

*Beschreibung
der Betriebsart
"Gleitkomma"
beim ER 56*

*für RWG 5603
+ 5604 mit GK-Zusackeneinrichtung*



STANDARD ELEKTRIK LORENZ

Informatik

IS/AI Jacobs
Ausgabe b, Jan. 60

Beschreibung der Betriebsart "Gleitkomma".

=====

	Seite
<u>Zahlendarstellung in Gleitkomma</u>	1
Vorzeichenstelle	2
Zahlenfaktor	
Exponent und Charakteristik	
<u>Transport von Gleitkommazahlen</u>	3
<u>Doppelt lange Gleitkommazahlen</u>	4
Zusammenfügen von Gleitkommazahlen	5
Trennen von Gleitkommazahlen in abspeicherbare Teile	6
<u>Normalisieren</u>	7
nach rechts: Befehl 20,0	8
nach links: Befehl 20,0	9
Runden: Befehl 21,2	
<u>Die echte Null als Ersatzzahl</u>	10
bei Exponentenunterlauf	
Rechnen mit der echten Null	
Mult.; Div.	
Prüfen auf echte Null	11
<u>Bereichsuntersuchung</u>	12
Exponenten-Überlauf	
Exponenten-Unterlauf	
Taste Exponentenstop am Bedienungspult	
<u>Gleitkomma-Befehle</u>	13
Übersicht und Beschreibung der einzelnen Befehle	
Transportbefehle	15
Arithmetische Operationen	Add.: 17; Mult.: 27; Div.: 31
Exponentenregister-Befehle	35
Vergleichsbefehle	37
Schieben, Runden und Normalisieren	39
Ausgabevorbereitung	46
Betriebsartwahl	47

Zahlendarstellung in Gleitkomma

Vor allem bei wissenschaftlichen Berechnungen kommen oft als Eingangsgrößen Zahlen vor, die um mehrere Zehnerpotenzen schwanken können. Damit kann man bei umfangreichen Rechnungen nicht einmal die Größenordnung des Ergebnisses abschätzen.

Man verwendet deshalb eine Zahlendarstellung, die diese Nachteile vermeidet. Dabei wird, ähnlich wie beim logarithmischen Rechnen, das Vorzeichen und der Zahlenfaktor (entspricht dem Ziffernwert bei Festkomma) angegeben und ein dazugehöriger, ganzzahliger Zehnerexponent, der die Stellung des Kommas, bezogen auf den Zahlenfaktor angibt. Diese Zahlendarstellung kann man halblogarithmisch nennen.

Da ein Exponentenbereich von -49 bis $+49$ für die Mehrzahl aller Rechnungen völlig ausreicht, kann man die Vorzeichenstelle des Exponenten dadurch sparen, daß man den Exponenten um 50 erhöht. Diesen um 50 erhöhten Exponenten nennt man Charakteristik.

Durch die Exponentenverschiebung erhält man (immer positive) Charakteristiken von 1 bis 99.

Exponent	-49	0	+49
Charakteristik	1	50	99

Eine Gleitkommazahl $Z = \pm \underline{Z} \cdot 10^{z-50}$ besteht also aus drei Teilen, die in drei verschiedenen Rechenwerken verarbeitet werden:

Vorzeichen der Zahl, verarbeitet im Vorzeichenrechenwerk
Zahlenfaktor \underline{Z} , verarbeitet im Hauptrechenwerk (mit A oder M oder A,M);
Charakteristik z (der um 50 erhöhte Exponent) der Zahl, verarbeitet im Exponenten-Rechenwerk.

Die Vorzeichenstelle der Zahl hat
 verschlüsselungen und Bedeutungen:

folgende Dezimal-

Dez. Bedeutung

- 1 + positive Zahl
- 2 - negative Zahl
- 4 +Q positive gekennzeichnete Zahl
- 5 -Q negative gekennzeichnete Zahl

0,7,8

- 9 ω als ω -Wort (Blockendezeichen), ...

Das Komma steht, wie bei Festkomma, hinter dem Vorzeichen, vor der ersten Dezimalen.

Eine Gleitkommazahl heißt normalisiert, wenn der Zahlenfaktor zwischen 0,1 und 1 liegt.

Der Zahlenfaktor ist 11-stellig für Zahlen in \bar{A} oder \bar{M} , 22-stellig bei der Benutzung beider Register zusammen: \bar{A}, \bar{M} .

Zum Akkumulator A gehört das zweistellige Exponentenregister a; zum Multiplikator-Register M gehört das Exponentenregister m und zum Hilfsregister N, das dem Programmierer nicht zugänglich ist, gehört das Exponentenregister e.

Beispiel für eine Gleitkommazahl Z im Akkumulator:

$$Z = \pm \underline{z} \cdot 10^{\pm z}$$

$$= \pm \underline{z} \cdot 10^{(z+50)-50} = \underline{z} \cdot 10^{z-50}$$

Gleitkommazahl Z =

Vz	1	2	3	4	5	6	7	8	9	10	11	1	2
----	---	---	---	---	---	---	---	---	---	----	----	---	---

Vorzeichen

Zahlenfaktor

Charakteristik z

= Exponent z + 50

z = z + 50

Vz

z

Zahl	0,01	0,1	1	10	100 ...
	$0,1 \cdot 10^{-1}$	$0,1 \cdot 10^0$	$0,1 \cdot 10^1$	$0,1 \cdot 10^2$	$0,1 \cdot 10^3$

Exponent	-1	0	+1	+2	+3
----------	----	---	----	----	----

Charakteristik	1....49	50	51	52	53....99
----------------	---------	----	----	----	----------

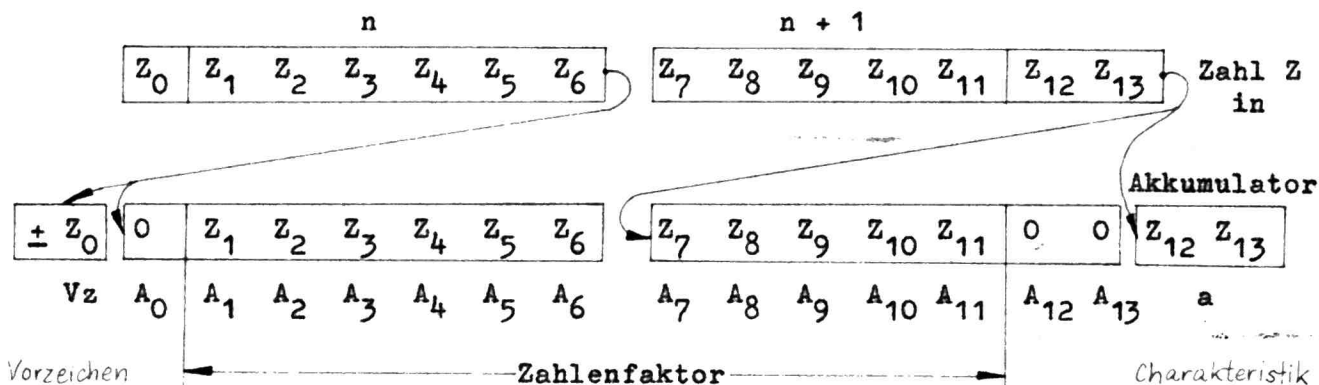
Beispiele für normalisierte Zahlen

	Vz	Zahlenfaktor	Ch
- 0,5 = -0,5 · 10 ⁰	2	5 0 0 0 0 0 0 0 0 0 0 0	50
+ π = + 3,14... = +0,31 · 10 ¹	1	3 1 4 1 5 9 2 6 5 3 5	51
-150,0 = -0,15 · 10 ³	2	1 5 0 0 0 0 0 0 0 0 0 0	53
+ 0,004 = +0,4 · 10 ⁻²	1	4 0 0 0 0 0 0 0 0 0 0 0	48

	Vz	Zahlenfaktor	Ch
nicht normalisiert: +0,004	1	0 0 4 0 0 0 0 0 0 0 0 0	50

Transport von Gleitkommazahlen

Beim Einlaufen der Zahl Z vom Speicher in den Akkumulator werden die beiden Teilwörter nacheinander verarbeitet.



Zuerst läuft der höherwertige Teil der Zahl Z in den Akkumulator A in der Reihenfolge Z₆, Z₅, Z₄, Z₃, Z₂, Z₁. Dabei wird die linke Hälfte des Akkumulators von links her gefüllt und der gesamte Inhalt dann jeweils um 1 Stelle nach rechts geschoben. Z₀ als Vorzeichenstelle gelangt dann in den Vorzeichenrechner, während gleichzeitig eine Null in die höchste Akkumulatorstelle A₀ eingedrückt wird.

Nun wird der hintere Teil der Zahl Z eingeschrieben.

Die Charakteristik der Zahl gelangt in das Exponenten-Register a, wobei gleichzeitig die letzten beiden (rechten) Stellen des Akkumulators (A₁₂, A₁₃) mit Nullen gefüllt werden. Darauf folgen die letzten Stellen der Zahl Z, zuerst Z₁₁, dann Z₁₀...Z₇. Damit steht die gesamte Zahl Z im Akkumulator.

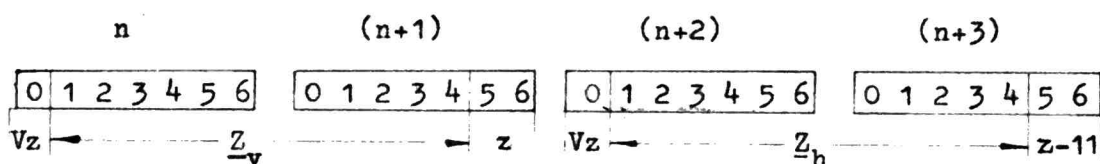
Entsprechend werden beim Abspeichern die Zahlenteile der Rechenergebnisse aus den Registern zusammengefügt und zum Kernspeicher geleitet.

Doppelt lange Gleitkommazahlen

Die Gleitkommabefehle werden so ausgeführt, daß der Benutzer auch Zahlen mit 22-stelligem Zahlenfaktor und Charakteristik a verarbeiten kann.

$$Z = \pm (\underline{z}_v \cdot 10^{z-50} + \underline{z}_h \cdot 10^{(z-11)-50})$$

Im Speicher stehen dann in 4 Zellen n bis n+3 zwei normale Doppelwörter je mit Vorzeichen, 11-stelligem Zahlenfaktor und Charakteristik. Beide Vorzeichen sind gleich, die Charakteristik des zweiten Doppelworts ist um 11 kleiner als die des ersten. Jedes Doppelwort enthält also vollständige Angaben und kann auch selbständig benutzt werden.



Im Akkumulator A, M stehen die Zahlenfaktoren am Ende jeder Gleitkomma-Operation mit Rücksicht auf etwa nachfolgendes Abspeichern in der den Speicherstellen entsprechenden Anordnung, nicht benutzte Akkumulatorstellen enthalten meist Nullen. Während des Gleitkomma-Rechenablaufs müssen aber alle 22 Stellen des Zahlenfaktors aneinander anschließen.

Das ist erreicht durch das Zusammenfügen vor jeder doppelt-langen arithmetischen Gleitkommaoperation und das Trennen nach jeder der genannten Operationen. Zusammenfügen und Trennen erfolgt automatisch im Rechenwerk und wird bei Addition, Subtraktion und Division (Trennen auch bei Multiplikation) sowie Runden, Normalisieren und den doppelt langen Schiebebefehlen 20,3 und 20,6 (nicht 20,9) durchgeführt.

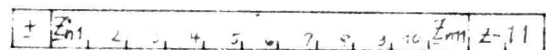
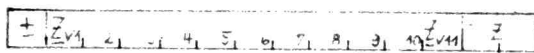
Zusammenfügen von Gleitkommazahlenfaktoren

Arbeiten A und M zusammen, dann werden automatisch die aus dem Speicher kommenden Zahlen so geschoben, daß die mittleren drei Nullen (im $A_{12, 13}$ und M_0) verschwinden und ein 22-stelliger Zahlenfaktor entsteht.

Ins Register a gelangt die Charakteristik z; (A) wird um 2 Stellen nach rechts, (M) um eine Stelle nach links geschoben und Null eingedrückt. Die Ziffern (meist Null) aus $A_{12, 13}$ und M_0 laufen ins Leere, die Zahlenhälften passen an der Nahtstelle richtig zusammen (Z_{n1} schließt an Z_{v11} an), sind aber um 2 Stellen zu weit nach rechts geschoben. Das wird rückgängig gemacht durch Schieben von (A, M) um 2 Stellen nach links. In M laufen hinten 2 Nullen ein.

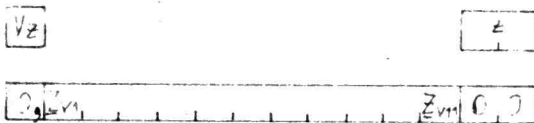
Speicherzelle n, n + 1

Speicherzelle n+2, n+3



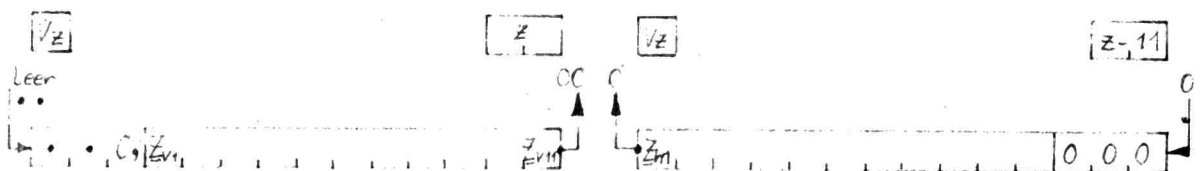
Akkumulator A

Multiplikatorregister M

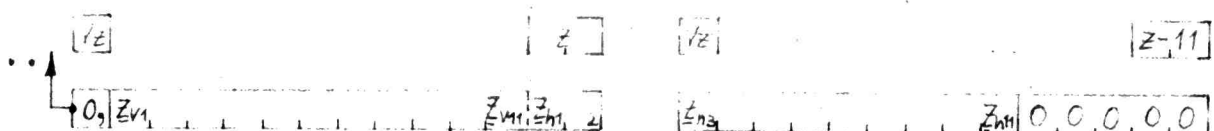


(A) um 2 Stellen rechts, →

← (M) um 1 Stelle links schieben!



