

# ELEKTRONISCHES RECHENGERÄT

Z 22

II. DER INTERN-CODE

AUSGABE JANUAR 1958

ZUSE K.G.  
BAD HERSFELD

V O R W O R T

=====

Das vorliegende Heft soll den Benutzern des elektronischen Rechengerätes Z 22 eine Einführung in den Intern-Code geben. Da die Befehlsstruktur der Z 22 eine sehr große Zahl von Befehlskombinationen erlaubt, können hier nur einige Beispiele erläutert werden. Doch hoffen wird, daß es nach Studium dieser Seiten möglich ist, die Wirkung jeder beliebigen Befehlskombination zu erklären.

Bei den Befehlen wird die Form gewählt, die man auch am Bedienungspult eintasten kann, wobei allerdings feste Adressen in dezimaler Darstellung angegeben werden. Nur bei einigen Kombinationen wird neben der internen Schreibweise auch eine abgekürzte (externe) benutzt. Das Leseprogramm des Freiburger Codes liest beide Formen ein.

Es handelt sich um die folgenden Abkürzungen:

S als Abkürzung für AS (Vergl. 3.5)

L als Abkürzung für LLR (Vergl. 3.5)

Das Befehlszeichen K, es soll nur die Schnellspeicheradresse wirksam sein, ist dort nicht mit angegeben, wo es vom Leseprogramm automatisch gesetzt wird (Vergl. 3.4).

Für die Wiederholungsbefehle wird die abgekürzte externe Schreibweise (z. B. F 16 - 3 für EPK 15 + 8189) gewählt, da man dieser Kombination besser ansehen kann, wie oft die Wiederholung erfolgen soll (Vergl. 4.1).

I N H A L T S V E R Z E I C H N I S

=====

I. Vorwort	I
Inhalt	II
1. Aufbau der Z 22	1
1.1. Leitwerk	1
1.2. Speicher	2
1.3. Rechenwerk	4
1.4. Ein- und Ausgabe	4
2. Zahlen und Befehlsdarstellung	5
2.1. Zahlendarstellung	5
2.2. Befehlsdarstellung	5
2.3. Symbolik	6
3. Erklärung der Befehlszeichen	8
3.1. Grundbefehle E, A, U, I	8
3.2. Befehlskreislauf	10
3.3. Befehlszeichen, die den Akkumulator be- treffen S, N, LL, L, R, V	11
3.4. Weitere Befehlszeichen F, C, K, H, Z,	16
3.5. Bedingungszeichen PP, QQ, P, Q, Y	20
3.6. Adressensubstitution: G mit Kombination	21
3.7. Löschungen in Befehlstexten (Zusammen- fassung von den schon erwähnten)	24
4. Zählungen und Verzweigungen	
4.1. Wiederholungsbefehle	25
4.2. Abzählen in Schleifen	26
4.3. Verzweigungen	27
5. Zugriffszeiten und deren Vermeidung	29

## 1. Aufbau der Z 22

=====

Bei der Konstruktion der Z 22 wurde Wert auf einen möglichst einfachen Aufbau gelegt.

So führt die Maschine nur ganz elementare Operationen (Verschieben, Speichern usw.) aus und setzt aus diesen die arithmetischen Operationen zusammen. Die Maschine enthält kein gesondertes Multiplikationswerk, sondern löst die Multiplikation durch eine Reihe von elementaren Befehlen.

Jede Dualstelle eines Befehls löst nur eine elementare Operation aus, daher ist eine Entschlüsselung der Befehlstexte nicht erforderlich. Die einzelnen elementaren Operationen können aber in fast beliebiger Weise miteinander kombiniert werden, wodurch über 10.000 sinnvolle Befehlsarten gebildet werden können.

Vor der Beschreibung der Wirkung der einzelnen Befehlszeichen soll zunächst eine Beschreibung des Aufbaus der Z 22 folgen.

Die Z 22 besteht aus 4 Hauptteilen:

1. Leitwerk
2. Speicher
3. Rechenwerk
4. Ein- und Ausgabe

1. Vom Leitwerk werden durch die einlaufenden Befehle die Operationen in der ganzen Maschine gesteuert. Das Leitwerk besteht aus 2 Flip-Flop-Registern und einer Kernspeicherzelle.

