

He 6 Zuse XIII

PROGRAMMGESTEUERTE
ELEKTRONISCHE RECHENANLAGE
ZUSE Z 22 und Z 22 R
Programmierungsanleitung

Ausgabe Juli 1960

Z U S E K G · Bad Hersfeld
Western Germany

2763

222 Red v. t.

V o r w o r t

Die vorliegende Programmieranleitung für das elektronische Rechengerät Zuse Z 22 befaßt sich mit dem Freiburger Code, eine von vielen möglichen Formen, in der der Benutzer mit der Maschine verkehren kann.

Durch ihren speziellen Aufbau erhält die Zuse Z 22 ihre besondere Flexibilität. Fest verdrahtet liegen in der Maschine nur einige elementare Operationen vor, wie Addition, Subtraktion, Links- und Rechtsverschiebungen u. ä., die aber in einen Befehl fast beliebig kombiniert werden können.

Die arithmetischen Operationen (wie z.B. Multiplikation, Division) werden durch Programme, die sich aus den elementaren Operationen zusammensetzen, gelöst. Den jeweiligen Erfordernissen entsprechend können diese Programme aufgestellt werden. Für jeden gewählten Extern- Code müssen das Leseprogramm, das die Übersetzung in den internen Code übernimmt, die Programme für die arithmetischen Operationen sowie das Schreibprogramm zur Ausgabe der Resultate speziell erstellt werden.

Um den Benutzern diese Arbeit abzunehmen, werden mit der Maschine diese Grundprogramme für einen einfachen, in der Praxis bewährten Extern- Code, den Freiburger Code, mitgeliefert.

Der Freiburger Code ist wie folgt charakterisiert:

1. Zahlendarstellung im Dezimalsystem
2. Gleitendes Komma :
Genauigkeit 9 Dezimalstellen
Zahlenbereich 10^{-20} bis 10^{+19}
3. Einfache Zahlenlänge
4. Einfache der Mathematik entnommene Funktionsbefehle
(+, -, X, :)
5. Aufnahme und Ausgabe von Klartext.

In den Kapiteln 5 - 6 wird der Freiburger Code ausführlich behandelt.

Die ersten beiden Kapitel geben eine kurze Einführung in den Aufbau der Zuse Z 22 und eine knappe Darstellung der elementaren Operationen.

In Kapitel 7 finden sich einige Tabellen und Übersichten. Für den Programmierungsspezialisten, der beispielsweise Grundprogramme für einen anderen Extern-Code entwickeln oder besondere Probleme bearbeiten will, wie z.B. das Aufstellen von interpretierenden Programmen (Überführung eines Codes in einen anderen), Programmen für die automatische Programmfertigung oder auch Chiffrierung oder Dechiffrierung von Klartexten, wird eine besondere Programmierungsanleitung "Der Intern-Code Zuse Z 22" herausgegeben.

Eine Kenntnis des Intern-Codes ist für die Programmierung der üblichen mathematischen Probleme nicht erforderlich.

Bad Hersfeld, Juli 1960

III

I n h a l t

Vorwort	I
1. Einleitung	
1.0 Übersicht über den Aufbau der Zuse Z 22	1
1.1 Das Leitwerk	1
1.2 Der Trommelspeicher	1
1.3 Der Schnellspeicher	2
1.4 Das Rechenwerk	2
1.5 Ein- und Ausgabe	3
2. Die Zahlen und Befehle intern	
2.0 Die ganzen Zahlen	4
2.1 Die Befehle	4
2.2 Der Befehlskreislauf	6
3. Der Freiburger Code	
3.0 Vorbemerkungen	9
3.1 Zahlendarstellung im gleitenden Komma	9
3.2 Zahlen für Adressenrechnen und Zählzwecke (Strichzahlen)	12
3.3 Klartextdarstellung	13
3.4 Darstellung von Befehlen	14
3.5 Lese- und Speicherbefehle	15
3.6 Sprung- und Rufbefehle	16
3.7 Arithmetische Befehle (im gleitenden Komma)	18
3.8 Der Ausgabebefehl	21
1Angaben zur Tabellierung	22
2Bemerkungen zur Ausgabeform	24
3.9 Befehle zur Tabellenanordnung	27
3.10Stopbefehle	29
3.11Bedingungszeichen	30
3.12Bandbefehle	31
4. Programmieren im Freiburger Code	
4.0 Einleitung	34
4.1 Mathematische Vorbereitung	34
4.2 Strukturdiagramm	35
4.3 Speicherliste	36
4.4 Das Zeitdiagramm	38

4.5 Lochstreifenherstellung	40
4.6 Rechenzeiten	45
5. Adressenrechnen und Zählen	
5.0 Vorbemerkungen	46
5.1 Addition und Subtraktion ganzer Zahlen, Inter- sektion	46
5.2 Konstantenzuführung aus dem Befehlsregister	47
5.3 Befehle für Adressensubstitution	48
5.4 Zählbefehle	50
6. Weitere Beispiele	
6.0 Vorbemerkung	51
6.1 Programm für die trigonometrischen Funktionen	52
6.2 Skalarprodukt zweier n dimensionaler Vektoren	58
6.3 Matrizenprodukt	62
6.4 Reelle Nullstellen eines Polynoms n ten Grades	66
6.5 Beispiel aus der Optik	72
6.6 Lösung einer Differentialgleichung nach dem Ver- fahren von Runge - Kutta	75
6.7 Gauß'scher Algorithmus	81
7. Tabellen	
7.0 Blockschema Zuse Z 22	
7.1 Fernschreibcode in Zuse Z 22	
7.2 Befehlsliste	
7.3 Schnellspeicherzellen Zuse Z 22	

1. Einleitung

1.0 Übersicht über den Aufbau der ZUSE Z 22.

Die Z 22 besteht aus dem Leit-, Speicher- und Rechenwerk und der Ein- und Ausgabe. Vergl. 7.0, Blockschema Z 22.

Die in der Maschine verarbeiteten Zahlen und Befehle werden durch Impulsserien dargestellt, die 38-stelligen Dualzahlen entsprechen. Alle Befehle und Zahlen (kurz „Worte“ genannt) werden in den Speichern, einer Magnettrommel und einem Magnetkernspeicher, aufbewahrt. Fünfstellige Dualzahlen oder Fernschreibsymbole können über einen Pufferspeicher in die Fernschreibmaschine und einen Locher geschickt werden oder von einem Lochstreifenabtaster über den Pufferspeicher in den elektronischen Teil der Maschine gelangen.

1.1 Das Leitwerk.

Alle Transporte und Umformungen von Worten werden vom Leitwerk gesteuert. Diese Steuerung besteht darin, daß das Steuerregister (eine Gruppe von 32 Flipflops) alle Stromtore der Maschine nach Maßgabe des jeweiligen Befehls öffnet oder schließt.

Ein Befehl gelangt über das Befehlsregister parallel in das Steuerregister.

1.2 Der Trommelspeicher.

Die Magnettrommel enthält auf 256 Spuren, welche in je 32 Sektoren unterteilt sind, 8192 Speicherzellen. Jede Speicherzelle hat 38 Dualstellen (Wortlänge) und kann 1 Wort, d.h. eine Zahl, einen Befehl oder 7 Klartextzeichen aufnehmen. Jeweils 8 Spuren, also 256 Speicherzellen, bilden einen Block und sind, um Zugriffszeiten zu vermeiden, nach einem besonderen Schema nummeriert. Diese Nummern werden Adressen genannt. Untereinander sind alle 32 Blöcke gleich, lediglich die Adressen erhöhen sich um 256 von Block zu Block.

Jedes in eine Speicherzelle eingeschriebene Wort bleibt dort, auch bei Abschaltung der Maschine, solange erhalten, bis ein anderes Wort in die gleiche Zelle eingeschrieben wird.

Spuren mit solchen Programmen, die über längere Zeiträume unbe-

dingt erhalten bleiben sollen, können durch Umlegen von Blockierungsschaltern vor dem versehentlichen Überschreiben geschützt werden. Dies gilt vor allem für die 32 Spuren, die die arithmetischen Grundprogramme, das Lese- und das Schreibprogramm tragen.

Die Magnettrommel benötigt für einen Umlauf 10 msec, so daß eine Wortzeit in der Z 22 etwa 0,5 msec beträgt. Die mittlere Zugriffszeit, d.h. das Warten auf eine beliebige Speicherzelle, liegt bei 5 msec. Die Zugriffszeiten lassen sich jedoch weitgehend vermeiden.

1.3 Der Schnellspeicher (siehe auch 7.3.)

Ohne Zugriffszeit steht ein Magnetkernspeicher mit 14 bzw. 25 Zellen zur Verfügung. Davon dient eine, mit der Adresse 4, als Akkumulator, d.h. als Register für alle Rechenresultate und liefert stets einen der Summanden für Addition, Subtraktion oder Intersektion.

Die Adressen 0 - 31 werden als Schnellspeicheradressen gewertet, wobei einigen Adressen Sonderfunktionen zugeordnet sind (vergl. 7.2.)

1.4 Das Rechenwerk

Das Addierwerk hat 2 Eingänge. Die vom Akkumulator kommende Zahl geht über Verzögerungsglieder, die diese Zahl links verschoben (L oder LL), rechts verschoben (R), ungeändert oder gar nicht (N) an das Addierwerk übergibt. In den zweiten Eingang des Addierwerks läuft die durch die Adresse gerufene Zahl.

Das Addierwerk kann Additionen (A), Subtraktionen (S) (durch Komplementbildung der aus den Speichern kommenden Zahlen) und die logische Intersektion (I) bewirken. Das Resultat einer Rechenoperation geht in den Akkumulator, bei der Zuse Z 22 R bei bestimmten Befehlen auch direkt in den Schnellspeicher (siehe Kap. 5.1). Der Akkumulator ist die Zelle 4 des Kernspeichers. Die Kernspeicherzelle 3 kann mit dem Akkumulator

so verkoppelt werden (V), daß deren Inhalt, eine Zahl doppelter Länge darstellend, gemeinsam verschoben werden kann. Bei Linksverschiebung gelangt dann die höchste Stelle von 3 in den Akkumulator und bei Rechtsverschiebung die unterste Stelle des Akkumulators in die höchste Stelle von 3.

1.5 Ein- und Ausgabe

Zur Eingabe von Zahlen, Befehlen und Klartexten dient ein handelsüblicher Fernschreiblochstreifenabtaster, zur Ausgabe eine Fernschreibmaschine mit angebautem Locher, der durch einen Schalter der Schreibmaschine angekuppelt werden kann. Die Ausgabegeräte dienen gleichzeitig zur Lochstreifenherstellung außerhalb des Rechengertes. (Näheres hierzu unter 4.6, Lochstreifenherstellung).

Neben dem mechanisch arbeitenden Abtaster kann ein schneller photoelektrischer Abtaster angeschlossen werden, um die Eingabegeschwindigkeit zu erhöhen. Zur schnelleren Ausgabe kann zusätzlich ein Locher angeschlossen werden, der nur Lochstreifen liefert, die unabhängig von der Rechenanlage von Fernschreibern in Klarschrift niedergeschrieben werden können.

2. Die Zahlen und Befehle intern

2.0 Die ganzen Zahlen.

Zahlen bestehen aus 38 Dualstellen, die verschieden interpretiert werden können. Die einfachste Konvention benutzt die oberste Stelle zum Kennzeichen für positive (oberste Stelle = 0) und negative Zahlen (oberste Stelle = 1) und setzt die übrigen 37 Stellen gleich den ganzen Zahlen von 0 bis $2^{37} - 1$. Der zeitlich zuerst einlaufende Impuls wird mit der kleinsten Stelle identifiziert und als 1' bezeichnet. In dieser Konvention sind die Trommeladressen von 0 bis 8191 in den Befehlstexten enthalten. Die Kernspeicheradressen werden als Vielfache von $2^{13} = 8192$ notiert.

Für den bequemen Gebrauch der Maschine zu mathematischen Rechnungen wurde ein anderer Code entwickelt, den wir den Freiburger Code nennen wollen (s.3.0).

2.1 Die Befehle.

Die Befehle enthalten 38 Dualstellen, die auf folgende Weise aufgeteilt sind:

Anzahl der Dualstellen

2	5	13	5	13
---	---	----	---	----

10 = Befehls- Bedingungs- Operations- Schnellspei- Trommelspeicher-
kennzeichen zeichen zeichen cheradressen adressen

Es ist für die Maschine wesentlich, daß jede Dualstelle eines Befehlstextes bestimmte Transportwege öffnet oder schließt. Die Stellen sind für den Gebrauch des Programmierers mit Buchstaben benannt, die die bewirkten Operationen andeuten. Die Befehlsstruktur sichert der Zuse Z 22 ihre große Flexibilität; sie gestattet über 10 000 Befehlsarten.

Im folgenden sollen die einzelnen Befehlszeichen in ihrer Wirkung kurz gekennzeichnet werden.

Die Grundbefehle

a) Sprung- und Rufbefehle

- Em Zuführung des Befehls aus der Zelle m ins Befehlsregister (vgl.2.2, Befehlskreislauf)
- Fm wie Em, nur zusätzlich Aufbewahrung des Inhalts des Befehlszählregisters in Schnellspeicherzelle 5

b) Arithmetische Befehle

- Am Addiere den Inhalt der Speicherzelle m zum Akkumulatorinhalt
- Sm Subtrahiere den Inhalt der Speicherzelle m vom Akkumulatorinhalt
- Um Speichern des Akkumulatorinhalts nach Speicherzelle m
- Im Intersektion des Akkumulatorinhalts mit dem Inhalt der Speicherzelle m.

Zusätze zu den Grundbefehlen

a) Befehlszeichen, die den Akkumulator betreffen

- N Nullsetzen des Akkumulators
 bei A vor Ausführung des Befehls
 bei U nach Ausführung des Befehls
- LL Linksverschieben des Akkumulatorinhalts um 2 Dualstellen (Multiplikation mit 4)
- L Linksverschieben des Akkumulatorinhalts um 1 Dualstelle (Multiplikation mit 2)
- R Rechtsverschieben des Akkumulatorinhalts um 1 Dualstelle (Multiplikation mit $1/2$)

b) Bedingungszeichen

Die Ausführung eines jeden Befehls kann von der Erfüllung einer oder mehrerer Bedingungen abhängig gemacht werden.

- | | | | |
|------|----------------------------------|---|--|
| PP | } Führe den Befehl nur aus, wenn | { | der Akkumulatorinhalt positiv ist |
| QQ | | | der Akkumulatorinhalt negativ ist |
| PPQQ | | | der Akkumulatorinhalt =0 ist |
| P | | | der Inhalt der Schnellspeicherzelle 2 positiv ist |
| Q | | | der Inhalt der Schnellspeicherzelle 2 negativ ist |
| Y | | { | in der untersten Stelle der Schnellspeicherzelle 3 eine '1' steht. |

c) weitere Befehlszeichen

- C Adressenteil aus dem Befehlsregister dem Addierwerk zuführen (vergl. Kap. 5)
- V Verkopplung von Schnellspeicherzelle 3 mit dem Akkumulator (für Links- und Rechtsverschiebung)

- H Einführen einer '1' in die unterste Stelle von Schnellspeicherzelle 3
- K Ausschließliche Benutzung der Schnellspeicherzellen
- Z Stop
- G Adressensubstitution (vergl. Kap 5)

2.2 Der Befehlskreislauf

Ein E-Befehl (ein Befehl der weder A noch U enthält) wirkt als Sprungbefehl z. B. E2000. Der E-Befehl holt aus der durch die Adresse 2000 gekennzeichneten Speicherzelle den Befehlstext in das Befehlsregister und bringt gleichzeitig den nächsten E-Befehl E2001 in das Befehlszählregister. Der aus der Zelle 2000 geholte Befehl wird ausgeführt; und wenn dieser Befehl kein Sprungbefehl ist, wird gleichzeitig der neue Befehl E2001 ins Befehlsregister gebracht. Dieses Zweitaktverfahren wird solange fortgesetzt, bis der aus dem Speicher geholte Befehl ein neuer Sprungbefehl z.B. E1709 ist. Dann beginnt durch diesen E-Befehl ausgelöst ab Zelle 1709 wieder ein linearer Programmablauf, d.h. die Ausführung der Befehle in den Zellen 1709, 1710, 1711 usw.

Durch E-Befehle lassen sich beliebige Sprünge auf der Trommel ausführen.

Ein F-Befehl, auch Ruf-Befehl genannt, hat die gleiche Wirkung wie ein E-Befehl, nur wird zusätzlich der gerade im Befehlszählregister stehende E-Befehl in Schnellspeicherzelle 5 aufbewahrt. Z.B. geht der in Zelle 2002 stehende Befehl F800 nach Aufruf der Zelle 800 ins Befehlsregister, als E-Befehl E801 ins Befehlszählregister, während der im Befehlszählregister stehende Befehl E2003 im Schnellspeicher 5 notiert wird. (vgl. untenstehendes Beispiel) Durch einen solchen F-Befehl können an beliebiger Stelle Unterprogramme aufgerufen werden. Durch einen Befehl E5 am Ende des Unterprogramms erfolgt dann automatisch ein Rücksprung ins Oberprogramm.

B e i s p i e l

In den Zellen 2000 bis 2005 und 800 bis 802 stehen folgende Befehle:

Hauptprogramm

Unterprogramm

2000 A1790

800 LLA4

2001 A1800

801 LA0

2002 F800

802 E5

2003 U1515

2004 A1825

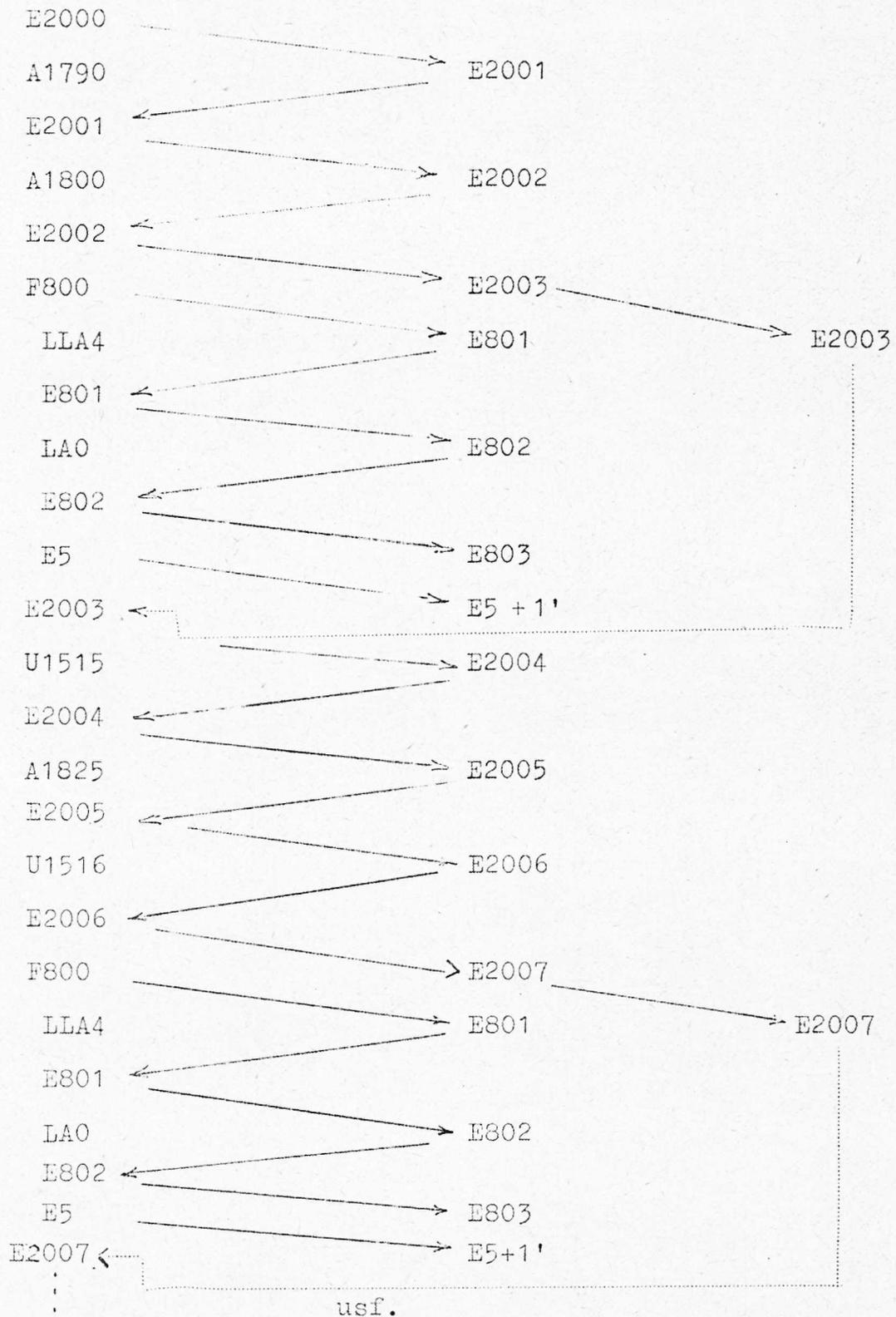
2005 U1516

2006 F800

... ..

Befehlsregister (b)

Bef.-Zählregister (c) Schnellsp. 5



3. Der Freiburger Code

3.0 Vorbemerkungen

In Arbeitsgemeinschaft des Mathematischen Instituts der Universität Freiburg und der wissenschaftlichen Abteilung der Firma Z U S E K.-G. wurde ein Code entwickelt, der den Gebrauch der Maschine für den Mathematiker äußerst einfach gestaltet. Bei Verwendung dieses Codes werden Programme und Zahlen von der Fernschreibmaschine aus durch Zeichen eingegeben, die der üblichen Formelsprache des Mathematikers entnommen sind. Zahlen werden in gleitendem Komma notiert, so daß auf Größenordnungen der errechneten Zahlen und auf Bereichsüberschreitungen beim Aufstellen der Programme keine Rücksicht genommen zu werden braucht.

Die Verwendung dieses einfachen Codes wird dadurch ermöglicht, daß folgende Grundprogramme mit der Maschine mitgeliefert werden:

- 1) Das Leseprogramm
welches die Fernschreibzeichen in die interne Befehlsprache der Maschine und Dezimalzahlen in Dualzahlen übersetzt. Es gestattet außerdem Klartext (Überschriften etc.) in der Maschine zu speichern.
- 2) Die arithmetischen Programme
bewirken Addition, Subtraktion, Multiplikation, Division, Quadratwurzelziehen in gleitendem Komma. Sie sind durch einen einzigen Befehl aufzurufen und an beliebigen Stellen des Hauptprogramms einzuschalten (F-Befehle).
- 3) Das Schreibprogramm
gestattet die Rechenresultate und erläuternden Klartext auf die Fernschreibmaschine in beliebiger Formularanordnung zu geben.

3.1 Zahlendarstellung im gleitenden Komma

Der im vorigen Abschnitt erklärte und mit der Zuse Z 22 mitge-

lieferte Extern-Code stellt die gewöhnlichen Zahlen in gleitendem Komma dar:

$$x = b \cdot 10^a$$

Bei Eingabe dieser Zahlen ist folgendes zu beachten:

- 1) Größter Betrag einer Zahl: $0,7 \cdot 10^{19}$
Es können noch Zahlen bis zu einem Betrag von $0,858993459 \cdot 10^{19}$ eingelesen werden, jedoch müssen dabei alle Dezimalstellen (9 gültige Stellen) angegeben werden.
- 2) Kleinster Betrag einer Zahl: $0,4 \cdot 10^{-19}$
Auch Zahlen bis zum Betrag von $0,343597384 \cdot 10^{-19}$ können mit 9 gültigen Stellen eingelesen werden.
- 3) Die Mantisse darf maximal 9 gültige Dezimalstellen besitzen.
- 4) Anstelle der Basis 10 wird ein Schrägstrich eingegeben. (z.B. aus $- 3,14 \cdot 10^{-3}$ wird $- 3,14/-3$), es wird also die halblogarithmische Form gewählt.
- 5) Innerhalb einer Zahl darf kein Zwischenraum vorhanden sein.

Sind diese Bedingungen nicht erfüllt, so geschieht folgendes:

Zu 1) Treten bei der Eingabe größere Werte auf, so reagiert die Maschine ebenso wie bei Bereichsüberschreitung bei den Rechenoperationen. Es wird dann auf der Fernschreibmaschine die Erläuterung BUEB (Bereichsüberschreitung) und die Stelle des Leseprogramms, wo die Bereichsüberschreitung festgestellt wurde, ausgeschrieben:

BUEB EO+198 bzw. EO+541 bzw. EO+292

je nach Art des Leseprogramms.

