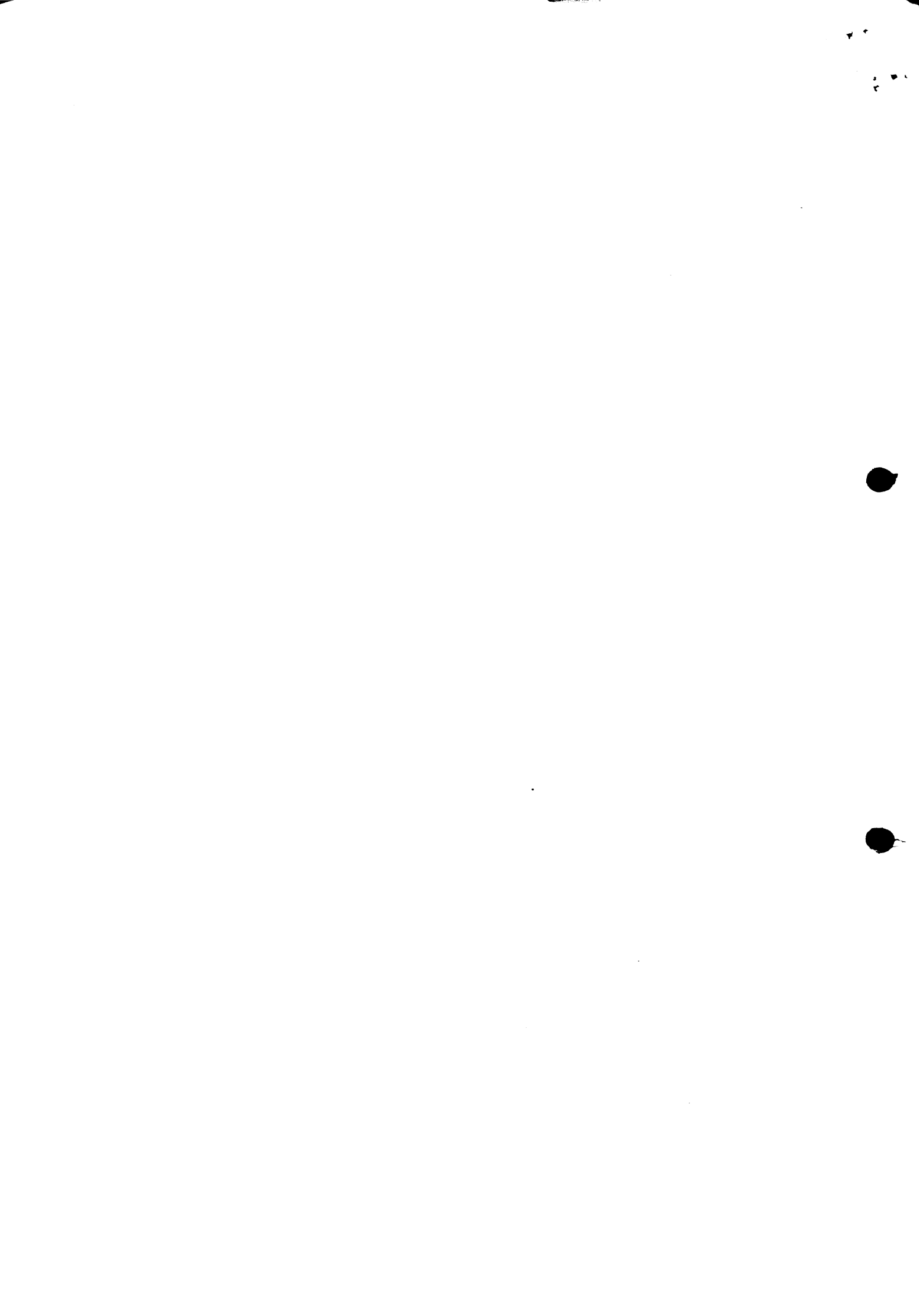


Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

#1- 24.1

Inhaltsübersicht

		Seite
Teil I	ALLGEMEINES	2
Teil II	BESCHREIBUNG DES HAUPTPROGRAMMS 24.1	4
	A. Die Register	4
	B. Format der Gleitkommazahlen	4
	C. Pseudo-Befehle	6
	D. Dateneingabe	11
	E. Datenausgabe	15
	F. Unterprogramme zur Berechnung spezieller Funktionen	16
Teil III	BESCHREIBUNG DES HILFSPROGRAMMS 24.1A	18
	A. Funktion	18
	B. Eingabe	18
	C. Ausgabe	19
	D. Vorgehensweise	19
Teil IV	BENUTZUNG DES ERFP-SYSTEMS	23
	Übersicht über Befehle und Operations- zeiten	26
Teil V	BEISPIELE	27



Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

Teil I ALLGEMEINES

Die Programmierung im Gleitkomma weist verschiedene Vorzüge gegenüber der Festkommaprogrammierung auf. Sie ist schneller, erfordert keine genaue Kenntnis der Größenordnung der Variablen und zieht keine zu große Verfälschung der kleineren Werte nach sich, wenn der Zahlenbereich groß ist.

Das Gleitkommasystem hat die Aufgabe, die Festkommabefehle des LGP-30 derart zu interpretieren, daß mit ihm so gerechnet werden kann, als ob er verdrahtete Gleitkommabefehle hätte. Es entfällt somit für den Programmierer bei Benutzung des Systems das sonst notwendige Skalieren. Die Interpretation wird bewirkt durch:

1. Die Bereitstellung eines Multiplizierregisters, eines Adressenregisters und eines Gleitkomma-Akkumulators. Der Gleitkomma-Akkumulator wird im folgenden als Pseudo-Akkumulator bezeichnet.
2. Die Bereitstellung von mehr Befehlstypen, die im folgenden Pseudo-Befehle genannt werden, einschließlich der kumulativen Multiplikation, des Vorzeichenwechsels und der Funktionserzeugung.
3. Eine Erweiterung der Funktion bestimmter Befehle wie des Eingabe- und des Druckbefehls.
4. Die Möglichkeit der Formatwahl für die zu speichernden Zahlen.

Es sind 44 Befehle im System vorgesehen. Alle diese Befehle - mit Ausnahme von Ein- und Ausgabe, Sinus, Cosinus, Arctang, Logarithmus und Exponentialfunktion - sind in dem Teil des Systems enthalten, der als interpretierendes Gleitkommaprogramm mit der Nr. 24.1 bezeichnet ist. Dieses Programm belegt nur 11 von den 64 vorhandenen Spuren und läßt 3392 Worte für das Problemprogramm und das Speichern von Daten frei. Für die Eingabe werden 7, für die Ausgabe 4 und für die speziellen Funktionen 6 Spuren benötigt. Somit läßt das Gesamtsystem 2304 Speicherplätze für das Problemprogramm und die Daten frei. Im allgemeinen benötigt ein Gleitkommaprogramm etwa 15 bis 25 mal soviel Zeit wie das

entsprechende Festkommaprogramm. Eine diesbezügliche Tabelle für die Operationszeiten der einzelnen Befehle steht am Ende von Teil IV. Es bestehen zwei Möglichkeiten für die Datenspeicherung. Diese zwei Möglichkeiten werden als Format 1 und Format 2 bezeichnet. In Teil II B wird eine vollständige Beschreibung der Zahlendarstellungen in den beiden verschiedenen Formaten gegeben. Hier sollen nur einige Vorbe-merkungen gemacht werden.

Format 1

Für 3 Zahlen werden 4 Speicherzellen benötigt. Drei aufeinanderfolgende Zellen enthalten die Mantissen von 3 aufeinanderfolgenden Zahlen. Die vierte darauf folgende Zelle enthält die 3 Exponenten. Das Programmieren der absoluten Adressen der in diesem Format gespeicherten Zahlen ist sehr umständlich. Es werden daher sowohl in der Programm-Niederschrift wie auch in den Dateneingabe-Streifen nur relative Adressen benutzt, die dann durch ein Hilfsprogramm in absolute Adressen umgerechnet werden. Für die Befehlsfolge in der Programm-Niederschrift wird dies von dem Hilfsprogramm 24.1A besorgt, während die Umrechnung der Adressen der Eingabedaten vom Eingabeprogramm 11.5 ausgeführt wird.