Das eine Flip-Flop-Register (Befehlsregister) nimmt den einlaufenden Befehl auf und gibt ihn parallel an das 2., das Steuerregister, weiter. Von jedem Flip-Flop des Steuerregisters werden eine Reihe von Gates gesteuert, so dass durch den im Steuerregister stehenden Befehl ein bestimmter Weg in der Maschine frei wird.

Der im Befehlsregister stehende Befehl geht über das +1-Addierwerk mit um 1' erhöhter Adresse ins Befehlszählregister (Kernspeicherzelle) vergl. Befehlskreislauf.

## 2. Speicher

Das Zentrum der Maschine ist eine Magnettrommel.

Die Trommel enthält auf 256 Spuren 8192 Speicherzellen, von denen jede ein Wort, d. h. eine Zahl oder einen Befehl aufnehmen kann. Die einzelnen Speicherzellen sind nummeriert, die Nummern sind die Adressen, unter denen diese Zellen aufgerufen werden können.

Die Anordnung der Zellen erfolgt nach einem bestimmten Schema. Je 256 Zellen sind zu einem Block zusammengefasst, alle Blöcke (0 bis 31) sind in ihrer Anordnung gleich, nur erhöhen sich die Adressen von Block zu Block um je 256.

Jeder Block ist in 8 Spuren mit je 32 Sektoren aufgeteilt, wobei jeder Sektor eine Speicherzelle enthält. Die Anordnung der Zellen erfolgt so, dass weitgehend Wartezeiten vermieden werden können (vergl. Kap. 5)

Zur Aufnahme von Zwischenwerten, die schnell wieder gebraucht werden, besitzt die Z 22 noch einen Schnell-speicher, dessen Zellen ohne Wartezeit zur Verfügung stehen. Die Kernspeicherzellen 2 bis 15 haben eine Speicherkapazität von 38 Dualstellen, können also je ein Wort aufnehmen.

Eigenschaften der Schnellspeicherzellen nach Adressen geordnet:

- 0 endet leer, liefert "0" in jeder Dualstelle
- 1 Liefert "1" in der obersten Stelle (Vorzeichen- bzw. Befehlskennzeichenstelle), in den anderen Stellen "0", das ist gleichbedeutend mit dem Befehl EC + 0.  
0 und 1 nehmen bei Speicherungen nichts auf.
- 2 Vorzeichentestregister <sup>x</sup>, sonst wie Speicher 6 bis 15
- 3 Testspeicherzelle für die unterste Stelle. Er kann mit dem Akkumulator zu gemeinsamen Verschiebungen verkoppelt werden. Einleitung einer '1' in der untersten Stelle bei H.

<sup>x</sup>Der Test bezieht sich nur auf die oberste Dualstelle.

- 4 der Akkumulator a mit vorgeschaltetem Addierwerk mit Intersektionseinrichtung und Komplementwerk Verzögerungsgliedern für Rechts- und Linksverschiebung, Vorzeichentest.
- 5 Rückkehradressenregister, wird bei F mit Inhalt des Befehlszählregisters c belegt. Sonst wie Speicherzellen 6 bis 15.
- 6-15 Zellen ohne Besonderheiten.
- 16=5 Die Speicherzelle 16 ist mit Zelle 5 identisch, kann unter der Adresse 16 aber nur gelesen werden.
- 17 In Abhängigkeit von einem Schalter kann unter Adresse 17 eine 1 in der untersten Dualstelle aufgerufen werden: z.B.
- B17 Schalter ein 1' → a  
Schalter aus 0 → a
- 20 Ein- und Ausgabepufferspeicher.

Kapazität: 5 Dualstellen, beim Lesen in den untersten, beim Speichern in den obersten 5 Stellen.  
Wird Schnellspeicher 20 gelesen, so wird dasjenige Fernschreibzeichen ausgeliefert, das sich gerade unter den Lesekontakten des Lochstreiferabtasters befindet. Anschliessend wird automatisch die nächste Fünfer-Kombination des Lochstreifens unter die Abtasterkontakte transportiert. Wird 20 kurz hintereinander gelesen, so dass der Lochstreifentransport des vorhergehenden Lesebefehls noch nicht beendet ist, wartet die Z 22 automatisch so lange, bis der Lochstreifen transportiert ist und führt anschliessend den neuen Lesebefehl durch. Der Lochstreifentransport dauert etwa 75 msec.