(A, M) um 2 Stellen nach links schieben!



Durch vorhergehende Schiebefehle, Einschreiben in Festkomma, durch logisches Einschreiben oder durch arithmetische Festkomma-Operationen können in

$$A_0; A_{12, 13} \text{ oder } M_0 \text{ und } M_{12, 13}$$

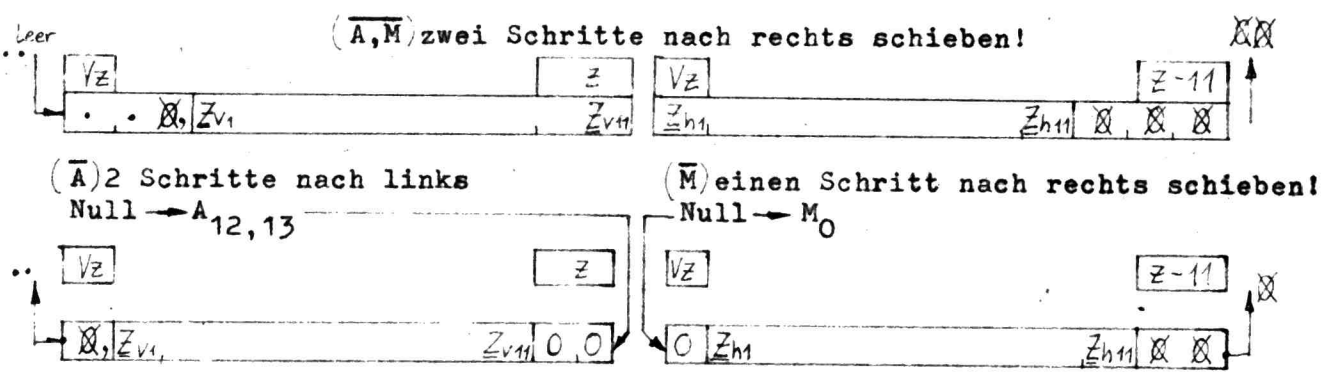
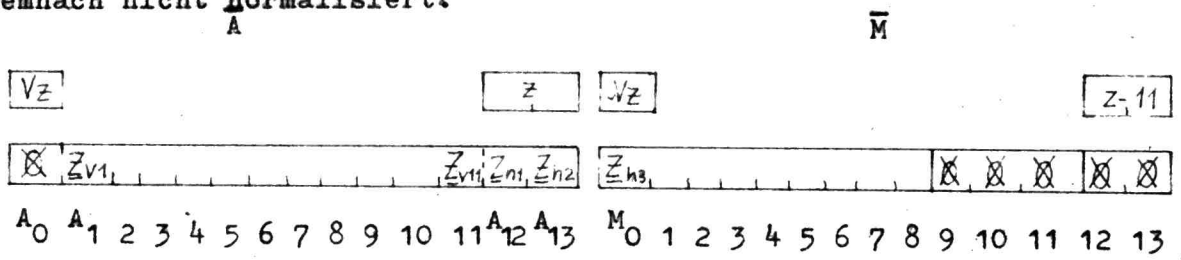
von Null verschiedene Ziffern auftreten. ($\otimes = 0$ oder \times)

Alle genannten Ziffern gehen beim Abspeichern (Befehle 32;34 und 42,44) verloren.

$(A_{12, 13})$ und (M_0) gehen auch bei allen arithmetischen Operationen einschließlich Runden und Normalisieren verloren, dagegen bleiben (A_0) und $(M_{12, 13})$ erhalten.

Trennen von Gleitkommazahlen in abspeicherbare Teile

Am Ende einer Gleitkommaoperation steht die Zahl Z in $\overline{A, M}$ mit Vorzeichen (im Vorzeichenregister von A), 22-stelligem Zahlenfaktor (in A von $A_1 \dots A_{13}$ und in M von M_0 bis M_8) und 2-stelliger Charakteristik (im Exponentenregister a), Nullen oder durch Schieben oder hervorgerufene Ziffern stehen im A_0, M_9 bis M_{13} . Unmittelbar anschließend werden die Zahlenfaktorteile im allgemeinen automatisch normalisiert (bis 10 Links-Normalisierschritte oder 1 Rechts-Normalisierschritt). Bei mehr als 10 Nullen hinter dem Komma ist das Ergebnis in A, M demnach nicht normalisiert.



Rechnet man jetzt weiter, so bleiben die letzten beiden Ziffern in $M_{12,13}$ erhalten, beim Abspeichern gehen sie jedoch verloren, an ihre Stelle tritt die Charakteristik.

Beispiel:

Abspeichern nach $n, n+1$
Vorzeichen und Charakteristik a
zufügen

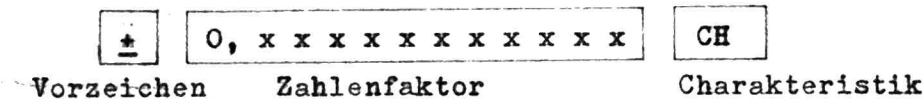
Abspeichern nach $n+2, n+3$
Vorzeichen und Charakteristik m
zufügen

\pm	Z_{-v1}	Z_{-v11}	z
-------	-----------	------------	-----

\pm	Z_{-h1}	Z_{-h11}	$z-11$
-------	-----------	------------	--------

Normalisieren von Gleitkommazahlen

Unter Normalisieren versteht man einen Schiebevorgang, der die Zahl in eine Normalform bringt. Der Zahlenfaktor der Gleitkommazahl wird dabei in den Bereich $0,1 \leq Z \leq 1$ gebracht.



Das Normalisieren geschieht durch den Befehl $20,0$: "Normalisieren und Zählen der Schritte" sowie am Ende aller arithmetischen Gleitkommaoperationen für A, M .

Von einem Zähler k wird die Anzahl der ausgeführten Links- Normalisierungsschritte aufgenommen.

Dieser k -Zähler ist für den Programmierer erreichbar durch den Befehl $21,0$: Anzahl der Normalisierungsschritte in ein Indexregister $J_{i;3,4}$ (nach $J_{i, 1,2}$ werden Nullen gebracht).

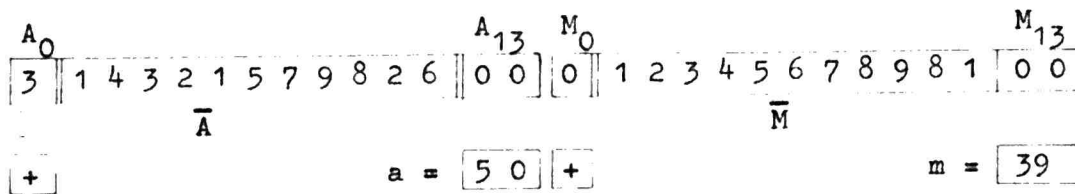
Man unterscheidet dabei Normalisieren nach rechts und nach links: Die Stellen A_0 und A_1 des Akkumulators werden auf größer oder gleich Null abgefragt.

Nach jeder arithmetischen Gleitkomma-Operation werden automatisch maximal 10 Links-Schiebeschritte oder ein Rechts-Schiebeschritt durchgeführt.

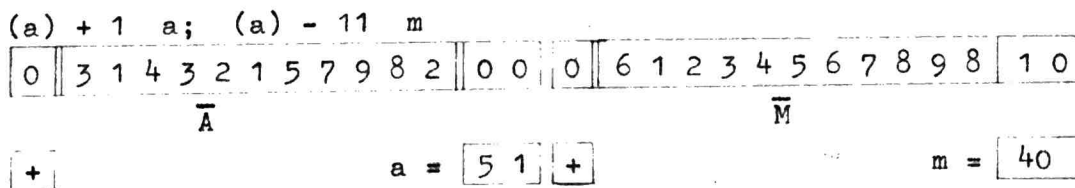
Nach rechts normalisieren $20,0$ (Fall 1)

Durch Überlauf kann $(A_0) \neq 0$ entstanden sein. Die Zahl in A, M wird um einen Schritt nach rechts geschoben. Der k -Zähler wird auf $k = 99 \hat{=} - 1$ gesetzt.

Beispiel: Nicht-normalisierte Zahl in $\overline{A, M}$: $+ 3,14 \dots = 3,14 \cdot 10^0$



Normalform: Zahlenfaktor im Bereich $0,1 \dots 1$. = $0,314 \cdot 10^{+1}$;
einen Schritt nach rechts schieben. Die Charakteristik wird um 1 erhöht:



Null Schritte normalisieren:

Die Zahl ist bereits in Normalform: $A_0 = 0$ und $A_1 = 0$. Beispiele s.o.
Der Normalisierzähler (k-Zähler) steht am Ende des Normalisiervorganges
auf $k = 0$, d.h. es brauchte nicht normalisiert zu werden.

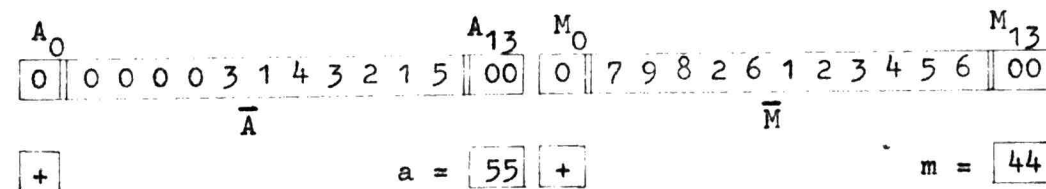
Nach links normalisieren: 20,0 (Fall 2)

Wurde im ersten Prüfschritt festgestellt, daß $A_0 = 0$ und $A_1 = 0$ sind,
so wird so lange nach links geschoben, bis $A_1 \neq 0$ ist.

Die Anzahl der Links-Normalisier-Schiebeschritte (max. 25) wird im
k-Zähler gezählt und die Charakteristik später um die Anzahl entspre-
chend erniedrigt.

Nach jeder arithmetischen Gleitkomma-Operation werden maximal 10 Links-
Schiebeschritte automatisch durchgeführt.

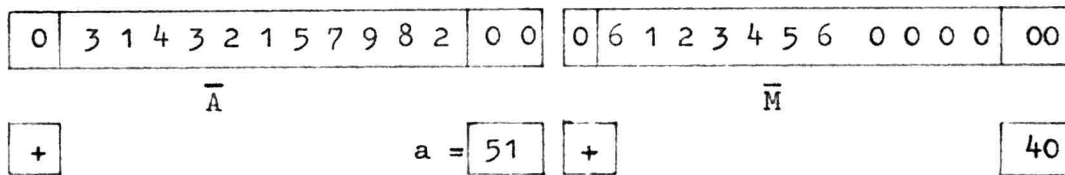
Beispiel: Nicht-normalisierte Zahl in $\overline{A, M}$: $0,0000 314 \dots \cdot 10^{+5} = 0,314 \cdot 10^1$



Normalform: Zahlenfaktor im Bereich $0,1 \dots 1 = 0,314 \dots \cdot 10^{+1}$;

4 Schritte nach links schieben. Die Charakteristik wird um 4 erniedrigt:

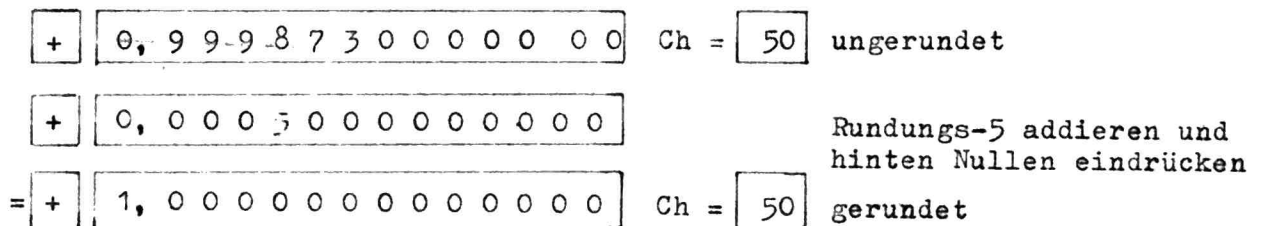
$a - 4 \rightarrow a; \quad m - 4 \rightarrow m$



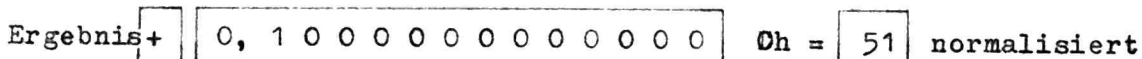
Ausnahme: A M Runden, Befehl 21.2

Nach dem Runden wird nicht nach links normalisiert, wohl aber nach rechts. Dies ist durch den Wunsch der Programmierer bedingt, Gleitkommazahlen auf eine feste Stellenzahl runden zu können, die dann für die Ausgabe in fester Form, Komma unter Komma, bereitstehen, wenn z.B. alle Charakteristiken einer Tabelle usw. gleich sind. Bei vorlaufenden Nullen steht dann ebenfalls Komma unter Komma.