Format 2

Für die Darstellung einer Zahl werden zwei aufeinanderfolgende Zellen verwendet. Die erste Zelle enthält die Mantisse, die zweite den Exponenten. Die Adressenumrechnung entfällt, da bei diesem Format die absoluten Adressen bequem zu programmieren sind.

Lochstreifen

Das Programm 24.1 ist auf zwei Lochstreifen gelocht. Streifen A enthält das oben erwähnte Hilfsprogramm 24.1A (in Teil III näher beschrieben), Streifen B enthält das Hauptprogramm 24.1 sowie alle wichtigen Unterprogramme:

Dateneingabe 11.5

Sin-Cos. 14.3

Logarithmus 18.2

Datenausgabe 12.5

Arctang 16.3

Exponentialfunktion 17.2

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

Teil II BESCHREIBUNG DES HAUPTPROGRAMMS 24.1

A. Die Register

Das Programmierungssystem 24.1 enthält 3 besondere Register:

1. Der Pseudo-Akkumulator belegt zwei Speicherplätze; der erste ist für die Mantisse, der zweite für den Exponenten bestimmt. Der Pseudo-Akkumulator hält ebenso wie der Festkomma-Akkumulator die Zwischenergebnisse. Die Mantisse wird bei $q = 1$ gespeichert, der Exponent bei $q = 29$.
2. Vom Multiplikationsregister werden ebenfalls zwei Speicherzellen, eine für die Mantisse und eine für den Exponenten, belegt. Das Multiplikationsregister enthält den Multiplikator für einfache Multiplikationen und kumulative Multiplikationen.
3. Der Adressen-Akkumulator belegt eine Speicherzelle für eine Adresse oder ein Kennzeichen. Die Gestalt der Adressen ist dieselbe wie bei den Festkomma-Operationen.

Die Registerinhalte werden nur durch Einschreiben von neuen Ergebnissen geändert. So bleibt z. B. das M-Register nach einer Multiplikation ungeändert. Ebenso wird ein anderer Speicherplatz nur dann geändert, wenn der betreffende Befehl dies ausdrücklich vorsieht.

B. Format der Gleitkomma-Zahlen

1. Definitionen

Jede Zahl A läßt sich darstellen in der Form $A = \pm a \cdot 2^{\pm b}$.
Dabei ist a die Mantisse und b der Exponent von A zur Basis 2.
Legt man fest, daß stets $0,5 \leq a < 1,0$ ist, so ist die Darstellung eindeutig, sofern A ungleich Null ist. Für $A = 0$ wird festgelegt $a = 0$ und $b = 0$.

Beispiele:

$$\begin{aligned} 2,000 &= 0,5 \cdot 2^2 \\ -5,000 &= -0,625 \cdot 2^3 \\ 1,500 &= 0,750 \cdot 2^1 \\ 0,300 &= 0,600 \cdot 2^{-1} \end{aligned}$$

Mantisse und Exponent werden gesondert gespeichert; die Art der Speicherung bestimmt sich aus der Wahl des Formates 1 oder 2.

2. Format 1

In diesem Format werden für je 3 Zahlen 4 aufeinanderfolgende Speicherzellen benutzt. Die ersten 3 Zellen enthalten je eine der 3 Mantissen (Vorzeichen + 30 Bits) und die vierte enthält die 3 Exponenten. Die Mantissen werden bei $q = 0$ und die Exponenten bei $q = 11, 19$ und 27 gespeichert, so daß für jeden Exponenten 8 Bits (Vorzeichen + 7 Bits) zur Verfügung stehen. Für den Exponenten b ergibt sich somit die Einschränkung: $-127 \leq b \leq 127$. $b > 127$ führt zu einem programmierten Stop; bei $b < -127$ wird die zugeordnete Zahl durch Null ersetzt.

Beispiel für die Zahlendarstellung im Format 1:

Die dezimalen Zahlen

$$\begin{aligned} 0,5 &= 0,5 \cdot 2^0 \\ -1,25 &= -0,625 \cdot 2^1 \\ 0,125 &= 0,5 \cdot 2^{-2} \end{aligned}$$

ergeben, wenn sie im Format 1 in den Zellen 4500 ff gespeichert werden, folgende Darstellung (in hexadezimal):

Zelle	Inhalt hex.	Inhalt dezimal
4500	40000000	0,5
4501	g0000000	-0,625
4502	40000000	0,5
4503	00001wq0	0 1 -2

} $q = 0$

$q = 11 \mid 19 \mid 27$

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

3. Format 2

Für jede Zahl werden zwei aufeinanderfolgende Speicherzellen benötigt. Die Mantisse wird in der 1. Zelle bei $q = 0$, der Exponent in der 2. Zelle bei $q = 29$ gespeichert. Obgleich somit der Exponent 29 Bits haben kann, ist er für die Ein- und Ausgabe doch beschränkt auf $-127 \leq b \leq 127$. Beim Speichern des Inhalts des Pseudo-Akkumulators hält der Rechner allerdings im Gegensatz zu Format 1 nicht an, auch wenn der Exponent die Schranken ± 127 überschreitet.

Beispiel für die Zahlendarstellung im Format 2:

Dieselben Zahlen wie im vorigen Beispiel sollen im Format 2 in den Zellen 4500 ff gespeichert werden. Es ergibt sich:

Zelle	Inhalt hex.	Inhalt dezimal	
4500	40000000	0,5	$q = 0$
4501	00000000	0	$q = 29$
4502	g0000000	-0,625	$q = 0$
4503	00000004	1	$q = 29$
4504	40000000	0,5	$q = 0$
4505	wwwwwww8	-2	$q = 29$

C. Pseudo-Befehle

Es stehen 44 Pseudo-Befehle zur Verfügung. Im folgenden steht der Einfachheit halber stets "Akkumulator" anstelle von "Pseudo-Akkumulator" und "Befehl" anstelle von "Pseudo-Befehl".

1. Arithmetische Befehle für Format 1

Die Speicherzelle xxxx sei im folgenden die Adresse der Mantisse einer Gleitkommazahl im Format 1.

1) Bxxxx, Bringen

Der Inhalt des Akkumulators wird durch den Inhalt der Speicherzelle xxxx ersetzt.

2. Axxxx, Addieren
Der Inhalt der Zelle xxxx wird zu dem Akkumulatorinhalt addiert und das Ergebnis im Akkumulator gespeichert.
3. Sxxxx, Subtrahieren
Der Inhalt der Zelle xxxx wird vom Akkumulatorinhalt subtrahiert; das Ergebnis wird im Akkumulator gespeichert.
4. Dxxxx, Dividieren
Der Akkumulatorinhalt wird durch den Inhalt der Zelle xxxx dividiert; das Ergebnis wird im Akkumulator gespeichert.
5. Pxxxx, Placieren
Der Inhalt des Multiplikationsregisters (M-Register) wird durch den Inhalt der Zelle xxxx ersetzt.
6. Mxxxx, Multiplizieren
Der Inhalt des M-Registers wird mit dem Inhalt der Speicherzelle xxxx multipliziert; das Ergebnis wird im Akkumulator gespeichert.
7. Nxxxx, kumulatives Multiplizieren
Der mit dem Inhalt des M-Registers multiplizierte Inhalt der Speicherzelle xxxx wird zu dem Inhalt des Akkumulators addiert; das Ergebnis wird im Akkumulator gespeichert.
8. XD000y, rechts Schiften
Der durch 2^y dividierte Akkumulatorinhalt tritt an die Stelle des Akkumulatorinhaltes. Es gilt $0 \leq y \leq 9$.
9. XM000y, links Schiften
Der mit 2^y multiplizierte Akkumulatorinhalt wird im Akkumulator gespeichert, wobei wieder $0 \leq y \leq 9$ gilt.
10. Hxxxx, Halten
Der Akkumulatorinhalt wird in der Zelle xxxx gespeichert.
11. Cxxxx, Löschen
Der Akkumulatorinhalt wird in der Zelle xxxx gespeichert; gleichzeitig wird der Akkumulator gelöscht.