- Zu 2) Kleinere Werte werden als Null in die Maschine aufgenommen.
- Zu 3) Mehr als 9 gültige Dezimalstellen können nicht aufgenommen werden, da sonst der vordere Teil verlorenght.

Zu 5) Ein Zwischenraum oder Zeilentransport wird als Schlußzeichen für eine Zahl aufgefaßt. Wird innerhalb einer Zahl ein Zwischenraum gegeben, so nimmt die Maschine die Teile, die durch den Zwischenraum getrennt sind, einzeln als Zahlen in verschiedenen Speicherzellen auf.

Im übrigen ist eine sehr universelle Eingabe möglich. So ist folgendes erlaubt:

- 1) Werden weniger als 9 Mantissenstellen eingegeben, so brauchen anstelle der fehlenden keine Nullen mit eingetastet zu werden.
- 2) Das Komma darf innerhalb der Mantisse an beliebiger Stelle stehen oder auch ganz fehlen.
- 3) Die positiven Vorzeichen der Mantisse und des Exponenten dürfen fortgelassen werden.
- 4) Der Exponent darf vollständig fehlen; in diesem Falle ist auch kein Schrägstrich erforderlich.

Resultate werden in folgender Anordnung herausgeschrieben:

A) Gleitkomma-Ausgabe

- 1) Vorzeichenstelle, Zwischenraum oder -
- 2) Die Mantisse, die stets mit einer gültigen Stelle vor dem Komma beginnt und im ganzen soviel Dezimalstellen enthält, wie herausgeschrieben werden sollen.
- 3) Ein Schrägstrich
- 4) Nach dem Schrägstrich das Vorzeichen + oder - des Exponenten.
- 5) Nach dem Exponentenvorzeichen die 2 Dezimalstellen des Exponenten. Ist der Exponent = 0, so erscheint: $\dots./+00$. Ist die herausgeschriebene Zahl = 0, schreibt die Maschine nur eine 0.

B) Festkomma Ausgabe

Hierbei wird der Exponent durch Verschiebung des Kommas berücksichtigt. (vergl. 3.8. 2B)

Diese Angaben werden im allgemeinen genügen. Für den Programmierungsspezialisten sei noch die Verteilung auf die Dualstellen innerhalb des Rechengertes angegeben:

Anzahl der Dualstellen:	2	29	7
Zahl positiv:	00	Mantisse \hat{x}	Exponent (um 64 erhöht) \bar{x}
Zahl negativ:	11	Mantissenkomplement $1 - \hat{x} $	

Es gilt folgende Konvention:

$$x = \hat{x} \cdot 2^{\bar{x}-64}$$

Hierbei ist stets:

$$0,5 \leq |\hat{x}| < 1$$

$$0 \leq \bar{x} \leq 127$$

also

$$0,5 \cdot 2^{-64} \leq |x| < 1 \cdot 2^{+63}$$

Die zwei Vorzeichenstellen der Mantisse dienen erstens dazu, daß beim Überlaufen (Addition) das Vorzeichen nicht verloren geht und gestatten zweitens eine Unterscheidung zwischen Zahlen, Befehlen und Klartext nach folgender Vereinbarung:

die obersten beiden Dualstellen	Bedeutung
00	Positive Zahl
11	Negative Zahl
01	Klartext
10	Befehl

Bei Vorzeichentests werden somit stets Klartext als positiv, ein Befehl als negativ gewertet.

3.2 Zahlen für Adressenrechnen und Zählzwecke (Strichzahlen)

Es handelt sich hier um ganze Zahlen, da in der Zuse Z 22 weder Adressen noch Zählergebnisse als gebrochene Zahlen auftreten.

Solche Zahlen werden durch Anfügen eines Apostrophs hinter der Zahl ein- und ausgegeben.

Für den Zahlenbereich gilt:

$$-(2^{35}-128) - < k' < 2^{35}-1$$

*erste 3 Stellen
+ 000
- 111*

Ein positives Vorzeichen kann, ein negatives muß angegeben werden. Im internen Code der Zuse Z 22 befinden sich diese Stellen am untersten Ende eines Wortes.

3.3 Klartextdarstellung

Die Zuse Z 22 kann sämtliche Buchstaben, Ziffern, Zeichen sowie Zeilentransport, Zwischenraum, Ziffern-Buchstaben-Umschaltung usw. des Fernschreibalphabetes auch als zusammenhängenden Klartext aufnehmen. Es lassen sich jeweils 7 Fernschreibzeichen in einer Speicherzelle unterbringen. Im mitgelieferten Extern-Code wird Klartext zusammenhängend eingegeben und kann zellenweise ausgegeben werden. Bei der Eingabe wird lediglich am Anfang und Ende eines Textes die Kombination \mathfrak{R} gesetzt, bei der Ausgabe ertönt an diesen Stellen ein Klingelzeichen. Das Zeichen \mathfrak{R} wird nicht mitgespeichert.

Für den Programmierungsspezialisten sei noch die interne Verteilung innerhalb der 38 Dualstellen einer Speicherzelle angegeben.

Anzahl der Dualstellen:

2	1	5	5	5	5	5	5	5	5
---	---	---	---	---	---	---	---	---	---

01
=Kennzeichen
für Klartext

1=Fernschreib-
maschine auf "Buch-
staben" stellen
0=Fernschreib-
maschine auf "Zif-
fern" stellen

7 beliebige Fernschreib-
zeichen

Bei Vorzeichentest wird eine Klartext enthaltende Speicherzelle als positiv gewertet.

3.4 Darstellung von Befehlen

Im Freiburger Code gilt folgende Konvention:

Jeder Befehl muß einen Operationsteil (mindestens einen Buchstaben) und eine Adresse (mindestens eine Ziffer) enthalten.

Ausnahmen bilden folgende Symbole

+ , - , X , : , W , HW , D , M .

Diese Zeichen müssen ohne Adressenangabe eingegeben werden.

Zu allen Befehlen können Bedingungszeichen hinzutreten, wobei die Reihenfolge

Bedingungszeichen - Operationszeichen

einzuhalten ist.

Es können in Befehlen eine oder zwei Adressen auftreten (zwei Adressen siehe Kap. 5)

Die Adressen unter 32 werden als Schnellspeicheradressen gewertet, die größeren Adressen (bis 8191) als Trommelspeicheradressen.

In den nächsten Abschnitten werden die Befehle erläutert, mit denen der Programmierende in den meisten Fällen auskommen wird. (weitere Befehle vgl. Kap. 5)

Zur übersichtlichen Erläuterung der Wirkung der Befehle werden die nachstehenden Symbole verwendet:

- a Akkumulator (=Schnellspeicher 4)
- b Befehlsregister,
nimmt die einlaufenden Befehlstexte auf, um sie anschließend durch das Steuerregister r ausführen zu lassen.
- c Befehlszählregister,
enthält den zuletzt im Befehlsregister b ausgeführten Befehl mit um 1' (Bedeutung siehe unten) erhöhter Adresse.
- n,m Allgemeine Adressen
- t Trommelspeicheradresse
- s Schnellspeicheradresse

- $\langle m \rangle$ Inhalt der Speicherzelle, die unter der Adresse m zu erreichen ist.
- $\langle m \rangle_{p..q}$ Der Inhalt der Dualstellen p bis q der Speicherzelle m .
- $\langle m \rangle_t$ Inhalt der Trommelspeicheradressenstellen der Speicherzelle m , d.h. die Stellen 26 bis 38.
- $\langle m \rangle_s$ Inhalt der Schnellspeicheradressenstellen der Speicherzelle m , d.h. die Stellen 21 bis 25.
- $\langle m \rangle \rightarrow n$ Der Inhalt von m geht nach der Speicherzelle n . Beim Lesen bleiben die Speicherinhalte erhalten: $\langle m \rangle \rightarrow m$.
- $\dots \Rightarrow \dots$ "ergibt" - Zeichen. Der links stehende Ausdruck enthält die Anweisung, wie aus schon bekannten Werten die rechts stehende Größe gebildet wird.
- z.B.
- $\langle m \rangle_t \Rightarrow \tilde{t}$ Der Inhalt der Trommeladressenstellen der Zelle m "ergibt" die neue Trommeladresse \tilde{t} .
- $k+1 \Rightarrow k$ Der um 1 erhöhte Index k "ergibt" den neuen Index k .
- $\{y\}$ Die Adresse der Speicherzelle, deren Inhalt = y ist.
- k' Eine ganze Zahl, aufbauend auf der untersten Dualstelle.

3.5 Lese- und Speicherbefehle

B_m = Speicherlesebefehl. (Bringe den Inhalt der Zelle m in den Akkumulator)

Dieser Befehl bewirkt gleichzeitig:

- 1) $\langle m \rangle \rightarrow a$
- 2) $\langle c \rangle \rightarrow b$

- 1) Mit dem B-Befehl wird der vorher im Akkumulator stehende Wert gelöscht und durch den in der Speicherzelle m stehenden Wert ersetzt.
- 2) Bei allen Befehlen außer den Sprung- oder Rufbefehlen wird der im Befehlszählregister c stehende Steuerbefehl in das Befehlsregister b gebracht, um dort den nächsten Befehl aufzurufen. (Siehe 3.6 Sprung- und Rufbefehle).

T_m = Speicherbefehl mit Akkumulatorlöschung

Wirkung:

- 1) $\langle a \rangle \rightarrow m$
- 2) $0 \rightarrow a$
- 3) $\langle c \rangle \rightarrow b$

Der T-Befehl bewirkt die Speicherung des Akkumulatorinhalts in die Speicherzelle m unter anschließender Löschung des Akkumulators. Bei gelöschtem Akkumulator ist sein Inhalt gleich der Zahl 0.

U_m = Speicherbefehl ohne Akkumulatorlöschung

Wirkung:

- 1) $\langle a \rangle \rightarrow m$
- 2) $\langle a \rangle \rightarrow a$
- 3) $\langle c \rangle \rightarrow b$

Der U-Befehl bewirkt die Speicherung des Akkumulatorinhaltes in die Speicherzelle m unter gleichzeitiger Erhaltung des Wertes im Akkumulator.

3.6 Sprung- und Rufbefehle

Die Wirkung der Sprung- und Rufbefehle wurde schon im Abschnitt 2.2 behandelt.

- | | | | | | |
|-----------|----|-------------------------|----|-----------------------------------|--------------------------------------|
| E_{m+1} | 1) | $E_{m+1} \rightarrow c$ | 2) | $\langle m \rangle \rightarrow b$ | |
| F_m | 1) | $E_{m+1} \rightarrow c$ | 2) | $\langle m \rangle \rightarrow b$ | 3) $\langle c \rangle \rightarrow 5$ |

- 1) Der Befehl E_{m+1} zum Aufruf des nächsten Befehls wird im Befehlszählregister bereit gestellt.
- 2) Der Inhalt der Speicherzelle m geht ins Befehlsregister.

Nur bei F

- 3) Der im Befehlszählregister stehende E-Befehl wird in Schnell-speicherzelle 5 notiert (Rückkehrbefehl).

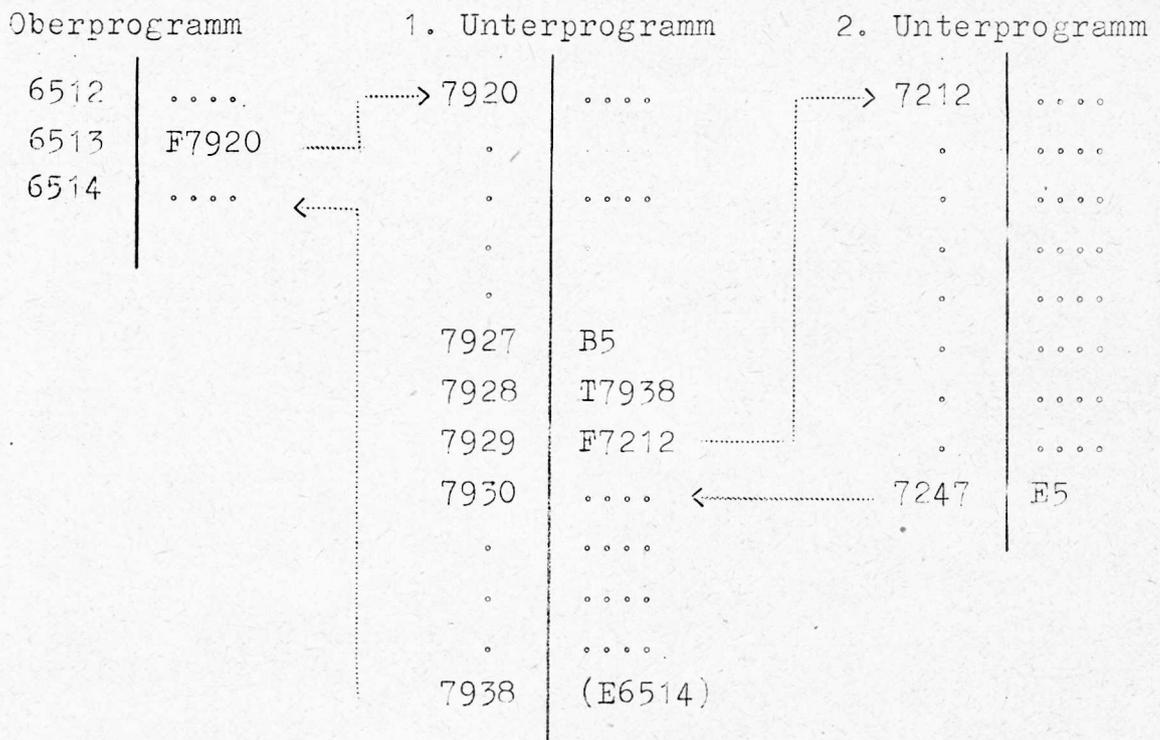
Die unter 1) bis 3) angegebenen Transporte erfolgen gleichzeitig während einer Wortzeit.

Der F-Befehl wird speziell zum Einschalten von Unterprogrammen an verschiedenen Stellen in übergeordnete Programme benutzt. Es kann auch eine mehrfache Ineinanderschachtelung von Unter-

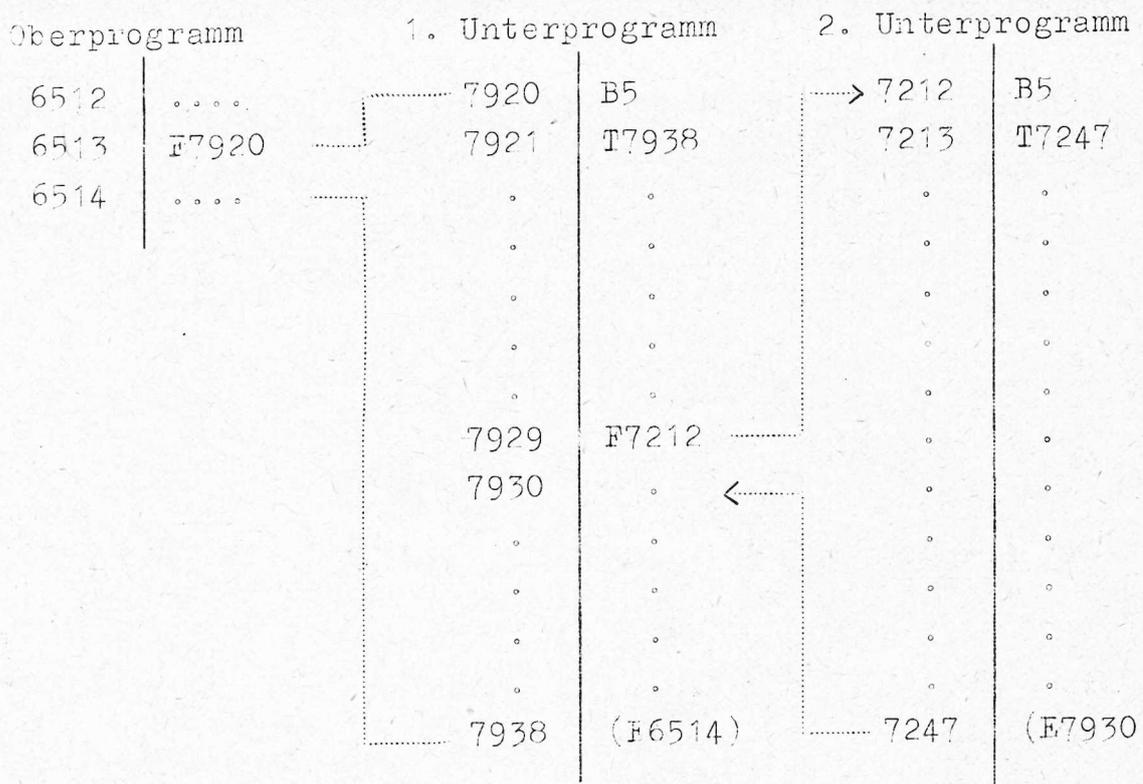
programmen erfolgen, nur muß der Rückkehrbefehl ins Oberprogramm vor Aufruf des neuen Unterprogramms aus Schnellspeicherzelle 5 herausgeholt und ans Ende des 1. Unterprogramms gespeichert werden

{ B 5
{ Tm

z.B.



Es ist noch zu beachten, daß die Befehle +, -, :, X, I, W, HW, M (vgl. die folgenden Abschnitte) in der Maschine ebenfalls Rufbefehl (F-Befehle) sind, bei denen der Inhalt von Schnellspeicherzelle 5 überschrieben wird. Treten solche Befehle auf, so ist wie im 1. Unterprogramm zu verfahren, d.h. der Rückkehrbefehl (Inhalt von Schnellspeicherzelle 5) wird umgespeichert. Die Umspeicherung erfolgt zweckmäßigerweise am Anfang des Unterprogramms.



Die Zellen 7938 und 7247 sind zunächst freizulassen und werden erst im Verlauf der Rechnung mit dem entsprechenden Rücksprungbefehl gefüllt.

3.7 Arithmetische Befehle (im gleitenden Komma)

Die Rechenbefehle werden durch das Leseprogramm in bestimmte Rufbefehle umgewandelt, da die Rechenoperationen in gleitendem Komma durch Unterprogramme gelöst werden. Die Rückkehr zum Hauptprogramm erfolgt automatisch.

Bei diesen Unterprogrammen werden die Schnellspeicherzellen 3 bis 10 gebraucht. Dort vorhandene Werte gehen also verloren.

+ = Addition

Wirkung: 1) $\langle 6 \rangle + \langle a \rangle \rightarrow a, 6$

2) $\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$

Zu 1) In Schnellspeicherzelle 6 steht der erste und im Akkumulator der zweite Operand. Am Ende der Operation werden in beiden Schnellspeicherzellen die Operanden

durch das Resultat ersetzt. Dieses gilt für alle 4 Grundoperationen.

Zu 2) Auch bei den Rechenbefehlen wird nach der Befehlsausführung der nächste Steuerbefehl des laufenden Hauptprogramms in das Befehlsregister geholt, wenn auch mit dem Umweg über die Schnellspeicherzelle 5.

- = Subtraktion

Wirkung: 1) $\langle 6 \rangle - \langle a \rangle \rightarrow a, 6$
2) $\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$

X = Multiplikation

Wirkung: 1) $\langle 6 \rangle \times \langle a \rangle \rightarrow a, 6$
2) $\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$

: = Division

Wirkung: 1) $\langle 6 \rangle : \langle a \rangle \rightarrow a, 6$
2) $\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$

Tritt bei den Rechenoperationen eine Bereichsüberschreitung auf, so reagiert die Maschine folgendermaßen:

Bei Werten, deren Betrag $\geq 2^{63}$ ist, stoppt die Maschine und schreibt auf einer neuen Zeile:

BUEB Es + t

wobei BUEB Bereichsüberschreitung und s + t die Rückkehradresse ins Oberprogramm bedeuten. Werden die Werte absolut kleiner als 2^{-65} , so rechnet die Maschine mit 0 weiter.

W = Quadratwurzelziehen

Wirkung: 1) $\sqrt{\langle a \rangle} \rightarrow a, 6$
2) $\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$

Zu 1) Der Radikand steht im Akkumulator, das Resultat gelangt wieder sowohl in den Akkumulator als auch in der Schnellspeicherzelle 6.

Falls der Radikand negativ ist, so stoppt die Maschine und schreibt zur Erläuterung auf einer neuen Zeile.

IM Es + t,

wobei IM imaginär

und s + t die Rückkehradresse ins Oberprogramm bedeuten.

$$HW = \sqrt{1 - x^2}$$

Wirkung: 1) $\sqrt{1 - \langle a \rangle^2} \rightarrow a, 6$
2) $\langle c \rangle \Rightarrow 5 \rightarrow b$

Zu 1) Die Quadratbildung und Subtraktion wird mit doppelter Stellenzahl durchgeführt.

Falls der Inhalt des Akkumulators beim Aufruf des Unterprogramms dem Betrage nach größer als 1 ist ($|\langle a \rangle| > 1$), so stoppt die Maschine und schreibt auf eine neue Zeile

IM

Es+t

Die Programme für \sqrt{x} und $\sqrt{1 - x^2}$ liegen außerhalb der Zellen 0 bis 1023, sind also nicht vor dem Überschreiben geschützt. Falls man nur das Wurzelprogramm braucht, kann man ein Programm einlesen lassen, das die Zellen von 1039 bis 1095 benötigt.

Falls man \sqrt{x} und $\sqrt{1 - x^2}$ benötigt, so muß man das entsprechende Programm einlesen lassen, das die Zellen von 1039 bis 1215 belegt.

Mit diesem Programm wird gleichzeitig ein Programm eingegeben, das die Umwandlung festes Komma in gleitendes Komma bewirkt. Die entsprechende ganze Zahl muß im Akkumulator stehen. Durch F1200 wird die Umwandlung vorgenommen. Die Zahl im gleitenden Komma steht im Akkumulator und in Schnellspeicherzelle 6.

M = Multiplikation mit -1

Wirkung: 1) $\langle a \rangle \times (-1) \rightarrow a, 6$
2) $\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$

Mit Hilfe des M-Befehls kann man auch den absoluten Betrag einer Zahl bilden, indem man M das Bedingungszeichen QQ (vgl. 3.10, Bedingungszeichen) voransetzt.

QQM = Absolutwertbildung

Wirkung:

a) bei erfüllter Bedingung, d.h. wenn der Inhalt des Akkumulators negativ ist.

1) $|\langle a \rangle| \rightarrow a, 6$

2) $\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$

b) bei nicht erfüllter Bedingung, d.h. wenn der Inhalt des Akkumulators positiv ist:

Es wird sofort der nächste Befehl des Programms aufgerufen.

In diesem Fall steht der ja schon positive Wert im Akkumulator, ist aber nicht in die Speicherzelle 6 gelangt, es sei denn, daß $\langle a \rangle$ das Ergebnis einer unmittelbar vorangehenden Rechenoperation ist.

Weitere Rechenbefehle, z.B. sin, cos, Wurzelziehen, und Potenzieren mit beliebigen Exponenten usw., können durch Schaffung zusätzlicher Unterprogramme jederzeit aufgestellt und durch Rufbefehle aufgerufen werden. Nach kleinen Ergänzungen des Leseprogramms können auch anstelle der Rufbefehle neue Symbole eingegeben werden, in ähnlicher Weise wie bei den bisher vorhandenen Rechenbefehlen.

3.8 Der Ausgabebefehl

Auch der Ausgabebefehl ist ein Rufbefehl, da die Ausführung durch Unterprogramme erfolgt. Sämtliche ausgegebenen Zeichen gehen auf die Fernschreibmaschine und, wenn der angeschlossene Locher eingeschaltet ist, auf Fernschreiblochstreifen. Das Leseprogramm kann alle auf diese Weise gelochten Angaben jederzeit über den Parallelabtaster wieder aufnehmen und beispielsweise für anschließende Rechnungen verwerten. (Äußeres Gedächtnis)

Das Unterprogramm zur Ausgabe benutzt wie die arithmetischen Programme die Schnellspeicher 3 - 10, dort vorhandene Werte gehen also verloren. Bei Rückkehr ins Oberprogramm steht das ausgeschriebene Wort im Akkumulator und Schnellspeicher 6

D = Ausgabebefehl

- Wirkung: 1) $\langle a \rangle \rightarrow$ Fernschreibmaschine
2) $\langle a \rangle \rightarrow a, 6$
3) $\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$

Der Akkumulatorinhalt wird zunächst automatisch untersucht, ob es sich um

- I. eine Zahl im gleitenden Komma
- II. eine Zahl für Adressenrechnen und Zählzwecke (Strichzahl)
- III. einen Befehl oder
- IV. um Klartext handelt.