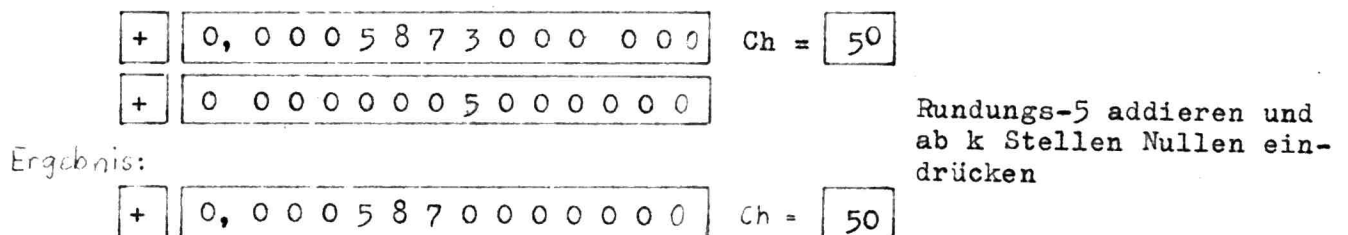
Beispiel: Zahl heißt +0,999 873, runden auf k=3 Stellen hinter dem Komma



1 Schritt nach rechts schieben



Beispiel: Zahl heißt + 0,0005873, runden auf k=6 Stellen hinter dem Komma:



Echte Null

Für die Zahl $Z = 0$, die in Gleitkommaform nicht darstellbar ist, ist eine Sonderkennzeichnung vorhanden:

Echte Null $\bar{0}$: Zahlenfaktor 0...0, Charakteristik 00.

Bei Gleitkommaoperationen sorgt die nachfolgend beschriebene Sondersteuerung dafür, daß die so dargestellte Null den üblichen Rechenregeln entsprechend verarbeitet wird:

$$Z + \bar{0} = Z \qquad \bar{0} \cdot Z = \bar{0} \qquad \bar{0} : Z = \bar{0}$$

Dabei genügt es, zum Prüfen auf Echte Null nur die Charakteristik zu untersuchen (vgl. Seite 11). Der Zahlenfaktor 0 entsteht beim Rechenprozeß von selbst.

Die Nullprüfung nur auf die Charakteristik gestattet es, bei Unterlauf des Exponentenbereichs (Charakteristik $(a) < 01$) automatisch eine Ersatzzahl anstelle der unterlaufenden Zahl einzusetzen, so daß der Rechenprozeß ohne Unterbrechung weiterlaufen kann. Die Ersatzzahl behält den Zahlenfaktor der ersetzten Zahl, hat aber die Charakteristik 00. Sie setzt die genannte Sondersteuerung ebenso in Gang wie die Echte Null, die Ergebnisse sind aber, von Ausnahmen abgesehen, falsch.

Wenn Exponentenunterlauf nicht vorkommen soll, kann der Rechner durch Einlegen der Taste "Exponentenstop" bei jedem Unterlauf ebenso zum Halten gebracht werden, wie das bei Exponentenüberlauf $((a) > 99)$ auf alle Fälle geschieht.

Bei Exponentenunterlauf in m ($(m) < 01$) ^{wird} bei nicht gedrückter Taste "Exponentenstop" ebenfalls die Ersatzcharakteristik 00 in m eingesetzt. Bei gedrückter Taste wird Exponentenstop nur ausgelöst, wenn aus M abgespeichert oder links geschoben wird (Befehle 42; 44 oder 20,3).

Prüfen auf echte Null

Vor dem Einlaufen der Exponenten ins Einheitsaddierwerk des Exponentenrechenwerks, zwischen Schalter und Codewandler werden die Exponenten auf echte Null-Charakteristik 00 geprüft.

2 aus 5 Code					Dezimal
V	IV	III	II	I	
0	0	0	1	1	0
		1		1	1
		1	1		2
	1			1	3
	1		1		4
	1	1			5
1				1	6
1			1		7
1		1			8
1	1				9

Ein Merker "echte Null" wird mit dem Schlußzeichen der vorhergehenden Operation auf "Echte Null" gesetzt.

Die drei bits III, IV und V werden auf Nicht-Eins geprüft. Wenn eines dieser drei bits kommt, dann war keine Null, sondern irgendein gültiger Exponent vorhanden.

Der Merker "echte Null" der das Eindrücken der Null im Ergebnis bewirkt, wird bei \neq 00 gelöscht.

Multiplikation und Division

Der Merker "echte Null" wird zu Beginn der Multiplikation oder Division gesetzt und mit Prüftakten bei Multiplikation oder Division gelöscht oder nicht.

Nach der Multiplikation oder Division werden beim Normalisieren bei echter Null 00 \rightarrow a und 00 \rightarrow m gebracht.

Bereichsuntersuchung

Die Hunderterstelle des Registers a wird auf Exponenten-Über- oder Unterlauf geprüft.

Es wird "Exponentenstop" signalisiert, wenn eines der drei bits III, IV oder V erscheint, d. h. wenn keine Null vorliegt (s.a. Taste "Exp.stop").

Nach dem Normalisieren werden die Charakteristik (a) und die Charakteristik (m) auf Bereichs-Über- oder Unterschreitung geprüft. Außerdem wird geprüft, ob ^{eine} der Charakteristiken zufällig 00 geworden ist (gilt als Bereichs-Unterschreitung).

Bereichs-Über- oder Unterschreitung im Register m führt zu Exponentenstop nur bei Befehlen, die das Register m betreffen, also
20,3 Schieben links
42; 44 Abspeichern aus M, da man sonst mit der falschen Charakteristik weiterrechnen würde.

Bereichs-Unterschreitung ist nur bei gedrückter Taste "Exponentenstop"-möglich.

Taste "Exponentenstop" = "Exp.Stop" am Bedienungspult

Die Maschine hält bei gedrückter Taste bei Exponenten-Überlauf und Exponenten-Unterlauf von (a) an.

Die Maschine hält bei nicht gedrückter Taste nur bei Exponenten-Überlauf von (a) an. Bei Exponenten-Unterlauf von (a) hält die Maschine nicht an, die Charakteristik (a) wird jedoch zu Null gemacht, die Charakteristik (m) nur dann, wenn auch (m) < 01 geworden ist. Es wird mit der gekennzeichneten Ersatzzahl weitergerechnet.

Bei Exponenten-Unterlauf von (m) hält die Maschine ebenfalls nicht an, die Charakteristik (m) wird jedoch zu Null gemacht.

Gleitkomma-Befehle

Transportbefehle

Befehls-Nr.	Bedeutung	
31	+ (n,n+1) → \bar{A}	Positives Einschreiben nach \bar{A}
33	-	Negatives
41	+ (n,n+1) → \bar{M}	Positives Einschreiben nach \bar{M}
43	-	Negatives
32	+ (\bar{A}) → n,n+1	Positives Abspeichern aus \bar{A}
34	-	Negatives
42	+ (\bar{M}) → n,n+1	Positives Abspeichern aus \bar{M}
44	-	Negatives

Arithmetische Operationen

35	+ (\bar{A}, \bar{M}) (n,n+1) → (\bar{A}, \bar{M})	Addieren
36	-	Subtrahieren
37	(\bar{A}) . (n,n+1) → (\bar{A}, \bar{M})	Multiplizieren
38	(\bar{A}, \bar{M}) : (n,n+1) → (\bar{A}, \bar{M})	Dividieren
39	(n,n+1) : (\bar{A}) → (\bar{A}, \bar{M})	Umgekehrtes Dividieren
30	(\bar{A}) mit (n,n+1) ausblenden.	Ausblenden (Stellenweises Multiplizieren)

Exponentenregister-Befehle

21,4	Einschreiben	$n_{3,4}$ → a
21,5	Einschreiben	($A_{12,13}$) → a
21,6	Abspeichern	(a) → $J_{1,3,4}$; Null → $J_{1,1,2}$
21,7	Abspeichern	(a) → $A_{12,13}$; Null → $A_{0\dots 11}$; Vz +.
21,8	Vergleichen	(a) mit $n_{3,4}$ vergleichen; Mark.setzen
21,9	Vergleichen	(a) mit ($A_{12,13}$) " "

Vergleichs-Befehle für \bar{A}

- 28 Vergleiche (\bar{A}) mit $(n, n+1)$
- 29 Vergleiche $|\bar{A}|$ mit $|(n, n+1)|$

Schieben, Runden und Normalisieren

- 20,1 (A) um k Stellen nach links schieben Ch.unverändert
 - 20,2 (M) um k Stellen nach links schieben Ch.unverändert
 - 20,3 (A,M) um k Stellen nach links schieben (m) \rightarrow a, m "
 - 20,4 (A) um k Stellen nach rechts schieben Ch.unverändert
 - 20,5 (M) um k Stellen nach rechts schieben Ch.unverändert
 - 20,6 (\bar{A}, \bar{M}) um k Stellen nach rechts schieben (a) \rightarrow m, a "
 - 20,7 (A) um k Stellen nach rechts im Ring schieben
 - 20,8 (M) um k Stellen nach rechts im Ring schieben
 - 20,9 (\bar{A}, \bar{M}) um k Stellen nach rechts im Ring schieben
- } Ch.unverändert
- 21,2 (\bar{A}, \bar{M}) Runden auf k Stellen hinter dem Komma
 - 20,0 (A,M) Normalisieren und Schritte zählen, Char. anpassen
 - 21,0 Anzahl der Links-Normalisierschritte $\rightarrow J_{i;3,4}$; Null $\rightarrow J_{i;1,2}$.
 - 21,1 Normalisierprüfwert ab jetzt $k[\leq 10]$
 - 15 Sprung, wenn Überlauf-Flipflop aus
 - 17 " " Normalisier " "

Ausgabevorbereitung

- 22 Besonderer Ausgabevorbereitungs-Befehl:
 $(\bar{A}) \rightarrow n, n+1$ mit 10-stelligem Zahlenfaktor, Exponentenvorzeichen
und zweistelligem Exponenten (nicht Charakteristik).

Betriebsartwahl

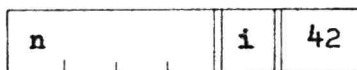
- 19,2 ab jetzt Betriebsart Gleitkomma.



Einschreiben in \bar{M}

Gleitkomma: $(n, n + 1) \rightarrow \bar{M}$

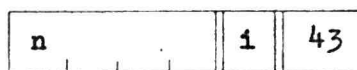
Der Inhalt von $Z_{12} Z_{13}$ wird in das Exponentenregister m gebracht und in $Z_0, M_{12} M_{13}$ werden Nullen eingefüllt. Die elfstellige Mantisse der Gleitkommazahl und das Vorzeichen werden in das Multiplikatorregister bzw. das Vorzeichenregister von M gebracht.



Abspeichern aus \bar{M}

Gleitkomma: $(\bar{M}) \rightarrow n, n + 1$

Die im Multiplikatorregister stehende Gleitkommazahl wird in zwei aufeinanderfolgende Speicherzellen übertragen. Der Inhalt des Vorzeichenregisters von M ersetzt den Inhalt von Z_0 , der Inhalt von M_1 bis M_{11} ersetzt den Inhalt von Z_1 bis Z_{11} , und der Inhalt des Exponentenregisters m ersetzt den Inhalt von $Z_{12} Z_{13}$.



Negatives Einschreiben in \bar{M}

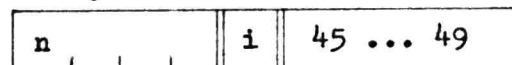
Gleitkomma: $-(n, n + 1) \rightarrow \bar{M}$

Das Einschreiben einer in zwei aufeinanderfolgenden Speicherzellen stehenden Gleitkommazahl in das Multiplikatorregister erfolgt entsprechend dem Befehl 41 mit einer Umkehrung des Vorzeichens beim Übertragen von Z_0 in das Vorzeichenregister von M . Der Inhalt der Speicherzellen bleibt unverändert.



Gleitkomma: $-(\bar{M}) \rightarrow n, n + 1$

Das Abspeichern des Inhaltes des Multiplikatorregisters in zwei aufeinanderfolgende Speicherzellen erfolgt entsprechend dem Befehl 42 mit einer Umkehrung des Vorzeichens beim Übertragen des Inhaltes des Vorzeichenregisters nach Z_0 . Der Inhalt des Multiplikatorregisters bleibt unverändert.



In der Betriebsart "Gleitkomma" existieren die Befehle 45 bis 49 nicht.

n

i

3 5

Addition

$$\overline{(A, M)} + (n, n+1) \rightarrow \overline{A, M}$$

Der Befehl 35 bewirkt, daß zu der in $\overline{A, M}$ stehenden Zahl die in zwei aufeinanderfolgenden Speicherzellen stehende Gleitkommazahl addiert wird.

Die in $\overline{A, M}$ stehende doppelt lange Gleitkommazahl besteht aus: Vorzeichen im Vorzeichenregister von A; ebenfalls im Vorzeichenregister von M; Zahlenfaktor in A_1 bis A_{11} und M_1 bis M_{11} ; in $M_{12, 13}$ können durch Schieben, möglicherweise auch durch Rechnung in GK von Null verschiedene Ziffern hineingelangen und durch Schieben wieder herausgeholt werden. In M_0 können Ziffern $\neq 0$ nur durch Schieben, in $A_{12, 13}$ können Ziffern $\neq 0$ durch Schieben oder arithmetische Operationen in Festkomma hineingelangen. $A_0, A_{12}, A_{13}, M_0, M_{12}, M_{13}$ werden nicht abgespeichert.

Das Ergebnis der Addition steht in $\overline{A, M}$. Anschließend an die Addition läuft ein Normalisierungsvorgang für die Zahl in $\overline{A, M}$ ab, der allerdings auf maximal 10 Links-Schiebeschritte beschränkt ist. (Vergl. dazu Befehl 20,0). Es kann demnach als Ergebnis auch eine nicht normalisierte Zahl in Erscheinung treten. Die Anzahl der Normalisierungsschritte wird, wie durch das Normalisieren mittels Befehl 20,0, in einem internen Zähler festgehalten, dessen Inhalt durch Befehl 21,0 nach einem Indexregister gebracht werden kann. Das Normalisier Flipflop wird auf "ein" gesetzt, wenn die Anzahl der Links-Schiebeschritte gleich oder größer als der mittels Vorbefehl 21,1 eingestellte Normalisierprüfwert ist. Stand der Normalisier-Flipflop bereits auf "ein", so bleibt es auf "ein".