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

2. Arithmetische Befehle für Format 2

Der Speicherplatz xxxx sei die Adresse der Mantisse einer Zahl im Format 2. Die folgenden 9 Befehle, die sich auf Daten im Format 2 beziehen, unterscheiden sich nur durch das Vorzeichen von den entsprechenden Befehlen für Format 1. Ihre Interpretation ist mit jenen völlig identisch.

- | | | |
|-----|----------|--------------------------|
| 12. | 800Bxxxx | Bringen |
| 13. | 800Axxxx | Addieren |
| 14. | 800Sxxxx | Subtrahieren |
| 15. | 800Dxxxx | Dividieren |
| 16. | 800Pxxxx | Placieren |
| 17. | 800Mxxxx | Multiplizieren |
| 18. | 800Nxxxx | kumulativ Multiplizieren |
| 19. | 800Hxxxx | Halten |
| 20. | 800Cxxxx | Löschen |

3. Sprung- und Testbefehle

21. Uxxxx, Unbedingter Sprung

Der nächste Befehl, der ausgeführt werden soll, befindet sich in der Zelle xxxx. Dieser Befehl kann nur innerhalb des Gleitkommasystems angewandt werden. Soll aus dem Gleitkommasystem herausgesprungen werden, so muß Befehl 36 benutzt werden.

22. Txxxx, Test

Falls der Akkumulatorinhalt negativ ist, wird der nächste auszuführende Befehl aus der Zelle xxxx genommen, andernfalls wird zu der folgenden Zelle übergegangen.

23. 800Txxxx, Test Sprungtaste

Falls der Akkumulatorinhalt negativ oder die Sprungtaste gedrückt ist, wird der nächste auszuführende Befehl aus der Zelle xxxx genommen, andernfalls wird in der normalen Reihenfolge fortgefahren.

4. Befehle zur Adressenmodifikation

24. Exxxx, Adressenersatz

Der Inhalt des Adressen-Akkumulators wird durch den Adressenteil der Zelle xxxx ersetzt.

25. Ixxxx, Adressenänderung

Der Inhalt des Adressen-Akkumulators wird um xxxx vermehrt. Dieser Befehl kann zur Verminderung des Inhalts des Adressen-Akkumulators benutzt werden, indem im Adressenteil des Befehls Ixxxx das entsprechende Komplement verwendet wird. Dieser Befehl muß benutzt werden, wenn die Adresse sich auf im Format 1 gespeicherte Daten bezieht. Es werden so automatisch alle die absoluten Adressen übersprungen, die Exponenten enthalten.

26. 800Ixxxx, Adressenänderung

Der Befehl hat die gleiche Funktion wie Befehl 25; er bezieht sich aber auf die im Format 2 gespeicherten Zahlen. In diesem Fall werden keine Adressen übersprungen.

27. Yxxxx, Adresse speichern

Der Adressenteil des Adressen-Akkumulators ersetzt den Adressenteil der Zelle xxxx.

28. Zxxxx, Nullprüfung

Die Adresse xxxx wird vom Inhalt des Adressen-Akkumulators abgezogen. Falls das Ergebnis ungleich Null ist, wird der nächste Befehl interpretiert. Ist das Ergebnis Null, so wird der übernächste Befehl interpretiert.

29. 800Zxxxx, Nullprüfung

Dieser Befehl hat die gleiche Funktion wie der Befehl 28. Zur Berechnung der absoluten Adressen wird aber der "/" Modifikator benutzt (s. Ablaufprogramm II für 24.1A).

30. Rxxxx, Adressenrückkehr

Die Zelle, in der der Befehl Rxxxx steht, wird um zwei vermehrt und im Adressenteil der Zelle xxxx gespeichert.

5. Hilfsbefehle

Die Adressen der folgenden Befehle müssen Null sein.

31. U0000, Registervertauschung

Die Inhalte des M-Registers und des Akkumulators werden vertauscht.

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

32. B0000, Betragsbildung
Der Akkumulatorinhalt wird positiv gemacht, falls er nicht schon positiv ist.
33. T0000, negativer Betrag
Der Akkumulatorinhalt wird negativ gemacht, falls er nicht schon negativ ist.
34. Y0000, Vorzeichenumkehr
Der Akkumulatorinhalt erhält das entgegengesetzte Vorzeichen.
35. Z0000, Halt
Die Rechnung wird unterbrochen, wenn nicht "Kein Halt 16" gedrückt ist. Nach Start Fortsetzung der Rechnung beim nächsten Befehl.
36. E0000, Ausgang
Das Gleitkommasystem wird verlassen. Der auf E0000 folgende Befehl wird als Elementarbefehl interpretiert.
6. Ein- und Ausgabebefehle
- Die Adressen der folgenden Befehle müssen ebenfalls Null sein.
37. I0000, Eingabe
Das Dateneingabe-Programm 11.5 wird aufgerufen. Es liest dezimale Zahlen vom Band, rechnet sie in duale Gleitkommazahlen um und speichert sie. Das geschieht so lange, bis das entsprechende, aus dem Dateneingabe-Programm herausführende Codewort vom Band gelesen wird. Danach wird der nächste Befehl interpretiert. (Einzelheiten s. Abschn. D dieses Kapitels).
38. P0000, Ausgabe
Die im Akkumulator stehende Zahl wird mit Hilfe des Unterprogramms 12.5 ausgedruckt. Der Akkumulatorinhalt bleibt dabei erhalten. (Einzelheiten s. Abschn. E dieses Kapitels).

7. Befehle für spezielle Funktionen

Auch diese Befehle haben die Adresse Null. Sie rufen Unterpläne auf, die nicht im Grundsystem enthalten sind. Eine Ausnahme davon bildet der Befehl 39, da das Wurzelprogramm im Grundsystem direkt enthalten ist.

39. R0000, Wurzel

Der Befehl veranlaßt, daß aus dem Akkumulatorinhalt die Quadratwurzel gezogen wird. Nach der Ausführung des Befehls steht das Ergebnis im Akkumulator.

40. S0000, Sinus

Es wird der Sinus des Akkumulatorinhalts berechnet und im Akkumulator gespeichert. Das Argument muß im Bogenmaß gegeben sein (Unterplan 14.3).

41. C0000, Cosinus

Der Cosinus des Akkumulatorinhalts wird berechnet und im Akkumulator gespeichert. Das Argument muß im Bogenmaß gegeben sein (Unterplan 14.3).

42. A0000, Arcustangens

Der Arcustangens des Akkumulatorinhalts wird berechnet und im Akkumulator gespeichert. Das Ergebnis erscheint in Bogenmaß (Unterplan 16.3).

43. N0000, Logarithmus

Der natürliche Logarithmus des Akkumulatorinhalts wird berechnet und im Akkumulator gespeichert (Unterplan 18.2).

44. H0000, Exponentialfunktion

Der Akkumulatorinhalt x wird durch e^x ersetzt (Unterplan 17.2).