3. 8. 1 Angaben zur Tabellierung

Die Zahlen im gleitenden Komma (I) und die Strichzahlen (II) können in jeder gewünschten Tabellenform von der Zuse Z 22 ausgeschriebenen werden. Der Benutzer muß jedoch vor Anfang des Programms die folgenden Angaben machen.

Zur Ausgabe im gleitenden Komma (vgl. 3.8.2.A):

A) wieviel gültige Stellen l die Mantisse haben soll.

Zur Ausgabe im festen Komma (vgl. 3.8.2.B):

B) wieviel Stellen α vor dem Komma die Zahlen haben sollen

C) wieviel Stellen β nach dem Komma die Zahlen haben sollen.

Zur Ausgabe allgemein:

D) wieviel Spalten k die Tabellen haben sollen

E) wieviel Zwischenräume z (=Zeichenbreite) zwischen den Spalten zwecks besserer Übersicht bleiben sollen.

F) wieviel Stellen n die Strichzahlen haben sollen

G) Ein Kennzeichen, ob im gleitenden Komma oder im festen Komma ausgegeben werden soll

Alle Angaben A bis G werden von der Maschine in den Zellen 1030 bis 1036 als Strichzahlen benötigt.

1030	l'	Anzahl der Stellen der Mantisse im gleitenden Komma	
1031	\mathcal{L}	Anzahl der Stellen vor dem Komma	} bei festem Komma
1032	β	" " " nach dem Komma	
1033	k'	Spaltenzahl	
1034	z'	Anzahl der Zwischenräume	
1035	n'	" " Stellen von Strichzahlen	
1036	+1'	Gleitkommaausgabe	
	bzw. -1'	Festkommaausgabe	

Für den Zahlenbereich der einzelnen Angaben gibt:

$$\begin{array}{l} 2 \leq l \leq 9 \\ 1 \leq \mathcal{L} \leq 9 \\ 0 \leq \beta \leq 8 \end{array} \left. \vphantom{\begin{array}{l} 2 \leq l \leq 9 \\ 1 \leq \mathcal{L} \leq 9 \\ 0 \leq \beta \leq 8 \end{array}} \right\} \mathcal{L} + \beta \leq 9$$
$$\begin{array}{l} 1 \leq k \\ 0 \leq z \end{array} \left. \vphantom{\begin{array}{l} 1 \leq k \\ 0 \leq z \end{array}} \right\} \text{je nach } l \text{ bzw. } \mathcal{L} \text{ und } \beta \text{ sowie } z \text{ bzw. } k$$
$$1 \leq n \leq 11$$

Die Beschränkung der Stellenzahl $l \leq 9$ bzw. $\mathcal{L} + \beta \leq 9$ ergibt sich daraus, daß in der Maschine die Zahlen nur mit dieser 9stelligen Genauigkeit angegeben sind. Außerdem könnte man Zahlen, die mit mehr Stellen ausgegeben werden, nicht wieder in die Maschine einlesen.

Die Beschränkung der Stellenzahl bei Strichzahlen $n \leq 11$ liegt an der größten Strichzahl $2^{35}-1 = 34359738367$.

Bei k=Anzahl der Spalten und z=Anzahl der Zwischenräume ist keine Begrenzung nach oben angegeben. Doch ist bei der Einteilung der Tabell zu beachten:

1. Die Zeilenlänge der Fernschreibmaschine ist 104 Zeichen
2. Die Zahlen im gleitenden Komma erscheinen mit 1 + 6 Anschlägen (1 Stellen, dazu Vorzeichen, Komma, Schrägstrich und der zweistellige Exponent mit Vorzeichen)

3. Die Zahlen im festen Komma erscheinen mit $\mathcal{L} + \beta + 2$ Anschlägen (\mathcal{L} Stellen vor und β Stellen nach dem Komma dazu Vorzeichen und Komma)
bei $\beta = 0$ nur mit $\mathcal{L} + 1$ Anschlägen (kein Komma)
4. Die Strichzahlen erscheinen mit $n + 2$ Anschlägen
(n Stellen dazu Vorzeichen und Apostroph nach der Zahl)

3.8.2. Bemerkungen zur Ausgabeform

I. Zahlen im gleitenden Komma

A) Ausgabe im gleitenden Komma ($\langle 1036 \rangle \geq 0$)

Die Zahlen werden in der unter 3.1 (Zahlendarstellung im gleitenden Komma) beschriebenen Form ausgegeben.

Es erscheint also stets das negative Vorzeichen bzw. ein Zwischenraum und eine gültige Stelle vor dem Komma, und nach dem Komma noch $l - 1$ Stellen; danach der Schrägstrich, das Vorzeichen des Exponenten und der zweistellige Exponent:

z.B. ($l = 5$)

3,7895/+04

-9,4810/+12

aber 0

Bei Null erscheint nur an der Stelle vor dem Komma eine 0, ohne Vorzeichen und anschließend entsprechend viele Zwischenräume wie die sonstigen Zahlen haben sollen.

B) Ausgabe im festen Komma ($\langle 1036 \rangle < 0$)

Der Exponent wird durch entsprechende Verschiebung der Zahl berücksichtigt. Es erscheinen maximal \mathcal{L} Stellen vor dem Komma, unwesentliche Nullen werden unterdrückt, dafür erscheinen Zwischenräume.

Das negative Vorzeichen steht direkt vor der ersten Ziffer, für das positive Vorzeichen folgt ein Zwischenraum. Nach dem Komma erscheinen noch β Ziffern. Bei $\beta = 0$ wird auch kein Komma gesetzt.

z.B. ($\mathcal{L} = 3$, $\beta = 4$)

346,7830

5,4321

-0,0014

aber 0

Bei Null erscheint nur an der Stelle vor dem Komma eine 0 ohne Vorzeichen mit entsprechend vielen Zwischenräumen danach.

Wird bei der Berücksichtigung des Exponenten durch Verschiebung der Zahl festgestellt, daß mehr als \mathcal{L} Stellen vor das Komma kommen, so wird die Zahl im gleitenden Komma mit 1 gültigen Stellen ausgegeben.

Ist die Zahl absolut kleiner als $10^{-\beta}$ aber nicht = 0, so erscheint bei der Ausgabe

($\mathcal{L} = 3$, $\beta = 4$)

0,0000

II. Strichzahlen

Die Zahlen für Adressenrechnen und Zählzwecke (Strichzahlen) werden mit negativem Vorzeichen bzw. Zwischenraum vor der Zahl und einem Apostroph hinter der Zahl ausgegeben. Unwesentliche Nullen werden unterdrückt, dafür erscheinen Zwischenräume.

z.B. ($n' = 4$)

4036'

7'

-48'

Hat die Zahl mehr als n Stellen so wird die Zahl automatisch mit der maximalen Stellenzahl also 11 Stellen ausgegeben.

III. Befehle

Befehle werden teilweise nicht in der üblichen vereinfachten Schreibart des Extern-Codes, sondern in einer dem Intern-Code der Maschine nahestehenden Form herausgeschrieben.

Im einzelnen fallen folgende Vereinfachungen fort:

Vereinfachte Form	Herausgeschriebene Form
B	NA
T	NU
S	AS
L	LLR
I	UA

Bei Befehlen, die nur auf eine Schnellspeicheradresse wirken sollen, wird ein K hinzugesetzt. Die Schnellspeicheradresse und die Trommelspeicheradresse werden stets durch ein + verbunden herausgeschrieben, selbst wenn die erste = 0 ist.

Ist die Trommeladresse = 0, dann wird nur die Schnellspeicheradresse ausgegeben.

Zu I, II und III

Nach den Zahlen im festen oder gleitenden Komma, den Strichzahlen und Befehlen wird automatisch die angegebene Anzahl z von Zwischenräumen bzw., wenn die letzte (kte) Spalte beschrieben ist, Wagen- und Zeilentransport gegeben.

Es besteht ferner die Möglichkeit, die Druckanordnung während des Programms zu ändern, indem einfach die Speicher 1030 f.f. im Programm vor dem Druckbefehl entsprechend gefüllt werden. Die Änderung der Zelle 1033 wird jedoch erst nach dem nachfolgenden "Wagentransport Zeilentransport" des Fernschreibers also in der nächsten Zeile wirksam.

IV. Klartext

Klartext wird auch als solcher ausgegeben. Es muß, wenn der Klartext in mehreren Zellen gespeichert ist, nacheinander jede Zelle in den Akkumulator gebracht und durch den Druckbefehl ausgegeben werden. Die richtige Buchstaben- und Zifferumschaltung erfolgt automatisch auch bei zusammenhanglosem Herausschreiben einzelner Zellen. Am Anfang und Ende jeder Zelle

wird das Klingelzeichen ausgegeben (aber nicht gedruckt).

Eine Spaltenzählung wird bei Klartext nicht vorgenommen. Das Ausdrucken von Klartexten, die nur ein paar Zellen umfassen, wird man durch einzelnes Aufrufen und Drucken der Zelleninhalte vornehmen. Mehrere längere Überschriften wird man zweckmäßig durch ein Unterprogramm ausdrucken. Ein Beispiel möge das erläutern. Zum Verständnis der Befehle müssen wir jedoch auf spätere Kapitel verweisen.

TmT	$\hat{\pi}$	Querkraft	Moment	Biegung	$\hat{\pi}$ -1'
TnT	$\hat{\pi}$	x	y	y'	$\hat{\pi}$ -1'

Die Klartextzellen haben das Kennzeichen 01, sind also positiv zu werten. Zur Kennzeichnung, daß der Klartext beendet ist, wird in die letzte Zelle etwas Negatives gespeichert. Das Ausdrucken des Klartextes erfolgt vom Hauptprogramm aus nur durch zwei Befehle

Hauptprogramm	Unterprogramm
.	2000 CS1
.	1 T11
CBm	2 B5
F2000	3 T2009
.	4 CGKB11+1
.	5 0
.	6 QQE2009
.	7 D
CBn	8 E2004
F2000	9 0
.	10
.	
.	

3.9. Befehle zur Tabellenanordnung

F800 = Spaltensprung

- Wirkung:
- 1) Spaltensprung → Fernschreibmaschine
 - 2) <a> → a,6
 - 3) <c> → <5> → b

Es werden so viele Zwischenräume an die Schreibmaschine gegeben, daß diese an den Anfang der nächsten Spalte gerückt wird, d.h. eine Zahlenspalte überspringt. Die Spaltenbreite richtet sich nach den Zahlen im gleitenden Komma (bei $\langle 1035 \rangle \langle 0 \rangle$) bzw. im festen Komma (bei $\langle 1036 \rangle \geq 0$).

Das Unterprogramm benötigt die Schnellspeicher 3 - 6 und 9. Der bei Aufruf des Unterprogramms im Akkumulator stehende Wert steht bei Rückkehr ins Oberprogramm im Akkumulator und Schnellspeicher 6.

F840 = Weiterrücken der Schreibmaschine um die im Akkumulator angegebene Anzahl von Zeichenbreiten.

- Wirkung:
- 1) Weiterrücken der Schreibmaschine um i Schritte
 - 2) 0 \rightarrow a
 - 3) $\langle c \rangle \Rightarrow 5 \rightarrow b$

Die Angabe um wieviel Zeichenbreiten die Schreibmaschine weitergerückt werden soll, erfolgt durch eine negative Strichzahl. z.B. Einrücken um 7 Zeichen, wird durch die beiden Befehle bewirkt:

CNS7

F840

(CNS7 vgl. 5.2. (Konstantenzuführung aus dem Befehlsregister)

Benutzte Schnellspeicher 3, 5 und 9

F1000 = Wagenrücklauf mit Zeilentransport

- Wirkung:
- 1) Anweisung \rightarrow Fernschreibmaschine
 - 2) $\langle a \rangle \rightarrow a,6$
 - 3) $\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$

Dieser Befehl braucht nur dann gegeben zu werden, wenn die letzte oder die letzten Spalten einer Zeile übersprungen werden sollen oder mehr als ein Zeilentransport ausgeführt werden soll, sonst erfolgt der Wagenrücklauf mit Zeilentransport automatisch am Zeilenende. Bei diesem Unterprogramm werden die Schnellspeicher 3 - 6 verwendet.

Falls bei der Maschine eine schnelle Ausgabe (Creed- Locher) angeschlossen ist, so werden die Ausgabebefehle und Tabellierungsbefehle mit einem vorangestellten H versehen, wenn sie auf die schnelle Ausgabe ansprechen sollen.

D	<a>	→	Fernschreiber
HD	<a>	→	Locher
F800	Spaltensprung	→	Fernschreiber
HF800	"	→	Locher
F840	Einrücken um die im Akku angegebene Anzahl von Zeichenbreiten	→	Fernschreiber
HF840	Einrücken um die im Akku angegebene Anzahl von Zeichenbreiten	→	Locher
F1000	Wagen und Zeilentransport	→	Fernschreiber
HF1000	" " "	→	Locher

Bei der Zuse Z 22 R wirken die mit H versehenen oben angegebenen Befehle nur dann auf den Locher, wenn in der Zelle 1029 etwas Negatives angegeben ist. Durch die Abhängigkeit vom Inhalt der Zelle 1029 ist man in der Lage, beim Ausprüfen der Programme die Ergebnisse direkt auf den Fernschreiber zu bekommen, ohne die im Programm vorhandenen Befehle HD usw. abzuändern. Es braucht nur die Zelle 1029 richtig gefüllt zu werden.

3.10 Stopbefehle

Z0 = Stop (Buchstabe Z Ziffer Null)

- Wirkung: 1) <a> → a
2) Stop der Maschine

Der Stopbefehl wird einmal am Programmende gegeben, zum anderen an Stellen, an denen der Rechner den weiteren Ablauf des Programms beeinflussen will.

Z kann zusammen mit jedem sinnvollen Befehl programmiert werden. Lautet der Befehl z.B. EZt, so erfolgt nach Drücken der Starttaste der Ablauf des ab Zelle t stehenden Programmes. Insbesondere erfolgt beim Stopbefehl EZ0+1 neues Einlesen vom Lochstreifen, sobald die Starttaste gedrückt wird.

Bedingter Stop. Gibt man in eine Zelle die Zahl 0, so stoppt die Maschine an dieser Stelle, sofern die Taste "Bedingter Stop" eingedrückt ist.

Nach Drücken der Tasten "Weiter und Start" läuft das Programm an dieser Stelle weiter. Bei ausgeschaltetem bedingten Stop überläuft die Maschine diese Zelle mit 0. Benutzt man diese Möglichkeit nicht, schaltet aber den genannten Schalter ein, so hat man eine gewisse Kontrolle für Programmierungsfehler. Wenn nämlich das Programm so durcheinander läuft, daß Zahlen oder Klartext ins Befehlsregister gelangen, dann stoppt die Zuse Z 22.

3.11 Bedingungszeichen

Jeder Befehl der Zuse Z 22 kann unter bestimmte Bedingungen gestellt werden, von deren Erfüllung die Ausführung des Befehls abhängig ist.

Die Bedeutung der Bedingungszeichen ist:

- P = Führe den Befehl nur aus, wenn der Inhalt des Schnellspeichers 2 positiv ist: $\langle 2 \rangle \geq 0$
- Q = Führe den Befehl nur aus, wenn der Inhalt des Schnellspeichers 2 negativ ist: $\langle 2 \rangle < 0$
- PP = Führe den Befehl nur aus, wenn der Inhalt des Akkumulators positiv ist: $\langle a \rangle \geq 0$
- QQ = Führe den Befehl nur aus, wenn der Inhalt des Akkumulators negativ ist: $\langle a \rangle < 0$
- PPQQ = Führe den Befehl nur aus, wenn der Inhalt des Akkumulators Null ist $\langle a \rangle = 0$
- Y = Führe den Befehl nur aus, wenn die unterste Dualstelle im Schnellspeicher 3 = 1 ist, d.h. $\langle 3 \rangle$, als ganze Zahl aufgefaßt, ungerade ist.

Nur für Zuse Z 22 R

- PQQQ = Führe den Befehl nur aus, wenn im Akkumulator ein Befehl oder Klartext steht, d.h. wenn die ersten beiden Stellen nicht übereinstimmen.

Durch beliebige Kombination dieser Buchstaben können die Befehle auch gleichzeitig unter mehrere Bedingungen gestellt werden, z.B. PQQE1312 = Führe den Befehl E1312 nur aus, wenn $\langle 2 \rangle \cong 0$ und $\langle a \rangle < 0$.

Bei bedingten Sprungbefehlen werden vor der Übernahme aus dem Befehlsregister b ins Befehlszählregister c die Bedingungszeichen PP, P, QQ, Q und Y gelöscht.

Wenn bei einem bedingten Befehl die Bedingung nicht erfüllt ist, dann wird aus dem Befehlszählregister der nächste E-Befehl ins Befehlsregister gerufen, der den im Programm nach dem bedingten Befehl stehenden Text ins Befehlsregister holt. Ein Befehl mit nicht erfüllter Bedingung bleibt im Programmablauf unwirksam.

Bedingungsschalter:

Unter der Schnellspeicheradresse 17 kann in Abhängigkeit von einem Schalter am Bedingungspult eine 1 in der untersten Dualstelle, bzw. eine 0 geholt werden:

B17

Wirkung:	A) bei Schalter <u>ein</u>	1) 1' → a
		2) $\langle c \rangle \rightarrow b$
B) bei Schalter <u>aus</u>	1) 0 → a	
	2) $\langle c \rangle \rightarrow b$	

Mit diesem Schalter kann der Benutzer das Programm beeinflussen, ohne die Maschine zu stoppen.

3.12 Bandbefehle

Alle Programme und Zahlen werden vom Lochstreifen in die Maschine eingegeben. Die Worte werden durch das Leseprogramm in die interne Sprache der Maschine übersetzt und laufend gespeichert.

Die Maschine muß aber die Anweisung bekommen,

- 1) wohin das auf dem Lochstreifen stehende Programm gespeichert werden soll,
- 2) wo die Maschine nach Einlesen des Streifens mit der Rechnung beginnen soll.

Auch diese Anweisungen werden vom Lochstreifen aus durch sogenannte Bandbefehle gegeben.

Die erste Anweisung wird durch die Kombination

TmT gegeben, wobei m die Adresse der Speicherzelle ist, in die das erste Wort, das jetzt auf dem Lochstreifen folgt, gespeichert werden soll. Das Leseprogramm sorgt automatisch für die Erhöhung der Adresse auf m+1', m+2' usw., damit die folgenden Angaben fortlaufend gespeichert werden können.

Die zweite Anweisung wird durch

EmE gegeben, wobei m die Adresse der Zelle ist, in der der erste Befehl des Programms steht.

Durch Anfügen des Buchstaben E direkt hinter die Adresse irgendeines Befehls wird bewirkt, daß dieser Befehl sofort ausgeführt wird. Das ist auch bei den vercodeten Rufbefehlen (+, -, X usw.) möglich, nur ist bei den benutzten Buchstaben X, W, D, HW, M vor dem E zusätzlich die Buchstabentaste zu geben.

Wenn der Befehl (kein E-Befehl) ausgeführt bzw. das aufgerufene Unterprogramm durchlaufen ist, wird erneut vom Lochstreifen abgelesen.

Als spezieller Bandbefehl sei der "Stop vom Lochstreifen" angegeben: ZO+1E (Anwendungsbeispiel vgl. 4.5, Lochstreifenherstellung). Nach dem Stop vom Lochstreifen kann durch Betätigen der Starttaste am Bedienungspult auch wieder das Leseprogramm aufgerufen werden.

Es gibt noch einen weiteren Bandbefehl (. . . . Am), welcher bewirkt, daß zur Adresse des Befehls, an den der Bandbefehl Am angefügt ist, der Inhalt der Trommeladressenstellen der Zelle m hinzuaddiert und anschließend der Befehl . . . mit der neuen Adresse gespeichert wird, falls kein weiteres Bandbefehlszeichen folgt. So ist es möglich, jedes Programm zunächst mit beliebigem Anfangspunkt zu programmieren, etwa ab Zelle 0 (in allen Befehlen mit dem Bandbefehl Am wird die Adresse als Trommeladresse eingesetzt). Durch den angefügten Bandbefehl Am wird dann die Translation zur gewünschten Anfangszelle durchgeführt (vgl. Beispiel 6.6).

Die Schnellspeicherzelle 12, bei Schnellspeichererweiterung auch die Schnellspeicher 18, 22 - 31 oder jede freie Trommelspeicherzelle können zur Aufnahme der Anfangsadresse benutzt werden. (Schnellspeicherzelle 12 dient bei E-Bandbefehlen als Hilfsakkumulator).

Es ist zu beachten, daß die T- und E-Bandbefehle nur zur Aufnahme des Lochstreifens in die Maschine dienen und nicht wie alle anderen Befehle auf der Trommel gespeichert werden.

4. Programmieren im Freiburger Code

4.0 Einführung.

Um die Anwendung der einzelnen Befehle zu erläutern, sollen anhand eines Beispiels die Vorbereitung und die Programmierung eines Problems durchgeführt werden. Um die Übersichtlichkeit nicht zu gefährden, wurde absichtlich ein einfaches Beispiel gewählt: Es soll eine Funktion $f(y)$ in einem Intervall $y_0 \leq y \leq y_n$ mit einer Schrittweite Δy tabelliert werden. Die Funktion sei ein Polynom 4. Grades, das Intervall $0 \leq y \leq 1$, die Schrittweite $\Delta y = 0.025$. Die einzelnen Funktionswerte sollen in Tabellenform mit 6 gültigen Stellen, 4 Spalten und 2 Zwischenräumen zwischen den Spalten herausgeschrieben werden, bei Ausgabe im gleitenden Komma.

4.1 Mathematische Vorbereitung.

Jedes Problem bedarf einer eingehenden und sorgfältigen Vorbereitung, um es maschinengerecht zu machen.

- 1.) Zunächst muß das Problem, das möglicherweise in sehr allgemeiner Form (technischphysikalische Formulierung) gegeben ist, in eine mathematische Form gebracht werden.
- 2.) Muß für diese mathematische Form ein geeignetes numerisches Verfahren ausgesucht werden.
- 3.) Dieses Verfahren muß dann noch in die einzelnen Rechenoperationen aufgespaltet werden, wie es ja auch für die Durchrechnung auf einer Bürorechenmaschine notwendig ist.
- 4.) Ist diese Aufgliederung erfolgt, so muß das erhaltene Rechenprogramm noch in die Sprache der Maschine übersetzt werden.

Bei dem gestellten Problem ist der erste Punkt schon ausgeführt. Für den 2. Punkt kommt eine Entwicklung nach dem Horner'schen Schema in Frage:

$$\varepsilon_0 + y (a_1 + y (a_2 + y (a_3 + y a_4))) \Rightarrow f(y)$$

Die Durchführung der weiteren Punkte wird in den folgenden Abschnitten behandelt.

4.2 Strukturdiagramm

Um eine Übersicht über den Rechengang zu bekommen, ist es zweckmäßig, ein Strukturdiagramm aufzustellen. In einem solchen Diagramm werden in einzelnen Kästchen die Operationen angegeben. Die Verbindung der einzelnen Kästchen gibt den Ablauf des Programmes an.

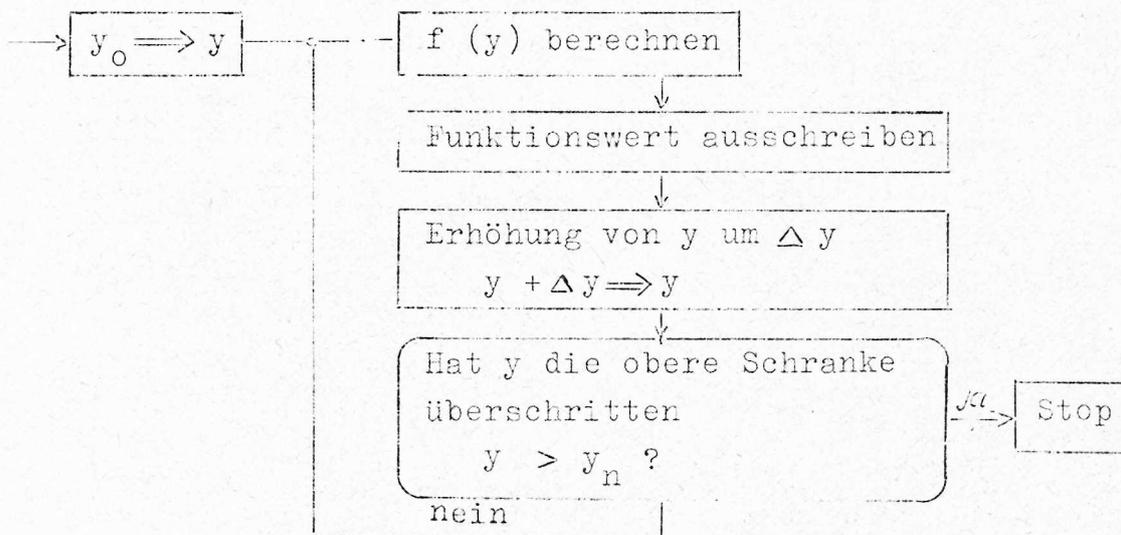
Es sollen dabei folgende Konventionen gelten:

- 1.) Jedes Kästchen hat nur einen Eingang.
- 2.) Jedes Kästchen mit unbedingter Operation hat auch nur einen Ausgang.
- 3.) Bei bedingten Operationen wird die abgefragte Bedingung mit einem Fragezeichen versehen und in ovale Kästchen gesetzt.