Wird durch Addition (A_0) erhöht (Überlauf), so wird das Überlauf-Flipflop auf "ein" gesetzt, auch wenn durch das nachfolgende Normalisieren der Überlauf wieder beseitigt wird. Das Überlauf-Flipflop wird mit Befehl 15 abgefragt, das Normalisier-Flipflop mit Befehl 17 (s.dort).

n

i

3 6

Subtraktion

$$\overline{(A, M)} - (n, n + 1) \rightarrow \overline{A, M}$$

Der Befehl 36 bewirkt, daß von der in $\overline{A, M}$ stehenden Zahl (Vorzeichen im Vorzeichenregister von A, Zahlenfaktor in A_1 bis A_{11} und M_1 bis M_{13} Charakteristik in a) die in zwei aufeinanderfolgenden Speicherzellen stehende Gleitkommazahl subtrahiert wird. Das Ergebnis der Subtraktion steht in $\overline{A, M}$. Über den Normalisierungsvorgang, der sich an die arithmetische Operation anschließt, siehe bei Befehl 35.

Addition und Subtraktion in Gleitkomma

n	i	3 5
---	---	-----

Addition: $(\overline{A, M}) + (n, n + 1) \rightarrow \overline{A, M}$

n	i	3 6
---	---	-----

Subtraktion: $(\overline{A, M}) - (n, n + 1) \rightarrow \overline{A, M}$

Die Subtraktion wird, wie bei Festkomma, durch Addition des Zehnkomplements ausgeführt.

1. Abschnitt: Operanden bereitstellen

Der Augend steht vor Beginn der Operation in $\overline{A, M}$.

Der Addend gelangt aus dem Speicher $n, n + 1$ in das Hilfsregister N , zuerst $(n) \rightarrow N1$, dann $(n + 1) \rightarrow N2$ (s. Transporte von GK-Zahlen).

2. Abschnitt: Exponenten sicherstellen

Bei Gleitkomma können die beiden Operanden mit verschiedenen Exponenten versehen sein, so daß ein Zurechtrücken der Operanden nötig ist.

Als erstes werden die Exponenten sichergestellt:

Die Charakteristik des Augenden in $\overline{A, M}$ befindet sich im Exponentenregister a , wird nach Register m gebracht: $(a) \rightarrow m$.

Die Charakteristik n des Addenden aus dem Speicher $n, n + 1$ gelangt in das zum Hilfsregister N gehörige Exponentenregister e und wird dort sichergestellt: $n \rightarrow e$.

3. Abschnitt: Exponentendifferenz:

Im gleichen Rechengang, in dem die Charakteristiken sichergestellt werden, wird im Exponentenrechenwerk die Differenz der Charakteristiken $(n - a)$ gebildet und nach Register a abgesetzt.

Diese Differenz $n - a = (N + 50) - (A + 50) = N - A$ Differenz der Exponenten N und A .

Damit wird festgestellt, ob die Zahl \overline{N} größer, im Bereich oder kleiner als die Zahl in $\overline{A, M}$ ist.

Die drei Fälle unterscheiden sich wie folgt:

Fall 1: Der Exponent N der Zahl im Speicher $n, n + 1$ ist größer oder gleich dem Exponenten A der Zahl $\overline{A, M}$. $N - A \geq 0$. } $ChN \geq ChA$

Beispiel: $N - A = 10 - 0 = 10$

Fall 2 a): Der Exponent n der Zahl im Speicher $n, n + 1$ ist im Exponentenbereich der vorderen Zahlenhälfte von (A, M) } $Ch N < Ch A$
aber höchstens
um 13
kleiner

$$-13 \leq n - m < 0 \qquad 02-13 \leq N$$

(oder $0 < m - n \leq 13$)

Beispiel: $n - m = -9 - 0 = -9$

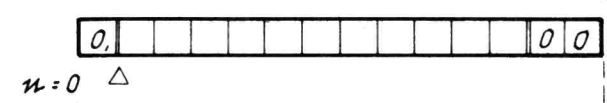
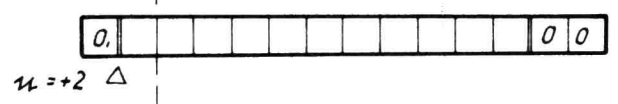
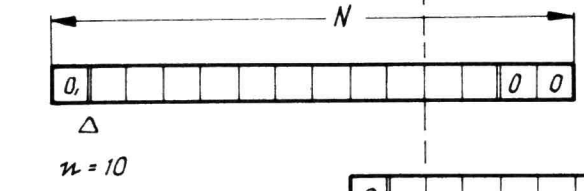
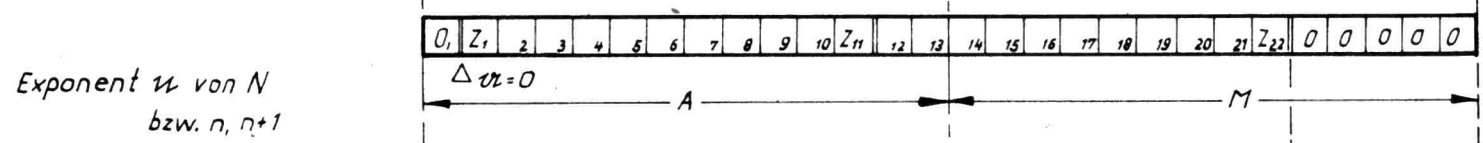
Fall 2 b): Der Exponent n der Zahl im Speicher $n, n + 1$ liegt im Exponentenbereich der hinteren Zahlenhälfte von (A, M) oder ist noch kleiner. } $Ch N$ um
mehr
als 13
kleiner
als $Ch A$

$$-13 > n - m$$

Beispiel: $n - m = -16 - 0 = -16$

GK
20

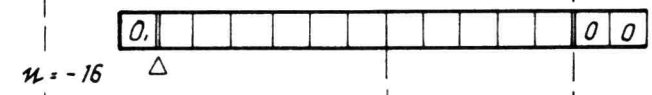
$\alpha = +10$ $+5$ 0 -5 -10 -15 -20 $-25 =$ Exponent α von (A, M)



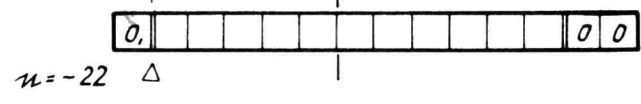
Fall 1: $\alpha - \alpha \geq 0$



Fall 2a: $-13 \leq \alpha - \alpha < 0$



Fall 2b: $-13 > \alpha - \alpha$



Beispiele für die 3 Fälle der Addition/Subtraktion bei Gleitkomma

Befehle: 35; 36: $(A, M) \pm (n, n+1) \rightarrow A, M$

4. Abschnitt: Prüftakt P 1: Fall 1 oder 2?

Die Differenz $(n) - (a)$ im Register a wird auf 0 oder 9 in der Hunderterstelle des Registers a abgefragt mit Prüftakt P 1.

Wenn $n - a \geq 0$; d.h. $\boxed{0 \ x \ x}$ im Register a, dann Fall 1.

Wenn $n - a < 0$; d.h. $\boxed{9 \ x \ x}$ " " " " 2.

Fall 1: k-Zähler fremdsetzen

Für Fall 1 wird der k-Zähler mit dem Signal w1 auf das zweistellige Neunerkomplement des Inhalts von Register a gesetzt.

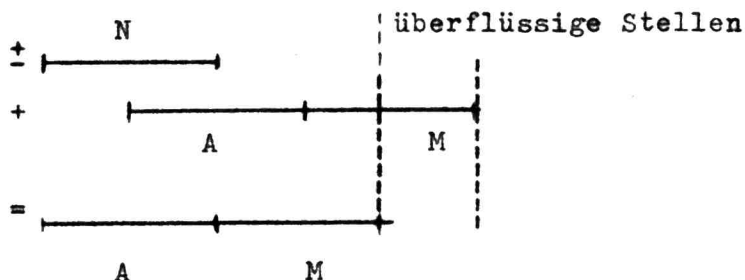
Beispiel: Exponenten $\mathcal{N} = +10$, $\mathcal{M} = 0$

Charakteristiken $(n) = 60$, $(a) = 50$. Differenz $(n) - (a) = 60 - 50 = 10$ im Register a als: $\boxed{0 \ 1 \ 0}$; mit P1 prüfen. 0 = Fall 1.

k-Zähler setzen auf $\boxed{8 \ 9}$

Der k-Zähler zählt später vorwärts von z.B. 89, 90 ... bis 00

Die Anzahl der Schritte entspricht dann den überflüssigen Stellen von (A, M) , da (N) größer ist (vorne heraushängt).



Fall 1: Ergebnis - Charakteristik absetzen

Da (N) die größere Zahl ist, wird die Charakteristik von (N) als Ergebnis-Charakteristik nach Register a abgesetzt. (Vorher $(n) \rightarrow e$ sicherstellen, jetzt: $(e) \rightarrow a$.)

5. Abschnitt: Prüftakt P2: Fall 2 a oder 2 b?

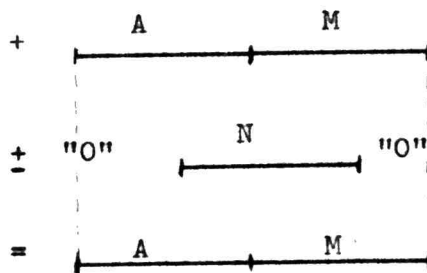
Zum Inhalt des Registers a wird 13 addiert, um festzustellen, ob (N) in den Bereich von (A, M) fällt oder kleiner ist (hinten herausragt).

Wenn $a + 13 \geq 0$, d.h. $\boxed{0 \ x \ x}$ im Register a, dann Fall 2 a.

Wenn $a + 13 < 0$, d.h. $\boxed{9 \ x \ x}$ " " " " 2 b.

Fall 2 a: Registerlänge berücksichtigen und k-Zähler fremdsetzen

Um Fall 2 a und 2 b gleich behandeln zu können, wurde nur 13 zu (a) addiert. Die Registerlänge von A ist aber 14. Deshalb wird jetzt nochmals eine 1 addiert. Der k-Zähler wird auf das Komplement gesetzt, er zählt später z.B. von 94 bis 0. Die Anzahl der Schritte entspricht der Versetzung von (N) gegenüber (A, M), d.h. so oft müssen Nullen zu (A, M) addiert werden, bis $(A, M) + (N)$ addiert wird. Nach 1+ Stellen (d.h. der k-Zähler zählt von 0 ... 13, springt zurück auf Null) werden wieder Nullen zu (A, M) addiert.



Beispiel: Fall 2 a.

Exponenten $\mathcal{N} = -9$, $\mathcal{M} = 0$; Charakteristiken $(n) = 41$, $(a) = 50$.

Differenz $n - a = 41 - 50 = -9$, im Register a als Zehnerkomplement:

991, Prüfen mit P 1 ergibt Fall 2. $(a) + 13 = -9 + 13 = +4$, im Register a als: 0 0 4. Prüfen mit P 2 ergibt Fall 2a. $(a) + 1 = 4 + 1 = 5$, im Register a als 0 0 5. K-Zähler auf Neunerkomplement: 9 4

Fall 2 b: Komplement bilden und k-Zähler fremdsetzen

Im Fall 2 b ist die Zahl N im Bereich (M) oder ragt "hinten hinaus". Die Prüfung mit P 2 ergibt immer noch einen negativen Registerinhalt von a, so daß man das Zehnerkomplement bilden muß. Dabei wird die Registerlänge berücksichtigt durch Addition einer 1 zu (a). Insgesamt also: $(a) + 1$ und Zehnerkomplement. Da die Komplementbildner aber nur Neunerkomplement bilden können, führt man entsprechend: $(a) + 0$ und Neunerkomplement durch.

Der k-Zähler wird mit dem Neunerkomplement von (a) fremdgesetzt.

Beispiel: Fall 2 b

Exponenten: $n = -16$, $u = 0$, Charakteristiken $n = 34$, $a = 50$.

Differenz $(n) - (a) = 34 - 50 = -16$, im Register a als $\boxed{984}$, mit P1 prüfen ergibt Fall 2; $(a) + 13 = -16 + 13 = -3$, im Register a als $\boxed{997}$. Prüfen mit P 2 ergibt Fall 2 b. Neunerkomplement bilden im Register a: $\boxed{002}$

k-Zähler auf Neunerkomplement setzen: $\boxed{97}$

Fall 2 a und 2 b: Ergebnis-Charakteristik absetzen.

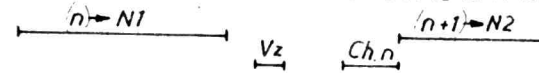
Da die Zahl in A, M größer ist als N, wird die Charakteristik von (A, M) als Ergebnis-Charakteristik nach Register a übernommen.

(Vorher $a \rightarrow m$ sicherstellen, jetzt: $m \rightarrow a$).

Rechenwerkstakte

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

1. Abschnitt: Operanden bereitstellen.



Exponenten-
Rechenwerkstakte

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

2. Abschnitt: Exponenten sicherstellen.



3. Abschnitt: Exponentendifferenz



4. Abschnitt: Prüftakt P1: Fall 1 oder 2?

k-Zähler fremdsetzen mit w 1.

Fall 1: Ergebnis-Charakteristik absetzen.

Fall 2: Bereich A oder M?



5. Abschnitt: Prüftakt P2: Fall 2a oder 2b?

Fall 2a: Registerlänge berücksichtigen.

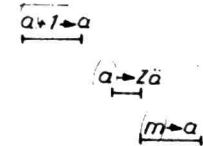
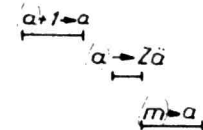
k-Zähler fremdsetzen mit w1.

Ergebnis-Charakteristik absetzen.

Fall 2b: Komplement bilden.

k-Zähler fremdsetzen mit w 1.

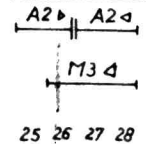
Ergebnis-Charakteristik absetzen.



