variable: A, B, . . . Z
(jeweils 1 Buchstabe)
- B) Festkomma-Variable (ganzzahlig): A', B' Z'
(jeweils 1 Buchstabe mit Apostroph)

2. Indizierte Variable

Die Variablen in gleitendem und festem Komma können indiziert werden (Der Index läuft stets ab 0 und kann nur positiv sein).

A) Numerisch angegebener Index:

z.B. wird für die Gl.K.V.

(Gleitkomma-Variable) a_7 $A7'$

und für die F.K.V.

(Festkomma-Variable) b_{17} $B'17'$

geschrieben.

B) Symbolisch angegebener Index:

a) einfacher Index

für die Gl.K.V.

a_i AI'

für die F.K.V.

b_k $B'K'$

b) zusammengesetzter Index

Ein zusammengesetzter Index wird stets in Klammern angegeben und hat die allgemeinste Gestalt:

$a_{n \cdot i + k}$ $A(N' \cdot I' + K')$

Es können hierbei einzelne Teile fehlen oder numerisch angegeben werden, z.B.:

a_{3i+k} $A(3' \cdot I' + K')$

a_{i+1} $A(I'+1')$

a_{n-1} $A(N'-1')$

Es ist zu beachten, daß der resultierende Index nur positiv sein darf.

C) Mehrbuchstabile Variablenbezeichnung

Um gewisse Variable besonders hervorzuheben oder um mehr Variablenbezeichnungen zu erhalten, können für Gleitkomma-variablen auch mehrbuchstabile Namen eingeführt werden. Solche Namen müssen in eckigen Klammern [] angegeben werden und können aus 1 bis 6 Zeichen bestehen (wobei Bu, Zi, leer und Zw unberücksichtigt bleiben und Wr und Zl nicht erlaubt sind):

z.B. [ALPHA]

[B1]

Namen können nicht indiziert werden.

Für die indizierten Variablen und die mit Namen bezeichneten Variablen müssen in der Vorbereitung Platzreservierungen vorgenommen werden (vergl. Ablochvorschrift).

3. Arithmetische Operationen

$A = B+C$	Addition
$A = B-C$	Subtraktion
$A = B.C$	Multiplikation
$A = B/C$	Division

Hier können A, B, C einfache oder indizierte Variablen (in Gleitkomma- oder Festkomma-Darstellung) sein. Vor und hinter dem Gleichheitszeichen brauchen keine Zwischenräume gelassen zu werden.

Die rechts stehenden Größen (hier B und C) müssen beide Gl.K.V oder F.K.V. sein. Dagegen kann links vom Gleichheitszeichen eine Festkomma-Variable verwendet werden, obwohl rechts zwei Gleitkomma-Größen stehen:

z.B. $A' = B.C$

(Der ganzzahlige Teil des Produkts der Gleitkomma-Variablen B und C wird als Festkomma-Variable A' gespeichert).

Rechts des Gleichheitszeichens können auch numerische Größen verwendet werden:

z.B. $A = 3/B$

Dem 1. Operanden kann ein Minuszeichen vorangesetzt werden, welches allerdings bei Addition und Subtraktion auf das Ergebnis angewendet wird:

d.h. $A = -B+C$
steht für $a = -(b+c)$

Rechts des Gleichheitszeichens kann auch nur eine einzige Größe stehen, deren Wert dann für die links angegebene Variable eingesetzt wird:

z.B. $A = B$
 $A' = 4'$

4. Funktionen

Es sind folgende Funktionen vorgesehen, die nur auf Gl.K.V. anwendbar sind:

SIN X	sin x
COS X	cos x
TAN X	tg x
CTG X	ctg x
ARCSIN X	arcsin x
ARCTAN X	arctg x

EXP X	e^x	
LOG X	$\ln x$	
BETR X	$ x $	
WURZ X	\sqrt{x}	
GANZ X	$[x]$	(ganzzahliger Teil von x)
MOD X	$x(\text{mod } 1)$	(Rest mod 1 von x)