D. Dateneingabe (Unterprogramm 11.5)

Die Dateneingabe wird ausgeführt, indem ein dezimal gelochter Streifen auf den Befehl I0000 hin eingelesen wird. Das Band enthält folgende Informationen:

1. Ein Kennwort
2. Mit Vorzeichen versehene Dezimalzahlen
3. Ein Endwort

1. Das Kennwort

Es bestimmt das Format und sieht folgendermaßen aus:

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

a) Für Format 1

$$T_1 T_2 S_1 S_2 i_1 i_2 i_3 i_4 ,$$

wobei $T_1 T_2 S_1 S_2$ die Anfangsadresse D_0 für einen im Format 1 gespeicherten Datenblock darstellt. D_0 ist identisch mit dem "-" Modifikator. Die Zellennummer $S_1 S_2$ muß stets ein Vielfaches von 4 sein (00, 04, 08 ... 56, 60). Die zweite Hälfte gibt an, um die wievielte Zahl innerhalb der gespeicherten Daten es sich handelt. Die Berechnung der Füllbeginn-Adresse F_0 für die zu speichernden Daten erfolgt dann nach der Beziehung

$$F_0 = D_0 + 4/3 (i_1 i_2 i_3 i_4).$$

Beispiel:

Das Kennwort heiße 20080093. Darin ist $D_0 = 2008$, nämlich Spur 20, Zelle 08. $i_1 i_2 i_3 i_4 = 0093$ bezeichnet die dem Kennwort folgende Zahl als die 93. des gesamten Blocks. Die Dateneingaberoutine berechnet F_0 aus:

$$4/3 \cdot 0093 = 0124 \text{ Zellen} = 1 \text{ Spur und } 60 \text{ Zellen.}$$

$$\text{Zu } D_0 \text{ addiert ergibt } F_0 = 2008 + 0160 = 2168 = 2204.$$

Falls kein D_0 auftritt, oder wenn es Null ist, wird das vorhergehende D_0 gebraucht.

b) Für Format 2 hat das Kennwort die Form

$$8000 T_1 T_2 S_1 S_2 ,$$

wobei $T_1 T_2$ die Spur und $S_1 S_2$ die Zelle angibt, in der die erste Zahl nach dem Kennwort gespeichert wird.

2. Mit Vorzeichen versehene Dezimalzahlen

Auf dem Band sind nach dem Kennwort eine oder mehrere Zahlen abgelocht, die vom Programm 11.5 ins Duale umgewandelt, und in dem durch das Kennwort bestimmten Format in die ebenfalls durch das Kennwort angegebenen Zellen gespeichert werden.

Jede Zahl ist auf dem Lochstreifen durch zwei Worte, auf die jeweils ein Stopcode folgt, wiedergegeben. Diese beiden Worte haben folgendes Aussehen:

Erstes Wort: $\pm d_1 d_2 d_3 d_4 d_5 d_6 d_7'$

Zweites Wort: $d_8 d_9 \pm p_1 p_2'$,

wobei die d_1 bis d_9 die einzelnen Dezimalziffern darstellen und die $p_1 p_2$ die zwei Ziffern des negativen dezimalen Zehner-Exponenten repräsentieren, wobei gilt $-21 \leq p_1 p_2 \leq +38$.

Beispiele:

$0,0000000521374209 = 521374209 \cdot 10^{-16}$ wird dargestellt als
5213742'09+16'

$5522829640000000 = 552282964 \cdot 10^{+7}$ wird dargestellt als
5522829'64-07'

3. Die Endworte

Der Lochstreifen muß ein Endwort enthalten, welches auf die letzte einzulesende Zahl folgt. Es gibt 2 verschiedene Endworte:

- a) "Group" wird gebraucht, falls einer eingegebenen Datengruppe eine weitere Datengruppe mit einem neuen Kennwort folgen soll.
- b) "Exit" wird gebraucht, wenn einer eingegebenen Datengruppe keine weitere folgt, wenn das Einlesen der Daten mit dem betreffenden I-Befehl also beendet ist. Die Buchstaben beider Schlußworte werden in dieser Form auch auf dem Band abgelocht.

Wenn "group" gelocht ist, liest das Programm als nächstes ein weiteres Kennwort ein; falls "exit" gelocht ist, führt das Programm einen Wagenrücklauf aus und interpretiert dann den auf den I0000-Befehl folgenden Befehl.

4. Beispiel für eine Dateneingabe

Die Niederschrift für die Eingabe von Daten, nach der dann der Lochstreifen abgelocht wird, sieht z. B. folgendermaßen aus:

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

Kennwort	stop	Dezimalzahl																
		±	d ₁	d ₂	d ₃	d ₄	d ₅	d ₆	d ₇	'	d ₈	d ₉	±	p ₁	p ₂	'		
8 0 0 0 5 5 0 0	'	-	3	3	0	0	0	0	0	'	0	0	+	0	4	'		
			3	1	4	1	5	9	2	'	6	5	+	0	8	'		
		-	0	0	0	0	5	9	2	'	3	4	+	1	1	'		
		g r o u p										'						
2 0 1 2 0 0 6 6	'									'						'		
									3	0	'	5	0	+	0	2	'	
									7	5	0	'	0	0	-	0	1	'
		-	0	1	0	0	0	0	0	0	'	0	0	-	0	6	'	
		e x i t										'						

Hier keine
Information
und keine
Stopcodes!

Das erste Kennwort sagt aus, daß die Zahlen -33000, 3,14159265 und -0,00000059234 im Format 2 in die Zellen ab 5500 gespeichert werden sollen:

Absolute Adresse	Inhalt
5500	Mantisse } von -33000,0
5501	
5502	Mantisse } von 3,14159265
5503	
5504	Mantisse } von -0,00000059234
5505	

Alle Exponenten werden bei $q = 29$ gespeichert. Das Wort "group" beendet das Lesen dieser ersten Gruppe und bewirkt, daß das Programm 11.5 das nächste gelesene Wort als Kennwort entschlüsselt. Dieses besagt, daß die folgenden Zahlen im Format 1 ab Zelle F_0 gespeichert werden; $D_0 = 2012$, $i = 66$.

$$F_0 = D_0 + 4/3 i = D_0 + 88 \text{ Worte} = D_0 + 1 \text{ Spur, } 24 \text{ Zellen}$$

$$F_0 = 2136$$

Die zweite, aus vier Zahlen bestehende Gruppe, wird folgendermaßen gespeichert:

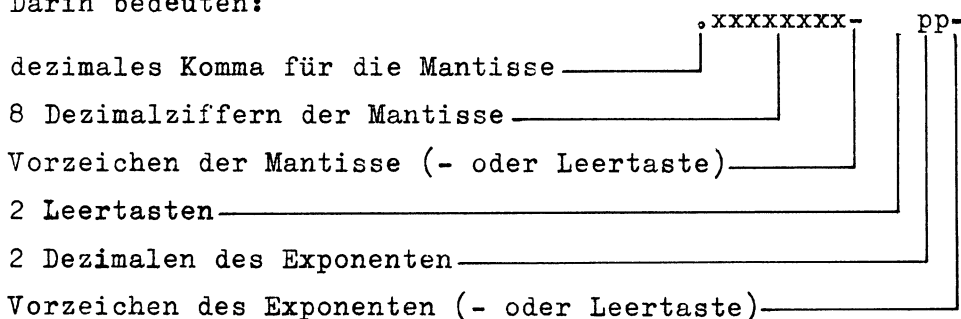
Absolute Adresse	Inhalt
2136	Mantisse von 0,0
2137	Mantisse von 30,5
2138	Mantisse von 750000
2139	Exponenten der 3 vorstehenden Zahlen
2140	$-10 \cdot 10^{12}$
2141	nichts
2142	nichts
2143	Exponent von $-10 \cdot 10^{12}$

Die zweite Datengruppe wird durch das Wort "exit" beendet.

E. Datenausgabe (Unterprogramm 12.5)

Ein gedrucktes Ausgabewort besteht aus einem dezimalen Punkt, dem acht dezimale Ziffern der Mantisse und ihr Vorzeichen folgen. Das Vorzeichen ist entweder ein "Minus" oder eine Leertaste. Auf das Vorzeichen der Mantisse folgen zwei Leertasten, danach werden die zwei dezimalen Ziffern des Zehner-Exponenten gedruckt, gefolgt von einem Minus-Zeichen, falls er negativ ist. Zum Schluß erfolgt noch ein Tabulatorsprung; danach wird der auf P0000 folgende Befehl im Problemprogramm ausgeführt. Somit sieht die Niederschrift einer Gleitkommazahl folgendermaßen aus:

Darin bedeuten:



Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

In der mathematischen Schreibweise bedeutet dieses:

\pm , xxxx xxxx . $10^{\pm pp}$

Beispiele:

Gedruckte Ausgabe	Numerischer Wert
.50600000 - 04	-5060,0
.93750000 02-	0,009375
.10000000 - 06-	-0,0000001
.10000000 30	$0,1 \cdot 10^{30}$

Die gedruckte Mantisse liegt innerhalb 0,10000000 und 0,99999999 oder sie ist Null. Für den Exponent gilt

$$-38 \leq pp \leq 39.$$

Ein zu kleiner Exponent verursacht das Drucken einer Null, ein zu großer dagegen einen Bereichsfehler (Stop).