Wird im Schnellspeicher 20 gespeichert, so wird diese Fünfer-Kombination anschliessend automatisch als Fernschreibzeichen von der Schreibmaschine herausgeschrieben, bzw. wenn eingeschaltet, gleichzeitig auf den Locher gegeben. Wird 20 gelesen oder wird nach 20 gespeichert, so lange ein in 20 bereits befindliches Zeichen noch nicht vollständig herausgeschrieben ist, wartet die Z 22 bis dieses geschehen, um anschliessend automatisch den betreffenden Befehl auszuführen. Das Herausschreiben eines Zeichens dauert 160 msec.

19 Schnelleser , Schnellspeicher

3. Das Rechenwerk besteht aus dem Addierwerk, dem Akkumulator und den Verzögerungsgliedern.

Das Addierwerk kann Addition, Subtraktion (durch Addition des Komplements) und Intersektionen durchführen. Das Addierwerk nimmt in dem einen Eingang den aus dem Akkumulator kommenden Wert unverändert, 1 mal (L) oder 2 mal (LL) links, oder 1 mal (R) rechts verschoben oder nichts (N) auf, während in den zweiten Eingang die aus der aufgerufenen Speicherzelle kommende Zahl einläuft. Das Ergebnis wird wieder in dem Akkumulator gespeichert. Der Schnellspeicher 3 kann mit dem Akkumulator verkoppelt werden (V), so dass der Inhalt - ein Wort doppelter Stellenzahl darstellend - 1 mal links (LV) oder rechts (RV) verschoben werden kann. (vergl. Blockschema)

4. Ein- und Ausgabe

Die Eingabe erfolgt mittels Fünflochstreifen über handelsübliche Lochstreifenabtaster. Als Ausgabegerät dient eine Fernschreibmaschine, die die Resultate sofort in Tabellenform niederschreibt. Gleichzeitig können durch Anschalten eines Lochers die Ergebnisse auch in Lochstreifen gestanzt werden.

Die Ausgabefernschreibmaschine kann auch zur Lochstreifenherstellung benutzt werden, sofern nicht zur besseren Ausnutzung der Maschine weitere Fernschreibmaschinen aufgestellt werden.



Jeder Befehl enthält ein Operationsteil (mindestens 1 Buchstabe) und einen Adressteil (mindestens 1 Ziffer). Die einzelnen Operationsszeichen können fast beliebig untereinander zu sinnvollen Befehlen zusammengefasst werden. Jedem Befehl kann eine oder mehrere Bedingungen zugesetzt werden, und damit die Ausführung von der Erfüllung der Bedingungen abhängig gemacht werden.

Die Schnellspeicheradressen sind gegenüber den Trommelspeicheradressen um 13 Stellen nach links verschoben. Diese Verschiebung der Adressen erfolgt im Leseprogramm automatisch, sobald eine Adresse kleiner als 32 ist.

### 2.3. Symbolik

Vor der Beschreibung der Wirkung der einzelnen Befehlszeichen, geben wir eine Zusammenstellung der einzelnen verwendeten Symbole:

- a Akkumulator (= Schnellspeicher 4)
- b Befehlsregister, nimmt die einlaufenden Befehle auf, um sie durch das Steuerregister ausführen zu lassen
- c Befehlszählregister, enthält den zuvor vom Befehlsregister ausgeführten Befehl mit um 1' erhöhter Adresse.
- m,n allgemeine Adressen
- s Schnellspeicheradressen
- t Trommelspeicheradressen
- $\langle m \rangle$  Inhalt der Zelle, die unter Adresse m zu erreichen ist
- $\{y\}$  Adresse der Zelle, deren Inhalt = y ist.
- also  $\langle \{y\} \rangle = y$  und  $\{ \langle m \rangle \} = m$
- $\langle m \rangle_p \dots q$  Inhalt der Stellen p bis q der Speicherzelle m
- $\langle m \rangle_t$  Inhalt der Trommeladressenstellen der Zellen m, d. h. die Stellen 26 bis 38
- $\langle m \rangle_s$  Inhalt der Schnellspeicheradressenstellen der Zelle s, d. h. der Stellen 21 bis 25
- $\langle m \rangle_{38}$  Inhalt der letzten Stelle des Speichers m
- $\langle m \rangle \rightarrow n$  Transportsymbol: Der Inhalt der Zelle m geht nach Zelle n beim Ablesen eines Speichers bleibt die Information dort erhalten ( $\langle m \rangle \rightarrow m$ )
- $\dots \Rightarrow \dots$  "ergibt"; der links stehende Ausdruck enthält die Anweisung, wie aus schon bekannten Werten die rechtsstehende Grösse gebildet wird. (Gleichung mit Richtungssinn).