5. Marken

Vor jede Formel oder sonstige Anweisung (s.u.) kann eine Marke gesetzt werden als Bezeichnung einer Stelle, auf die gesprungen werden soll. Marken sind numerisch anzugeben und mit dem Zeichen) zu versehen; die darauffolgende Formel ist von der Marke durch einen Zwischenraum zu trennen:

z.B. 2) A = B/C

6. Sprünge

A) Unbedingte Sprünge

SPR 7 Sprung auf die mit der Marke 7 versehene Formel

B) Bedingte Sprünge

WENN A-B 3 7 6

D.h., wenn $a - b < 0$, springe auf Marke 3

$a - b = 0$, springe auf Marke 7

$a - b > 0$, springe auf Marke 6

B kann auch numerisch angegeben werden:

WENN A-3,74 3 7 6

Es kann auch nur eine einzige Größe nach dem WENN stehen:

WENN A 3 7 6

C) Schalterbedingte Sprünge

Um ein Programm während des Ablaufs von außen beeinflussen zu können, kann vom Programm her der Bedingungsschalter am Bedienungspult abgefragt werden.

BED 3:7 5:8

heißt z.B.: Falls der Bedingungsschalter auf 3 steht, springe nach Marke 7, falls er auf 5 steht, springe nach Marke 8; sonst fahre im Programm fort.

Hinter dem Symbol BED können eine oder mehrere Angaben stehen. Durch Übergang auf eine neue Zeile wird das Schlußzeichen für BED gegeben.

D) Sprung auf ein Unterprogramm

UPR 13

Sprung auf das Unterprogramm, das an der Marke 13 beginnt.
Durch

RSP wird das Ende des Unterprogramms bezeichnet
(Rücksprung auf die Stelle nach dem Sprung ins Unter-
programm).

Will man z.B. das Produkt z_3 der beiden komplexen Größen z_1
und z_2 berechnen, also

$$z_3 = z_1 \cdot z_2 \quad \text{mit} \quad \begin{cases} z_1 = x_1 + iy_1 \\ z_2 = x_2 + iy_2 \\ z_3 = x_3 + iy_3 \end{cases}$$

so programmiert man:

UPR7

·
·
·

7) $R = X1' \cdot X2'$

$S = Y1' \cdot Y2'$

$X3' = R - S$

$R = X1' \cdot Y2'$

$S = Y1' \cdot X2'$

$Y3' = R + S$

RSP

7. Zyklen (maximal 12 Zyklen ineinander geschachtelt)

a) für Gleitkommavariablen

FUER A = B:C:D

Die Gl.K.V. A soll vom Anfangswert B mit der Schrittweite
C bis zum Endwert D variiert werden.

Hierbei darf die Schrittweite nur positive Werte haben. Der
Zyklus wird mit WDH A abgeschlossen.

Z.B.: Summe aller ganzen Zahlen von 1 bis 100

S = 0

FUER Z = 1:1:100

S = S+Z

WDH Z

b) für Festkommavariablen

FUER I' = A':B':C'

Formeln, die mit dem veränderlichen Index I' durchlaufen werden sollen, sind direkt unter dem FUER zu schreiben; mit WDH I' ist abzuschließen.

Die Größen rechts vom Gleichheitszeichen (B:C:D bzw. A':B':C') können als Variablen, indizierte Variablen oder numerisch angegeben werden.

8. Text

Für Überschriften, Erläuterungen usw. kann Text an beliebiger Stelle eingefügt werden. An diesen Stellen ist zu lochen:

TEXT ;.....;⌘

Der zwischen den ; eingeschlossene Text wird im Programmablauf an der betreffenden Stelle ausgeschrieben (; ist gleichbedeutend mit ⌘ = Klingelzeichen).

9. Einlesen

Durch die Anweisung

LIES A bzw. LIES I'

wird ein Wort vom Lochstreifen eingelesen und an die durch A bzw. I' bezeichnete Stelle gebracht.