6. Abschnitt: (A, M) zusammenziehen. A schieben

M schieben

Rechenwerkstakte 0 1



6. Abschnitt: Zusammenrücken von (A, M).

Am Ende der Exponentenprüfung läuft in allen 3 Fällen im Rechenwerk das Zusammenrücken von (A, M) ab, und zwar so, daß beide Operationen zur gleichen Zeit beendet sind.

7. bis 9. Abschnitt: Addition bzw. Subtraktion

Für die einzelnen Fälle sind die drei Abschnitte jeweils verschieden. Zuerst wird (A, M) oder (N) geschoben, so daß die Registerlänge 28 für (A, M) des Ergebnisses sichergestellt wird. Im Fall 2 a wird nur Null zu (A, M) addiert. Die eigentliche Addition bzw. Subtraktion erfolgt im 8. Abschnitt, 14 Schritte lang. Darauf folgt wieder eine Addition von Null zu (A, M).

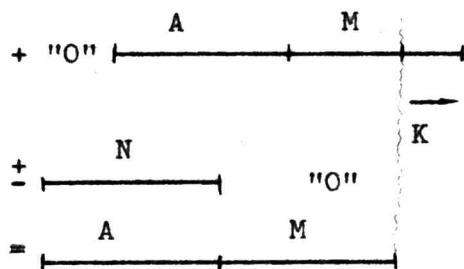
Der k-Zähler wurde auf das Komplement von Register a gesetzt, er gibt jeweils die Anzahl der Schiebeschritte oder Null-Additionsschritte an.

Fall 1:

7. Abschnitt: (A, M) nach rechts schieben. k-Zähler gibt Anzahl der Schiebeschritte an.

8. Abschnitt: (A, M) + "0" → A, M; 14 Schritte.

9. Abschnitt: (A, M) + (N) → A, M.

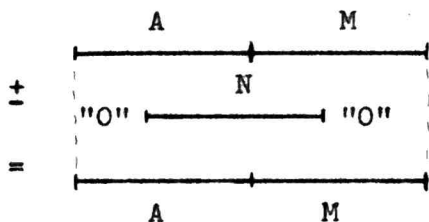


Fall 2 a

7. Abschnitt: (A, M) + "0" → A, M; Anzahl der Schritte vom k-Zähler bestimmt.

8. Abschnitt: (A, M) + (N) → A, M; 14 Schritte

9. Abschnitt: (A, M) + "0" → A, M.

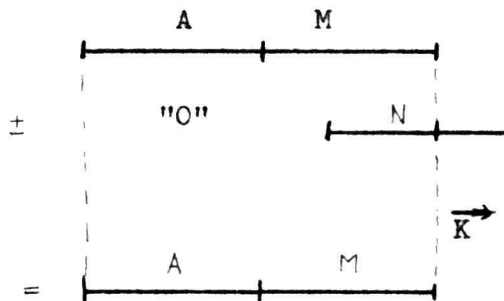


Fall 2 b:

7. Abschnitt: (N) um k Schritte nach rechts schieben

8. Abschnitt: $(A, M) + (N) \rightarrow A, M$; 14 Schritte

9. Abschnitt: $(A, M) + "0" \rightarrow A, M$;



10. Abschnitt: Komplementumlauf

Für alle 3 Fälle ist nach der Addition bzw. Subtraktion eine Komplementbildung vorgesehen, die durchgeführt wird, wenn sich das Ergebnis in A, M in Komplementform ergibt, d.h. wenn $(A_0) = 9$ ist. Für $(A_0) = 0$ (d.h. Ergebnis erscheint im Implement) wird nur ein Umlauf ohne Komplementbildung durchgeführt: $(A, M) \rightarrow A, M$.

11. Abschnitt: Normalisieren von $\overline{A, M}$

s. Normalisieren von Gk-Zahlen.

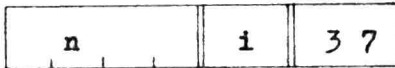
12. Abschnitt: Trennen in abspeicherbare Teile (s. dort)

n	1	37	Multiplikation
---	---	----	----------------

$(\overline{A}) \cdot (n, n + 1) \rightarrow \overline{A, M}$

Der Befehl 37 bewirkt, daß die im Akkumulator stehende Gleitkommazahl (Vorzeichen im Vorzeichenregister von A, Mantisse in A_0 bis A_{13} ; Charakteristik im Exponentenregister a) mit einer in zwei aufeinanderfolgenden Speicherzellen stehenden Gleitkommazahl multipliziert wird. Das Ergebnis der Multiplikation steht in $\overline{A, M}$ (Vorzeichen im Vorzeichenregister von A, Mantisse in A_1 bis A_{11} und M_1 bis M_{11} ; A_{12} A_{13} und M_{12} M_{13} enthalten Nullen; Charakteristik im Exponentenregister a). Anschließend an die Multiplikation läuft wie bei allen arithmetischen Gleitkommaoperationen ein Normalisiervorgang ab. Vergl. dazu die Bemerkungen bei Befehl 35. Zahlenfaktor

Multiplikation in Gleitkomma



Multiplikation:

$$(\bar{A}) \cdot (n, n + 1) \rightarrow A, M$$

Zahlenfaktoren: $11...14 \times 11$ -stellig \rightarrow ~~21~~²⁵-stellig.

Vor Beginn der Multiplikation muß der Multiplikand in \bar{A} stehen.

1. Abschnitt: Bereitstellen der Faktoren: $(A) \rightarrow N$; $-(n, n + 1) \rightarrow M$

Vordere Zahlenhälfte zuerst: "0" \rightarrow A1 \rightarrow A2 \rightarrow N1; $(n) \rightarrow$ M1.

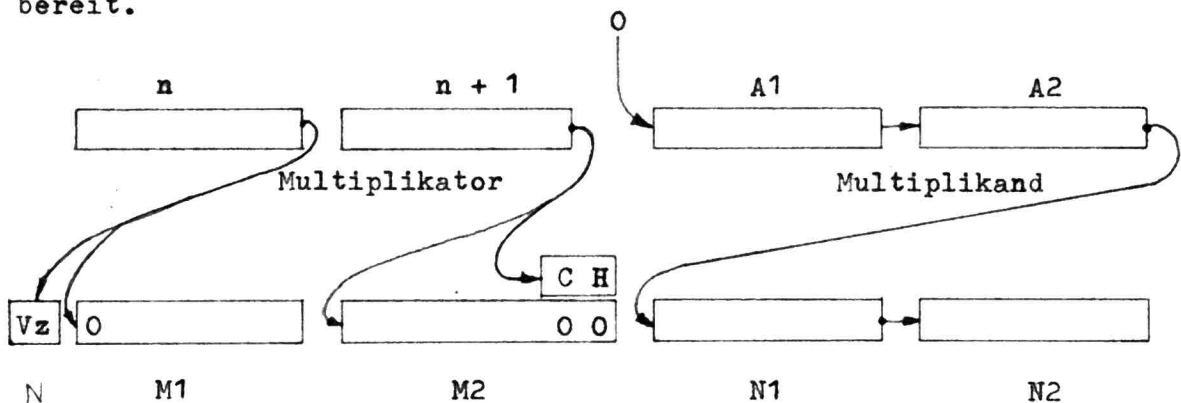
Multiplikand: Die vordere Hälfte von A, A1, wird in die hintere Hälfte A2 gebracht, die hintere Hälfte A2 gelangt in die vordere Hälfte des Hilfsregisters N, N1.

Multiplikator: Der Inhalt der Speicherzelle n wird, mit der letzten Ziffer Z_6 beginnend, in die vordere Hälfte des Multiplikatorregisters M, M1, gebracht. Der Inhalt von Z_0 gelangt in das Vorzeichenregister von N, N_0 wird durch eine Null aufgefüllt.

Hintere Zahlenhälfte: "0" \rightarrow A1 \rightarrow A2 \rightarrow N1 \rightarrow N2; $(n + 1) \rightarrow$ M2

Multiplikand: Die Zahl wird so geschoben, daß der Multiplikand im Hilfsregister N bereitsteht.

Multiplikator: Die hintere Hälfte der Zahl in $n + 1$ geht in die hintere Hälfte des Multiplikatorregisters M 2. Dabei werden zuerst anstelle der Charakteristik, die ins Exponentenregister m geleitet wird, zwei Nullen eingedrückt. Der gesamte Multiplikator steht dann in M bereit.



2. Abschnitt

Ergebnis-Charakteristik errechnen: $(n_{12,13}) + (a) \rightarrow a$; $(a) - 50 \rightarrow a$.
 Gleichzeitig mit der Verarbeitung der Zahlenfaktoren im Rechenwerk werden die beiden Charakteristiken im Gleitkomma-Rechenwerk verarbeitet. Während die Charakteristik des Multiplikanden (a) von $A_{12,13}$ kommt, wird

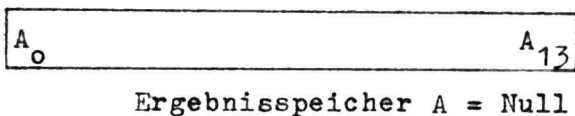
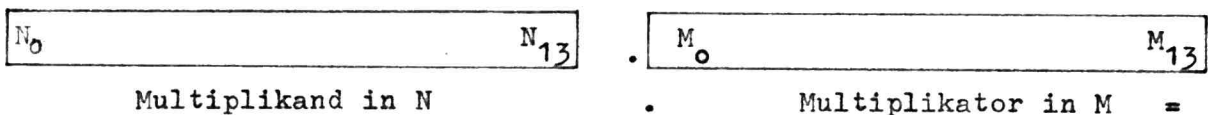
zur gleichen Zeit die Charakteristik des Multiplikators vom Speicher her kommend ($n_{12, 13}$) zu (a) addiert.

Die Charakteristik ist der um 50 erhöhte Zehnerexponent der Zahl. Bei der Multiplikation werden die Exponenten addiert. Man hat also jetzt zweimal 50 zuviel addiert, muß also 50 wieder abziehen, um die Ergebnis-Charakteristik zu erhalten. Damit ist die Bearbeitung des Exponenten (bis auf das Normalisieren am Ende der Multiplikation) beendet.

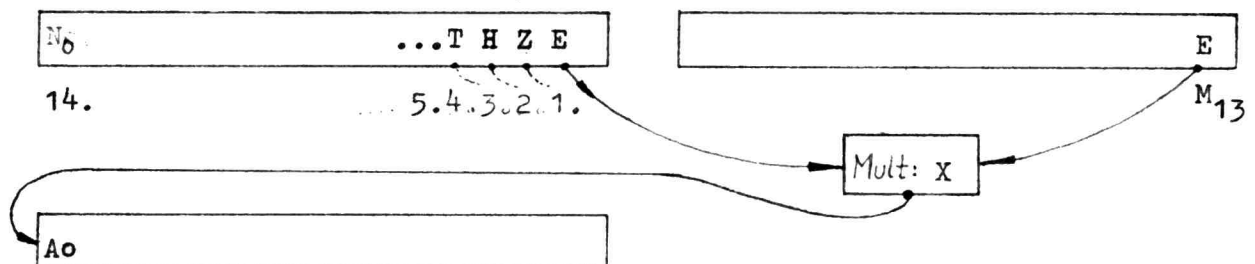
$$\begin{array}{r}
 \boxed{} \boxed{CH} \cdot \boxed{} \boxed{CH} = \boxed{} \boxed{CH} \\
 X \cdot 10^{x+50} \cdot Y \cdot 10^{y+50} = X \cdot Y \cdot 10^{(x+50)+(y+50)-50} \\
 = Z \cdot 10^{z+50}
 \end{array}$$

3. Abschnitt: Zahlenfaktoren miteinander multiplizieren.

Ausgangsstellung:



1. Schritt: $(M_{13} \cdot (N_{13,12,11\dots 0}) \rightarrow A_0 \rightarrow \dots A_{13}, M_1$

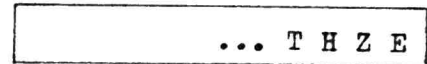
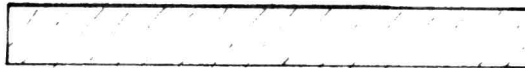


Die Multiplikatorziffer wird laufend aus M_{13} aufgerufen. Der Multiplikand im Hilfsregister N wird rechts im Ring geschoben und dabei ziffernweise mit M_{13} multipliziert. Das Ergebnis wird nach A, beginnend mit A_0 , abgesetzt und jeweils zwischen den Ziffernmultiplikationen $M_{13} \cdot N$ nach rechts geschoben. Nach 14 Ziffernmultiplikationen $M_{13} \cdot N$ steht das Ergebnis in A, die höchste Stelle in A_0 . Nur noch der Übertrag aus der letzten Multiplikation $M_{13} \cdot N_0$ ist um eine Dezimale verzögert abzusetzen.

ü

erstes Teilprodukt (14stellig)

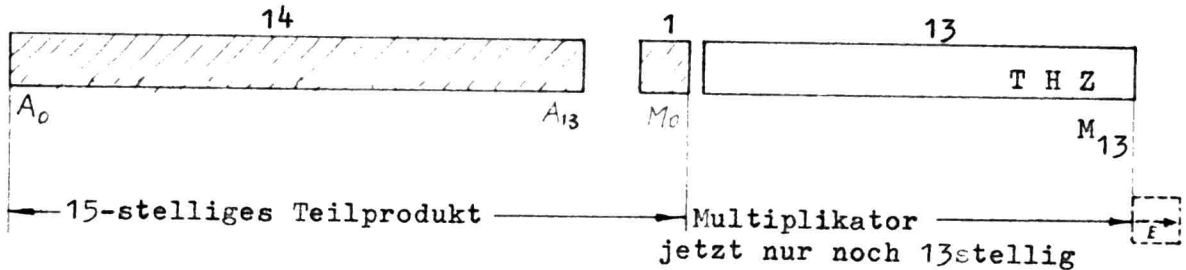
Multiplikator 14stellig



A

M

Die ganze Zahl A, M wird beim 15. Schritt (Übertrag $\rightarrow A_0$) um einen Schritt nach rechts geschoben

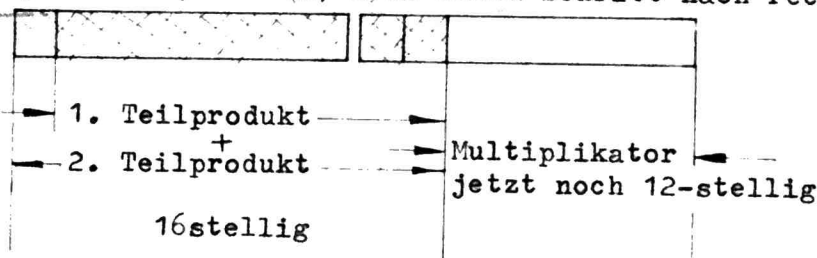


Damit erhält man das 15-stellige Teilprodukt $M_{13} \cdot N$ in A von A_0 bis A_{13} und in M_0 .