F. Unterprogramme zur Berechnung spezieller Funktionen

Mittels dieser Unterprogramme wird zu einem gegebenen Argument der zugehörige Funktionswert berechnet. Vor Aufruf des Unterprogramms muß das Argument im Pseudo-Akkumulator stehen. Der Aufruf selbst und damit der Beginn der Rechnung erfolgt durch den entsprechenden Pseudo-Befehl. Nach beendeter Rechnung erscheint der Funktionswert im Pseudo-Akkumulator. Einzelheiten über Speicherbelegung, Zeit, Genauigkeit und programmierte Stops finden sich in Teil IV.

1. Unterprogramm 14.3 Sinus-Cosinus

Dieses Unterprogramm dient zur Berechnung des Sinus bzw. Cosinus eines Arguments, das im Bogenmaß vorliegen muß. Es verwendet zur Approximation ein Polynom neunten Grades.

Es wird durch den Befehl S0000 bzw. C0000 aufgerufen. Die folgende Tabelle gibt eine Übersicht über die vom Unterprogramm gelieferten Ergebnisse in Abhängigkeit vom Exponenten:

Exponent	sin x	cos x
+26 bis +127	Argument zu groß	Argument zu groß
-8 bis 25	Berechnung mit Polynom 9. Grades	
-20 bis -9	sin x = x	Berechnung mit Polynom 9. Grades
-127 bis -21	sin x = 0	cos x = 1

2. Unterprogramm 16.3 Arcustangens

Dieses Unterprogramm berechnet den zu einem gegebenen Wert gehörigen Arcustangens. Die Rechnung wird durch den Befehl A0000 eingeleitet. Das Ergebnis erscheint in Bogenmaß im Akkumulator.

3. Unterprogramm 18.2 Logarithmus

4. Unterprogramm 17.2 Exponentialfunktion

Die Berechnung des Logarithmus wird durch den Befehl N0000, die der Exponentialfunktion durch den Befehl H0000 eingeleitet.

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

Teil III BESCHREIBUNG DES HILFSPROGRAMMS 24.1A

A. Funktion

Das Programm 24.1A leistet folgendes:

1. Berechnung der absoluten Adressen für Befehle des Problemprogramms, falls diese relative Adressen enthalten, welche sich auf Daten im Format 1 beziehen.
2. Lochen eines hexadezimalen Streifens, der enthält:
 - a) Eine Programm-Eingaberoutine (Spur 62) ohne Prüfsumme
 - b) Das Codewort "+00U6200", mittels dessen der Sprung vom Eingabeprogramm 10.4 in das neue hexadezimale Eingabeprogramm auf Spur 62 veranlaßt wird
 - c) Das Problemprogramm nach ausgeführter Adressenumrechnung mit anschließender Prüfsumme
3. Drucken des Problemprogramms in dezimaler Form
4. Lochen von x0000000' nach beendetem Ausdrucken, falls die Sprungtaste gedrückt ist.

B. Eingabe

Die Eingabe besteht aus einem Streifen, der für jedes mit dem Hilfsprogramm zu bearbeitende Programm die 5 folgenden Typen von Informationen enthält:

1. Ein Füllbeginn-Codewort
2. Zwei Modifikations-Codeworte:
 - a) Das normale vom Typ /000xxxx'
 - b) Ein spezielles vom Typ -000xxxx.

Der Sektorteil des letzteren muß 00 oder ein Vielfaches von 4 sein: 00, 04, 08 ... 56, 60. Dieses Codewort modifiziert alle die Befehlsadressen, die sich auf die gespeicherten Eingangsdaten beziehen.

$$F_0 = D_0 + 4/3 i = D_0 + 88 \text{ Worte} = D_0 + 1 \text{ Spur, } 24 \text{ Zellen}$$

$$F_0 = 2136$$

Die zweite, aus vier Zahlen bestehende Gruppe, wird folgendermaßen gespeichert:

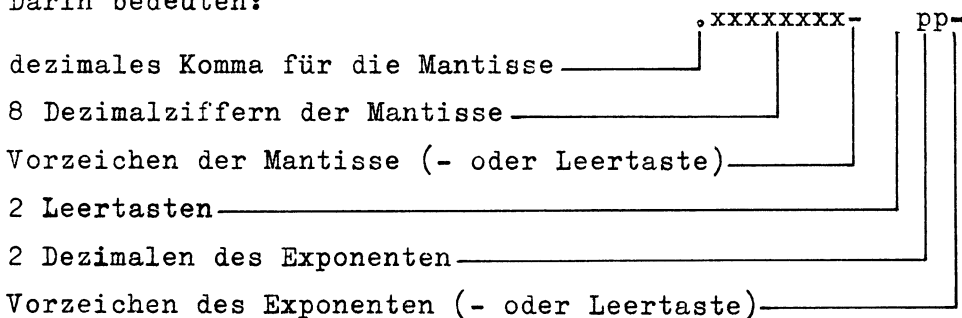
Absolute Adresse	Inhalt
2136	Mantisse von 0,0
2137	Mantisse von 30,5
2138	Mantisse von 750000
2139	Exponenten der 3 vorstehenden Zahlen
2140	$-10 \cdot 10^{12}$
2141	nichts
2142	nichts
2143	Exponent von $-10 \cdot 10^{12}$

Die zweite Datengruppe wird durch das Wort "exit" beendet.

E. Datenausgabe (Unterprogramm 12.5)

Ein gedrucktes Ausgabewort besteht aus einem dezimalen Punkt, dem acht dezimale Ziffern der Mantisse und ihr Vorzeichen folgen. Das Vorzeichen ist entweder ein "Minus" oder eine Leertaste. Auf das Vorzeichen der Mantisse folgen zwei Leertasten, danach werden die zwei dezimalen Ziffern des Zehner-Exponenten gedruckt, gefolgt von einem Minus-Zeichen, falls er negativ ist. Zum Schluß erfolgt noch ein Tabulatorsprung; danach wird der auf P0000 folgende Befehl im Problemprogramm ausgeführt. Somit sieht die Niederschrift einer Gleitkommazahl folgendermaßen aus:

Darin bedeuten:



Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

Bemerkung: Wenn der Flexowriter zur Ein- und Ausgabe benutzt wird,
darf KH 16 nicht gedrückt sein.

Auf den beiden folgenden Seiten ist das Ablauf-Programm für 24.1A
in zwei Teilen wiedergegeben (Abb. 1 und 2).

Der Teil I stellt den Programm-Ablauf dar, nachdem das Codewort
.xxxxxxx gefunden wurde, die Umrechnung aller Adressen im Problem-
programm also beendet wurde. Dieser Teil ist für den Bediener
besonders wichtig, weil er alle notwendigen Manipulationen zum Aus-
löchen des hexadezimalen Programmstreifens enthält.

Der Teil II stellt die Adressenumrechnungen selbst in Abhängigkeit
von der ersten gelesenen Tetrade eines jeden Wortes dar.