Ein waagerechter Ausgang (evtl. zusätzliche Bezeichnung: ja) gilt bei Erfüllung der Bedingung, ein senkrechter Ausgang (zusätzliche Bezeichnung: nein) bei Nichterfüllung der Bedingung.

Um einen groben Überblick zu erhalten, wird man zunächst in den einzelnen Kästchen ganze Programmteile zusammenfassen, die man später noch näher aufgliedert.

Im vorgelegten Beispiel, oder etwas allgemeiner bei der Berechnung einer Funktion $f(y)$ im Intervall $y_0 \leq y \leq y_n$ mit einer Schrittweite von Δy , hat das Strukturdiagramm folgende Gestalt:



Für den Fall der Berechnung eines Polynoms wird diese Aufteilung schon genügen. Bei komplizierteren Funktionen wird man das Kästchen $f(y)$ berechnen noch näher in eine Reihe von Einzel-

kästchen aufgliedern.

Für Kontrollzwecke und Erläuterung eines Programms kann man nach der Aufstellung der Speicherliste (vgl. nächsten Abschnitt) nochmals ein Strukturdiagramm aufstellen, das jeden einzelnen Schritt berücksichtigt, und dabei in jedem Kästchen die Adressen der Zellen mitangeben, in denen die entsprechenden Befehle stehen.

4.3 Speicherliste.

Hat man durch das Strukturdiagramm einen Überblick über den Ablauf des Rechengangs erhalten, kann man an die Aufstellung der Speicherliste gehen.

In dieser Liste ist angegeben, welche Zahlen bzw. Befehle in den einzelnen Speicherzellen stehen. Eine erste Spalte enthält die Speicheradresse, eine zweite den Inhalt der Zelle.

Zweckmäßig ist es, noch weiteren Platz für Zusätze und Erläuterungen zu lassen. Die Speicherliste für das Programm zur Berechnung des Polynoms hat die folgende Gestalt; das Programm soll etwa ab Zelle 1450 eingespeichert werden.

Speicher Nr.	Inhalt	Bemerkung
1450	a_0	} Koeffizienten
1	a_1	
2	a_2	
3	a_3	
4	a_4	
5	y_0	Anfangswert
6	$y_n + \Delta y$	Obere Schranke + Schrittweite
7	Δy	Schrittweite
8	y	Argument
9	B1455	
1460	U1458	$y_0 \Rightarrow y$
1	T6	$y \rightarrow 6$
2	B1454	
3	X	$a_4 \cdot y \rightarrow a, 6$
4	B1453	
5	+	$a_3 + ya_4 \rightarrow a, 6$
6	B1458	
7	X	$y(a_3 + ya_4) \rightarrow a, 6$
8	B1452	
9	+	$a_2 + y(a_3 + ya_4) \rightarrow a, 6$
1470	B1458	
1	X	$y(a_2 + y(a_3 + ya_4)) \rightarrow a, 6$
2	B1451	
3	+	$a_1 + y(a_2 + y(a_3 + ya_4)) \rightarrow a, 6$
4	B1458	
5	X	$y(a_1 + y(a_2 + y(a_3 + ya_4))) \rightarrow a, 6$
6	B1450	
7	+	$a_0 + y(a_1 + y(a_2 + y(a_3 + ya_4))) \Rightarrow f(y) \rightarrow a, 6$
8	D	ausschreiben
9	B1458	
1480	T6	$y \rightarrow 6$
1	B1457	
2	+	$y + \Delta y \rightarrow a, 6$
3	T1458	$y + \Delta y \Rightarrow y$
4	B1456	} wenn $y - (y_n + \Delta y) \geq 0$, Stop
5	-	
6	PRZO	
7	B1458	$y \rightarrow a$
8	E1461	Sprung nach 1461

4.4. Das Zeitdiagramm.

Eine weitere Möglichkeit, den Ablauf eines Programmes zu verfolgen, bietet das Zeitdiagramm. Es gibt zu jeder Zeit an, was in den einzelnen Zellen steht. Es brauchen dabei natürlich nur die Zellen mit veränderlichem Inhalt aufgeführt zu werden. So kann man kontrollieren, daß keine Doppelbelegung von Speicherzellen erfolgen, und andererseits feststellen, zu welcher Zeit etwa noch ein Schnellspeicher zur Aufnahme eines Zwischenergebnisses zur Verfügung steht.

In einer Spalte werden der Inhalt des Befehlsregisters, in einer zweiten der Inhalt des Akkumulators und in weiteren die übrigen verwendeten Zelleninhalte angegeben.

Für unser Beispiel erhält man folgendes Zeitdiagramm:

b	a	6	3.5,7-10	1458
	<a>	<6>	<s>	<1458>
B1455	y_0			
U1458				y_0
T6	0	y_0		
B1454	a_4			
X	Multiplikation			
	$a_4 y$	$a_4 y$		
B1453	a_3			
+	Addition			
	$a_3 + a_4 y$	$a_3 + a_4 y$		
B1458	y			
X	Multiplikation			
	$y(a_1 + y(a_2 + y(a_3 + ya_4)))$	$f(y)$		
B1450	a_0			
+	Addition			
	$f(y)$	$f(y)$		
D	Rückübersetzung (Drucken)			
	$f(y)$	$f(y)$		
B1458	y			
T6	0	y		
B1457	Δy			
+	Addition			

	Addition		
T1458	$y + \Delta y$	$y + \Delta y$	
B1456	0		$y + \Delta y \Rightarrow y$
B1456	$y_n + \Delta y$		
	Subtraktion		
PPZ0	$y - (y_n + \Delta y)$	$y - (y_n + \Delta y)$	
B1458	y		
E1461	Sprung nach 1461		

Ist die Bedingung PP erfüllt, also $y \geq y_n + \Delta y$, so wird durch PFZO die Maschine gestoppt.

Durch die langen waagerechten Striche ist angedeutet, daß die Unterprogramme +, X, - , D die Schnellspeicher 3 bis 10 benutzen.

4.5 Lochstreifenherstellung.

Die Lochstreifen, die im internationalen Fernschreibcode (vgl. 7.1) die Rechenprogramme, Zahlenwerte und andere Angaben enthalten, werden auf dem Programmierungstisch hergestellt oder, falls die Rechenanlage nicht voll ausgelastet ist, auch auf der Ausgabeeinrichtung der Z 22.

Die auf der Fernschreibmaschine eingetasteten Buchstaben, Ziffern und Zeichen werden dabei sofort in den Lochstreifen gestanzt und gleichzeitig als Protokoll niedergeschrieben.

Bei der Lochstreifenherstellung sind folgende Regeln zu beachten:

- 1.) Als erster Befehl auf dem Lochstreifen ist der Bandbefehl TmT zu geben, der das Leseprogramm anweist, in welche Zellen der folgende Text zu speichern ist.
Man kann nur in die Trommel einlesen (nicht in Sammlerspeicher!)
- 2.) Innerhalb eines Befehls oder einer Zahl darf kein Zwischenraum oder Zeilentransport gegeben werden, da diese Zeichen als Schlußzeichen für eine solche Angabe betrachtet werden.
Am Schluß einer jeden Angabe ist mindestens ein solches Schlußzeichen zu geben.
- 3.) Zwischen zwei Worten können beliebig viele Betriebszeichen der Fernschreibmaschine (Zw = Zwischenraum, Zl = Zeilentransport, Wtr = Wagentransport oder leer = Leerzeichen) gegeben werden.
Bei Wagen- und Zeilentransport ist zu beachten, daß die Reihenfolge Wtr Zl eingehalten wird.
- 4.) Als Irrung ist (vor dem Schlußzeichen) zweimal die Buchstabentaste zu drücken (BuBu). Das angefangene Wort, in dem ein Fehler gemacht wurde, wird dann nicht mit eingelesen.* Bei Klartext (Kennzeichnung durch das Zeichen Ω am Anfang und am Ende) kann Irrung nicht berücksichtigt werden. Doch kann man den falsch gelochten Teil des Streifens mit Bu Bu Bu überlochen.

Diese Folge von Bu Bu . . . wird nicht mit in die Zellen aufgenommen.

* Gilt nur für Buchst. und Zahlen. Zeichen werden durch Bu Bu nicht gelöscht!

5.) Am Ende des Programms ist dann der Bandbefehl EmE zu geben, der die Maschine anweist, wo mit der Rechnung zu beginnen ist. Vom Lochstreifen sind auch die Angaben für die Druckanordnung zu geben (vgl. 3.8.1)

Im gegebenen Beispiel sollen

l = 6 gültige Stellen ausgeschrieben werden, die Tabelle

k = 4 Spalten enthalten,

z = 2 Zwischenräume zwischen den Spalten gelassen und im gleitenden Komma ausgegeben werden.

Wenn die Koeffizienten a folgende Werte haben:

$a_0 = 51,68$

$a_1 = -76,541$

$a_2 = 2,41$

$a_3 = 276$

$a_4 = -1,456$

lautet die Befehlsfolge auf dem Lochstreifen:

T1030T 6' 3' 5' 4' 2' 2' 1'

T1450T 51,58 -76,541 2,41 276 -1,456

O 1,025 0,025 0

B1455 U1458 T6 B1454 X B1453 + B1458 X

B1452 + B1458 X B1451 + B1458 X B1450 +

D B1458 T6 B1457 + T1458 B1456 - PPZO

B1458 E1461 E1459E

Für die Druckanordnung gibt man zweckmäßig alle Angaben, auch die im jeweiligen Fall nicht erforderlichen. (Hier die Angabe der Stellen bei Festkommalausgabe und die Stellenzahl für Strichzahlen.)

Am Anfang und am Ende eines Lochstreifens gibt man zweckmäßig eine Reihe von Leerzeichen (nur Transportloch). Leerzeichen werden beim Einlesen ignoriert. Der so hergestellte Lochstreifen kann nun in den Abtaster eingelegt werden, und zwar so, daß das Bu vor dem ersten T noch nicht auf den Abtastknöpfen liegt. Durch Einstellung des Befehls E1' (erste und letzte Befehlsstelle, da das gleichzeitige Nichtvorhandensein von A und U als E aufgefaßt wird) am Bedienungspult und Betätigung der Tasten: Befehlsübernahme und Start wird das Leseprogramm aufgerufen, das Programm wird in die Maschine eingelesen, und sofort beginnt die Rechnung. Ist das Programm durchgerechnet, erfolgt automatisch Stop.

Will man ein Polynom mit verschiedenen Koeffizienten berechnen, so braucht man nicht das ganze Programm neu einzugeben, sondern nur die Koeffizienten zu ändern.

Man gibt an Stelle des Befehls PPZO den Befehl PPEO+1, d.h.

Sprung ins Leseprogramm und auf dem Lochstreifen nach dem letzten B1459E die neuen Koeffizienten. In diesem Fall muß dann auch

vom Streifen her der Stop-Befehl kommen (Z0+1E).

Bei dem vorliegenden Beispiel soll zunächst das Polynom mit den Koeffizienten

$$a_0 = 51,68$$

$$a_1 = -76,541$$

$$a_2 = 2,41$$

$$a_3 = 276$$

$$a_4 = -1,456$$

berechnet werden, und anschließend die Koeffizienten in

$$a_0 = 7,384/-2$$

$$a_1 = 9,416/-1$$

$$a_2 = 2,139/-3$$

$$a_3 = 5,41/-3$$

$$a_4 = 4,113/-1$$

abgeändert werden. Ist das Polynom mit diesen Koeffizienten berechnet, sollen nochmals die Koeffizienten a_2 bis a_4 in

$$a_2 = 4,312/1$$

$$a_3 = 6,197/2$$

$$a_4 = 7,199/1$$

geändert werden.

Die Worte, die in die Zellen 1450 bis 1485 eingelesen werden, bleiben wie vorher. Für die weiteren Angaben erhält man auf dem Lochband:

```
... B1456 - PPEO+1 B1458 E1461 E1459E
T1450T 7,384/-2 9,416/-1 2,139/-3 5,41/-3
4,113/-1 E1459E
T1452T 4,312/1 6,197/2 7,199/1 E1459E
Z0+1E
```

Im oben angeführten Beispiel wurde gezeigt, wie man ein Programm, das sich auf der Trommel befindet, ändern kann. Es brauchen dann nur über Lochstreifen die entsprechenden Bandbefehle und die neuen Worte (Zahlen und Befehle) gegeben werden. Natürlich kann ein so geändertes Programm anschließend herausgeschrieben und gelocht werden, um Unterlagen bzw. Lochstreifen für die Programmbibliothek zu erhalten. Man wird hiervon besonders bei der Beseitigung von Programmierungsfehlern, die erst während der Rechnung festgestellt werden, Gebrauch machen, um die korrigierten Programme für spätere Wiederholung aufheben zu können.

Das Herausschreiben von Programmen, die auf der Trommel stehen, erfolgt am besten durch folgendes kleines Programm:

```
2100 CB (Anfangsadresse -1)
      1 T11
      2 CGKB11 + 1
      3 0
      4 D
      5 B11
      6 CS (Endadresse)
      7 QQE2102
      8 EZ0
```

(Wirkung der Befehle CBn, CGKB11+1, CSn vgl. Kapitel 5). Will man z.B. ein Programm, das in den Zellen 1700 bis 1938 steht, herausschreiben, so sind in Zelle 2100 der Befehl CB1699 und in Zelle 2106 der Befehl CS1938 einzusetzen.

Zum Kopieren und Ändern von Lochstreifen benutzt man den Programmierungstisch mit angebautem Schrittstopgerät. Beim Kopieren braucht man nur den zu kopierenden Lochstreifen in den Abtaster des Programmierungstisches zu legen und den Abtaster anzustellen. Der Streifen wird laufend abgetastet und entsprechend gelocht, wobei gleichzeitig noch ein Protokoll in Klarschrift geschrieben wird.

Beim Ändern verfährt man ähnlich. Bis in die Nähe der zu ändernden Stelle läßt man laufend kopieren, dann schaltet man das Schrittstopgerät auf schrittweises Abtasten. Durch Betätigen einer Taste wird jeweils ein Zeichen abgetastet und gelocht. Bei abgeschaltetem Locher wird der falsche Text abgelesen; der richtige Text wird auf der Fernschreibmaschine eingefügt. Danach kann durch entsprechende Einstellung am Schrittstopgerät wieder laufend kopiert werden.

Ist dagegen kein gesonderter Programmierungstisch vorhanden, so kann man das Kopieren und Ändern von Lochstreifen auch mit Hilfe der Z 22 durch das folgende kleine Programm durchführen.

```
TmT
B20
RVAO
RVAO
RVAO
RVAO
RVAO
RVAO
B3
T20
B17
EPQCEa
EZm
EZmT
```

Bemerkungen:

- 1) 20 ist die Adresse unter der die Ein- und Ausgabe aufgerufen wird.
- 2) Die Rechtsverschiebung unter Verkopplung von Akkumulator mit Schnellspeicher 3(V), erfolgen, weil bei der Eingabe (E20) die untersten 5 Stellen belegt werden, während bei der Ausgabe (T20) die obersten 5 Stellen an die Fernschreibmaschine gehen.
- 3) Wenn der Bedingungsschalter am Bedienungspult auf "aus" steht, wird durch den Befehl B17 eine 0 in den Akkumulator geholt; es ist dann also die Bedingung $\langle a \rangle = 0$ erfüllt und der Befehl PPQQEm wird ausgeführt. Steht dagegen der Bedingungsschalter auf "ein", so kommt durch B17 eine 1 in den Akkumulator und für den Befehl PPQQEm ist die Bedingung nicht erfüllt. Es gelangt danach der nächste Befehl EZm ins Befehlsregister, durch den die Maschine gestoppt wird. Nach Start wird der Sprungbefehl Em ausgeführt.

Der Lochstreifen für dieses Programm wird eingelesen, am Ende stoppt die Maschine mit dem Befehl EZm im Befehlsregister. Danach legt man den zu kopierenden oder zu ändernden Lochstreifen ein und startet ohne etwas neues einzutasten durch Drücken der Starttaste.

Steht der Bedingungsschalter auf "aus", so wird der Streifen laufend herausgeschrieben und bei eingeschaltetem Locher gleichzeitig kopiert. Bei Bedingungsschalter auf "ein" wird jeweils ein Fernschreibzeichen eingelesen und wieder ausgegeben, dann stoppt die Maschine. Durch Betätigung der Starttaste wird dann das nächste Zeichen verarbeitet.

Man kann also das Kopieren bis in die Nähe der Stelle, die geändert werden soll, laufend durchführen. Dann stellt man den Bedingungsschalter auf "ein" und tastet sich schrittweise (bei Start wird jeweils 1 Zeichen kopiert) an die zu ändernde Stelle heran. Während die alten auszulassenden Zeichen geschrieben werden, schaltet man den Locher rechtzeitig ab und schreibt dann bei wiedereingeschaltetem Locher die neuen einzufügenden Zeichen auf der Fernschreibmaschine. Danach kann man wieder durch Bedingungsschalter auf "aus" laufend kopieren.

4.6 Rechenzeiten

Im Folgenden sind die Rechenzeiten angegeben, die die gebräuchlichen Befehle im Freiburger Code zu ihrer Ausführung benötigen. Die Zeiten für die Befehle +, -, X, :, W, HW, D enthalten die reinen Ausführungszeiten.

Die Zeit der Befehle Em und Fm enthält die mittlere Zugriffszeit für das Aufsuchen von m.

Die Ausführungszeiten der Befehle Bm, Tm, Um enthalten keine Zugriffszeiten, da diese bei optimaler Programmierung oder bei Anwendung auf Schnellspeicheradressen entfallen. Tritt bei diesen 3 Befehlen eine Trommeladresse auf, die nicht optimal liegt, so sind jeweils 10 msec Zugriffszeit je Befehl hinzuzurechnen.

Befehl	Ausführungszeit in msec	
+	32,5 - 52,5	
-	34 - 54	
X	30	
:	70	
M	3,5	
W	135	
HW	200	
D	je nach Anzahl der Stellen	
bzw. HD	auf Fernschreiber auf Locher	
	Gleitkomma-Ausgabe: } 10 Zeichen/sec 20 Zeichen/sec	
	Festkomma-Ausgabe: }	
Fm	5,3	
Fm	5,3	
Bm	0,6	
Tm	0,6	
Um	0,6	

Bei der Eingabe eines Lochstreifens in die Zuse Z 22 werden bei Verwendung eines mechanischen Abtasters etwa 15 Fern-

schreibzeichen in der Sekunde aufgenommen und durch das Leseprogramm verarbeitet. Bei Verwendung eines schnelleren Abtasters könnten durch das Leseprogramm 30 bis 40 Zeichen pro Sekunde eingelesen werden.

5. Adressenrechnen und Zählen

5.0 Vorbemerkungen

Die Möglichkeit der Zuse Z 22, Adressen umzurechnen und damit selbsttätig Programme abzuändern, macht die Maschine noch wendiger. Die hier behandelten Befehle können stets durch eine Folge der einfachen Befehle des 3. Kapitels ersetzt werden. Sie bringen jedoch in vielen Fällen eine Einsparung an Rechenzeit, vor allem aber eine Verringerung der Anzahl der Befehle des Programms, wodurch Speicherraum und Zeit für die Eingabe eingespart werden.

Für die Durchführung von Adressenänderungen und Zählungen hat die Zuse Z 22 zwei Möglichkeiten:

- 1) Addition oder Subtraktion von ganzen Zahlen (vgl. 5.1 und 5.2)
- 2) Benützen von Index- bzw. Zähl-Registern (vgl. 5.3 und 5.4)

5.1 Addition und Subtraktion ganzer Zahlen, Intersektion

Am = Addition ganzer Zahlen

- Wirkung: 1) $\langle a \rangle + \langle m \rangle \rightarrow a$
2) $\langle c \rangle \rightarrow b$

Es wird hier die Addition des Inhalts der Speicherzelle m zum Akkumulatorinhalt ausgeführt, wobei $\langle a \rangle$ und $\langle m \rangle$ als ganze Zahlen aufgefaßt werden. Es sei ausdrücklich darauf hingewiesen, daß das Zeichen A und das folgende S nicht für die Addition und Subtraktion im gleitenden Komma verwendet werden können.

Sm = Subtraktion ganzer Zahlen

- Wirkung: 1) $\langle a \rangle - \langle m \rangle \rightarrow a$
2) $\langle c \rangle \rightarrow b$

Es wird hier vom Akkumulatorinhalt der Inhalt der Speicherzelle m subtrahiert.

Durch das zusätzlich gegebene Zeichen N wird der Akkumulator vorher gelöscht. NSm = $\langle m \rangle \rightarrow a$, NAm = Bm.

Im = Intersektion

- Wirkung: 1) $\langle a \rangle \wedge \langle m \rangle \rightarrow a$
2) $\langle c \rangle \rightarrow b$

Es wird hier die dualstellenweise Konjunktion durchgeführt, d.h. in den Dualstellen, in denen sowohl im Akkumulator als auch in der Speicherzelle m eine 1 steht, erscheint nach der Intersektion eine 1, in den anderen Stellen, wo im Akkumulator oder in der Zelle m oder in beiden eine Null steht, erscheint eine Null. Durch die Intersektion kann man beliebige Teile von irgendwelchen Angaben herauschneiden.

In der Zuse Z 22 R besteht die Möglichkeit, in sämtliche Schnellspeicher von der Trommel aus direkt hinein zu addieren oder subtrahieren, ohne dass der Akkumulator dabei berührt wird.
Z.B.

A7+1360 $\langle 7 \rangle + \langle 1360 \rangle \rightarrow 7$, $\langle c \rangle \rightarrow b$
 $\langle a \rangle \rightarrow a$
S12+1208 $\langle 12 \rangle - \langle 1208 \rangle \rightarrow 12$, $\langle c \rangle \rightarrow b$
 $\langle a \rangle \rightarrow a$

Ein in einem solchen Doppeldressbefehl stehendes N (=Nullsetzen) wirkt auf den angegebenen Schnellspeicher

B6+1400 $\langle 1400 \rangle \rightarrow 6$, $\langle c \rangle \rightarrow b$
(B-NA) $\langle a \rangle \rightarrow a$
NS13+8186 $-\langle 8186 \rangle \rightarrow 13$, $\langle c \rangle \rightarrow b$
 $\langle a \rangle \rightarrow a$

5.2 Konstantenzuführung aus dem Befehlsregister

Durch das Befehlszeichen C ist es möglich, die letzten 19 Dualstellen aus dem Befehlsregister dem Rechenwerk zuzuführen. Diese Stellen, die sonst die Schnellspeicher- und Trommelspeicher-

adressen enthalten, sind bei Befehlen mit C als Adresse unwirksam. Das Befehlsregister wirkt hier als Speicher, aus dem die Zahl geholt wird. Auf diese Art kann man ganze Zahlen bis $n = 524287 (= 2^{19} - 1)$ aus dem Befehlsregister holen, die sich somit als Befehl verkleidet direkt in das Programm einflechten lassen oder am Bedienungspult eingestellt werden können.

CAn	$\langle a \rangle + n' \rightarrow a$	}	$\langle c \rangle \rightarrow b$
CSn	$\langle a \rangle - n' \rightarrow a$		
CBn	$n' \rightarrow a$		
CNSn	$- n' \rightarrow a$		
CIIn	$\langle a \rangle \wedge n' \rightarrow a$		
CISn	$\langle a \rangle \wedge (-n') \rightarrow a$		

z.B.