Der Multiplikator ist nun noch 13stellig, die letzte Ziffer in M, die nicht mehr zum Multiplizieren gebraucht wird, ist durch das Schieben verschwunden. In M_{13} steht die nächsthöhere Ziffer, mit der im 2. Schritt multipliziert wird.

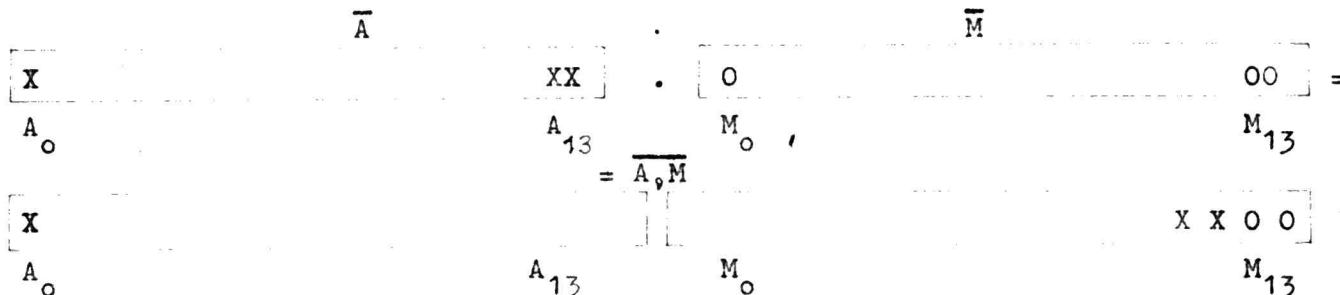
2. Schritt: $M_{12} \cdot N + A \rightarrow A, M_{0:1}$

Wie beim ersten Schritt wird die Zahl, die in der Anfangsstellung in M_{12} stand, der Reihe nach Ziffer für Ziffer mit N multipliziert, beginnend mit N_{13} , endend mit N_0 . Das erste Teilprodukt wird laufend addiert. Beim 15. Schritt, in dem der Übertrag der letzten Ziffernmultiplikation geschieht, wird A, M um einen Schritt nach rechts geschoben.



3. bis 14. Schritt

Stellenweise Multiplikation und Addition der Teilprodukte. Zum Schluß steht in $(\overline{A}, \overline{M})$ das fertige Produkt.



Das Produkt hat mindestens 2 Nullen in den niedrigsten Stellen, da $M_{12}, M_{13} \equiv 0$ vor Beginn der Operation, außerdem war $M_0 \equiv 0$; A_0 kann durch Schieben oder logisches Einschreiben $\neq 0$ gewesen sein.

4. Abschnitt

Normalisieren (nach rechts oder links), Schritte zählen, Charakteristik anpassen. Siehe "Normalisieren einer GK-Zahl". Dabei im Gleitkomma Exponentenrechenwerk: (a)-(Zähler) \rightarrow a; (a) - 11 \rightarrow m.

Trennen in abspeicherbare Teile.

n	i	38
---	---	----

Division

$$(\overline{A}, \overline{M}) : (n, n + 1) \rightarrow \overline{A} \text{ und Rest } \rightarrow \overline{M}$$

Vor Ausführung der Division wird der Inhalt von $\overline{A}, \overline{M}$ um einen Schritt nach rechts geschoben und gleichzeitig der in einem Eingangsregister N (dieses Register ist ein internes Hilfsregister und einer Programmierung nicht zugänglich) stehende Divisor normalisiert. Auf diese Weise wird verhindert, daß der Quotient größer als eins wird und die Maschine auf Divisionsstop läuft. Im übrigen verläuft die Division in der Betriebsart "Gleitkomma" analog zur Betriebsart "Festkomma, lang". Das Ergebnis der Division steht in \overline{A} , der Rest in \overline{M} . Anschließend an die Division läuft wie bei allen arithmetischen Gleitkommaoperationen ein Normalisiervorgang ab, der sich hierbei aber nur auf (A) bezieht, während der Rest in M unverändert bleibt.

Division in Gleitkomma

n	i	3 8
---	---	-----

Division: $(\overline{A}, \overline{M}) : (n, n+1) \rightarrow \overline{A}$; Rest $\rightarrow \overline{M}$

Zahlenfaktoren: 22stellig: 11stellig \rightarrow 11-stellig,
Rest 11-stellig

1. Abschnitt: Faktoren bereitstellen.

Der Dividend muß vor Beginn der Operation in $\overline{A}, \overline{M}$ stehen. Dabei kann von der vorangegangenen Operation her noch $A_0 \neq 0$ sein.

Der Divisor läuft von $n, n+1$ nach Hilfsregister N ein. Zuerst die vordere Hälfte der Zahl, mit der niedrigsten Stelle zuerst: $(n) \rightarrow N_1$, das Vorzeichen wird abgespalten und ins Vorzeichenregister gebracht, statt dessen in N_0 eine Null eingedrückt, dann folgt die hintere Hälfte der Zahl: $(n+1) \rightarrow N_2$. Dabei wird die Charakteristik im Eingangsregister e abgesetzt: $n_{12, 13} \rightarrow e$; dafür laufen in $N_{12, 13}$ Nullen ein.

Die Zahl $(\overline{A}, \overline{M})$ wird zusammengefügt (s. Absatz: Zusammenfügen von Gk-Zahlen).

2. Abschnitt: Faktoren zurechtrücken

Die Division kann richtig nur ablaufen, wenn (wie bei Festkomma) der Dividend kleiner als der Divisor ist: $\frac{D'_{end}}{D'_{or}} < 1$. Deshalb wird der Dividend in A, M um eine Stelle nach rechts geschoben. Damit wird sein Zahlenfaktor um 10 kleiner

Der Divisor kommt aus dem Speicher, ist also bestimmt ohne Überlauf, $N_0 \equiv 0$, muß aber noch nicht normalisiert sein. Deshalb wird jetzt der Divisor in N links normalisiert, die Schritte werden im Zähler gezählt.

Damit kann der Divisor höchstens 0, 9 9 ... 9 werden.

Es wird erreicht, daß der Dividend (möglichst) immer kleiner als der Divisor ist.

Divisionsstop tritt dann auf, wenn durch Schieben oder andere Betriebsart in A_0 ein Überlauf stand

oder wenn der Zahlenfaktor des Divisors

= 0 ist.

Beispiel: Dividend in A: 6,xxx wird normalisiert (1 nach rechts schieben) 0,6 xxx; wenn der Divisor in $n, n+1 \rightarrow N = 0,5xxx$ ist, dann Divisionsstop.

3. Abschnitt Exponentenrechnung

Beim Dividieren werden die beiden Charakteristiken (die um 50 erhöhten Exponenten) subtrahiert, so daß das Charakteristik-Ergebnis um 50 zu klein wäre. Deshalb addiert man vorher noch 50. Außerdem muß man noch 1 addieren, da zwar der Zahlenfaktor des Dividenden um 1 Schritt nach rechts geschoben wurde, nicht aber die Charakteristik verändert wurde. Insgesamt wird also $(a) + 51 \rightarrow a$ ausgeführt.

Die Charakteristik m ist normalerweise um 11 kleiner als a : $a - 11 = m$. Vorher wurde aber der Zahlenfaktor von $\overline{A, M}$ um 1 nach rechts geschoben, ist also um den Faktor 10 kleiner. Die Charakteristik a ist um 1 zu klein, man braucht statt 11 nur noch 10 abzuziehen: $a - 10 \rightarrow m$; m ist die Charakteristik des Restes.

Im k -Zähler steht die Anzahl der Links-Normalisierschritte für das \underline{N} -Register (\underline{N} enner = Divisor). Diese Anzahl wird zur Charakteristik a addiert: $a + \text{Zä} \rightarrow a$, und man erhält die Quotienten-Charakteristik.

Beispiel: Exponentenrechnung

a) Exponenten-Darstellung

$$\begin{array}{rcl}
 0, x \cdot 10^{+4} & : & 0,000 x \cdot 10^{18} \\
 \downarrow \quad \downarrow & & \downarrow \quad \downarrow \\
 .10^{-1} \cdot 10^{+1} & : & .10^{+3} \cdot 10^{-3} \\
 \hline
 = 0,0 x \cdot 10^5 & : & 0, x \cdot 10^{15} = 0, x \cdot 10^{5-15} = 0, x \cdot 10^{-10}
 \end{array}$$

b) Charakteristik-Darstellung

$$\boxed{0, x \dots \quad 54} : \boxed{0,000 x \dots \quad 68} = \boxed{0, x \quad 40}$$

$= a$ $\text{Zä}=3$ $= n$ Quotient $= a$

- 1) $n = 68 \rightarrow e$
- 2) $a - 10 = 44 \rightarrow m$ (Rest-Charakteristik)
- 3) $a + 51 = 105 \rightarrow a$
- 4) $a - e = 105 - 68 = 37 \rightarrow a$
- 5) $a + \text{Zä} = 37 + 3 = 40 \rightarrow a$ (Quotienten-Ch)

$$+ \boxed{0, x \quad 44}$$

Rest $= m$

4. Abschnitt: Division der Zahlenfaktoren.

Die Zahlenfaktoren werden wie Festkommazahlen dividiert. Die Quotientenziffern entstehen, durch das Rechenverfahren bedingt, in der Schutzstelle des Dividenden im Multiplikatorregister.

5. Abschnitt: Vertauschung von Quotient und Rest.

Am Ende der Division der Zahlenfaktoren steht in A der Rest, in M der Quotient. Durch Ringschieben werden (A) und (M) vertauscht:



6. Abschnitt:

Nur (A) wird normalisiert, nicht aber der Rest (M).

Der Rest kann ein anderes Vorzeichen besitzen als der Quotient, nämlich immer das des Dividenden.

Umgekehrte Division



$$(n, n+1) : (\bar{A}) \rightarrow \bar{A}, \text{ Rest} \rightarrow \bar{M}$$

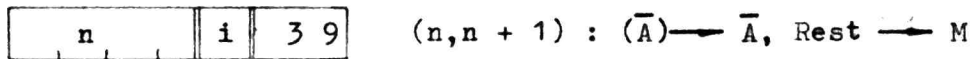
Vor der Division wird das Multiplikatorregister mit Nullen gefüllt. Nach Ablauf der Division geht der Rest in das Multiplikatorregister. Da dieser Divisionsrest durch Hinzunahme von Nullen zur Verlängerung des 11-stelligen Dividenden entstanden ist, wird er sinnvollerweise nur zum Runden der

Zahlenfaktors

11stelligen **Mantisse** des im Akkumulator stehenden Quotienten verwendet, so daß schließlich ein betragskleinster Rest entsteht.

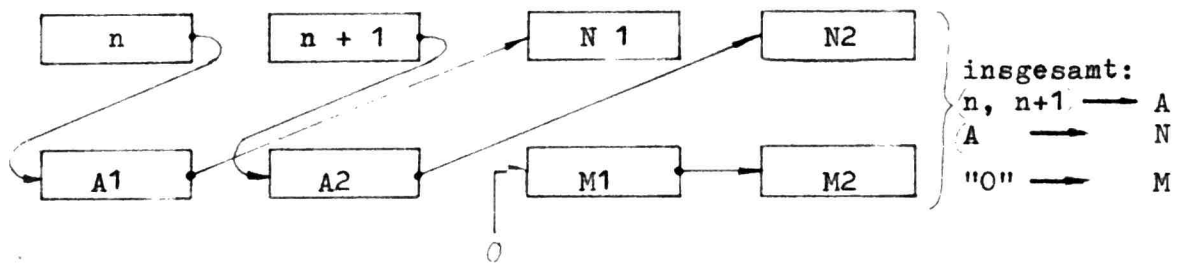
Diese Rundung ist in der Betriebsart "Festkomma, normal" bereits ausgeführt.

Umgekehrte Division bei Gleitkomma



Zahlenfaktoren: 11 : 11 \rightarrow 11-stellig, Rest: 11-stellig

Bei der umgekehrten Division werden nur zu Beginn der Operation die Operanden in umgekehrte Register gebracht wie bei der Division, sonst ist der Ablauf der Operationen gleich.



Überlauf im Divisor verboten.

Exponentenregister - Befehle

4	-	k		i		2	1
---	---	---	--	---	--	---	---

 Einschreiben $n_3, n_4 \rightarrow a$

Die in $n_3 n_4$ stehende zweiziffrige Zahl k wird in das Exponentenregister a des Akkumulators gebracht. Der Inhalt von $n_3 n_4$ bleibt erhalten.

Anwendung: Aus einer im Akkumulator stehenden Festkommazahl kann eine Gleitkommazahl erzeugt werden. Bringt man die zur Festkommazahl gehörige Charakteristik (in $n_3 n_4$) nach a , so steht die gewünschte Gleitkommazahl für arithmetische Operationen zur Verfügung.

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

 Einschreiben $(A_{12} A_{13}) \rightarrow a$

Die in $A_{12} A_{13}$ stehende Zahl wird in das Exponentenregister a des Akkumulators gebracht. Der Inhalt von $A_{12} A_{13}$ bleibt erhalten.