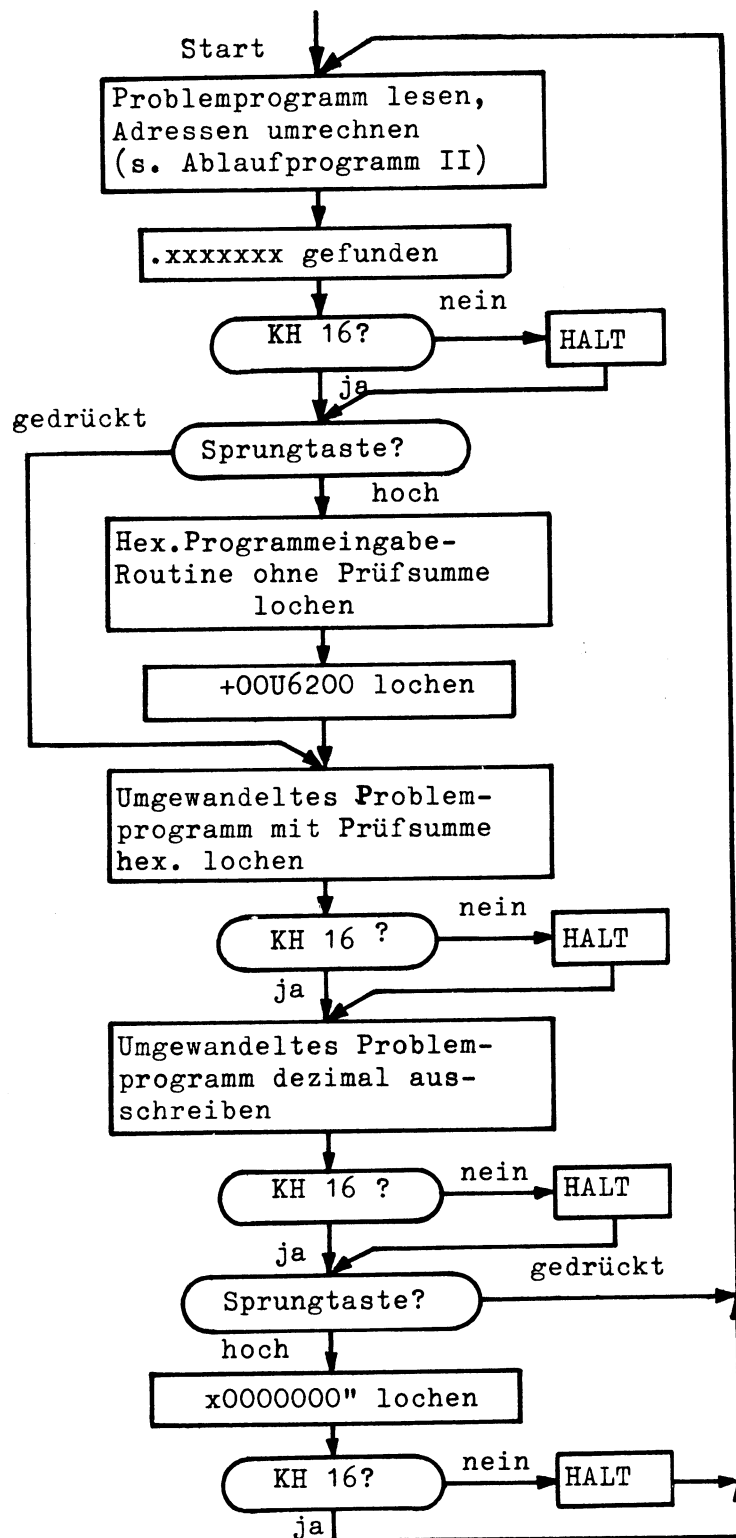


Abb. 1 Ablaufprogramm I für 24.1A

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

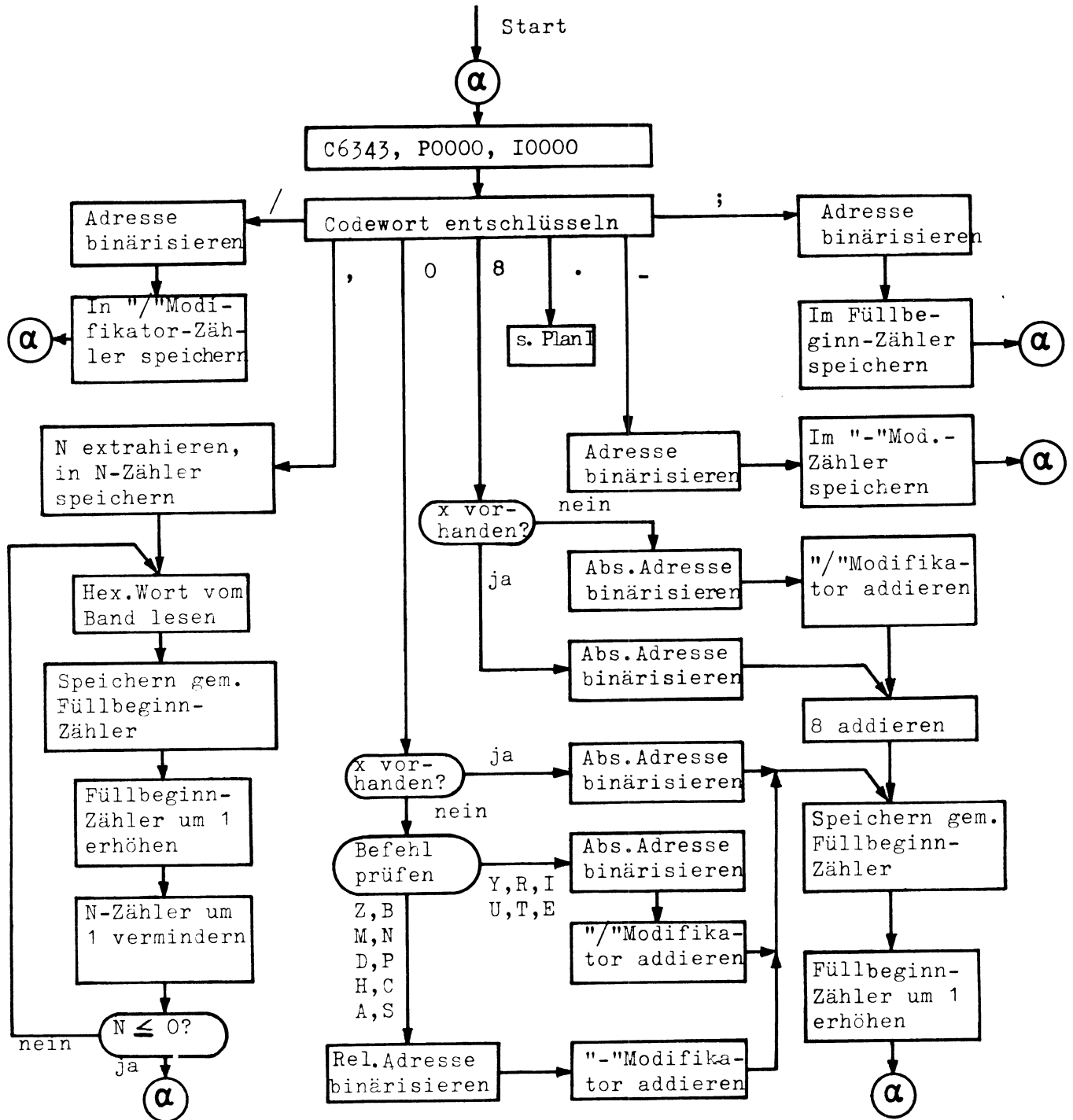


Abb. 2 Ablaufprogramm II für 24.1A

11



Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

Teil IV BENUTZUNG DES ERFP-SYSTEMS

1. Eingabe

Eingabedaten sind entweder auf Band gelochte oder bereits gespeicherte Gleitkommazahlen oder Zahlen in Pseudo-Registern, die das Ergebnis vorhergehender Rechnungen sind.