z. B.  $f(x) \Rightarrow y$  mit dem gegebenen  $x$  wird  $f(x)$  gebildet und ergibt den Funktionswert  $y$   
 $x + \Delta x \Rightarrow x$  das um die Schrittweite  $\Delta x$  erhöhte Argument  $x$  ergibt das neue Argument  
 $k + 1 \Rightarrow k$  der um 1 erhöhte Index  $k$  ergibt den neuen Index  $k$   
 $\langle m \rangle_t \Rightarrow t$  Inhalt der Trommelspeicheradressenstellen der Zelle  $m$  ergibt die Trommeladresse  $t$ .

$k'$  das  $k$ -fache der untersten Dualstelle.  
 $a \wedge b$  Konjunktion von  $a$  und  $b$  ( $a$  "und"  $b$ )

$a$	$b$	$a \wedge b$
0	0	0
1	0	0
0	1	0
1	1	1

Disjunktion von  $a$  und  $b$  ( $a$  "oder"  $b$ )

$a$	$b$	$a \vee b$
0	0	0
1	0	1
0	1	1
1	1	1

$\bar{a}$  Negation von  $a$  ("nicht"  $a$ )  
 $\overline{\bar{a}} = a$

Rechenregel:

$$\overline{(a \wedge b)} = \bar{a} \vee \bar{b}$$

$$\overline{(a \vee b)} = \bar{a} \wedge \bar{b}$$

Bei Auflösung einer Negation:  
 Konjunktion in Disjunktion verwandeln  
 und Grösse durch ihre negierten Werte  
 ersetzen.

### 3. ERKLÄRUNG DER BEFEHLSZEICHEN

#### 3.1. Grundbefehle

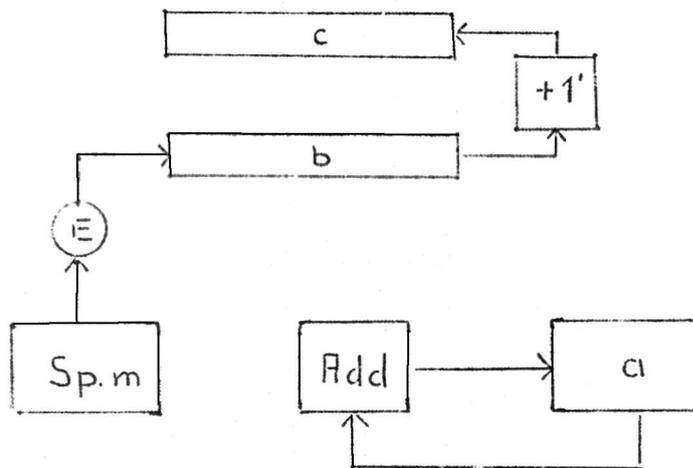
Die wichtigsten der 38 Dualstellen sind die Stellen 12 und 13, die mit U und A bezeichnet sind. Die vier möglichen Kombinationen dieser Stellen ergeben die Grundbefehle, während alle übrigen Operationszeichen Zusätze zu diesen sind.

Stelle 12	Stelle 13	bezeichnet mit:
0	0	E
0	1	A
1	0	U
1	1	UA = I

Em = Sprungbefehl

Wirkung:  $\langle m \rangle \rightarrow b$        $\langle b \rangle + 1' \rightarrow c$

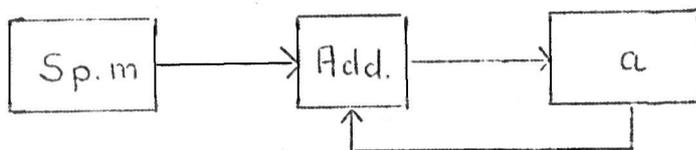
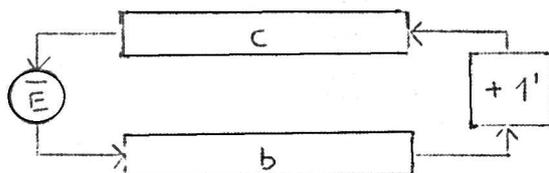
Es wird der Inhalt der Zelle m ins Befehlsregister geholt, gleichzeitig geht der im Befehlsregister stehende Befehl (nämlich Em) über das 1-Addierwerk mit um 1' erhöhter Adresse ins Befehlszählregister (vergl. Befehlskreislauf)