10. Ausgabe

Die Ausgabe errechneter Werte kann in verschiedener Form erfolgen, weshalb jeweils eine Druckanweisung gegeben werden muß.

a) Ausgabe in Gleitkommaform:

DRU A 9'

Der Zahlenwert A soll im gleitenden Komma mit 9stelliger Mantisse ausgegeben werden (z.B. -,314159265/+01). Die Anzahl der Stellen kann auch symbolisch angegeben werden:

DRU A N'

b) Ausgabe in Normalform:

DRU B 3' 4'

Der Zahlenwert B soll mit 3 Stellen vor und 4 Stellen nach dem Komma ausgeschrieben werden (z.B. -146,1359). Hierbei kann die 2. Angabe auch 0' sein, muß aber gemacht werden.

c) Ausgabe von Festkommazahlen:

DRU I' 4'

Die Festkommazahl I' soll 4stellig ausgegeben werden (z.B. -6149').

Falls keine Druckanweisung vorliegt, wird die Zahl in der Form gedruckt, wie zuletzt ausgegeben wurde. Wenn überhaupt keine Angabe gemacht worden ist, so erscheinen die Gleitkommazahlen mit 9 Mantissenstellen und Festkommazahlen mit 4 Stellen.

11. Tabellenform

Hinter jeder Zahl werden 2 Zwischenräume freigelassen. Will man mehr Zwischenräume haben bzw. etwas einrücken, so kann man dies durch

ZWI n angeben, wobei n die Anzahl der Zwischenräume angibt.

ZLN 1 Wagen- und Zeilentransport

ZLN 5 5mal Zeilentransport

12. Start und Stop

Durch STOP kann an jeder beliebigen Stelle ein Zwischenstop in den Programmablauf eingeführt werden. Nach dem Betätigen der Starttaste kann das Programm wieder fortgesetzt werden.

Am Ende des Programmstreifens ist

START n

zu geben, wobei n die Marke bezeichnet, an der der Programmablauf beginnen soll.

13. Test

Wenn an einer Programmstelle der Befehl TEST steht, wird beim Einlesen, sofern Schalter 14 auf 1 geschaltet ist, für jede folgende Formel ein zusätzlicher Druckbefehl eingeführt. Während des Programmablaufs erfolgt jedoch das Ausdrucken nur, falls Schalter 14 auf "aus" steht. Die Ergebnisse der Formeln werden dann jeweils auf einer Zeile mit 9 Stellen im gleitenden Komma ausgegeben.

Auf diese Weise ist es möglich, von einem neu aufgestellten Programm ein direktes Rechenprotokoll zu bekommen. Ist das Programm getestet, kann man es anschließend ohne die zusätzlichen Druckbefehle aufbauen lassen, indem man beim Einlesen Schalter 14 ausschaltet.

Als weitere Testmöglichkeit kann man beim Einlesen Schalter 14 auf 2 stellen. Dann wird für jeden bedingten Sprung (WENN) ein Druckbefehl eingeführt, der die Marke, auf die gesprungen wird, mit folgendem) Zeichen ausschreibt.

Ablochvorschrift

Für den Programmstreifen ist abzulochen:

- 1) Ein Gleichheitszeichen als Anfangszeichen;
- 2) die größte im Programm auftretende Marke;
- 3) eine Liste der mehrbuchstabigen Variablen (jeder Name in eckigen Klammern);
- 4) eine Liste der jeweils mit dem höchsten Index versehenen indizierten Variablen (dadurch Platzreservierung vom Index 0 bis zur angegebenen Indexzahl);
- 5) ein \times als Abschlußzeichen der Vorgaben (identisch mit Fragezeichen);
- 6) das Programm, wobei jede Formel auf eine Zeile zu schreiben ist (zwischen Funktionsbezeichnung und Variablen ist jeweils ein Zwischenraum zu lochen; Marken sind von den folgenden Formeln oder Anweisungen durch Zwischenraum zu trennen).

Der Programmstreifen wird mit EO+1 eingelesen.