6	-	-	-		i		2	1
---	---	---	---	--	---	--	---	---

 Abspeichern $(a) \rightarrow J_{i;3,4}$ und Null $\rightarrow J_{i;1,2}$

Der Inhalt des Exponentenregisters a wird in die 3. und 4. Stelle und Nullen in die 1. und 2. Stelle des angegebenen Indexregisters übertragen. Der Inhalt des Exponentenregisters a bleibt ungeändert.

Man beachte die Bemerkungen zu Befehl 21,0 (Anzahl der Links-Normalisierungsschritte $\rightarrow J_i$), die auch für den Befehl 21,6 gelten.

Anwendung: Umrechnung der Charakteristik im Adressenrechenwerk.

7	-	-	-		i		2	1
---	---	---	---	--	---	--	---	---

 Abspeichern $(a) \rightarrow A_{12} A_{13}$, Null $\rightarrow A_0$ bis A_{11}
Akkumulatorvorzeichen plus

Der Inhalt des Exponentenregisters a wird nach $A_{12} A_{13}$ übertragen. Gleichzeitig wird das Vorzeichenregister von A auf plus gesetzt und in A_0 und A_{11} Nullen eingefüllt. Der Inhalt des Exponentenregisters a bleibt erhalten.

Anwendung: Umrechnung der Charakteristik im Rechenwerk.

Befehl 21,7 gilt als arithmetische Operation, so daß mit den Befehlen 18,1 bis 18,4 Merk-Flipflops vom Operationsergebnis abhängig gemacht werden können.

8	-	n_3	n_4	i	2	1
---	---	-------	-------	---	---	---

Charakteristiken

vergleichen: (a) mit n_3, n_4

und Vergleichsmarkierungen setzen (Auswertung durch die Befehle 11 bis 14). Der Inhalt des Exponentenregisters a wird mit der zweiziffrigen Zahl im Adressenteil $n_3 n_4$ verglichen. Abhängig vom Vergleichsergebnis $((a) - n_3, n_4)$ werden Vergleichsmarkierungen gesetzt, die mit den Befehlen 11 bis 14 zur Programmverzweigung ausgenutzt werden können. Die Vergleichsmarkierungen bleiben bis zum nächsten Vergleichsbefehl gespeichert. Der Inhalt von a bleibt durch den Befehl 21,8 ungeändert. Vergleichsbefehle sind die Befehle 21,8; 21,9; 25 bis 29 sowie 88, 87, 98 und 97.

Charakteristiken

9	-	-	-	i	2	1
---	---	---	---	---	---	---

vergleichen: (a) mit (A_{12}, A_{13})

und Vergleichsmarkierungen setzen (Auswertung durch die Befehle 11 bis 14). Der Inhalt des Exponentenregisters a wird mit der in $A_{12} A_{13}$ stehenden zweiziffrigen Zahl verglichen. Der Inhalt von a und der Inhalt des Akkumulators A_0 bis A_{13} bleiben ungeändert. Der Inhalt des Vorzeichenregisters wird in den Vergleich nicht einbezogen. Abhängig vom Vergleichsergebnis $((a) - (A_{12}, A_{13}))$ werden Vergleichsmarkierungen gesetzt. Siehe dazu die Erläuterungen zu Befehl 21,8.

Vergleichsbefehle für \bar{A}

28 vgl. (\bar{A}) mit (n,n+1)

29 vgl. (\bar{A}) " (n,n+1)

Vergleicher

Der Vergleicher wird sowohl mit den Ergebnisregistern A und a des Rechenwerkes als auch mit dem Kommandowerk zusammen gebraucht. Insbesondere dient er zum Vergleich des Akkumulatorinhalts mit dem Inhalt von Kernspeicherzellen, ohne den Akkumulatorinhalt zu zerstören, also ohne Subtraktion. Die Zahlen werden von der niedrigsten Stelle beginnend ziffernweise verglichen, wobei das jeweilige Ergebnis das Ergebnis der vorlaufenden Ziffer korrigiert und ständig 2 Flipflops steuert, eins für die Angabe =/≠, eins für \geq /<. Beide Flipflops werden vor Beginn des Vergleichs auf = bzw. \geq gesetzt.

Die Vergleichsergebnisse dienen als Sprungbedingungen, sie werden am Bedienungsfeld angezeigt.

n	i	28
---	---	----

 (\bar{A}) mit (n,n+1) vergleichen

Beim Vergleich mit dem Inhalt der beiden Speicherzellen wird der Inhalt des Vorzeichenregisters und der Inhalt von A_0 bis A_{11} sowie die im Exponentenregister a stehende Charakteristik herangezogen. Die zum Vergleich herangezogenen Zahlen bleiben unverändert. Zunächst werden die Charakteristiken verglichen. Bei ungleichen Charakteristiken werden die Zahlenfaktoren nur auf Normalisierung geprüft. Ist durch den Vergleichsbefehl, also ohne Veränderung des Akkumulatorinhalts, keine Aussage zu gewinnen (ungleiche Charakteristik, beide $\neq 0$, nicht normalisierte Zahlenfaktoren, gleiche Vorzeichen), so werden sämtliche vier Vergleichsmarkierungen (=, \neq , \geq , <) zugleich gesetzt. In diesem Fall kann der Vergleich über eine Subtraktion durchgeführt werden.

Man beachte: Trägt der Inhalt der zum Vergleich herangezogenen Speicherzellen eine Markierung (α -, ω - oder Q-Zeichen), so wird gleichzeitig mit der Durchführung des Vergleichs das entsprechende Eingangs-Flipflop beim Rechenwerk gesetzt.

n	i	29
---	---	----

|(\bar{A})| mit |(n,n+1)| vergleichen

Beim Vergleich mit dem Inhalt der beiden Speicherzellen werden der Inhalt von A_1 bis A_{11} sowie die im Exponentenregister a stehende Charakteristik herangezogen. Die zum Vergleich herangezogenen Zahlen bleiben unverändert. Man beachte die Bemerkungen bei Befehl 28 über Normalisierungen im Gleitkomma-Vergleich und über Markierungen.

Links schieben

20,1	(A) ◁	$k \leq 12$	V_z u. Ch. von A bleiben
20,2	(M) ◁	$k \leq 14$	" " M "
20,3	($\overline{A, M}$) ◁	$k \leq 25$	" " M " " " M \Rightarrow " " A Null $\Rightarrow A_{12}, A_{13}, M_0$

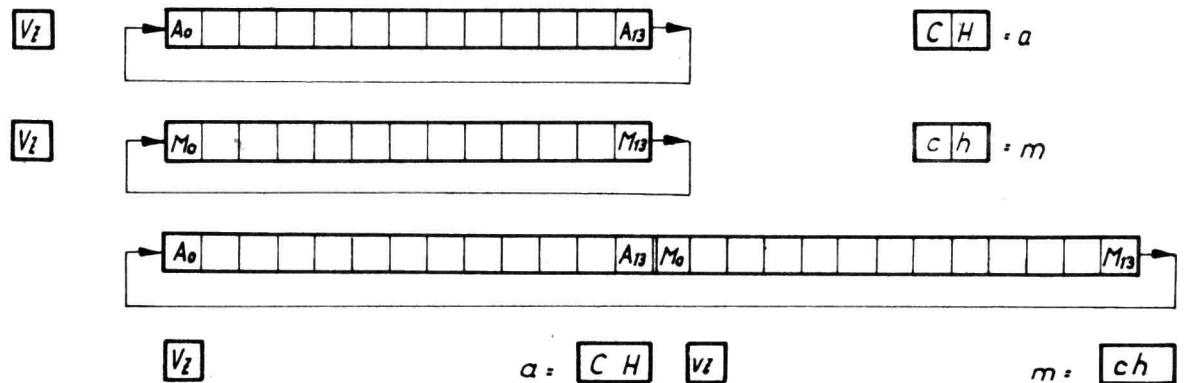
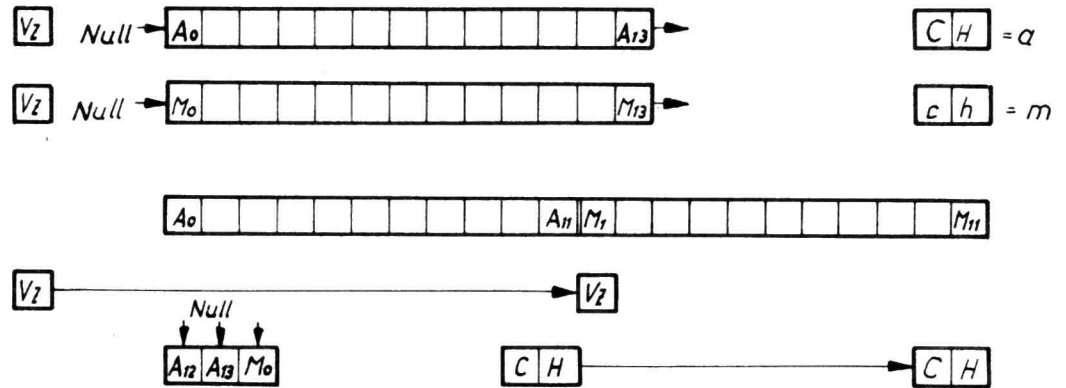
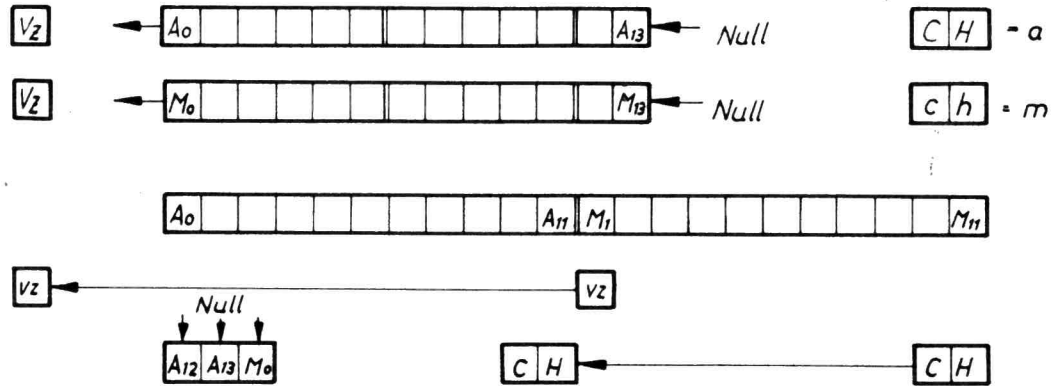
Rechts schieben

20,4	(A) ▷	$k \leq 14$	V_z u. Ch von A bleiben
20,5	(M) ▷	$k \leq 14$	" " M "
20,6	($\overline{A, M}$) ▷	$k \leq 25$	" " A " " " A \Rightarrow " " M Null $\Rightarrow A_{12}, A_{13}, M_0$

Im Ring schieben

20,7	(A) ↻	$k \leq 13$	V_z u. Ch. von A bleiben
20,8	(M) ↻	$k \leq 13$	" " M "
20,9	(AM) ↻	$k \leq 27$	" " M " " " A "

**Gleitkomma - Beschreibung
Schiebebefehle**



Schiebebefehle bei Betriebsart Gleitkomma.

Links schieben

1	-	k		i		2	0
---	---	---	--	---	--	---	---

(A) um k Stellen nach links schieben

Der Inhalt von A_0 bis A_{13} wird um die durch k festgelegte Stellenzahl nach links geschoben. Die über A_0 hinausgeschobenen Ziffern gehen verloren. Von rechts her werden Nullen eingefüllt, angefangen bei A_{13} . Das Vorzeichen und die Charakteristik a bleiben erhalten. Es ist nur eine Angabe $k \leq 12$ sinnvoll. Bei $k > 12$ erhält A nur noch Nullen.

2	-	k		i		2	0
---	---	---	--	---	--	---	---

(M) um k Stellen nach links schieben

Der Inhalt von M_0 bis M_{13} wird um die durch k festgelegte Stellenzahl nach links geschoben. Die über M_0 hinausgeschobenen Ziffern gehen verloren. Von rechts her werden Nullen eingefüllt, angefangen bei M_{13} . Das Vorzeichen von M und die Charakteristik m bleiben erhalten. Es ist nur eine Angabe $k \leq 14$ sinnvoll. bei $k > 14$ stehen im M lauter Nullen.

3	-	k		i		2	0
---	---	---	--	---	--	---	---

(A,M) um k Stellen nach links schieben

Der Inhalt von A_0 bis A_{11} und M_1 bis M_{13} wird um die durch k festgelegte Stellenzahl nach links geschoben. Die über A_0 nach links hinausgeschobenen Ziffern gehen verloren. Die in M_1 stehende Ziffer wechselt nach A_{11} über und von rechts her werden Nullen eingefüllt, beginnend bei M_{13} . Das Vorzeichen im Vorzeichenregister von M und die Charakteristik im Exponentenregister m werden beim Schieben in das Vorzeichenregister von A und das Exponentenregister a übertragen, unabhängig von der Anzahl der Schiebeschritte. Der Inhalt des Vorzeichenregisters von M und der Inhalt des Exponentenregisters m bleiben erhalten. Nach dem Schiebevorgang stehen in A_{12} , A_{13} und M_0 Nullen. Es ist nur eine Angabe von $k \leq 25$ sinnvoll. Bei $k > 25$ stehen in A,M nur noch Nullen.

Rechts schieben GLK

4	-	k		i		2	0
---	---	---	--	---	--	---	---

(A) um k Stellen nach rechts schieben

Der Inhalt von A_0 bis A_{13} wird um die durch k festgelegte Stellenzahl nach rechts geschoben. Die über A_{13} hinausgeschobenen Ziffern gehen verloren. Von links her werden Nullen eingefüllt, beginnend bei A_0 . Das Vorzeichen und die Charakteristik a bleiben erhalten. Es ist nur eine Angabe von $k \leq 14$ sinnvoll. Bei $k > 14$ stehen in A lauter Nullen.