CI8191 bewirkt, daß die Trommeladressenstellen herausgeschnitten werden ($8191 = 2^{13} - 1$). CIS262144 bewirkt das Heraustrennen des Operationsteiles. ($262144 = 2^{18}$)

5.3 Befehle für Adressensubstitution

Mit Hilfe des Befehlszeichens G kann man verschiedene Arten von automatischen Adressenänderungen vor Ausführung eines Befehls bzw. automatischer Zählung bewirken, je nachdem, ob G allein oder in Verbindung mit K, CK oder C auftritt. Da die Adressenänderung eine Wortzeit in Anspruch nimmt, wird zur Vermeidung von Wartezeiten die Adresse des im Befehlszählregisters stehenden E-Befehl, der den G-Befehl ins Befehlsregister geholt hat, noch ein zweites mal um 1' erhöht. Es wird also nach einem G-Befehl als nächstes der Inhalt der übernächsten Zelle ins Befehlsregister gerufen. Beim Programmieren ist also auch nach einem G-Befehl eine Zelle frei zu lassen, die aber selbstverständlich zur Aufnahme von irgendwelchen Hilfsgrößen benutzt werden kann.

Der Inhalt des Akkumulators bleibt bei der Adressenänderung unverändert erhalten.

G...m bewirkt vor Ausführung des Befehls..., daß der Inhalt der Trommeladressenstellen der Zelle m als neue Adresse \tilde{t} für den Befehl..., eingesetzt wird. (m ist also die

Adresse, unter der die richtige Adresse zu finden ist). Im Befehlszählregister wird um 1 weitergezählt, das Zeichen G wird gelöscht.

Danach wird der Befehl ... mit der Adresse \tilde{t} ausgeführt.

- 1. Schritt $\langle m \rangle_t \Rightarrow \tilde{t} \quad \langle c \rangle + 1 \rightarrow c \quad G \text{ wird gelöscht}$
- 2. Schritt Befehl ... \tilde{t} ausführen.

Als nächster Befehl wird der Inhalt der übernächsten Zelle aufgerufen.

GK...s+t Bei diesem Befehl wirkt die Schnellspeicherzelle s als Indexregister mit konstantbleibendem Inhalt. Vor der Ausführung des Befehls ... wird der Inhalt der Schnellspeicherzelle s zu t hinzuaddiert; von dieser Summe werden die letzten 13 Stellen (Trommeladressenstellen) als neue Adresse \tilde{t} eingesetzt. G, K und die Schnellspeicheradresse s werden gelöscht und im Befehlszählregister wird um 1 weitergezählt.

Danach wird der Befehl ... \tilde{t} ausgeführt.

- 1. Schritt $(\langle s \rangle + t)_t \Rightarrow \tilde{t} \quad \langle c \rangle + 1 \rightarrow c \quad G, K, s \text{ löschen}$
 $\langle s \rangle \rightarrow s$
- 2. Schritt Befehl ... \tilde{t} ausführen

Als nächster Befehl wird wieder der Inhalt der übernächsten Zelle aufgerufen.

Eine hier aufgerufene Schnellspeicherzelle s spielt somit die Rolle eines sogenannten Indexregisters, dessen Inhalt, eine positive oder negative ganze Zahl aufgebaut auf der untersten Dualstelle, die Trommelspeicheradresse t entsprechend verändert. Der Inhalt dieses Indexregisters bleibt dabei dort unverändert erhalten.

Bei $t = 0$ hat der Befehl GK. s+t die gleiche Wirkung wie der einfache G-Befehl mit Schnellspeicheradresse,

$$GK...s+0 \quad G...s$$

CGK..s+t Bei diesem Befehl wirkt die Schnellspeicherzelle s als Indexregister mit sich veränderndem Inhalt. Die Wirkung ist die gleiche wie bei dem oben beschriebenen GK-Befehl, nur wird zusätzlich der Inhalt der Schnellspeicherzelle s um t erhöht.

1.Schritt $\langle s \rangle + t \rightarrow s$
 $\langle s \rangle + t \Big|_t \Rightarrow \tilde{t} \quad \langle c \rangle + 1' \rightarrow c \quad C, G, K, s \text{ löschen}$

2.Schritt Befehl ... \tilde{t} ausführen

Der Inhalt der übernächsten Zelle wird als nächster Befehl aufgerufen.

Der einzige Unterschied dieses Befehls gegenüber dem vorher beschriebenen GK-Befehl ist der, daß der Inhalt der aufgerufenen Speicherzelle s nicht konstant bleibt, sondern um die im Befehl zunächst enthaltene Trommelspeicheradresse t erhöht wird. Beim CGK-Befehl wird also die aufgerufene Schnellspeicherzelle als Indexregister mit sich änderndem Inhalt verwendet.

5.4 Zählbefehle

Die Kombination CG bewirkt reine Zählung. Wir unterscheiden zwei Arten:

CGU s+t Es wird hier t zum Inhalt des Zählregisters addiert und die Summe wieder in s notiert. Im Befehlszählregister wird um 1' weitergezählt sowie G und s gelöscht. Der übrigbleibende Befehl $CU\tilde{t}$ ist ein wirkungsloser Befehl.

1.Schritt $\langle s \rangle + t \rightarrow s \quad \langle c \rangle + 1' \rightarrow c \quad G, s \text{ löschen}$
2.Schritt $CU\tilde{t}$ wirkungslos

Der Akkumulatorinhalt bleibt erhalten.

CGA s+t Auch hier wird im Schnellspeicher s um t weitergezählt, anschließend aber die Trommeladressenstellen des Zählergebnisses zum Akkumulator hinzuaddiert.

1.Schritt $\langle s \rangle + t \rightarrow s$
 $\langle s \rangle + t \Big|_t \Rightarrow \tilde{t} \quad \langle c \rangle + 1' \rightarrow c \quad G, s \text{ löschen}$

2.Schritt $CA\tilde{t}$ ausführen

Der einzige Unterschied zu dem oben beschriebenen CGU-Befehl ist der, daß das neue Zählergebnis nicht nur im aufgerufenen Zählregister s sondern auch die Trommeladressenstellen dieses Zählergebnisses (nach Ausführung des 2. Schrittes $CB\tilde{t}$) im Akkumulator zur Verfügung stehen.

Dieser Befehl ist auch mit B, S, NS an Stelle von B möglich so daß im 2. Schritt

$CB\tilde{t}$
bzw $CS\tilde{t}$
" $CNS\tilde{t}$ ausgeführt wird. (Vgl. 5.2)

Ferner ist noch zu beachten, daß im Falle $\langle s \rangle + t < 0$ in der zweiten Wortzeit des G-Befehles nicht $(\langle s \rangle + t)$ in den Akkumulator kommt bzw. hinzuaddiert oder subtrahiert wird, sondern der Befehl in der zweiten Wortzeit mit der Konstanten $(8192 - [\langle s \rangle + t])$ ausgeführt wird, weil diese Konstante gerade die Trommeladressenstellen der negativen Zahl $\langle s \rangle + t$ darstellt.

6. Weitere Beispiele

6.0 Vorbemerkung

In diesem Kapitel soll an einigen Beispielen die Anwendung der einzelnen Befehle gezeigt werden. Das erste Beispiel wird allein die in Kapitel 3 erklärten Befehle benutzen, die weiteren Beispiele auch die Befehle aus Kapitel 5.

Im allgemeinen ist es empfehlenswert, wenn in jedes Programm Kontrollen eingebaut werden, z.B. durch Rückwärtsrechnen, Einsetzen oder Rechnen auf zwei verschiedenen Wegen, schon um Programmierungsfehler aufzudecken.

Bei den folgenden Beispielen wurde auf solche Kontrollen verzichtet, um die Übersichtlichkeit nicht zu gefährden.

Für die Beispiele werden das Strukturdiagramm und die Speicherliste aufgestellt.

Über die Eingabe der Programme in die Maschine vergleiche 4.6 Lochstreifenherstellung.

6.1 Programm für die trigonometrischen Funktionen

Für das Intervall $0 \leq x \leq \frac{\pi}{2}$, kann man mit Hilfe der Tschebyscheff'schen Polynome eine Näherungsfunktion für die Sinusfunktion mit 7 Gliedern aufstellen, die $\sin x$ auf 9 Dezimalstellen genau annähert.

Für $\sin x$ ergibt sich

$$\sin x = \sum_{i=0}^6 a_{2i+1} x^{2i+1} + R$$

$$a_1 = 1$$

$$a_3 = -1,66666666/-1$$

$$a_5 = 8,33333331/-3$$

$$a_7 = -1,98412648/-4$$

$$a_9 = 2,75568256/-6$$

$$a_{11} = -2,50259203/-8$$

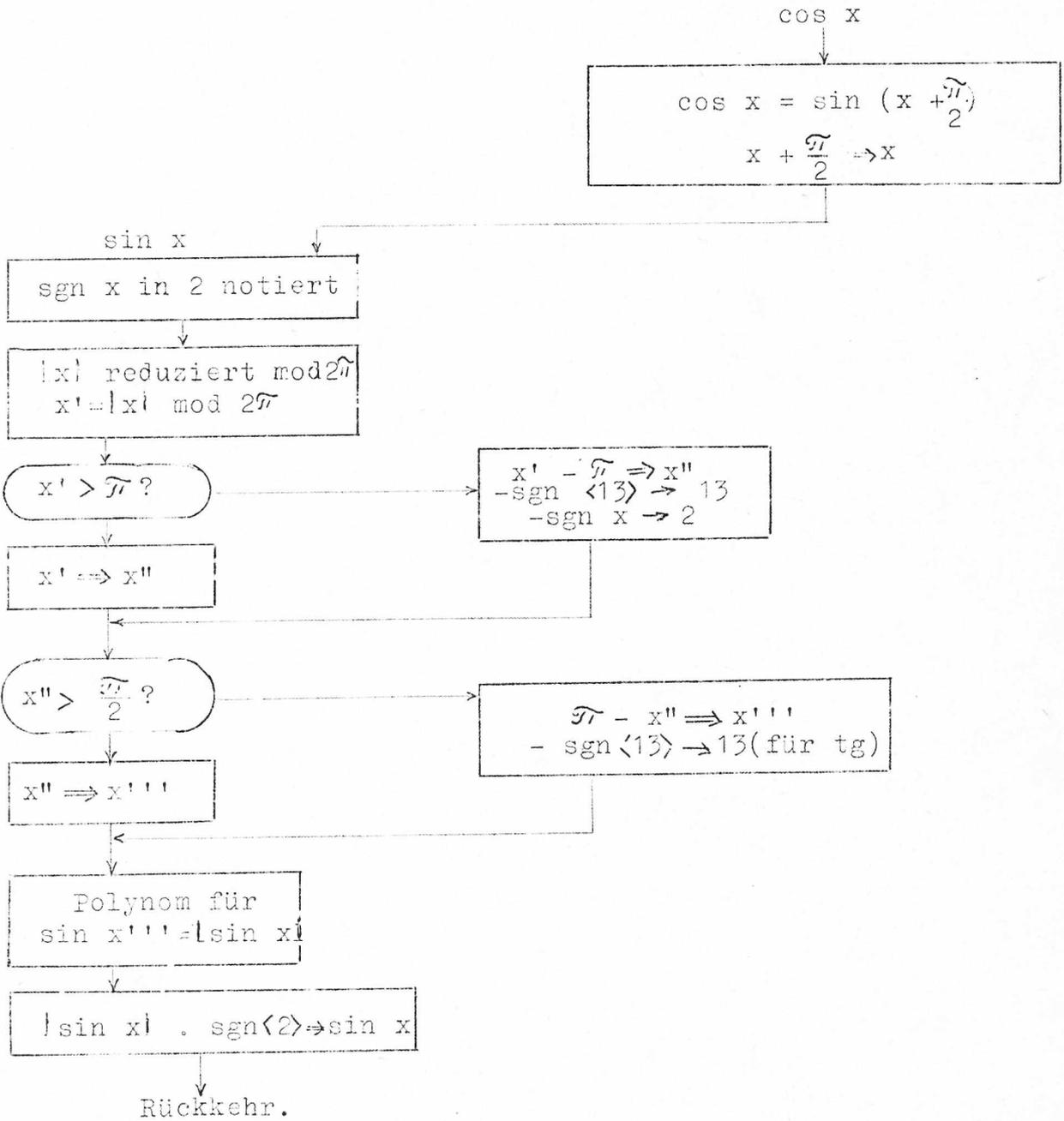
$$a_{13} = 1,53514915/-10$$

$$|R| \leq 6,7 \cdot 10^{-12}$$

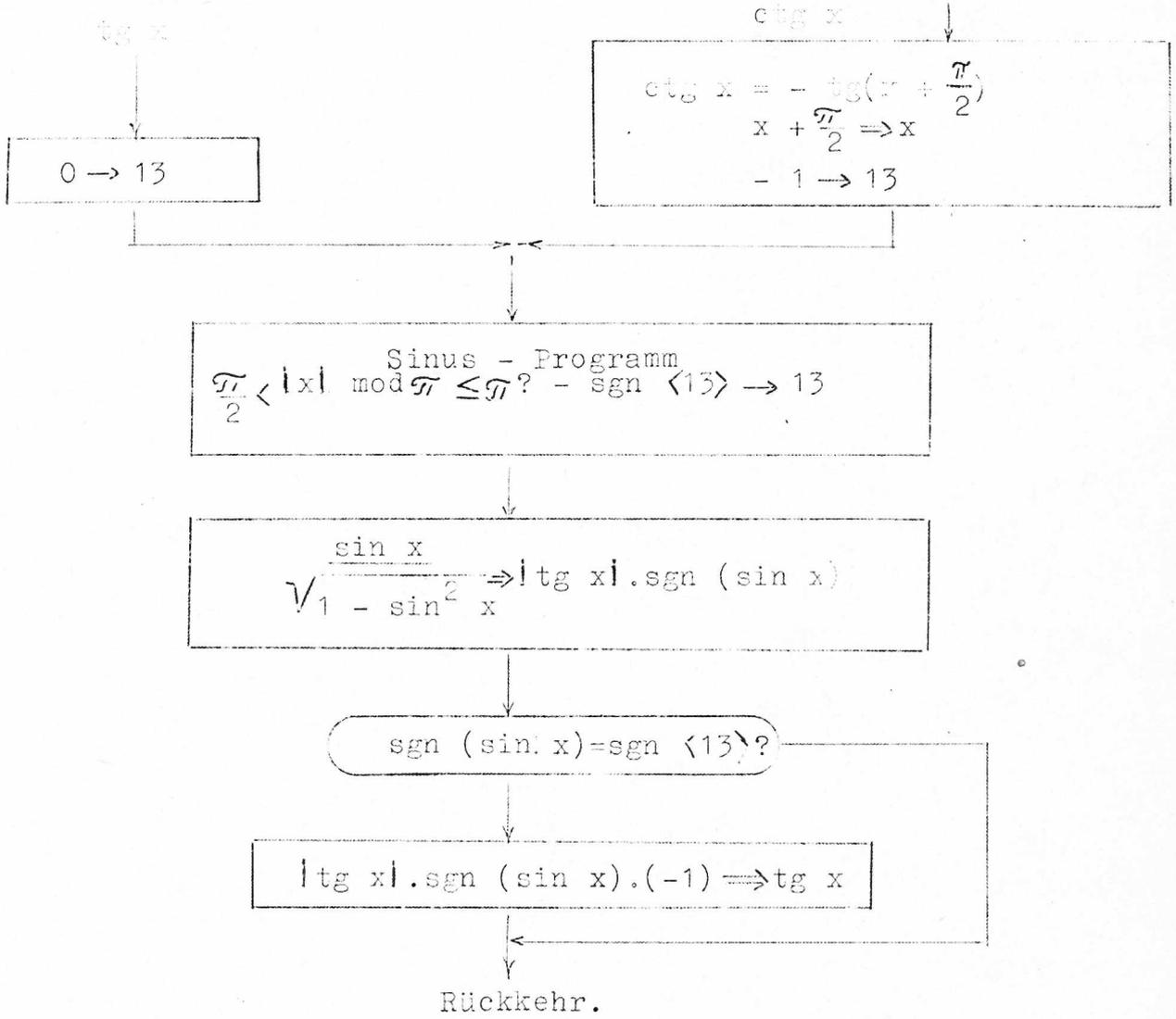
Die übrigen trigonometrischen Funktionen werden auf $\sin x$ zurückgeführt.

Der Wert x , für den die trigonometrischen Funktionen berechnet werden sollen, möge im Akkumulator stehen. x kann beliebige Größe haben, im Programm wird $x \bmod 2\pi$ reduziert und für die Berechnung dem oben angegebenen Intervall angepasst.

Strukturdiagramm für sin x, cos x.



Strukturdiagramm für tg x, ctg x



Speicherliste für die trigonometrischen Funktionen.

tgx →	1310	T2	x → 2
	1	B5	Rückkehradresse umspeichern
	2	T1327	
	3	T13	
	4	F1337	Sinus-Programm
	5	U12	sin x → 12
	6	HW	cos x → 7
	7	T7	
	8	B12	sin x → 6
	9	T6	
	1320	B13	<13> → 2
	1	T2	
	2	B7	$\frac{\sin x}{ \cos x } \Rightarrow \operatorname{tg} x \cdot \operatorname{sgn}(\sin x)$
	3	:	
	4	QQQE1327	
	5	PPPE1327	sgn <2> = sgn (sin x)
	6	M	
	7	E(Rückkehr)	
ctgx →	8	T6	
	9	B5	Rückkehradresse umspeichern
	1330	T1327	
	1	B1427	$x + \frac{\tilde{\pi}}{2} \Rightarrow x$
	2	+	
	3	T2	
	4	B1414	-1
	5	E1313	Tangensprogramm
sin x →	6	T2	x → 2
	7	B5	
	8	T1399	Rückkehradresse umspeichern
	9	B2	
	1340	QQM	x

1341	U6		}		
2	T11				
3	B1423	2π			$x' = x \bmod 2\pi \rightarrow 11$
4	-				
5	PPE1341				
6	B1442	π	}		
7	+				$x' - \pi < 0? x' \Rightarrow x''$
8	QQE1356				
9	T11				$x' - \pi \geq 0? x' - \pi \Rightarrow x'$
1350	B13		}		
1	QQNE1353				$-\text{sgn} \langle 13 \rangle \rightarrow 13$
2	B1416	-1			
3	T13		}		
4	PE1434	-1			$-\text{sgn } x \rightarrow 2$
5	T2		}		
6	B1420	$\frac{\pi}{2}$			
7	T6				
8	B11				$\frac{\pi}{2} - x'' \geq 0; x'' \Rightarrow x'''$
9	-				
1360	PPE1368		}		
1	B1409	$\frac{\pi}{2}$			$\frac{\pi}{2} - x'' < 0; \pi - x'' \Rightarrow x'''$
2	+				
3	T11		}		
4	B13				
5	QQNE1367				$-\text{sgn} \langle 13 \rangle \rightarrow 13$
6	B1416	-1			
7	T13		}		
8	B11				
9	U6				$x^2 \rightarrow 12,6$
1370	I		}		
1	T12				
2	B1436	a_{13}	}	$a_{13} \cdot x^2 \rightarrow a,6$	
3	X				
4	B1422	a_{11}	}	$\langle 6 \rangle + a_{11} \rightarrow a,6$	
5	+				
6	B12		}	$\langle 6 \rangle \cdot x^2 \rightarrow a,6$	
7	X				
8	B1410	a_9	}	$\langle 6 \rangle + a_9 \rightarrow a,6$	
9	+				
1380	B12		}		
1	X				$\langle 6 \rangle \cdot x^2 \rightarrow a,6$
2	E1130	a_7			$\langle 6 \rangle + a_7 \rightarrow a,6$
3	+				

1384	B12		} <6> . x ² → a,6
5	X		
6	B1418	a ₅	} <6> + a ₅ → a,6
7	+		
8	B12		} <6> . x ² → a,6
9	X		
1390	B1438	a ₃	} <6> + a ₃ → a,6
1	+		
2	B12		} <6> . x ² → a,6
3	X		
4	B1426	a ₁	} <6> + a ₁ → a,6
5	+		
6	B11		} <6> . x → a,6 = sin x
7	X		
8	QM		sin x . sgn <2> ⇒ sin x
9	E {Rückkehr}		
cos x	1400	U5	x → 6
1	B5		Rückkehradresse umspeichern
2	T1399		
3	B1419	$\frac{\pi}{2}$	
4	+		x + $\frac{\pi}{2}$ → a,2
5	U2		
6	E1340	Sinus Programm	
7			
8	$\frac{\pi}{2}$	1426	a ₁
9	$\frac{\pi}{2}$	7	$\frac{\pi}{2}$
1410	a ₉	8	$\frac{\pi}{2}$
1		9	
2		1430	a ₇
3		1	
4	-1	2	
5		3	
6	-1	4	-1
7		5	
8	a ₅	6	a ₁₃
9	$\frac{\pi}{2}$	7	
1420	$\frac{\pi}{2}$	8	a ₃
1		9	
2	a ₁₁	1440	
3	$\frac{\pi}{2}$	1	
4		2	$\frac{\pi}{2}$
5			

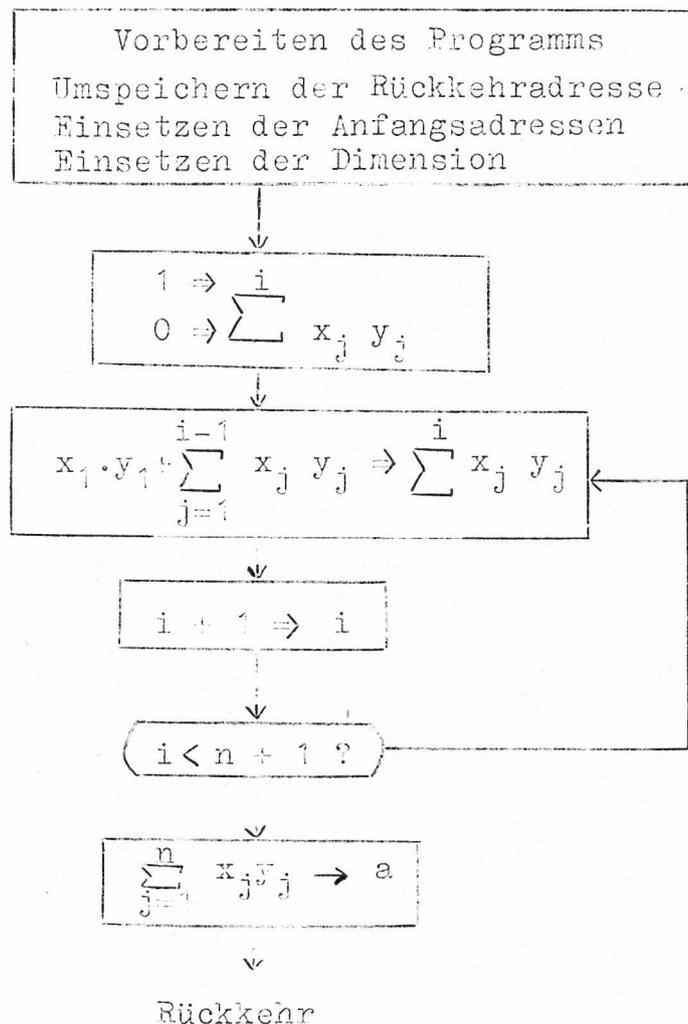
Bei diesem Programm wurde speziell auch Wert auf optimale Programmierung gelegt, deshalb liegen die Koeffizienten a_j so verstreut und andere Werte ($\frac{\pi}{2}$ und -1) werden mehrfach gespeichert, um Zugriffszeiten zu vermeiden.

6.2 Skalarprodukt zweier n-dimensionalen Vektoren.

$$(B, \eta) = \sum_{j=1}^n x_j y_j$$

Das Programm soll als Unterprogramm aufgestellt werden mit beliebiger Dimension n. Ferner sollen auch die Komponenten der Vektoren an beliebiger Stelle der Trommel stehen. Beim Sprung in dieses Unterprogramm müssen dann jeweils Dimension und Anfangsadressen von den Komponenten der Vektoren angegeben werden. Das Beispiel wird zeigen, wie man Adressen umrechnen kann.

Strukturdiagramm



Die Angaben über Dimension und Anfangsadressen mögen in den Schnellspeichern 6 bis 8 stehen.

<6> = n' Dimension

<7> = k' Anfangsadresse der Komponenten von \mathcal{B}

<8> = l' Anfangsadresse der Komponenten von \mathcal{N}

Als erstes soll das Programm mit Hilfe der Befehle für die Adressenrechnung, als zweites unter Benutzung von Indexregistern aufgestellt werden.