Am = Additionsbefehl

Wirkung:  $\langle a \rangle + \langle m \rangle \rightarrow a$        $\langle c \rangle \rightarrow b$

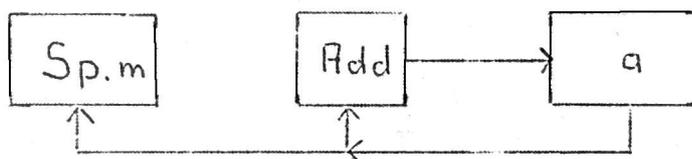
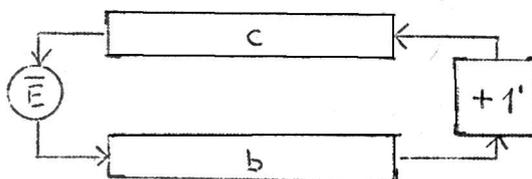
Der Inhalt der Zelle m wird zu dem Inhalt des Akkumulators addiert, das Ergebnis geht wieder in den Akkumulator. Gleichzeitig gelangt der Inhalt des Befehlszählregisters wieder in das Befehlsregister (vergl. Befehlskreislauf)



Um = Speicherbefehl

Wirkung:  $\langle a \rangle \rightarrow m$        $\langle a \rangle \rightarrow a$        $\langle c \rangle \rightarrow b$

Der Inhalt des Akkumulators wird in der Zelle m gespeichert, der Inhalt des Akkumulators bleibt erhalten und gleichzeitig gelangt der Inhalt des Befehlszählregisters ins Befehlsregister (Vergl. Befehlskreislauf)



Im = Intersektion

Wirkung:  $\langle a \rangle \wedge \langle m \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$

Es wird ~~du~~alstellenweise die Konjunktion von Akkumulatorinhalt und Inhalt der Zelle m gebildet, d. h. im Akkumulator erscheint nach der Intersektion nur in den Stellen eine Eins, in denen sowohl im Akkumulator als auch in der Speicherzelle m eine Eins stand, sonst Nullen.

Der Inhalt des Befehlszählregisters gelangt wieder gleichzeitig ins Befehlsregister. (vergl. Befehlskreislauf). Durch die Intersektion können beliebige Teile aus irgendwelchen Angaben herausgeschnitten werden.

z. B. Herausschneiden des Trommeladressenteils aus einem Befehl im Akkumulator.

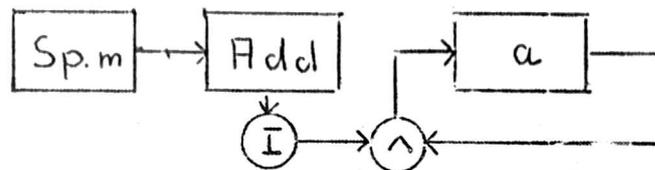
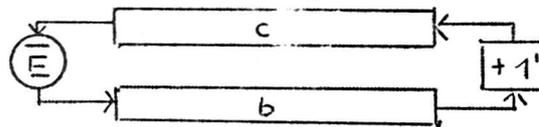
$\langle a \rangle$  E 536 = 100..... 000001000011000

$\langle m \rangle$  8191 = 000..... 0011111111111111

nach der Intersektion Im

$\langle a \rangle$  536 = 000..... 000001000011000

(vergl. CI-Befehle)