5	-	k		i		2	0
---	---	---	--	---	--	---	---

(M) um k Stellen nach rechts schieben

Der Inhalt von M_0 bis M_{13} wird um die durch k festgelegte Stellenzahl nach rechts geschoben. Die über M_{13} hinausgeschobenen Ziffern gehen verloren. Von links her werden Nullen eingefüllt, beginnend bei M_0 . Das Vorzeichen von M und die Charakteristik m bleiben erhalten. Es ist nur eine Angabe von $k \leq 14$ sinnvoll. Bei $k > 14$ stehen in M lauter Nullen.

6	-	k		i		2	0
---	---	---	--	---	--	---	---

(A,M) um k Stellen nach rechts schieben

Der Inhalt von A_0 bis A_{11} und M_1 bis M_{13} wird um die durch k festgelegte Stellenzahl nach rechts geschoben. Die über M_{13} hinausgeschobenen Ziffern gehen verloren. Die in A_{11} stehende Ziffer wechselt nach M_1 über (A_{12} , A_{13} und M_0 werden übergangen) und von links her werden, beginnend bei A_0 , Nullen eingefüllt. Das Vorzeichen im Vorzeichenregister von A und die Charakteristik im Exponentenregister a werden beim Schieben in das Vorzeichenregister von M und das Exponentenregister m übertragen, unabhängig von der Anzahl der Schiebeschritte. Der Inhalt des Vorzeichenregisters von A und des Exponentenregisters a bleiben erhalten. Nach dem Schiebevorgang stehen in A_{12} , A_{13} und M_0 Nullen. Es ist nur eine Angabe von $k \leq 25$ sinnvoll. Bei $k > 25$ stehen in A,M nur noch Nullen.

Im Ring schieben GLK

7	-	k		1		2	0
---	---	---	--	---	--	---	---

(A) um k Stellen nach rechts im Ring schieben

Der Inhalt von A_0 bis A_{13} wird um die durch k festgelegte Stellenzahl nach rechts im Ring geschoben. Die über A_{13} nach rechts hinausgeschobenen Ziffern laufen in A_0 ein. Es gehen keine Ziffern verloren. Das Vorzeichen und die Charakteristik a bleiben erhalten. Es ist nur eine Angabe $k \leq 13$ sinnvoll.

8	-	k		1		2	0
---	---	---	--	---	--	---	---

(M) um k Stellen nach rechts im Ring schieben

Der Inhalt von M_0 bis M_{13} wird um die durch k festgelegte Stellenzahl nach rechts im Ring geschoben. Die über M_{13} nach rechts hinausgeschobenen Ziffern laufen in M_0 ein. Es gehen keine Ziffern verloren. Das Vorzeichen und die Charakteristik in m bleiben erhalten. Es ist nur eine Angabe $k \leq 13$ sinnvoll.

9	-	k		1		2	0
---	---	---	--	---	--	---	---

(A,M) um k Stellen nach rechts im Ring schieben

Der Inhalt von A_0 bis A_{13} und M_0 bis M_{13} wird um die durch k festgelegte Stellenzahl^x Die über M_{13} nach rechts hinausgeschobenen Ziffern laufen nach A_0 ein; die in A_{13} stehende Ziffer wechselt beim Schieben nach M_0 über. Die Vorzeichen in den Vorzeichenregistern von A und M sowie die Charakteristiken in a und m bleiben erhalten. Es ist nur eine Angabe von $k \leq 27$ sinnvoll.

^x nach rechts im Ring geschoben

Runden GLK

2	-	k	i	2	1
---	---	---	---	---	---

Runden auf k Stellen hinter dem Komma.

Die in $(\overline{A, M})$ stehende Zahl wird auf die durch k festgelegte Stellenzahl gerundet. Es schließt sich M_1 an A_{11} an.

Vz	A ₁	2	3	4	5	6	7	8	9	10	11	M ₁	2	3	4	5	6	7	8	9	10	11
----	----------------	---	---	---	---	---	---	---	---	----	----	----------------	---	---	---	---	---	---	---	---	----	----

Vorzeichen

Ziffernwert

Zahlenfaktor

CH
a

Charakteristiken

ch
(m) = (a) - 11

Beispiel: k = 16

Runden auf 16 Stellen heißt hier, daß eine 5 in M_6 addiert und anschließend in M_6 bis M_{13} Nullen eingefüllt werden. Entsteht durch das Runden ein Überlauf, so wird dieser beseitigt unter Erhöhung der Charakteristik um 1. Es ist nur eine Angabe von $k \leq 24$ sinnvoll. Es wird nicht nach links normalisiert, wohl aber nach rechts (um Überlauf zu beseitigen).

Normalisieren GLK

0	-	-	-	i	2	0
---	---	---	---	---	---	---

(A, M) normalisieren mit Schrittzählung

Unter Normalisieren wird ein Schiebevorgang verstanden, der den Zahlenfaktor der Zahl $(\overline{A, M})$ in den Bereich von 0,1 bis 1 bringt. Das Normalisieren beseitigt auch einen durch arithmetische Operationen oder durch Schieben entstandenen Überlauf. Die Anzahl der Normalisierungsschritte ist auf 25 beschränkt. Dadurch wird ggf. auch die letzte Ziffer (die 24.), wenn sie auch eine Null ist, 2 Stellen links über die erste Akkumulatorstelle A_1 hinausgeschoben.

Der Inhalt von A_0 bis A_{11} und M_1 bis M_{13} wird in der angegebenen Weise geschoben und die Schrittzählung durchgeführt. Zusätzlich hierzu findet ein Anpassen der Charakteristik statt, so daß der Betrag der Zahl durch das Normalisieren nicht geändert wird. Nach dem Normalisiervorgang stehen in A_{12} , A_{13} und M_0 Nullen.



Normalisier - Flipflop



Anzahl der Linksnormalisierschritte

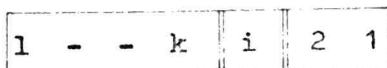
→ $J_{i; 3,4}$ und Nullen → $J_{i; 1,2}$

Der Inhalt des Zählers, der die ^xLinks-Schiebeschritte nach dem Normalisieren im Anschluß an jede Gleitkommaoperation und nach Ausführung des Normalisierbefehls 20,0 enthält, wird in die dritte und vierte Stelle des Indexregisters i übertragen; die erste und zweite Stelle des Indexregisters werden Nullen eingefüllt. x Anzahl der

Man beachte:

Der Index i wird im Befehl 21,0 für zwei Funktionen gebraucht. Zunächst findet eine Adressenumrechnung mit Hilfe des angegebenen Indexregisters statt. Soll keine Adressenumrechnung vor der eigentlichen Ausführung des Befehls 21,0 erfolgen, so muß dafür gesorgt werden (eventuell durch einen entsprechenden Vorbefehl), daß im angegebenen Indexregister entweder eine Null oder zumindest eine Zahl enthalten ist, die die Null im Adressenteil n_4 des Befehls 21,0 nicht verändern kann.

Andererseits dient der Index i im Befehl 21,0 etwa wie in den Befehlen der 90er Gruppe als "zweite Adresse".



Normalisierprüfwert ab jetzt $k = n_4 \leq 10$

Als Normalisierprüfwert wird die Zahl k genommen.

Beim Ablauf aller nachfolgenden Normalisierbefehle und nach arithmetischen Gleitkommaoperationen wird eine Prüfung durchgeführt. Ist die Anzahl der Normalisierschritte größer oder gleich dem Prüfwert, so wird das Normalisier-Flipflop auf "ein" gesetzt, ist sie kleiner, so bleibt das Flipflop in seiner augenblicklichen Stellung. Der Wert $k = 0$ entspricht einem Normalisierprüfwert von 10.



Sprung, wenn das Normalisier-Flipflop auf "aus" steht und zurücksetzen, wenn es auf "ein" steht.

Die Maschine fährt mit dem Programm bei der angegebenen Adresse fort, wenn das Normalisier-Flipflop auf "aus" steht. Steht das Flipflop hingegen auf "ein", so wird im Programm, nachdem das Normalisier-Flipflop auf "aus" gesetzt wurde, mit dem unmittelbar folgenden Befehl fortgefahren. Das Normalisier-Flipflop wird auf "ein" gesetzt, wenn eine vorangegangene Normalisierung, die durch Befehl 20,0 ausgelöst oder im Anschluß an eine Gleitkomma-Operation automatisch abläuft, ein Schieben von genau oder mehr als k Plätzen zur Folge hatte.

Vgl. dazu die Erläuterungen zu Befehl 21,0 und 21,1.

Anwendung:

Der Befehl 17 erlaubt eine sofortige Programmverzweigung, wenn beim Normalisieren genau oder mehr als k Nullen von rechts eingezogen werden. Man kann so verhindern, daß mit Zahlen weitergerechnet wird, von denen nur noch wenige Ziffern richtig sind, bzw. sich wenigstens einen Überblick über die Genauigkeit der Rechnung verschaffen.

Überlauf - Flipflop

n		i	1	5
---	--	---	---	---

Sprung, wenn das Überlauf-Flipflop auf "aus" steht und zurücksetzen, wenn es auf "ein" steht.

Die Maschine setzt das Programm mit der angegebenen Adresse fort, wenn das Überlauf-Flipflop auf "aus" steht, d. h. wenn kein Überlauf aufgetreten ist. Steht hingegen das Überlauf-Flipflop auf "ein", so geht die Maschine zum unmittelbar nächsten Befehl im Programm weiter, nachdem das Überlauf-Flipflop auf "aus" gesetzt wurde.

Man beachte:

Das Überlauf-Flipflop wird nur beeinflusst durch Überlauf, der von arithmetischen Operationen einschließlich Runden (21,2) herrührt, also nicht durch "Überlauf", der durch Schieben oder logisches Einschreiben entsteht. Die Division kann keinen Überlauf erzeugen. Wird der berechnete Quotient größer gleich 1, so wird nicht das Überlauf-Flipflop beeinflusst, sondern die Maschine angehalten. Der bei Gleitkomma-Additionen kurzfristig auftretende Überlauf wird durch das evtl.

automatische Rechtsschieben sofort wieder rückgängig gemacht; das Überlauf-Flipflop wird trotzdem auf "ein" gesetzt. Außerdem wird der Rechtsnormalisierschritt durch negatives Zählen des im Befehl 20,0 erwähnten Schrittzählers angezeigt. Da durch Abfragen des Überlauf-Flipflops dieses aus einer "Ein"-Stellung wieder auf "Aus" gestellt wird, kann der Überlauf im Akkumulator weiterhin vorhanden sein, ohne daß das Überlauf-Flipflop eingeschaltet ist. Es wird erst erneut eingeschaltet, wenn eine Betragserhöhung der ersten Ziffer stattfindet, wenn z. B. aus der Zahl 1,3... durch Addition etwa die Zahl 2,2... wird; das Überlauf-Flipflop wird nicht eingeschaltet, wenn aus der Zahl 1,3... durch Addition beispielsweise die Zahl 1,4... entsteht.

n	i	2	2
---	---	---	---

(\bar{A}) \rightarrow n,n+1 Gleitkomma-Ausgabevorbereitung.

mit 10-stelligem Zahlenfaktor, Exponentenvorzeichen und zweistelligem Exponenten (nicht Charakteristik).

Nur in Gleitkomma sinnvoll. Eine modifizierte Gleitkommazahl Z_0 bis Z_{13} wird aus dem Akkumulator in zwei aufeinanderfolgende Speicherzellen übertragen:

1. Aus dem Vorzeichenregister von A geht das Vorzeichen nach Z_0 .
2. Aus dem Akkumulator A_1 bis A_{10} wird der 10-stellige Zahlenfaktor nach Z_1 bis Z_{10} gebracht.
3. Aus dem Exponentenrechner wird der Exponent mit Vorzeichen (Z_{11} = Vorzeichen des Exponenten, Z_{12}, Z_{13} = Exponent) gebracht. Der Exponent wird intern im Exponentenrechner aus der um 50 verminderten Charakteristik gebildet.

Der Inhalt von \bar{A} bleibt unverändert, ebenso der des Exponentenregisters a.

Anwendung: Diese Art der Ausgabe von Gleitkommazahlen ist notwendig für das Ausdrucken von Ergebniswerten mit Exponent. Allerdings verliert man dabei eine Dezimale, denn das Vorzeichen des Exponenten tritt an die Stelle der 11. Dezimale des Zahlenfaktors. (An dieser Stelle muß die Formatsteuerung ein "Vorzeichen" lesen.) Das ist jedoch im allgemeinen kein echter Verlust, weil die Zuverlässigkeit der letzten Dezimalen bei Gleitkommazahlen ohnedies fraglich ist. Beim Normalisieren im Anschluß an Gleitkommaoperationen, vgl. Befehl 20,0, werden nämlich viel-

fach Ziffern aus M nach A übernommen, denen keine arithmetische Bedeutung zukommt.

Den umgekehrten Befehl, also die Eingabe von Zahlen mit Exponent und Vorzeichen anstelle der Charakteristik gibt es nicht. Gleitkommazahlen werden entweder mit Charakteristik eingegeben, oder man rechnet sie blockweise nach beendeter Eingabe oder auch einzeln unmittelbar vor der Verwendung um.

2	-	-	-	i	19
---	---	---	---	---	----

Ab jetzt Betriebsart Gleitkomma.

Übergang in Betriebsart Festkomma durch:

0	-	-	-	i	19
---	---	---	---	---	----

Ab jetzt Festkomma normal

1	-	-	-	i	19
---	---	---	---	---	----

Ab jetzt Festkomma lang

Nachtrag zu Blatt GK 17:

Wird durch Addition (A_0) erhöht (Überlauf), so wird das Überlauf-Flipflop auf "ein" gesetzt, auch wenn durch das nachfolgende Normalisieren der Überlauf wieder beseitigt wird. Das Überlauf-Flipflop wird mit Befehl 15 abgefragt, das Normalisier-Flipflop mit Befehl 17 (s. dort).