Gleitkommazahlen, die in den Formeln direkt auftreten, müssen in der Normalform abgelocht werden, z.B.:

3 3,14 0,768

Ein Zwischenraum bzw. Wagen- oder Zeilentransport sind Schlußzeichen für Zahlen.

Festkommazahlen, die in die Formeln eingehen oder die auf dem Datenstreifen stehen, müssen mit einem Apostroph direkt hinter der letzten Ziffer versehen sein.

Auf dem Datenstreifen dagegen können Zahlen entweder in der Normalform oder in Gleitkommadarstellung gelocht werden, wobei der Dezimal-Exponent durch einen Schrägstrich von der Mantisse getrennt wird, z.B.

68,76/+10 für die Zahl 687600000000

Bemerkungen zum Umfang des Übersetzerprogramms

Der "Kleine Formelübersetzer" wird ab Trommelzelle 1040 eingespeichert und benötigt 948 Speicherplätze. Unmittelbar darauf folgen die Funktionen und belegen 412 Plätze.

Das übersetzte Programm wird ab Zelle 2400 eingespeichert.

Für jede Anweisung im Formelcode werden vom Übersetzerprogramm durchschnittlich 3 Speicherzellen benötigt. Der

Speicherraum kann noch vergrößert werden, indem man zunächst nur den Übersetzer (ohne Funktionen) einliest und das übersetzte Programm ab Zelle 1988 speichert. Bevor der Ablauf des Programms beginnt, werden dann die Funktionen angefordert durch den Text:

BITTE UNTERPROGRAMME

Nach Einlesen der Unterprogramme kann mit dem Buchstaben J vom Lochstreifen her der Ablauf des Programms gestartet werden (oder in Trommelzelle 1040).

Falls die Funktionen

SIN, COS, TAN, CTG, ARCSIN, ARCTAN, EXP, LOG, GANZ und MOD nicht benötigt werden, kann gleich nach Ausdrucken des Textes

BITTE UNTERPROGRAMME

mit J bzw. 1040 gestartet werden (WURZ und BETR sind stets vorhanden).

Die Einteilung, von wo ab das übersetzte Programm eingespeichert wird, übernimmt die Maschine selbst.

Beispiele:

$$1) \quad z = c \sqrt{\frac{P_n(x)}{P_m(x)}} = c \sqrt{\frac{\sum_{i=0}^n a_i x^{n-i}}{\sum_{i=0}^m b_i x^{m-i}}}$$

Mit der Schrittweite 0,25 sollen für $0 \leq x \leq 5,0$ die Funktionswerte errechnet werden.

```

=
6'
A20'   B20'
X
1) LIES N'
   FUER I' = 0':1':N'
   LIES AI'
   WDH I'
   LIES M'
   FUER I' = 0':1':M'
   LIES BI'
   WDH I'
   LIES C
   FUER X = 0:0,25:5
   DRU X 1' 2'
   D = A
   FUER I' = 1':1':N'

```



```

D = D•X
D = D+AI'
WDH I'
E = B
FUER I' = 1':1':M'
E = E•X
E = E+BI'
WDH I'
WENN E 3 2 3
2) TEXT ; UNENDLICH; ✱
   SPR 6
3) D = D/E
   WENN D 4 5 5
4) TEXT ; IMAGINAER; ✱
   SPR 6
5) Z = WURZ D
   Z = C•Z
   DRU Z 9'
6) ZLN 1
   WDH X
   STOP

```

START 1

A ≡ A0' B ≡ B0'

D = Ergebnis des Zählerpolynoms

E = Ergebnis des Nennerpolynoms

I = Index der Zyklen; da 4 Zyklen nacheinander ablaufen,
konnte für alle der gleiche Index benutzt werden.

Die Polynomkoeffizienten müssen in der Reihenfolge

n' , a_0 , a_1 , a_2 . . . a_n ,

m' , b_0 , b_1 , b_2 . . . b_m ,

c

abgelocht sein.

Falls mehr als 20 Koeffizienten vorhanden sind, muß die
Angabe A20' B20' entsprechend geändert werden.