1700	B5	Rückkehradresse umspeichern
1	T1726	
2	B1734	Einsetzen der Dimension
3	A6	CSn \rightarrow 1723
4	T1723	
5	B1735	
6	A7	Einsetzen der Anfangsadresse von \mathcal{E}
7	T1713	Bk \rightarrow 1713
8	B1735	Einsetzen der Anfangsadresse von \mathcal{N}
9	A8	B1 \rightarrow 1715
1710	T1715	
1	T11	0 \Rightarrow i-1 \rightarrow 11
2	T12	0 \rightarrow $\sum x_j y_j$
3	(Bk)	
4	T6	$x_i \rightarrow 6$
5	(B1)	$x_i y_i \rightarrow a, 6$
6	X	
7	B12	
8	+	$x_i y_i + \sum_{j=1}^{i-1} x_j y_j \rightarrow 6, 12$
9	T12	
1720	B11	
1	CA1	<11> +1 \rightarrow 11
2	U11	
3	(CSn)	
4	QQE1727	
5	B6	$\sum_{j=1}^n x_j y_j \rightarrow a, 6$ Rücksprung
6	(E...)	
7	B1713	
8	CA1	B (k+1) \Rightarrow Bk
9	T1713	

1730	B1715	
1	CA1	$B(l+1) \Rightarrow l1$
2	T1715	
3	E1713	Sprung: Erneute Durchrechnung.
4	CS0	
5	B0+0	

Das gleiche Programm mit Verwendung von Indexregistern würde lauten:

1700	B5	Rückkehradresse umspeichern
1	T1727	
2	B1728	Lesebefehl für die x_i vorbereiten
3	A7	$GKB11+k \rightarrow 1713$
4	T1713	
5	B1728	Lesebefehl für die y_i vorbereiten
6	A8	$GKB11+l \rightarrow 1716$
7	T1716	
8	B1729	Einsetzen der Dimension
9	A6	$CSn \rightarrow 1724$
1710	T1724	
1	T11	$0 \Rightarrow i-1 \rightarrow 11$
2	T12	$0 \Rightarrow \sum x_j y_j \rightarrow 12$
3	(GKB11+k)	
4	0	$x_i \rightarrow 6$
5	T6	
6	(GKB11+l)	
7	0	$x_i y_i \rightarrow a,6$
8	X	
9	B12	
1720	+	$x_i y_i + \sum_{j=1}^{i-1} x_j y_j \rightarrow 6,12$
1	T12	
2	CGB11+1	$\langle 11 \rangle + 1 \rightarrow 11, a$
3	0	
4	(CSn)	
5	QQE1713	$\langle 11 \rangle < n$ nächste Komponente
6	B12	$\sum_{j=1}^n x_j y_j \rightarrow a,6$ Rücksprung.
7	(E.....)	
8	GKB11+0	
9	CS0	

Die eingeklammerten Befehle werden erst vom Programm selbst eingesetzt.

Ein Vergleich der beiden Programme zeigt deutlich den Vorteil, den die Benutzung der Adressensubstitution mit sich bringt:

- 1.) Enthält das zweite Programm weniger Befehle. (Die beiden in Zelle 1728 und 1729 stehenden Befehle zum Aufbau des Programms können auch in die hinter den G-Befehlen freigelassenen Zellen eingespeichert werden).
- 2.) Ist die Rechenzeit des zweiten Programms kürzer. Beim ersten Programm gehen durch die Abänderung der Befehle in den Zellen 1723 und 1715 je zwei Trommelumdrehungen verloren.

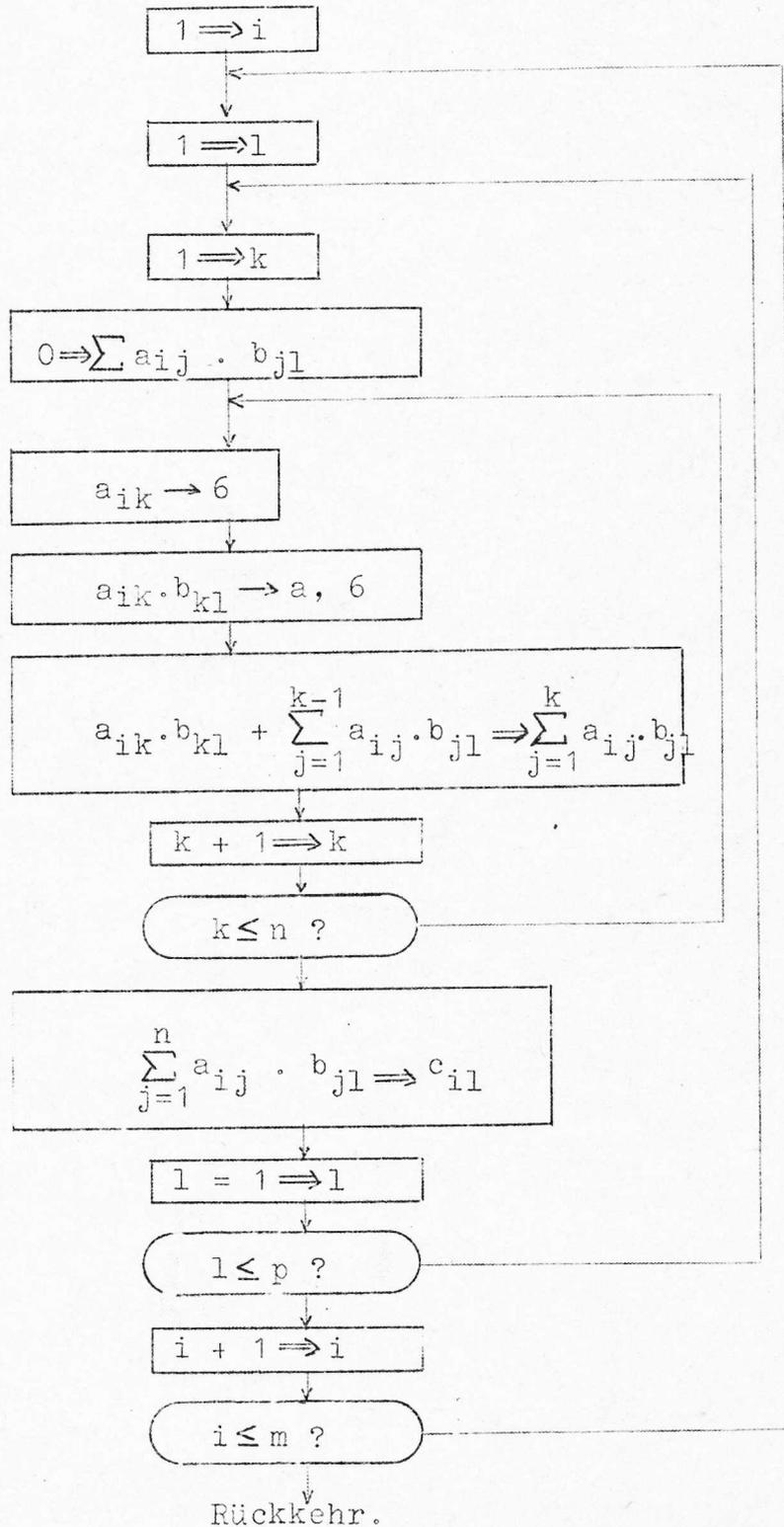
6.3 Matrizenprodukt.

$$A_{m,n} \cdot B_{n,p} = C_{m,p}$$

Als nächstes Beispiel sei das Matrizenprodukt zweier Rechteckmatrizen gegeben.

Strukturdiagramm

Vorbereitung des Programms:



Das Programm soll wiederum als Unterprogramm aufgerufen werden können. Beim Aufruf (F2001) sollen die folgenden Angaben gemacht sein.

Anfangsadresse der Elemente von \mathcal{A}	$\{a_{11}\} \rightarrow 11$
Anfangsadresse der Elemente von \mathcal{B}	$\{b_{11}\} \rightarrow 12$
Anfangsadresse der Elemente von \mathcal{C}	$\{c_{11}\} \rightarrow 13$
Spaltenzahl $\mathcal{A} \equiv$ Zeilenzahl \mathcal{B}	$n' \rightarrow 14$
Spaltenzahl \mathcal{C}	$p' \rightarrow 15$
Anzahl der Elemente von \mathcal{A}	$m' \cdot n' \rightarrow a$

Die Elemente seien zeilenweise fortlaufend gespeichert.

2001	A2037	
2	A11	
3	T2067	CS $\{a_{mn}\} \rightarrow 2067$
4	B2037	CS0 a,6
5	U6	
6	A14	CSn $\rightarrow 2061$
7	T2061	
8	B6	CSp $\rightarrow 2058$
9	A15	
2010	T2058	
1	B2050	
2	A15	CGKB12 + p $\rightarrow 2039$
3	T2039	
4	B2040	
5	A14	CAn $\rightarrow 2065$
6	T2065	
7	B11	
8	CS1	$\{a_{11}\} -1 \rightarrow 11$
9	U11	
2020	A14	
1	T14	$\{a_{1n}\} \rightarrow 14$
2	B12	
3	S15	$\{b_{11}\} -p \rightarrow 12, 2033$
4	U12	
5	T2033	
6	B13	
7	CS1	$\{c_{11}\} -1 \rightarrow 13$
8	T13	
9	B5	Rückkehradresse umspeichern

2030	T2073	
1	T2034	$0 \Rightarrow 1 - 1 \rightarrow 2034$
2	E2035	
3	$\{b_{11}\} - p$	
4	$\frac{1}{1} = \frac{1}{1}$	
5	T15	$0 \Rightarrow \sum a_{ij} b_{jl}$
6	CGKB11+1	} $\langle 11 \rangle + 1 \rightarrow 11$ } $a_{ik} \rightarrow 6$
7	CSO	
8	T6	
9	CGKB12+p	} $\langle 12 \rangle + p \rightarrow 12$ } $a_{ik} b_{kl} \rightarrow a, 6$
2040	CAO	
1	X	
2	B15	
3	+	$a_{ik} \cdot b_{kl} + \sum_{j=1}^{k-1} a_{ij} b_{jl} \rightarrow 15$
4	T15	
5	B11	$\langle 11 \rangle < a_{in}$ Sprung nach 2036
6	S14	
7	QQE2036	
8	B15	
9	CGKT13+1	$\langle 15 \rangle \Rightarrow c_{i1}$
2050	CGKB12+0	
1	B2034	
2	CA1	} $\{b_{11}\} - p \rightarrow 12$
3	U6	
4	A2033	
5	T12	
6	B6	$\langle 2034 \rangle + 1 \rightarrow a, 2034$
7	U2034	
8	CSp	wenn $1 \geq p$ Sprung nach 2064
9	PPE2064	
2060	B11	
1	CSn	$\langle 11 \rangle - n \rightarrow 11$
2	T11	
3	E2035	gleiche Zeile von \mathcal{C}_l mit nächster Spalte von \mathcal{E}
4	B14	$\langle 14 \rangle + n \rightarrow 14$
5	CAn	Umschalten auf neue Zeile von \mathcal{C}_l
6	U14	$(i + 1 \Rightarrow i)$
7	CS $\{a_{mn}\}$	} $\langle 14 \rangle \cong \{a_{mn}\}$ Rückkehr
8	PPE2073	
9	B2033	$\{b_{11}\} - p \rightarrow 12$

2070		T12		
1		T2034	$0 \Rightarrow 1$	- 1
2		E2035	Sprung nach 2035.	
3		E{Rückkehr}		

Da Elemente der Matrizen zeilenweise fortlaufend gespeichert sind, erhöht sich die Adresse

- a) beim Übergang von einer Spalte zur nächsten um 1
- b) beim Übergang zur nächsten Zeile dagegen um die Anzahl der Spalten.

Deshalb wird die Adresse für den Aufruf der a_{ik} jeweils um 1 erhöht: Befehl CGKB11+1 in Zelle 2036.

Für den Aufruf der b_{kl} wird die Adresse um p (= Anzahl der Spalten von \mathcal{L}) erhöht: Befehl CGKB12+p in Zelle 2039.

6.4. Reelle Nullstellen eines Polynoms n-ten Grades

$$f(y) = \sum_{i=0}^n a_i y^i$$

Für dieses Polynom sollen im Intervall

$$y_0 \leq y \leq y_n$$

die Nullstellen berechnet werden.

Mit Hilfe des Horner'schen Schemas wird zunächst das Vorzeichen von $f(y_0)$ bestimmt und weiter die Funktionswerte mit vergrößertem y berechnet. Wenn ein Vorzeichenwechsel von $f(y)$ gegenüber dem Anfangswert $f(y_0)$ auftritt, hat man mit dem y eine Näherung für eine Wurzel gefunden, die man nach Newton iterativ verbessert.

$$y_{r+1} = y_r - \frac{f(y_r)}{f'(y_r)}$$

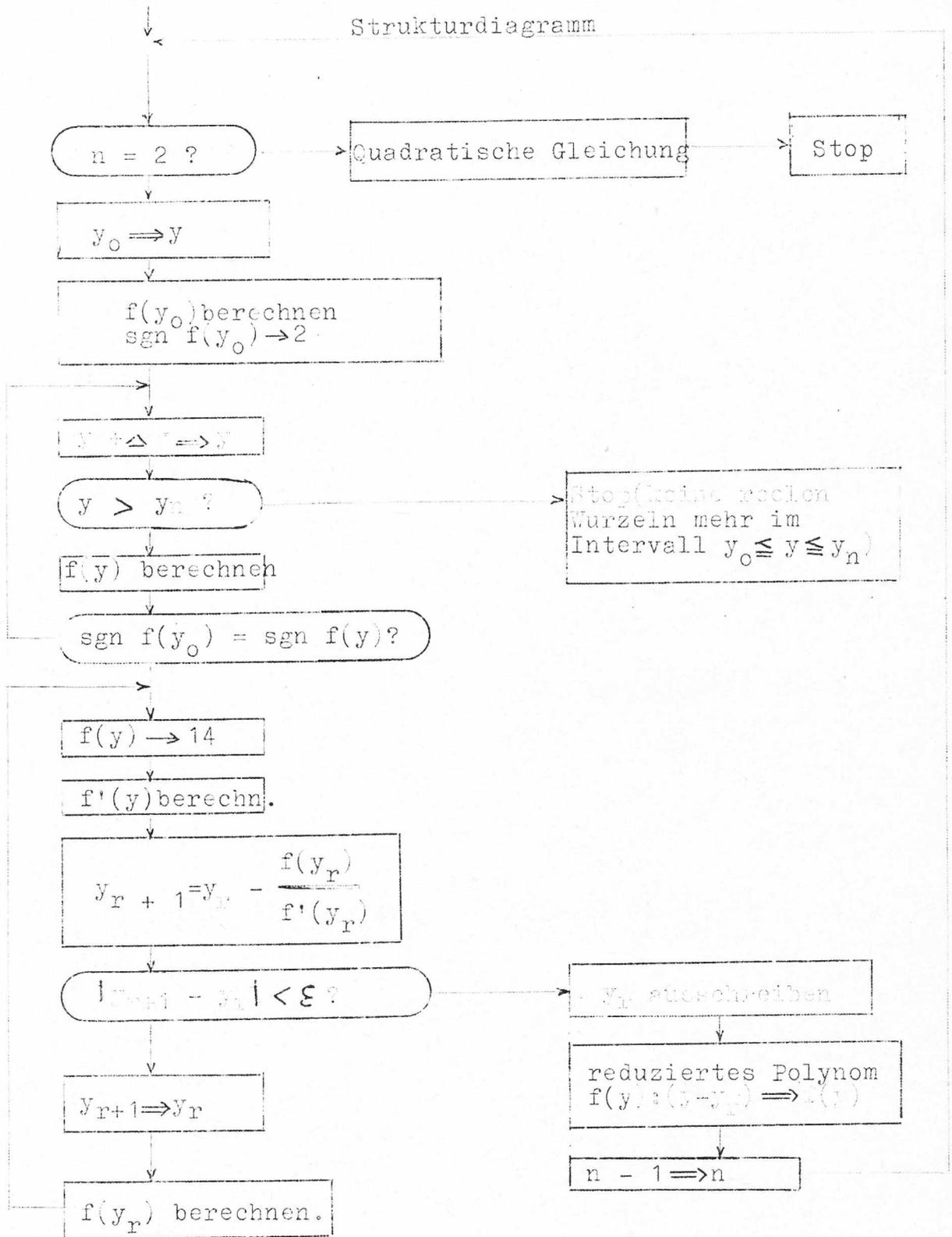
Da y_r eine Näherung für eine Wurzel ist, kann man für $f'(y_r)$ angenähert setzen

$$f'(y) \approx \frac{f(y)}{y - y_r}$$

Die Koeffizienten zur Berechnung von $-\frac{f(y)}{y - y_r}$ ergeben sich selbsttätig beim Horner'schen Schema.

Ist $|y_{r+1} - y_r| < \epsilon$ so wird y_{r+1} herausgeschrieben und für

$f(y) : (y - y_{r+1})$ erfolgt eine erneute Durchrechnung.



2500	B2630	}	$n \rightarrow 13$	
1	U13			
2	CS3	}	$n = 2 ? \rightarrow$ Quadratische Gleichung	
3	QQNE2596			
4	B2631			
5	T12	}	$y_0 \rightarrow 12$	
6	F2543			$f(y_0)$ berechnen
7	T2		$f(y_0) \rightarrow 2$	
8	B12	}		
9	T6			
2510	B2632			
1	+	}	$y + \Delta y \Rightarrow y$	
2	T12			
3	B2633	}		
4	-			wenn $y \geq y_n + \Delta y$ Stop
5	PPZO			
6	F2543		$f(y)$ berechnen	
7	U14		$f(y) \rightarrow 14$	
8	PPPE2508	}	wenn $\text{sgn } f(y_0) = \text{sgn } f(y)$	
9	QQQE2508			Sprung nach 2508
2520	E2561		$f(y) : (y - y_r)$ berechnen	
1	U7	}		
2	B14			
3	T6			
4	7		$\frac{f(y)}{f(y) : (y - y_r)} \Rightarrow \delta y \rightarrow 15$	
5	:	}		
6	T15			
7	B12			
8	T6	}		
9	B15			$y_r - \delta y \Rightarrow y_r + 1 \rightarrow 14$
2530	-	}		
1	T14			
2	B12			
3	-	}		
4	QQM			
5	B2634			wenn $ y_r + 1 - y_0 < \epsilon y_r$ zu schreiben
6	-	}		
7	QQE2576			
8	B14			
9	T12		$y_r + 1 \Rightarrow y_r$	

2540 F2543 f(y) berechnen
1 U14 f(y) \rightarrow 14
2 E2561 f(y):(y-y_r) berechnen, anschliessend Sprung
nach 2521

3 B5
4 T2560
5 T11
6 B2635
7 T6
8 B12
9 X

2550 GKB11+2636

1 0 } Berechnung von f(y)
2 +
3 GKT11+2660
4 0
5 CGB11+1
6 0
7 S13
8 QQE2548
9 B6

2560 E (Rückkehr)

1 CB1
2 T11
3 B2635
4 T6
5 B12
6 X
7 GKB11+2659

8 0 } Berechnung von
9 + f(y):(y-y_r)

2570 CGB11+1

1 0
2 S13
3 QQE2565
4 B6
5 E2521

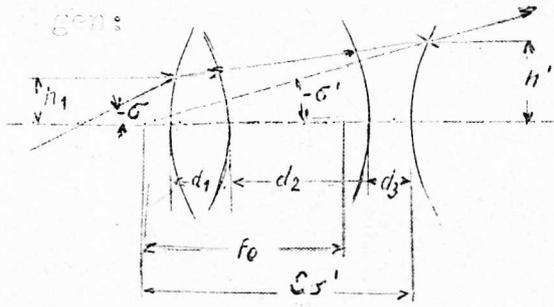
6 B14 }
7 D } y ausschreiben
8 CB1
9 T11

2580	GKB11+2659	}	$f(y):(y-y_r) \quad f(y)$
1	0		
2	GKT11+2635		
3	0		
4	CGB11+1		
5	0		
6	S13		
7	QQE2580	}	$n-1 \Rightarrow n$
8	B13		
9	CS1		
2590	E2501		
1	-2		
2	0		
3	0		
4	0		
5	0		
6	B2635		a_2
7	T6		
8	B2591		-2
9	X		
2600	T15		$-2a_2$
1	D2591		-2
2	X		
3	B2637		a_0
4	X		
5	T14		$4 a_0 a_2$
6	B2636		a_1
7	U6		
8	X		a_1^2
9	B14		
2610	-		$a_1^2 - 4a_0 a_2 < 0$ keine reellen Wurzeln. Stop
1	QQE20		
2	T		
3	B15		$\frac{\sqrt{a_1^2 - 4a_0 a_2}}{-2a_2} \rightarrow 14$
4	:		
5	T14		
6	B2636		
7	T6		
8	B15		$\frac{a_1}{-2a_2} \rightarrow 13$
9	:		

2620	T13	
1	B14	
2	+	$\frac{-a_1 - \sqrt{\dots}}{2a_2} = y$ ausschreiben
3	D	
4	B13	
5	T6	
6	B14	$\frac{-a_1 + \sqrt{\dots}}{2a_2} = y$ ausschreiben
7	-	
8	D	
9	Z0	Stop
2630	n'	Grad
1	y ₀	Anfangswert
2	Δ y	Schrittweite
3	y _n + Δ y	Obere Grenze
4	ε	Genauigkeitsschranke
5	a _n	
6	a _{n-1}	Koeffizienten
7	a _{n-2}	
.		
.		
.		
2660		
.		zur Aufnahme der Koeffizienten von
.		f(y):(y-y _r) bestimmt.
.		
2685		

6.5 Beispiel aus der Optik.

Zur Berechnung der Gauss'schen Bildebene und der Brennweite eines optischen zentrierten Linsensystems gelten die folgenden Gleichungen:



$$\sigma' = \frac{n}{n'} \sigma + \left(1 - \frac{n}{n'}\right) \frac{h}{r}$$

$$h' = h - d \sigma'$$

$$s_o' = h' / \sigma'_k$$

$$f_o' = h / \sigma'_k$$

Nehmen wir ein System von 8 Linsen an, dann sollen enthalten die Speicher

- 1120 bis 1127 die Radien r_1, r_2, \dots, r_8
- 1128 bis 1136 die Brechzahlen n_1, n_2, \dots, n_9
- 1137 bis 1143 die Dicken $d_{1,2}; d_{2,3}; \dots, d_{7,8}$

Die benötigten Eingangswerte $\sigma_1, h_1, 1, 8'$ (Anzahl der Flächen) sind gespeichert nach 1144 bis 1147. Das Programm beginnt mit Zelle 1150.

1120	r_1
.	r_2
.	.
.	.
.	.
1127	r_8
1128	n_1
.	.
.	.
.	.
1136	n_9
1137	$d_{1,2}$
.	.
.	.
.	.
1143	$d_{7,8}$
1144	σ_1
5	h_1
6	1
7	8'

Start 1150	B1144	σ_1	} Einlauf
1	T14		
2	B1145	h_1	
3	T15		
4	T11		
5	GKB11+1128	n	
6	0		
7	T6		
8	GKB11+1129	n'	
9	0		
1160	:	n/n'	
1	T12		
2	B14	σ	
3	X		
4	T13	$(\frac{n}{n'})\sigma$	
5	B1146	1	
6	T6		
7	B12		
8	-	$1 - \frac{n}{n'}$	
9	B15	h	
1170	X		
1	GKB11+1120	r	
2	0		
3	:	$(1 - \frac{n}{n'})^h / r$	
4	B13	$(\frac{n}{n'})\sigma$	
5	+	.	
6	T14	σ'	
7	GGB11+1	$v+1 \Rightarrow v$	
8	0		
9	S1147	$v-8 \geq 0? \rightarrow$ Schlussrechnung	
1180	PPE1191		
1	GKB11+1136	d	
2	0		
3	X		
4	T12	$d \cdot \sigma'$	
5	B15		
6	T6		
7	B12	$h - d\sigma'$	
8	-		
9	T15	h'	
1190	E1155	Wiederholung:nächste Fläche	

1191	B15	h
2	T6	
3	B14	ξ'
4	:	
5	U1148	$h/\xi' \Rightarrow s_0'$
6	D	
7	B1145	h_1
8	T6	
9	B14	ξ'
1200	:	
1	U1149	$h_1/\xi' \Rightarrow f_0'$
2	D	
3	Z0	

6.6 Lösung einer Differential Gleichung nach dem Verfahren von Runge und Kutta.

Die Differentialgleichung

$y' = f(x,y)$ mit dem Anfangswert $y(x_0) = y_0$ soll in dem Intervall $x_0 \leq x \leq x_n$ mit der gewählten Schrittweite $h = \frac{x_n - x_0}{n}$ gelöst werden. Der Einfachheit halber wurde Konstante Schrittweite gewählt. Häufig wird man mit ganzer und halber Schrittweite rechnen und sie dann entsprechend der Güte der Annäherung wählen.