2. Aufruf-Befehlsfolge

Speicherort	Befehl	Adresse	
α	R	Lo	Lo $\hat{=}$ Anfangsadresse von 24.1
$\alpha + 1$	U	Lo	
$\alpha + 2$.	.	} Gleitkommabefehle
.	.	.	
.	.	.	
.	.	.	
$\alpha + n-1$.	.	
$\alpha + n$	E	0000	} Ausgang aus Gleitkommasystem
$\alpha + n+1$	Fortsetzung des Hauptprogramms		

3. Speicherbelegung

Lo sei die Anfangsadresse des Unterprogramms 24.1. Die übrigen Unterprogramme müssen dann mit den in der folgenden Tabelle angegebenen Füllbeginn- und Modifikator-Schlüsselworten eingegeben werden:

Unterprogramm	Füllbeginn	Modifikator	Speicherbelegung
ERFP	24.1 Lo	Lo	11 Spuren
Dateneingabe	11.5 Lo + 1100	Lo + 1100	7 Spuren
Datenausgabe	12.5 Lo + 1800	Lo + 1800	4 Spuren
Sinus-Cosinus	14.3 Lo + 2200	Lo	2 Spuren
Arcustangens	16.3 Lo + 2400	Lo	83 Zellen
Logarithmus	18.2 Lo + 2519	Lo	45 Zellen
Exponentialfunktion	17.2 Lo + 2600	Lo	2 Spuren

4. Programm-Stops

Speicherort	Bedeutung	Abhilfe
Lo + 0312	Exponent ≥ 128 für H oder C	Bei Drücken der Start-Taste wird die Rechnung fortgeführt. Der H- oder C-Befehl wird ignoriert.
Lo + 0710	Division durch Null oder Zahl, die nicht in Gleitkommaform ist	Die Rechnung kann nicht fortgeführt werden
Lo + 1119	D ist kein Vielfaches von 4 (Dateneingabe)	D ₀ korrigieren
Lo + 1743	Der dezimale Exponent P ist außerhalb des Bereichs $-21 \leq P \leq 38$ (Dateneingabe)	P korrigieren
Lo + 2045	Der Exponent ist größer als 127 (Datenausgabe)	Das Drücken der Start-Taste veranlaßt das Unterprogramm, einen Tabulatorsprung auszuführen, ohne zu drucken und nach 24.1 zurückzugehen.
Lo + 2200	Das Argument für Sinus oder Cosinus ist $\geq 2^{26}$	START drücken zur Fortführung der Rechnung. Der S0000- oder der C0000-Befehl wird ignoriert
Lo + 2526	Das Argument von ln x ist ≤ 0	START drücken. Programm wird mit ln x = 0 fortgesetzt
Lo + 2729	Das Argument für e ^x ist ≥ 128	Nach Drücken der Start-Taste wird die Rechnung fortgesetzt. Der H0000-Befehl wird ignoriert

5. Zeitbedarf

S. Tabelle 1 auf Seite 26.

6. Arbeitsspeicher

Als Speicher für zeitweilige Zwischenergebnisse wird Spur 63 benutzt:

Unterprogramm		Benutzte Zellen
ERFP-System	24.1	06, 07, 09, 13, 15-17, 19, 22, 26, 27, 31, 32, 37-40, 42, 53, 54, 59, 63
Eingabe	11.5	01-03, 07, 08, 10, 17, 18, 20, 21, 25, 26, 28, 33, 37, 42-45, 47-49, 53, 57, 59, 60
Ausgabe	12.5	00, 03, 09, 15, 16, 32, 33, 36, 42, 45, 50, 56, 61, 62
Sinus-Cosinus	14.3	06, 08, 09, 16, 40, 42, 43, 45, 52, 55
Arcustangens	16.3	04, 20, 27, 29, 31, 33, 35, 37, 39, 45, 46, 51
Logarithmus	18.2	09, 11, 18, 20, 39
Exponentialfunktion		06, 15, 17-22, 24, 40
	17.2	
Hilfsprogramm	24.1A	22, 31-33, 35, 38, 43, 48

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

7. Genauigkeit

- a) Sinus-Cosinus: Der maximale Fehler beträgt $5 \cdot 10^{-8}$
- b) Datenausgabe : Der maximale Fehler beträgt eine Einheit der 8. gedruckten Dezimalstelle.

8. Ausgang aus dem Gleitkommasystem

Nach dem Ausgangs-Befehl E0000 wird der nächste unmittelbar folgende Befehl als normaler LGP-30-Befehl ausgeführt.

9. Bemerkungen:

- a) Die Inhalte der Register des Gleitkommasystems werden nicht zerstört, wenn man es verläßt oder wieder in es eintritt.
- b) Es wird empfohlen, daß die Anfangsadresse Lo des Systems die Zelle 00 einer Spur sei. Andernfalls sind viele der Adressen, die sich auf Spur 63 beziehen, nicht optimal.
- c) Für Speicherung in Format 1 muß der Exponent kleiner als 128 sein; ist er kleiner als -127, wird die Zahl als eine Null gespeichert. Für Format 2 kann jede Zahl, die der Akkumulator aufnehmen kann, auch im Hauptspeicher gespeichert werden.
- d) Alle Befehle mit der Adresse 0000 haben spezielle Bedeutung und beziehen sich nicht auf den Speicherplatz 0000. Das vorliegende Gleitkommasystem hat 14 Befehle mit der Adresse 0000 und außerdem zwei spezielle Schiftbefehle xD000Y und xM000Y, was insgesamt 16 Sonderbefehle ausmacht. Die zwei Schiftbefehle benutzen die 9 Adressen 0001 bis 0009; deshalb dürfen diese Adressen nicht in dem Dividierbefehl Dxxxx und im Multiplizierbefehl Mxxxx benutzt werden.

Befehl	Ergebnis, falls Adresse $\alpha \neq 0$	Zeit im Format 1 in msec	Zeit im Format 2 in msec	Ergebnis, falls Adresse $\alpha = 0$ bzw. $\alpha = 0001$ bis 0009 bei $D000y$ und $M000y$	Zeit in msec
Z	$\langle \text{Adr. Akk} \rangle - \alpha = 0?$ Ja: Sprung; Nein: kein Sprung	105	105	Halt, falls KH 32 nicht gedrückt	105
B	$\langle \alpha \rangle \rightarrow \text{Akk.}$	305	250	Mache $\langle \text{Akk} \rangle$ positiv	135
Y	$\langle \text{Adr. Akk} \rangle \rightarrow \alpha$	135	135	Bilde Komplement von $\langle \text{Akk} \rangle$	105
R	Ort von R + 2 $\rightarrow \alpha$	135	135	$\sqrt{\langle \text{Akk} \rangle} \rightarrow \text{Akk}$	500
I	$\langle \text{Adr. Akk} \rangle + \alpha \rightarrow \text{Adr. Akk}$	165	150	Eingabe von Gleitkommazahlen	25/min.
D	$\langle \text{Akk} \rangle : \langle \alpha \rangle \rightarrow \text{Akk}$	350	300	$\langle \text{Akk} \rangle : 2 \rightarrow \text{Akk}$	165
N	$\langle M \rangle \times \langle \alpha \rangle + \langle \text{Akk} \rangle \rightarrow \text{Akk}$	585	535	$\ln \langle \text{Akk} \rangle \rightarrow \text{Akk}$	500
M	$\langle M \rangle \times \langle \alpha \rangle \rightarrow \text{Akk}$	320	270	$\langle \text{Akk} \rangle \times 2 \rightarrow \text{Akk}$	135
P	$\langle \alpha \rangle \rightarrow \text{M-Register}$	315	265	Drucke $\langle \text{Akk} \rangle$	2,25 sec
E	Adr. Teil $\langle \alpha \rangle \rightarrow \text{Adr. Akk}$	150	150	Ausgang aus dem System	105
U	Nächster Befehl von α	90	90	$\langle \text{Akk} \rangle \rightarrow \text{M}, \langle \text{M} \rangle \rightarrow \text{Akk}$	180
T	Falls $\langle \text{Akk} \rangle$ neg. Sprung nach α	105	105	Mache $\langle \text{Akk} \rangle$ negativ	135
-T	Falls $\langle \text{Akk} \rangle$ neg. oder Sprung- taste gedrückt, Sprung nach α	105	105	Falls $\langle \text{Akk} \rangle$ neg. oder Sprungtaste gedrückt, Sprung nach α	105
H	$\langle \text{Akk} \rangle \rightarrow \alpha$	290	210	$e^{\langle \text{Akk} \rangle} \rightarrow \text{Akk}$	470
C	$\langle \text{Akk} \rangle \rightarrow \alpha; 0 \rightarrow \text{Akk}$	305	225	Cosinus $\langle \text{Akk} \rangle \rightarrow \text{Akk}$	570
A	$\langle \text{Akk} \rangle + \langle \alpha \rangle \rightarrow \text{Akk}$	500	450	Arcustangens $\langle \text{Akk} \rangle \rightarrow \text{Akk}$	550
S	$\langle \text{Akk} \rangle - \langle \alpha \rangle \rightarrow \text{Akk}$	515	465	Sinus $\langle \text{Akk} \rangle \rightarrow \text{Akk}$	570

Die angegebenen Zeiten sind näherungsweise und variieren mit dem Betrag der Abweichungen von einem angenommenen Wert. Die tatsächlichen Zeiten dürften etwas geringer sein als die angegebenen. Die Zeit reduziert sich im allgemeinen, wenn irgendein Faktor Null ist.