Mit den Werten

5'

2,5 1,43 0,2 3,12 2,0 1,25

4'

5,7 3,25 1,87 0,56 2,75

0,545

erhält man folgende Ergebnisse:

0	,367438924/-00
0,25	,434338161/-00
0,50	,474010722/-00
0,75	,475701536/-00
1,00	,469807635/-00
1,25	,473865160/-00
1,50	,487915232/-00
1,75	,508088675/-00
2,00	,531390558/-00
2,25	,556052197/-00
2,50	,581107500/-00
2,75	,606044239/-00
3,00	,630597859/-00
3,25	,654638573/-00
3,50	,678110311/-00
3,75	,700997389/-00
4,00	,723306096/-00
4,25	,745054506/-00
4,50	,766266337/-00
4,75	,786967776/-00
5,00	,807185530/-00

2) Integration einer durch Unterprogramm (Marke 9) gegebenen Funktion $f(x)$ im Intervall a, b nach der Simpson'schen Regel.

```

1) F=0
2) R=B-A
   N=2.N
   D=R/N
   M=0
   S2'=0
   X=A
   UPR 9
   S1'=Y
   W=-1,0
3) M=M+1,0
   R=D.M
   X=R+A
   UPR 9
   WENN W 4 5 5
4) R=4.Y
   S3'=R+S1'
   W=1,0
   SPR 3
5) S1'=Y
   S2'=Y+S2'
   S2'=S2'+S3'
   W=-1,0
   WENN M-N 3 6 6
6) R=D/3
   S3'=R.S2'
   R=S3'-F'
   R=R/S3'
   R=BETR R
   WENN R-0,00001 8 7 7
7) F=S3'

```

```

      SPR 2
8) RSP
9) Y=X.X
   Y=Y.0,75 (z.B.  $y=\frac{3}{4}x^2+3$ )
   Y=Y+3,0
   RSP

```

A = Anfang des Intervalls

B = Ende des Intervalls

N = Anzahl der Teilintervalle

F = berechnete Fläche

R = Zwischenspeicher

D = Intervallbreite $\frac{b-a}{2n}$

M = Zählung der Teilintervalle

x = Argument für die gegebene Funktion

y = f(x)

*W = Weiche

S1, S2, S3 = Zwischensummen

Vor Aufruf des obigen Unterprogramms (ab Marke 1) müssen vom Hauptprogramm (Marke 10) die Parameter a, b und n als Gleitkommazahlen in die Speicher A, B und N gebracht werden. Das gesamte Programm kann dann in der folgenden Form geschrieben werden:

```

      =
      10'
      S3'
      X
1) F=0
   ...
10) A=0
   B=4,0
   N=4
   UPR 1
   TEXT ;F=;
   DRU F 3' 3'
   STOP
   START 10

```

} (z.B. $0 \leq x \leq 4, \Delta x=1$)

Ergebnis F= 28,000

3) Skalarprodukt zweier n-dimensionaler Vektoren. Der Aufruf dieses Unterprogramms erfolgt mit UPR 5. Die benötigten Eingabedaten müssen in der Reihenfolge

n'

$x_1, x_2, x_3 \dots x_n,$

$y_1, y_2, y_3 \dots y_n$

abgelocht sein.

Das Ausdrucken des Ergebnisses erfolgt vom Hauptprogramm aus, da es im Unterprogramm nur gespeichert wird.

```
=  
5'  
X10' Y10'  
<
```

```
1) UPR 5  
   DRU S 3' 3'  
   ZLN 1  
   STOP
```

```
5) LIES N'  
   FUER I' = 1':1':N'  
   LIES XI'  
   WDH I'  
   FUER I' = 1':1':N'  
   LIES YI'  
   WDH I'  
   S = 0  
   FUER I' = 1':1':N'  
   A = XI'•YI'  
   S = S+A  
   WDH I'  
   RSP
```

```
START 1
```

Ist die Dimension n der Vektoren >10, muß im Hauptprogramm die Angabe X10' Y10' entsprechend geändert werden.