Die Runge-Kutta-Formeln geben die Rechenvorschrift:

$$x_i = x_0 + ih$$

$$k_1^{(i)} = h f(x_i, y_i)$$

$$k_2^{(i)} = h f(x_i + \frac{h}{2}, y_i + \frac{k_1}{2})$$

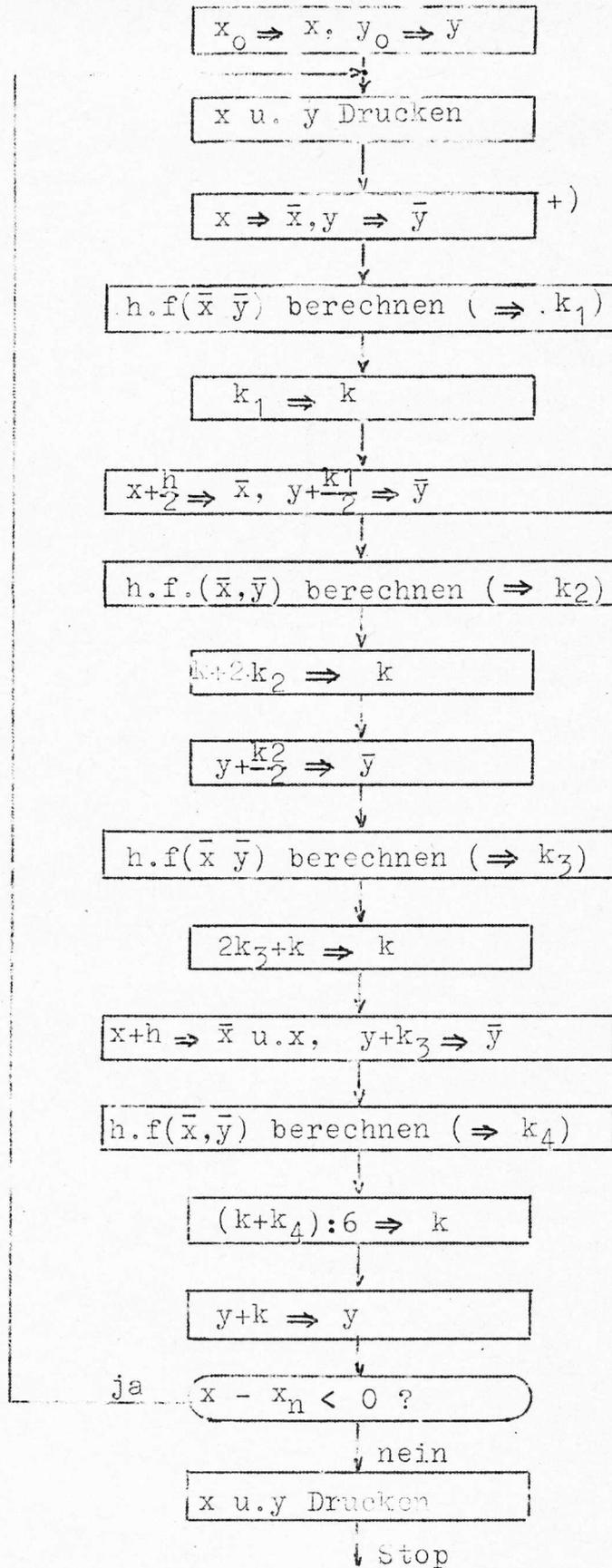
$$k_3^{(i)} = h f(x_i + \frac{h}{2}, y_i + \frac{k_2}{2})$$

$$k_4^{(i)} = h f(x_i + h, y_i + k_3)$$

$$k^{(i)} = \frac{1}{6}(k_1^{(i)} + 2k_2^{(i)} + 2k_3^{(i)} + k_4^{(i)})$$

$$y_{i+1} = y(x_{i+1}) = y_i + k^{(i)}$$

Daraus ergibt sich folgendes Strukturdiagramm:



+) Die Anweisung "f(x̄, ȳ) berechnen" stellt ein einziges Unterprogramm dar, das entsprechend der Differentialgleichung aufzustellen ist. x̄, ȳ stellen die Werte dar, mit denen das Unterprogramm jeweils die Funktion f(x̄, ȳ) bilden soll.

Es ist manchmal zu empfehlen, sich bei der Aufstellung eines Programmes nicht auf bestimmte Adressen festzulegen, sondern das Programm mit relativen Adressen zu vercoden. Man hat dann die Möglichkeit, das Programm in beliebige freie Speicherzellen zu bringen.

Das Programm wird dann so aufgestellt, als wenn es bei Zelle 0 beginnen würde. Alle Befehle, die sich auf Trommelzellen mit Relativadressen beziehen, sind mit einem Bandbefehl ...Am zu versehen, wobei mit m eine bestimmte Trommelzelle oder Schnellspeicher 12 (bzw. bei Schnellspeichererweiterung Schnellspeicher 18, 22 - 31) gemeint ist. (siehe Bandbefehle 3.12.) Vor Einlegen des Programmes in den Abtaster ist nur dafür zu sorgen, daß in die Zelle m diejenige Adresse als Konstante gespeichert wird, von der ab das Programm eingelesen werden soll. Soll z.B. das Programm von der Zelle n ab gespeichert werden, so wird vorweg ein Band mit folgenden Angaben eingelesen:

- a. wenn Schnellspeicher 12, b. Trommelspeicher m benutzt wird

C B n E

C B n E

T m E

(siehe E-Bandbefehle 3.12).

(Der Befehl T12E erübrigt sich bei a., weil bei E-Bandbefehlen Schnellspeicher 12 Hilfsakkumulator ist und dadurch die Konstante n sowieso schon dort gespeichert wird).

Das nachfolgende Programm benötigt ab Zelle 0 folgende Angaben

0	x_0	6	frei, Speicher für y
1	$x_a - \epsilon$ ($0 < \epsilon < h$)	7	0.5
2	h	8	0.166666667
3	$y_0 = y(x_0)$	9	E o+a'; a'=Adresse von wo ab das
4	$\frac{h}{2}$		Unterprogramm für $f(\bar{x}, \bar{y})$ gespeichert
5	frei, Speicher für x		ist.

Das Speichern der obenstehenden Werte ab Zelle n erfolgt dann durch ein Band mit den Angaben:

TOA12T

x₀

x_n

.

.

.

Soll das Programm z.B. ab Zelle n+20 folgen:

	T20A12T		
20	B0A12	}	$x_0 \Rightarrow x$
1	T5A12		
2	B3A12	}	$y_0 \Rightarrow y$
3	T6A12		
4	B5A12	}	$x \Rightarrow \bar{x} \rightarrow 13$
5	U13		
6	D		x Drucken
7	B6A12	}	$y \Rightarrow \bar{y}$
8	U14		
9	D		y Drucken
30	F9A12		berechne $f(\bar{x}, \bar{y})$
1	B2A12	}	$h f(\bar{x}, \bar{y}) \Rightarrow k_1$ (Es wird verlangt, dass das Unterprogramm
2	X		
3	T15		$k_1 \Rightarrow k$ $f(\bar{x}, \bar{y})$ nach a. 6 bringt.)
4	B5A12	}	
5	T6		
6	B4A12		$x + \frac{h}{2} \Rightarrow \bar{x} \rightarrow 13$
7	+		
8	T13		
9	B15		
40	T6		
1	B7A12	}	$y + \frac{k}{2} \Rightarrow \bar{y}$
2	X		
3	B6A12		
4	+		
5	T14		
6	F9A12		berechne $f(\bar{x}, \bar{y})$
7	B2A12	}	$h \cdot f(\bar{x}, \bar{y}) \Rightarrow k_2 \rightarrow 12$
8	X		
9	U12		
50	U6		
1	+	}	$2k_2 + k \Rightarrow k$
2	B15		
3	+		
4	T15		
5	B12		
6	T6		
7	B7A12		$y + \frac{k}{2} \Rightarrow \bar{y}$
8	X		
9	B6A12		
60	+		

61	T14	}	
2	F9A12		berechne $f(\bar{x}, \bar{y})$
3	B2A12	}	
4	X		$h f(\bar{x}, \bar{y}) \Rightarrow k_3 \rightarrow 12$
5	U12	}	
6	U6		
7	+	}	$2k_3 + k \Rightarrow k$
8	B15		
9	+	}	
70	T15		
1	B5A12	}	
2	T6		
3	B2A12	}	$x+h \Rightarrow \bar{x} \text{ u. } x$
4	+		
5	U13	}	
6	T5A12		
7	B6A12	}	
8	T6		
9	B12	}	$y+k_3 \Rightarrow \bar{y} \rightarrow 14$
80	+		
1	T14	}	
2	F9A12		berechne $f(\bar{x}, \bar{y})$
3	B2A12	}	
4	X		$h \cdot f(\bar{x}, \bar{y}) \Rightarrow k_4$
5	B15	}	
6	+		$(k+k_4) : 6 \Rightarrow k$
7	B8A12	}	
8	X		
9	B6A12	}	
90	+		$y+k \Rightarrow y$
1	T6A12	}	
2	B5A12		
3	T6	}	
4	B1A12		$x-x_n < 0?$
5	-	}	
6	QQE24A12		
7	B5A12	}	
8	D		
9	B6A12	}	
100	D		
1	Z0		

Ab Zelle a' ist anschliessend das Programm für $f(\bar{x}, \bar{y})$ zu speichern. Dabei ist zu berücksichtigen, dass \bar{x} u. \bar{y} im Schnellspeicher 13 und 14 steht und $f(\bar{x}, \bar{y})$ am Ende des Unterprogrammes in den Akkumulator und Schnellspeicher 6 gebracht wird.

6.7 Gauss'scher Algorithmus.

Es soll das Programm für die Lösung eines linearen Gleichungssystems mit n Unbekannten aufgestellt werden. Es wird hier nur der Gauss'sche Algorithmus durchgeführt, ohne die Lösung zu verifizieren. Bei grösserem n muss man möglicherweise noch durch eine Iteration oder ähnliches die mit dem Gauss'schen Algorithmus gefundenen Lösungen verbessern, bis die gewünschte Genauigkeit erreicht ist.

Bezeichnung:

$$\begin{array}{rcl}
 a_{11}x_1 + a_{12}x_2 + \dots & + a_{1n}x_n & = b_1 \\
 a_{21}x_1 + a_{22}x_2 + \dots & + a_{2n}x_n & = b_2 \\
 \cdot & & \cdot \\
 \cdot & & \cdot \\
 \cdot & & \cdot \\
 a_{n1}x_1 + a_{n2}x_2 + \dots & + a_{nn}x_n & = b_n
 \end{array}$$

Die Koeffizienten sollen zeilenweise fortlaufend gespeichert werden: $a_{11}, a_{12}, \dots, a_{1n}, b_1 (=a_{1,n+1}), a_{21}, a_{22}, \dots$. Es ist vorgesehen, dass das Programm ab Zelle 1510 gespeichert werden soll. Die Zellen 2000 bis $2000 + n^2 + n$ sind zur Aufnahme von Zwischenwerten bestimmt.

In den Speicherzellen 1490 - 1492 sollen folgende Angaben stehen:

- 1490 {a₁₁} = die Adresse von a₁₁
- 1491 n' = die Zahl der Unbekannten
- 1492 {x₁} = die Adresse, wo die errechneten x - Werte stehen sollen.

Als Hilfsspeicher dienen weiter die Zellen 1493 bis 1496.

1493	n + 1	1495	a _{ii}
1494	1	1496	{a _{nn} }

Speicherliste für den Gauss'schen Algorithmus.

1510	B1491	}
1	U12	
2	CA1	
3	U7	
4	U1493	
5	T6	
6	CB1999	
7	T14	
8	B1490	
9	A1654 GKB11+0	

1520 T1523
 1 T11
 2 T15
 3 GKB11+{a₁₁}
 4 0
 5 CGKT14+1
 6 0
 7 CGB11+1
 8 0
 9 S6

Umspeichern der Koeffizienten
 in die Zellen ab 2000

1530 QQE1523
 1 B7
 2 A6
 3 T6
 4 CGB15+1
 5 0
 6 S12
 7 QQE1523
 8 B1655 CS2000
 9 A12

1540 T1578
 1 CB1
 2 U1494
 3 S12
 4 T2
 5 CB2000
 6 T11
 7 B1654 GKB11+0
 8 A7
 9 U1561

Vorbereitung

1550 CA1
 1 T1569
 2 B1656 GKT11+0
 3 A7
 4 T1575
 5 GKB11+0
 6 0
 7 U6
 8 U1495
 9 PPNS4

a_{ii} → a,6,1495

1560 PPE1620
 a_{ii}=0 ? → Umordnen der Zeilen

1561	GKB11+k(n+1)	}	
2	0		
3	M		$-a_{k,i} \rightarrow a,6$
4	PPNS4		
5	PPE1586		$a_{k,i}=0 ? \rightarrow \text{Übergang } k+1 \Rightarrow k$
6	B7		
7	:		$-\frac{a_{ki}}{a_{ii}} \rightarrow 15$
8	T15		
9	CGKB11+1)		
1570	0		
1	X		
2	GKB11+k(n+1)		$a_{i,i+r} (-\frac{a_{ki}}{a_{ii}}) + a_{k,i+r} \Rightarrow a_{k,i+r}$
3	0		
4	+		
5	GKT11+k(n+1)		
6	0		
7	B11		
8	CS $\{a_{i,n+1}\}$	$i+r = n+1 ?$	
9	PPE1583		
1580	B15		
1	T6	$-\frac{a_{ki}}{a_{ti}} \rightarrow 6$	
2	E1569	erneut für $r+1 = r$ durchrechnen	
3	B11		
4	S1491	$0 \Rightarrow r$	
5	T11		
6	CGU2+1		
7	0	bei $k=n$ Übergang $i+1 \Rightarrow i$	
8	PE1602		
9	B1493		
1590	U7		
1	A1561		
2	T1561		
3	B1572		
4	A7 A7	$k+1 \Rightarrow k$	
5	T1572		
6	B1575		
7	A7		
8	T1575		
9	B1495		
1600	U7	$a_{ii} \rightarrow 7$	
1	E1561		

1602	B11	
3	A1493	
4	CA1	
5	T11	
6	B1491	
7	CS1	> i+1 ⇒ i
8	T1491	
9	B1494	
1610	CA1	
1	U1494	
2	S12	
3	PPE1664	i=n Berechnung der x_i
4	T2	
5	B1493	
6	U7	
7	A1578	i+1 ⇒ k
8	T1578	
<u>9</u>	<u>E1547</u>	
1620	B1561	
1	T1622	
2	GKB11+k(n+1)	
3	0	
4	PPNS4	
5	PPE1660	
6	B1622	
7	U1633	
8	CS191	
9	A1656 GKE11+0	
1630	T1638	
1	B1578	wenn ein $a_{ii}=0$
2	T1644	
3	GKE11+k(n+1)	
4	0	Umordnen der Zeilen.
5	T6	
6	GKB11+0	
7	0	
8	GKT11+k(n+1)	
9	0	
1640	B6	
1	GKT11+0	
2	0	
3	B11	

1644 CS{ $a_{i,n+1}$ }
5 PPE1649
6 CGU11+1
7 0
8 E1633
9 B11

1650 S1491
1 T11
2 E1555
3 0

4 GKB11+0
5 CS2000
6 GKT11+0
7 GKT13+0
8 GKB13+0
9 0

Hilfswert

1660 B1622
1 A1493
2 T1622
3 E1622
4 B11
5 T1496
6 T1494
7 B12
8 CA2
9 U1493

1670 CS3
1 U1491
2 A1492
3 U6
4 A1657
5 T1692
6 A6
7 A1658
8 T1710
9 T15

GKT13+0

Vorbereitung

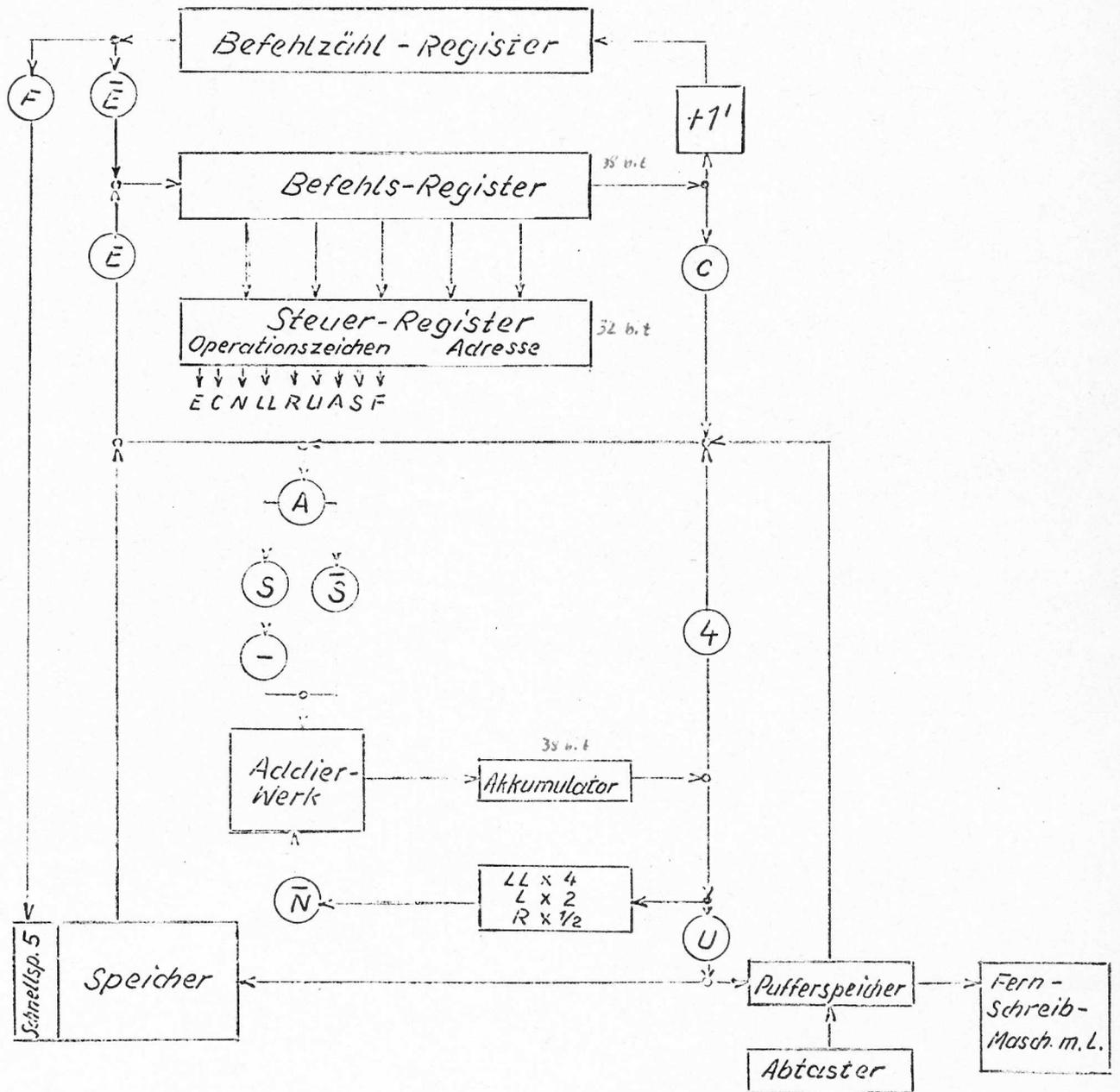
GKB13+0

1680	GKB11+0
1	0
2	T1495
3	B1494
4	T13
5	GKB11+1
6	0
7	T6
8	B15
9	-
1690	B1495
1	:
2	GKT13+ $\{x_n\}$
3	0
4	B1496
5	S1493
6	U1496
7	T11
8	T15
9	B1494
1700	CS1
1	U1494
2	A1491
3	QQE1721
4	GKB11+0
5	0
6	T1495
7	CGKB11+1
8	0
9	T6
1710	GKB13+ $\{x_n\}$
1	0
2	X
3	B15
4	+
5	T15
6	B13
7	PPE1683
8	CA1
9	T13
1720	E1707

Berechnung der x_i

1721	B1492	
2	A1658	GKB13+0
3	T1725	
4	T13	
5	GKB13+{x ₁ }	
6	0	
7	D	
8	CGB13+1	
9	0	
1730	S12	
1	QQE1725	
2	Z0	

Herausschreiben der
Lösungen



7.2 Befehlsliste

Befehl	Wirkung	intern
Sprung- und Rufbefehle		
Em	$\langle m \rangle \rightarrow b \quad \langle b \rangle + 1' \rightarrow c$	Em
Fm	$\langle m \rangle \rightarrow b \quad Em+1' \rightarrow c \quad \langle c \rangle \rightarrow 5$	Fm
NEm	$\langle m \rangle \rightarrow b \quad Em+1' \rightarrow c \quad 0 \rightarrow a$	NEm
Lese- und Speicherbefehle		
Bm	$\langle m \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$	NAm
Tm	$\langle a \rangle \rightarrow m \quad 0 \rightarrow a \quad \langle c \rangle \rightarrow b$	NUm
Um	$\langle a \rangle \rightarrow m \quad \langle a \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$	Um
Arithmetische Befehle		
		8011/20 8021/40 9801/20
+	1) $\langle 6 \rangle + \langle a \rangle \rightarrow a, 6$	F 417 F 294 F 348
-	1) $\langle 6 \rangle - \langle a \rangle \rightarrow a, 6$	F 418 F 422 F 426
X	1) $\langle 6 \rangle X \langle a \rangle \rightarrow a, 6$	F 272 F 304 F 367
:	1) $\langle 6 \rangle : \langle a \rangle \rightarrow a, 6 \quad \langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$	F 338 F 337 F 480
W	1) $\sqrt{\langle a \rangle} \rightarrow a, 6$	F 1055 F 1055 F 1055
HW	1) $\sqrt{1 - \langle a \rangle^2} \rightarrow a, 6$	HF1055 HF1055 HF1055
M	2) $\langle a \rangle x (-1) \rightarrow a, 6$	F 267 F 466 F 437
Bedingungszeichen		
PP . . .		$\langle a \rangle \geq 0$
QQ . . .		$\langle a \rangle < 0$
PPQQ . .		$\langle a \rangle = 0$
P . . .	Befehl . . .	nur ausf. $\langle 2 \rangle \geq 0$
Q . . .	wenn	$\langle 2 \rangle < 0$
Y . . .		$\langle 3 \rangle_{38} = 1$
nur Zuse Z 22 R		
PQQQ . .		$\langle a \rangle$ ein Befehl oder Klartext
Ausgabebefehle u. Tabellierungsbefehle		
D 1)	$\langle a \rangle \rightarrow$ Fernschreibmaschine	$\langle a \rangle \rightarrow a, 6$ F 644
F800 3)	Spaltensprung	$\langle c \rangle \Rightarrow \langle 5 \rangle \rightarrow b$
F1000 2)	Wagen- u. Zeilentransp.	
F840 3)	Einrücken um n Zeichenbreiten, wenn -n' im Akku	$0 \rightarrow a$

Ausgabebefehle mit vorangestelltem H (HD, HF800, HF1000, HF840) wirken auf die schnelle Ausgabe.

Befehl	Wirkung
	Stopbefehle
Z0	(Buchstabe Z Ziffer Null) Stop vom Programm her. <a> → a
0	Schalter bedingter Stop (Ziffer Null) Bei eingeschaltetem "Bedingter Stop" Stop der Maschine mit 0 im Befehlsregister. Nach Weiter und Start weiterer Ablauf des Programms. Bei ausgeschaltetem "Bedingter Stop" sofortiger weiterer Ablauf des Programms.
Z0+1E	Stop vom Lochstreifen
	Angaben zur Tabellierung
T1030T	
1'	Anzahl der Stellen bei Gleitkomma-Ausgabe
ℓ'	Anzahl der Stellen vor dem Komma
β'	Anzahl der Stellen nach dem Komma
k'	Anzahl der Spalten
z'	Anzahl der Zwischenräume
n'	Anzahl der Stellen bei Strichzahlen
+1' bzw. -1'	Kennzeichen für Gleitkomma-(+1') bzw. Festkomma (-1') - Ausgabe
	Bandbefehle
TmT	Das Folgende in Zelle m, m+1, m+2 ... speichern
.....E	Befehl . . . sofort ausführen
.....Am	<m> _t zur Adresse des Befehls addieren
	Aufruf des Leseprogramms
EO+1 4)	Eintasten oder Programmieren Leseprogramm mit mechanischem Abtaster
HEO+1 4)	" " " Leseprogramm mit photoelektrischem Abtaster.