### 3.2. Befehlskreislauf

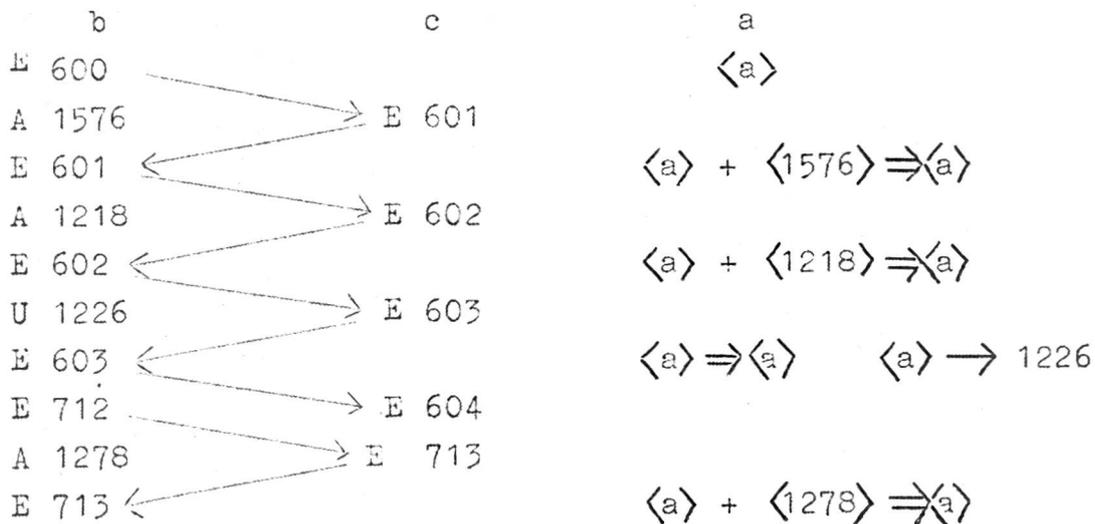
Ein Em-Befehl bewirkt einen linearen Programmablauf, d. h. die Ausführung der in den Zellen m, m + 1, m + 2 ...stehenden Befehle. Die Folge wird erst unterbrochen, wenn ein neuer E-Befehl aus einer der Zellen m + i ins Befehlsregister gelangt. Dieser ruft wiederum einen linearen Ablauf mit neuen Adressen hervor, d. h. der E-Befehl ist ein Sprungbefehl.

Ein Beispiel soll den Kreislauf Befehlsregister - Befehlszählregister verdeutlichen:

Zunächst sei die Speicherliste der Befehle gegeben. In der Speicherliste links die Adresse der Zelle und rechts daneben der Inhalt dieser Zelle. Rechts lässt man zweckmässig noch Platz für Bemerkungen über die Wirkung der Befehle.

<u>Adresse</u>	<u>Inhalt</u>
600	A1576
601	A1218
602	U1226
603	E712
⋮	
712	A 1278
713	...
714	...

In dem nun folgenden Schema wird der zeitliche Ablauf von oben nach unten dargestellt.



### 3.3. Befehlszeichen, die den Akkumulator betreffen

S als Zusatzzeichen zu A oder I bewirkt, dass die aus dem Speicher geholte Zahl negativ gemacht wird, ehe sie dem Rechenwerk zugeführt wird. Vom Leseprogramm wird bei dem Zeichen S automatisch AS gesetzt.

Sm =Subtraktion

Wirkung:  $\langle a \rangle - \langle n \rangle \rightarrow a$                        $\langle c \rangle \rightarrow b$

Vom Akkumulatorinhalt wird der Inhalt der Zelle m subtrahiert. Aus dem Befehlszahlregister gelangt gleichzeitig der nächste E-Befehl ins Befehlsregister.

ISM Intersektion mit dem negativen Inhalt der Zelle m

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

Der Inhalt der Zelle m wird negiert und mit diesem Wert wird die Intersektion gebildet.

N = Nullsetzen des Akkumulators

In Verbindung mit A vor der Addition:

NAm:  $\langle m \rangle \rightarrow a$                        $\langle c \rangle \rightarrow b$

In Verbindung mit U nach dem Speichern:

NUM:  $\langle a \rangle \rightarrow m$      $0 \rightarrow a$                        $\langle c \rangle \rightarrow b$

Wird das Zeichen N einem E-Befehl zugesetzt, so bewirkt es die Löschung des Akkumulators, es wird aber nicht mit ins Befehlszahlregister übernommen.

NEM:  $\langle m \rangle \rightarrow b$      $E_m + 1' \rightarrow c$      $0 \rightarrow a$

N in Verbindung mit I ist sinnlos, da die Intersektion mit der aus dem Akkumulator kommenden Null nur Null liefert.