Tabelle Zeitbedarf für Gleitkommabefehle

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

Teil V BEISPIELE

1. Berechnung eines Polynoms mit dem HORNER-Schema

Das Polynom

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

mit $a_0 = 0,09, a_1 = 0,078, a_2 = 0,056$
 $a_3 = 0,034, a_4 = 0,012$

soll für den Wert $x = 0,109$ nach dem HORNER-Schema berechnet werden:

$$P(x) = \left\{ \left[(a_4x + a_3) x + a_2 \right] x + a_1 \right\} + a_0$$

Die Eingangswerte werden im Format 1 in der Reihenfolge $x, a_4, a_3, a_2, a_1, a_0, 0$ in die Speicherzellen 4304 bis 4310 gesetzt.

Der Dateneingabe-Streifen sieht dann folgendermaßen aus:

Kennwort	Stop	Dezimalzahl Teil I	Stop	Dezimalzahl Teil II	Stop
4 3 0 4 0 0 0 0	'	+ 1 0 9 0 0 0 0	'	0 0 + 0 9	'
		+ 0 1 2 0 0 0 0	'	0 0 + 0 9	'
		+ 0 3 4 0 0 0 0	'	0 0 + 0 9	'
		+ 0 5 6 0 0 0 0	'	0 0 + 0 9	'
		+ 0 7 8 0 0 0 0	'	0 0 + 0 9	'
		+ 0 9 0 0 0 0 0	'	0 0 + 0 9	'
		e x i t	'		'

Die Befehlsfolge für Beispiel 1 sieht folgendermaßen aus:

Blatt A

LGP 30 - Rechenprogramm

Auftrag Nr. _____ Bearbeiter G.L. Waters Blatt von
 Programm Nr. 24.1 Geprüft _____ Datum 5. 6. 1958
 Problem Berechnung eines Polynoms 4. Grades Spur Nr. _____

Programmeingabe Schlüsselwort	Speicher Nr.	Befehl Operation	Adresse	Inhalt der Adresse	Bemerkungen
; 0 0 0 4 2 0 0	'				
/ 0 0 0 4 2 0 0	'				
- 0 0 0 4 3 0 4		x R	1 4 0 0		Sprung in
	0 0				
	0 1	x U	1 4 0 0		24.1
	0 2	x I	0 0 0 0		Zahlenwerte einlesen
	0 3	E	0 0 1 7	X 4305	Adresse f. 1. Koeffizienten
	0 4	Y	0 0 0 6	4305	a ₄ setzen
	0 5	B	0 0 0 6	Null (in 4310)	
	0 6	A []		a _n	Koeffizient a _n addieren
	0 7	x I	0 0 0 1	X (4305+n)	Adr. Akk um 1 erhöhen
	0 8	Y	0 0 0 6	(4305+n)	<Adr. Akk> → 0006
	0 9	Z	0 0 0 6		Ende?
	1 0	U	0 0 1 4		Nicht beendet
	1 1	x P	0 0 0 0	X	Beendet; Ergeb- nis drucken
	1 2	x E	0 0 0 0		Ausgang aus 24.1
	1 3	x Z	0 0 0 0		Halt (normaler Befehl)
	1 4	x U	0 0 0 0		<Akk> → M-Register
	1 5	M	0 0 0 0	X x (in 4304)	Mit x multipli- zieren
	1 6	U	0 0 0 6		zum nächsten Term
	1 7	Z	0 0 0 1	a ₄ (in 4305)	
. 0 0 0 X X X X	'				
	1 8				
	1 9				

Interpretierendes Gleitkommasystem
mit erweitertem Bereich (ERFP)

24.1

2. Berechnung des gleichen Polynoms in Format 2

Die Daten sind an den gleichen relativen Adressen gespeichert.

Der Dateneingabe-Streifen hat folgendes Aussehen:

Kennwort	Stop	Dezimalzahl Teil I	Stop	Dezimalzahl Teil II	Stop
8 0 0 0 . 4 3 0 4	'	+ 1 0 9 0 0 0 0 0	'	0 0 + 0 9	'
		+ 0 1 2 0 0 0 0 0	'	0 0 + 0 9	'
		+ 0 3 4 0 0 0 0 0	'	0 0 + 0 9	'
		+ 0 5 6 0 0 0 0 0	'	0 0 + 0 9	'
		+ 0 7 8 0 0 0 0 0	'	0 0 + 0 9	'
		+ 0 9 0 0 0 0 0 0	'	0 0 + 0 9	'
		e x i t	'		

Zum Einlesen des Problemprogramms ist es nicht notwendig, daß das Programm 24.1A gespeichert ist. Es wird mit 10.4 eingelesen und dann gleich nach 4200 gesprungen. Eine Adressenumrechnung der Daten entfällt, da im Programm nur absolute Adressen verwendet wurden. Deshalb ist auch der "-" Modifikator nicht vorhanden.

Die Niederschrift der Befehlsfolge befindet sich auf der nächsten Seite.

Man könnte auch beide Formate in einem Programm verwenden. Dann muß aber 24.1A verwendet werden und der "-" Modifikator angegeben sein. Alle Befehle, die sich auf Daten in Format 2 beziehen, müssen mit x versehen sein und die absoluten Adressen der Daten enthalten. Der Dateneingabe-Streifen ist in mindestens 2 Gruppen eingeteilt mit den jeweiligen das Format bestimmenden Kennworten.

Die Befehlsfolge für Beispiel 2 sieht folgendermaßen aus:

Blatt A

LGP 30 - Rechenprogramm

Auftrag Nr. _____ Bearbeiter G.L. Waters Blatt von _____
 Programm Nr. 24.1 Geprüft _____ Datum 5.6.1958 _____
 Problem Berechnung eines Polynoms 4. Grades Spur Nr. _____

Programmeingabe Schlüsselwort	Speicher Nr.	Befehl Operation	Adresse	Inhalt der Adresse	Bemerkungen
; 0 0 0 4 2 0 0					
/ 0 0 0 4 2 0 0	⊗				
	0 0	x R	1 4 0 0		Sprung in
	0 1	x U	1 4 0 0		24.1
	0 2	x I	0 0 0 0		Zahlenwerte ein- lesen
	0 3	E	0 0 1 8	⊗ 4306	} Adresse für 1. Koeffizienten a ₄ einsetzen
	0 4	Y	0 0 0 6	4306	
	0 5	8 0 0 B	0 0 1 3	Null	
	0 6	8 0 0 A	[]	a _n	Koeffizienten a _n addieren
	0 7	8 0 x I	0 0 0 2	⊗ (4306+2n)	Adr.Akk um 2 erhöhen
	0 8	Y	0 0 0 6	(4306+2n)	⟨Adr.Akk⟩ → 0006
	0 9	x Z	4 3 1 6		Beendet?
	1 0	U	0 0 1 5		Nicht beendet
	1 1	x P	0 0 0 0	⊗	Beendet, drucken
	1 2	x E	0 0 0 0		Ausgang aus 24.1
	1 3	x Z	0 0 0 0		Halt (normaler Befehl)
	1 4	x Z	0 0 0 0		
	1 5	x U	0 0 0 0	⊗	⟨Akk⟩ → M-Register
	1 6	8 0 x M	4 3 0 4		Mit x multipli- zieren
	1 7	U	0 0 0 6		zum nächsten Term
	1 8	x Z	4 3 0 6	a ₄	
0 0 0 4 2 0 0	1 9			⊗	
	2 0				