- 1) Die Unterprogramme benutzen die Schnellspeicher 3-10
- 2) Die Unterprogramme benutzen die Schnellspeicher 3-6
- 3) Die Unterprogramme benutzen die Schnellspeicher 3-6 und 9

4) Das Leseprogramm benutzt die Schnellspeicher
2-15

Befehl	Wirkung
Befehle für Adressenrechnen	
Am	$\langle a \rangle + \langle m \rangle \rightarrow a$
Sm	$\langle a \rangle - \langle m \rangle \rightarrow a$
NSm	$-\langle m \rangle \rightarrow a$
CAn	$\langle a \rangle + n' \rightarrow a$
CBn	$n' \rightarrow a$
CSn	$\langle a \rangle - n' \rightarrow a$
CNSn	$-n' \rightarrow a$
Im	$\langle a \rangle \wedge \langle m \rangle \rightarrow a$
CIn	$\langle a \rangle \wedge n' \rightarrow a$
} $\langle c \rangle \rightarrow b$	
nur Zuse Z 22 R	
As+t	$\langle s \rangle + \langle t \rangle \rightarrow s, \quad \langle a \rangle \rightarrow a$
Bs+t	$\langle t \rangle \rightarrow s, \quad \langle a \rangle \rightarrow a$
Ss+t	$\langle s \rangle - \langle t \rangle \rightarrow s, \quad \langle a \rangle \rightarrow a$
NSs+t	$-\langle t \rangle \rightarrow s, \quad \langle a \rangle \rightarrow a$
} $\langle c \rangle \rightarrow b$	
Adressensubstitution	
G...m	$\langle m \rangle_t \Rightarrow \tilde{t}$
GK...s+t	$(\langle s \rangle + t)_t \Rightarrow \tilde{t}$
CGK...s+t	$\langle s \rangle + t \rightarrow s, (\langle s \rangle + t)_t \Rightarrow \tilde{t}$
} $\langle c \rangle + 1' \rightarrow c$ Befehl mit neuer Adresse \tilde{t} ausführen	
Zählbefehle	
CGUs+t	$\langle s \rangle + t \rightarrow s, \quad \langle c \rangle + 1' \rightarrow c$ Befehl CUt (wirkungslos) ausführen
CGBs+t	$\langle s \rangle + t \rightarrow s, (\langle s \rangle + t)_t \Rightarrow \tilde{t}, \quad \langle c \rangle + 1' \rightarrow c$ Befehl (Auch mit A,S oder NS anstelle von B) CB \tilde{t} ausführen

Automatische Löschungen bei der Übernahme eines Wortes aus dem Befehlsregister (b) nach dem Befehlszählregister (c)

- 1) Vor Übernahme des Textes in das Befehlszählregister werden gelöscht die Bedingungszeichen (PP,P,QQ,Q,Y,)N,F und G
- 2) Im zweiten Schritt eines G-Befehles vor Übernahme $\langle b \rangle \rightarrow c$ werden gelöscht alle Bits außer der Befehlskennzeichnung und den Trommeladressenstellen.

7.3 Schnellspeicherzellen Zuse Z 22

Die Schnellspeicher der Zuse Z 22 stehen ohne Zugriffszeit stets zur Verfügung. Sie nehmen vor allem die Zwischenresultate während der Rechnung auf. Alle Schnellspeicherzellen außer 0, 1, 17, 19 und 20 haben die gleiche Stellenkapazität wie eine Trommelspeicherzelle.

Folgende Schnellspeicher enthält die Zuse Z 22:

- 0 endet leer, liefert "0" in jeder Dualstelle
- 1 liefert "1" in der obersten Stelle (Vorzeichen bzw. Befehlskennzeichenstelle), in den anderen Stellen "0", das ist gleichbedeutend mit dem Befehl EO+0
0 und 1 nehmen bei Speicherungen nichts auf.
- 2 Vorzeichentestspeicher ^{*)}, sonst wie Speicher 6-15.
- 3 Testspeicher für die unterste Stelle. Er kann mit Schnellspeicher 4 (Akkumulator) gekoppelt und dann wie dieser verschoben werden. Einleitung einer 1' in der untersten Stelle bei H.
- 4 Der Akkumulator a mit vorgeschaltetem Addierwerk, mit Intersektionseinrichtung und Komplementwerk (Negation mit Addition der flüchtigen Eins)
Verzögerungsglieder für Rechts- und Linksverschiebungen, Vorzeichentest ^{*)})
- 5 Rückkehradressenspeicher wird bei F mit Inhalt des Befehlszählregisters c belegt. Sonst wie Speicher 6 - 15.
- 6-15 Schnellspeicher ohne Besonderheiten.
- 16 = 5. Der Speicher ist mit Speicher 5 identisch, kann unter der Adresse 16 aber nur gelesen werden.
- 17 In Abhängigkeit von einem Schalter kann unter Adresse 17 eine 1 in der untersten Dualstelle aufgerufen werden: z.B.
B17 Schalter ein 1' → a
Schalter aus 0 → a
- 19 Pufferspeicher für schnelle Ein- und Ausgabe.
- 20 " " gewöhnliche Ein- und Ausgabe.
Kapazität: bei 19 und 20 5 Dualstellen, beim Lesen in den untersten, beim Speichern in den oberen 5 Stellen.

^{*)} Der Test bezieht sich nur auf die oberste Dualstelle.

21 Bedingungsdatte 0-32 18, 22-31 normale SSP

Wird ein Pufferspeicher gelesen, so wird dasjenige Fernschreibzeichen ausgeliefert, das sich gerade unter den Lesekontakten des Abtasters befindet. Anschließend wird automatisch die nächste Fünferkombination des Lochstreifens unter die Abtasterkontakte transportiert.

Wird ein Puffer kurz hintereinander gelesen, so daß der Lochstreifentransport des vorhergehenden Lesebefehls noch nicht beendet ist, wartet die Zuse Z 22 automatisch so lange, bis der Lochstreifen transportiert ist und führt anschließend den neuen Lesebefehl durch. Der Lochstreifentransport dauert bei Adresse 20 etwa 75 msec, bei Adresse 19 5 msec.

Wird in einen Pufferspeicher gespeichert, so wird diese Fünferkombination anschließend automatisch als Fernschreibzeichen von der Schreibmaschine herausgeschrieben, bzw. auf den Locher gegeben.

Der Pufferspeicher gibt für die Zuse Z 22 auch das Freisignal, wann der Puffer erneut angesprochen werden kann. Ist noch kein Freisignal vorhanden, so wartet die Zuse Z 22.

Das Herausschreiben eines Zeichens auf dem Fernschreiber dauert 100 msec, das Lochen 40 msec.

Mit Schnellspeichererweiterung stehen noch die Zellen

18

22 bis 31

zur Verfügung, die keine Besonderheiten aufweisen.

Beschreibung zum Programmierungstisch
 =====

Der Programmierungstisch besteht aus

- A) Fernschreiber mit Locher
- B) Abtaster mit Schritt-Stop-Gerät

Bei der Bedienung des Programmierungstisches sind die folgenden Punkte zu beachten.

A) Der Fernschreiber mit Locher

Die Tastatur umfasst

- 8 Tasten für Betriebszeichen der Fernschreibmaschine
- 26 Tasten für die Buchstaben
- 23 Tasten für Ziffern und Zeichen

Die Tasten der 1. Gruppe können stets bedient werden, während von den beiden letzten Gruppen stets eine blockiert ist; die Buchstaben, wenn die Taste "Ziffern und Zeichen" betätigt wurde; und umgekehrt die Zeichen und Ziffern, wenn die "Buchstaben-Taste" gedrückt wurde.

Die Betriebszeichen

- | | | |
|----|--------------|--|
| 1. | A... | (= Bu): Blockierung der Ziffern- und Zeichen-Tasten und Freigabe der Buchstaben-tasten |
| 2. | 1... | (= Zi): Freigabe der Tasten für Ziffern und Zeichen, Sperrung der Buchstabentasten |
| 3. | Zwischenraum | (die große Taste unten in der Mitte ohne Aufschrift):
Der Wagen wird um 1 Zeichenbreite weitergerückt |
| 4. | < | (= Wagenrücklauf): Der Wagen wird an den Anfang der Zeile gerückt. |
| 5. | ≡ | (= Zeilentransport): Das Papier wird um eine Zeilenhöhe nach oben transportiert |
| 6. | ⊖ | (= Leertaste): Auf dem Lochstreifen wird nur das Transportloch gelocht |

7. (= Dauer): Das zuletzt angeschlagene Zeichen wird solange wiederholt, bis die Taste wieder losgelassen wird. Nützlich zum Herstellen von längeren Stücken von Leer-Streifen am Anfang und Ende eines Lochstreifens.
8. R (Taste oberhalb der Lochstreifenausgabe): Rückstellung des Lochstreifens um eine Lochreihe, zum Überholen von falsch geschriebenem Text mit "Buchstaben" (der Buchstabentaste entspricht das Fernschreibzeichen mit 5 Löchern).

Beim Benutzen der Tasten A... , 1... ,  und R bleibt der Wagen unverändert stehen.

Die Buchstaben-Tasten umfassen das Alphabet. Sie können nur betätigt werden, wenn vorher die Taste A... gegeben wurde.

Die 23 Tasten für Ziffern und Zeichen umfassen die

10 Ziffern 0 bis 9
die 11 Zeichen () + = ? / ' , . - :

die alle das entsprechende Zeichen schreiben.

Ferner gehört dazu die Taste  (= Klingel), die ein Glockenzeichen ertönen läßt. Der Wagen bleibt bei Drücken dieser Taste unverändert stehen.

Bei der Taste  (= Wer da? aus dem Fernschreibdienst) wird kein Typenhebel angeschlagen, aber die Kombination wird auf den Lochstreifen gelocht.

Der Locher kann durch Drücken der Taste  ein- und der Taste  (oberhalb der Lochstreifenausgabe) ausgeschaltet werden. Bei eingeschaltetem Locher wird für jede angeschlagene Taste das entsprechende Fernschreibzeichen gelocht (auch "Leer", "Bu" , "Zi" usw.)

Die Taste "L" oberhalb der Lochstreifenausgabe dient zum Einlegen des Lochstreifens.

B) Abtaster mit Schritt-Stop-Gerät

Zum Kopieren und Ändern von Lochstreifen wird der Abtaster mit dem Schritt-Stop-Gerät benutzt. Das Schritt-Stop-Gerät (rechts neben der Tastatur) hat einen Schalter zur Einstellung für laufendes Kopieren (schwarzer Schalter nach unten, roter Punkt verdeckt) und Stop (schwarzer Schalter nach oben, daß roter Punkt sichtbar) sowie eine weiße Drucktaste, deren Betätigung bei Stopstellung das Abtasten von jeweils einem Fernschreibzeichen bewirkt.

Der Lochstreifen wird so eingelegt, daß die Transportzähne in die Transportlöcher eingreifen. Der Lochstreifen selbst ist um den Papierspannhebel zu legen, damit der Abtaster automatisch stoppt, wenn der Lochstreifen klemmt; so wird ein Reißen verhindert.

Nach Anschalten des Abtasters (Hebel links unten nach oben stellen) wird - sofern das Schritt-Stop-Gerät auf Start steht (roter Punkt nicht sichtbar) - der Lochstreifen abgetastet und in Klarschrift von der Schreibmaschine niedergeschrieben und bei eingeschaltetem Locher gleichzeitig kopiert. Sind in dem Streifen Änderungen vorzunehmen, so läßt man den Streifen bis in die Nähe der betreffenden Stelle laufend kopieren und stoppt durch das Schritt-Stop-Gerät (roter Punkt sichtbar). Durch Betätigung der weißen Taste am Schritt-Stop-Gerät wird jeweils 1 Zeichen abgetastet und geschrieben. So kann man sich schrittweise bis zur falschen Stelle herantasten. Bei abgeschaltetem Locher läßt man dann die zu ändernden Zeichen schrittweise abtasten (sie werden im Protokoll mitgeschrieben) und fügt bei eingeschaltetem Locher den neuen richtigen Text ein. Es ist dabei zu beachten, daß die Tasten des Fernschreibers nicht dem zuletzt abgetasteten Bu bzw. Zi entsprechend gesperrt sind, wohl aber der Schlitten entsprechend gestellt ist. So kann es vorkommen, daß bei Betätigung der Tasten für Buchstaben, Ziffern und Zeichen geschrieben werden. (Es waren vor dem Abtasten zuletzt Buchstaben geschrieben worden,

also die Ziffern-Taste blockiert, vom Abtaster aber als letztes Ziffern abgetastet und entsprechend der Schlitten angehoben.) Es ist also bei Einfügungen stets vor Buchstaben die Bu-Taste, vor Ziffern die Zi-Taste zu geben. Es empfiehlt sich deshalb, ein ganzes Wort neu einzufügen, nicht nur ein Zeichen, da man dann automatisch die richtige Buchstaben- und Ziffernumschaltung vornimmt.

Der Abtaster wird ausgeschaltet, wenn man den Hebel nach unten stellt.

Bei der Herstellung von Programmen beachte man die Wirkung der Betriebszeichen der Fernschreibmaschine.

1. Innerhalb eines Operationsteiles eines Befehls:
 - a) leer, Wagentransport, Zeilentransport, Zwischenraum wirkungslos
 - b) Zifferntaste = Schlußzeichen für Operationsteil und Übergang zum Adressenteil
 - c) Buchstabentaste = Irrung (Löschung der Befehlszeichen, die von dem behandelten Wort schon eingelesen sind.)
2. Innerhalb eines Adressenteils:
 - a) leer, Wagentransport, Zifferntaste wirkungslos
 - b) Zwischenraum oder Zeilentransport = Schlußzeichen
 - c) Buchstabentaste = Übergang zu Bandbefehlen
3. Innerhalb von Bandbefehlen:
 - a) Buchstabentaste = Irrung (Löschung des eingelesenen Wortes)
 - b) E, A, T Übergang zu den entsprechenden Programmteilen
 - c) Stop bei allen anderen Zeichen
4. Innerhalb einer Zahl:
 - a) leer, Wagentransport, Zifferntaste wirkungslos
 - b) Zwischenraum, Zeilentransport = Schlußzeichen
 - c) Buchstabentaste = Irrung
5. bei Strich-Zahlen wie unter 4 a) und c);
als Schlußzeichen gilt hier das Apostroph.

Bad Hersfeld, April 1960
Su/Sto

Das adressenlose Programmieren

Will man in einem Programm mit festen oder relativen Adressen einige Befehle abändern oder einschieben, so muß man einen großen Teil der vorbereitenden Adressen und Sprungbefehle ausbessern. Diese Arbeit kann vor allen Dingen bei großen Programmen sehr mühsam und zeitraubend sein. Um diesen Aufwand zu vermeiden, hat man die Möglichkeit vorgesehen, Adressen durch Namen zu ersetzen, die die Maschine von sich aus in einem Adressbuch bzw. Vormerkbuch mit der zugehörigen Adresse laufend ab speichert und im Programm durch die Adressen ersetzt. Dadurch erreicht man, daß man im Befehlstext nur die Namen der angesprochenen Adressen zu setzen braucht. Ein Programm (Adressierprogramm) ersetzt dann automatisch alle Namen, die beim Einspeichern benutzt wurden, über das Adressbuch durch die dazugehörigen Adressen.

Im Adressbuch A werden alle Namen mit schon bekannten Adressen fortlaufend abgespeichert, während im Vormerkbuch B alle die Namen mit dem Stand des Zählregisters 13 notiert werden, die keine Adresse haben und noch nicht unter A aufgeführt sind. Um vorgemerkte Texte zu kontrollieren, wird jedesmal, wenn ein Name allein auftritt, nach der Notierung des Namens ins Adressbuch das Vormerkbuch kontrolliert. Die laut Vormerkbuch zu ergänzenden Texte werden geändert, und die Vormerkung wird dann gelöscht.

Für das Adressierprogramm sind folgende Konventionen getroffen:

- I. Ein Name besteht aus maximal 5 Fernschreibzeichen, die in runde Klammern zu setzen sind. Der Name hat also folgendes Aussehen:

(Bu $f_1, f_2 \dots f_5$ Zi)

Man kann in einem Namen demnach entweder 5 Buchstaben angeben, oder 3 Buchstaben und 1 Ziffer, oder 2 Buchstaben und 2 Ziffern usw.

Die Konvention über die Namen sieht vor, daß das Zeichen „(" von Bu gefolgt wird und vor dem Schlußzeichen „)" ein Zi steht. Diese Zeichen, Bu am Anfang und Zi am Ende, zählen nicht mit zu den 5 Zeichen.

z.B. B6(KZ1), es muß gelocht werden:

BuBZi6(Bu K Z Zi 1 Zi); der Name besteht aus den
4 Zeichen KZzi1.

II. Jeder Name steht symbolisch für eine und nur eine Adresse, und zwar für das nächste einzuspeichernde Wort, Befehl, Zahl, Tabellenanfang usw. Die Adresse dieses Wortes "m" steht im Kernspeicher 13. Das Adressierprogramm sucht daher die erste freie Zelle im Adressbuch A und notiert den Namen und die Adresse "m". Ist der Name nicht in B vorgemerkt, so geht das normale Leseprogramm weiter.

III. Kommt ein Name innerhalb eines Befehls vor - B6(NAM) - so wird im Adressbuch A die zugehörige Adresse gesucht. Wird sie gefunden, so wird diese Adresse zu dem bisher fertigen Befehlstext hinzuaddiert. Konvention dabei ist, daß auch Zahlen unter 31 als Trommeladressen gewertet werden, z. B. wird {NAM} (Adresse von NAM) zu B0+6 addiert.

Wird der Name nicht im Adressbuch A gefunden, so ist auch die Adresse nicht bekannt, der Name wird dann im Vormerkbuch B notiert und diejenige Zellennummer vorgemerkt, in der später die Adresse nachgetragen werden muß. Die Zellennummer, die gerade in 13 steht, wird mit dem angegebenen Namen ins Vormerkbuch B gespeichert.

IV. Die Namen können auch innerhalb von Bandbefehlen verwendet werden, z. B. T(BETA) T oder E(ALFA)E..

Zum besseren Verständnis wird im folgenden auf das Adressierprogramm (Nr. 1A der Programmbibliothek) eingegangen.

I. Man muß, bevor man das Adressierprogramm auf die Trommel speichert, einen Vorstreifen eingeben mit

TmT

n₁' (n₁ Plätze für das Adressbuch A)

n₂' (n₂ Plätze für das Vormerkbuch B)

Z0+1E

Damit ist nun auch die Stelle angegeben, an die das Adressierprogramm gespeichert wird, und zwar von $m+2$ ab. Das Einlesen endet mit YZH ($m+97$) im Befehlsregister. Man muß nun noch einmal starten, um die Zellen zu löschen, die für die Adressbücher A und B vorgesehen sind.

- II. Die Relativadressen des Adressierprogramms laufen von 0 ab. Beim Zeichen "(" wird der Sprung vom Leseprogramm auf das Adressierprogramm ausgeführt, um Namen einzulesen und die Adresse, die dem Namen zugeordnet wird, einzusetzen bzw. die Stelle vorzumerken, in der die Adresse noch eingesetzt werden muß. Daraufhin erfolgt der Rücksprung ins Leseprogramm.

Will man von einem bestimmten Namen an, z.B. "(NAM)", das Adressbuch löschen, so gibt man auf einem Lochstreifen V(NAM) in die Maschine. War der angegebene Name noch nicht notiert, so stoppt die Maschine mit YEHZ($m+97$) im Befehlsregister und löscht nach Start das gesamte Adressbuch und Vormerkbuch. Zum Löschen der Adressbücher A und B benutzt man am besten V(Bu Zi) (Kein Name) (Stop der Maschine mit YEHZ($m+97$) Start). Danach erfolgt der Rücksprung ins Leseprogramm.

Der spezielle Bandbefehl ")", ohne daß vorher "(" gekommen ist, führt einen Sprung auf "Adressbuchdruck" aus. In der ersten Zeile schreibt die Maschine die Adresse der letzten Notierung im Adressbuch aus, darunter jeweils nach der Reihe die eingegebenen Namen mit der zugeordneten Adresse.

Durch Zeilensprung getrennt, wird in der gleichen Anordnung das Vormerkbuch ausgeschrieben, und zwar nur die Namen, die im Programm noch nicht eingesetzt wurden. Nach dem Druck der Adressbücher erfolgt der Sprung ins Leseprogramm.

Im folgenden soll das Verfahren an einem Beispiel erläutert werden. Ein Polynom 4. Grades

$$f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

soll als Unterprogramm mit der Schrittweite d tabelliert werden.

$$f(x) = (((a_4x + a_3)x + a_2)x + a_1)x + a_0$$

Zu Beginn des Programmes werden x_0 (Anfangswert), x_n (Endwert), d (Schrittweite) und die Koeffizienten abgespeichert. Aufruf des Unterprogrammes mit F(UP1).

T2000T

(X)	x_0
	x_n
	d
(K)	a_4
	a_3
	a_2
	a_1
	a_0
(N)	4'
(UP1)	B5
	T(FENDE)
	B(X)
	T11
(AZ)	B11
	D
	NS(N)
	T2
	CB(K)
	T13
	GKB13+0
	O
	T6
(IZ)	B11
	X
	CGKB13+1
	O
	+
	CGU2+1
	O
	QE(IZ)
	D
	B11
	T6
	B2(X)

```
+  
U11  
B1(X)  
-  
QQ(AZ)  
(ENDE) 0
```

Die Verarbeitung durch das Adressierprogramm geschieht folgendermaßen:

Zuerst wird der Name (X) eingelesen und mit der Adresse identifiziert, die die folgende Größe "x₀" erhält. Dann werden die Größen x₀, x_n, d eingelesen. Der Name (K) gibt die Anfangsadresse der Zellen an, in die die Koeffizienten abgespeichert werden.

(N) ist der Name der Stelle, in der der Grad des Polynoms gespeichert wird.

(UP1) ist die Bezeichnung des Anfangs des Unterprogramms.

Alle Namen (X), (K), (N), (UP1) werden mit den zugeordneten Adressen ins Adressbuch A eingeschrieben.

Die Rückkehradresse soll an das Ende des Programms abgespeichert werden. Der Befehl T(ENDE) wird im Adressierprogramm untersucht, (ENDE) kann im Adressbuch noch nicht gefunden werden. (ENDE) wird daher zusammen mit der Adresse des Befehlstextes in das Vormerkbuch B gesetzt.

Für (X) kann die richtige Adresse eingesetzt werden, da (X) ja schon im Adressbuch steht. Der Name (AZ) verweist auf den Beginn des äußeren Zyklus.

(AZ) wird im Adressbuch A aufgenommen.

Die Befehle NS(N), CB(K) (bringe die Adresse, in der der erste Koeffizient steht, in den Akku) können alle mit den richtigen Adressen versehen werden, da die Namen (N) und (K) schon im Adressbuch A stehen.

Der innere Zyklus ist mit (IZ) bezeichnet; der Befehl QQE(IZ) bringt den Rücksprung auf den Anfang des inneren Zyklus, also auf die bereits bekannte Adresse (IZ). In B2(X) und B1(X) kann

die schon notierte Adresse (X) eingesetzt werden. QQ(AZ) bringt den Rücksprung auf den Anfang des äußeren Zyklus. Der Name (ENDE) gibt die letzte Stelle im Programm an, (ENDE) wird im Adressbuch A notiert und das Vormerkbuch wird nach Befehlen durchsucht, in denen (ENDE) als Adresse auftritt.

Das so verschlüsselte Programm ist vollkommen unabhängig davon, an welcher Stelle der Trommel es eingespeichert wird, ist auch nicht an eine bestimmte Anzahl von Koeffizienten gebunden, wenn nur die entsprechenden Koeffizienten unter (K) und der Grad des Polynoms unter (N) als Strichzahl gespeichert werden.

Da beim Einlesen des Programms die richtigen Adressen eingesetzt und die Befehle mit den richtigen Adressen gespeichert werden, hat die Art der Programmierung mit festen Adressen oder symbolischen Adressen keinerlei Einfluß auf die Rechenzeit. Lediglich verlängert sich die Einlesezeit etwas, jedoch ist diese Zeit wesentlich kürzer als die von einem Programmierer benötigte Zeit, die Numerierung der Speicherplätze und Einsetzen der richtigen Adressen durchzuführen.

Bad Hersfeld, 14.10.1960