Verschiebungen: Um die Verschiebungen in ihrer Wirkung genau beschreiben zu können, sei an dieser Stelle noch folgende Bezeichnung eingeführt, die jede einzelne Dualstelle berücksichtigt.

$\langle a \rangle = (\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{37}, \alpha_{38},)$  = Inhalt des Akkumulators

$\langle \zeta \rangle = (\zeta_1, \zeta_2, \zeta_3, \dots, \zeta_{37}, \zeta_{38},)$  = Inhalt von Schnellspeicherzelle 3

LL Linkverschiebung des Akkumulatorinhalts um 2 Dualstellen (=Multiplikation mit 4)

Die im Akkumulator stehende Zahl wird zweimal links verschoben dem Addierwerk zugeführt, in den beiden untersten Stellen werden Nullen nachgeschoben, während die obersten beiden Stellen herausgeschoben werden.

Bemerkungen:

a) Da die Speicherung direkt aus dem Akkumulatorausgang erfolgt, wird bei LLU der unverschobene Wert gespeichert.

$$\text{LLAm} \quad 4 \langle a \rangle + \langle m \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$$

$$\text{LLUm} \quad \langle a \rangle \rightarrow m \quad 4 \langle a \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$$

Die Wirkung der Verschiebung auf die einzelnen Bits ist die folgende:

$$\text{LLAm} \quad (\alpha_3, \alpha_4, \dots, \alpha_{38}, 0, 0) + \langle m \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$$

$$\text{LLUm} \quad (\alpha_4, \dots, \alpha_{38}) \rightarrow m, \quad (\alpha_3, \alpha_4, \dots, \alpha_{38}, 0, 0) \rightarrow a \quad \langle c \rangle \rightarrow b$$

b) Bei Verbindung mit E ist zu beachten, dass LL mit ins Befehlszählregister gelangt, also auch im weiteren Ablauf stets wieder wirkt.

$$\text{LLEm} \quad \langle m \rangle \rightarrow b \quad \text{LLEm} + 1' \rightarrow c \quad 4 \langle a \rangle \rightarrow a$$

Die Bemerkungen a) und b) gelten für R und L analog.

R Rechtsverschieben des Akkumulatorinhalts um 1 Dualstelle (=Multiplikation mit 1/2)

Der rechts verschobene Akkumulatorinhalt wird dem Addierwerk zugeführt. Da verschobene Zahlen nach der Verschiebung noch das gleiche Vorzeichen behalten sollen, wird bei Rechtsverschiebung in die oberste frei werdende Stelle das gleiche gesetzt, was vor der Verschiebung in der ersten Stelle gestanden hat.

$$\text{RAm} \quad 1/2 \langle a \rangle + \langle m \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$$

$$\text{RUm} \quad \langle a \rangle \rightarrow m \quad 1/2 \langle a \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$$

$$\text{REm} \quad \langle m \rangle \rightarrow b \quad \text{REm} + 1' \rightarrow c \quad 1/2 \langle a \rangle \rightarrow a$$

$$\text{RAm} \quad (\alpha_1, \alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{37}) + \langle m \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$$

LLR Die Kombination LLR bewirkt eine Linksverschiebung des Akkumulatorinhalts (Multiplikation mit 2). Diese Kombination kann extern mit L abgekürzt werden.

Der einmal links verschobene Inhalt des Akkumulators geht in den einen Eingang des Addierwerks, wobei in der untersten Dualstelle eine 0 nachgeschoben wird, während die oberste Stelle herausgeschoben wird.

LAm  $2 \langle a \rangle + \langle m \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$   
 LUm  $\langle a \rangle \rightarrow m \quad 2 \langle a \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$   
 LEm  $\langle m \rangle \rightarrow b \quad \text{LEm} + 1' \rightarrow c \quad 2 \langle a \rangle \rightarrow a$   
 LAm  $(\alpha_2, \alpha_3, \dots, \alpha_{38}, 0) + \langle m \rangle \rightarrow a \quad \langle c \rangle \rightarrow b$

LL, L, R in Verbindung mit N ist sinnlos (vergl. aber unten LVN, RVN....)

Durch Kombination der Verschiebungen und gleichzeitigem Addieren des unverschobenen Akkumulatorinhaltes (Adresse 4) kann man mit einem einzigen Befehl den Inhalt des Akkumulators mit  $-1, -1/2, 1/2, 3/2, 2, 3, 4,$  oder  $5$  multiplizieren:

NS 4				$-\langle a \rangle \rightarrow a$
RS 4	$1/2 \langle a \rangle$	$-\langle a \rangle$	$=$	$-1/2 \langle a \rangle \rightarrow a$
RAO				$1/2 \langle a \rangle \rightarrow a$
RA 4	$1/2 \langle a \rangle$	$+\langle a \rangle$	$=$	$3/2 \langle a \rangle \rightarrow a$
LAO				$2 \langle a \rangle \rightarrow a$
A 4	$\langle a \rangle$	$+\langle a \rangle$	$=$	$2 \langle a \rangle \rightarrow a$
LA 4	$2 \langle a \rangle$	$+\langle a \rangle$	$=$	$3 \langle a \rangle \rightarrow a$
LLAO				$4 \langle a \rangle \rightarrow a$
LLA 4	$4 \langle a \rangle$	$+\langle a \rangle$	$=$	$5 \langle a \rangle \rightarrow a$

Das Zeichen V verkoppelt den Schnellspeicher 3 mit dem Akkumulator für Links- und Rechtsverschiebungen.

LVAm Die beiden Speicher werden miteinander verkoppelt und der Inhalt wird gemeinsam um eine Dualstelle nach links verschoben, wobei die oberste Stelle von Speicher 3 in die unterste des Akkumulators gelangt. In der untersten Stelle von 3 wird eine 0 nachgeschoben, während die oberste Stelle des Akkumulators herausgeschoben wird.

Die so erhaltenen Stellen des Akkumulators werden dem Addierwerk zugeführt und der Inhalt der Zelle m zuaddiert.

$(\alpha_2, \alpha_3, \dots, \alpha_{38}, \delta_1) + \langle m \rangle \rightarrow a$   
 $(\delta_2, \delta_3, \dots, \delta_{38}, 0) \rightarrow 3 \quad \langle c \rangle \rightarrow b$

RVAm Akkumulator und Speicher 3 werden verkoppelt und gemeinsam um eine Stelle nach rechts verschoben, wobei die unterste Stelle aus dem Akkumulator in die oberste Stelle von 3 gelangt. Die unterste Stelle von 3 wird herausgeschoben, während im Akkumulator in die oberste Stelle der vor der Verschiebung in der obersten Stelle stehende Wert gelangt.

Zu dem so erhaltenen verschobenen Akkumulatorinhalt wird der Inhalt von Zelle m zuaddiert.

$$(\alpha_1, \alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{37}) + \langle m \rangle \rightarrow a$$

$$(\alpha_{38}, \delta_1, \delta_2, \dots, \delta_{37}) \rightarrow 3 \qquad \langle c \rangle \rightarrow b$$

LLV Doppellinksverschiebung in Verbindung mit Verkoppelung ist nicht möglich.

Bei LV bzw. RV in Verbindung mit U, wird der unverschobene Inhalt des Akkumulators gespeichert.

LV bzw. RV in Verbindung mit E wird ins Befehlszählregister mitübernommen.

LVEm:  $\langle m \rangle \rightarrow b$       LVEm + 1'  $\rightarrow c$       Linksverschiebung von 4 und 3

RVEm:  $\langle m \rangle \rightarrow b$       RVEm + 1'  $\rightarrow c$       Rechtsverschiebung von 4 und 3

LV in Verbindung mit N bewirkt Verschiebung des Inhalts von 3 und Löschen des Akkus, auch die oberste Stelle von 3 gelangt nicht in den Akkumulator.

Bei RV in Verbindung mit N wird vor der Löschung des Akkumulators die unterste Stelle des Akkumulatorinhalts in die oberste Stelle von Speicher 3 übernommen.

(vergl. auch bei H    LVH..., RVH...)  
                   bei C    CV..., LVC..., RVC...)

### 3. 4 Weitere Befehlszeichen

F Das Zeichen F bewirkt, dass der Inhalt des Befehlszählregisters im Schnellspeicher 5 notiert wird.

EFm : auch kurz mit Fm bezeichnet, führt den Sprung an die Stelle m aus, notiert aber gleichzeitig den im Befehlszählregister stehenden E-Befehl im Speicher 5 und bringt Em + 1' ins Befehlszählregister, d. h. F wird vor Übernahme ins Befehlszählregister gelöscht.

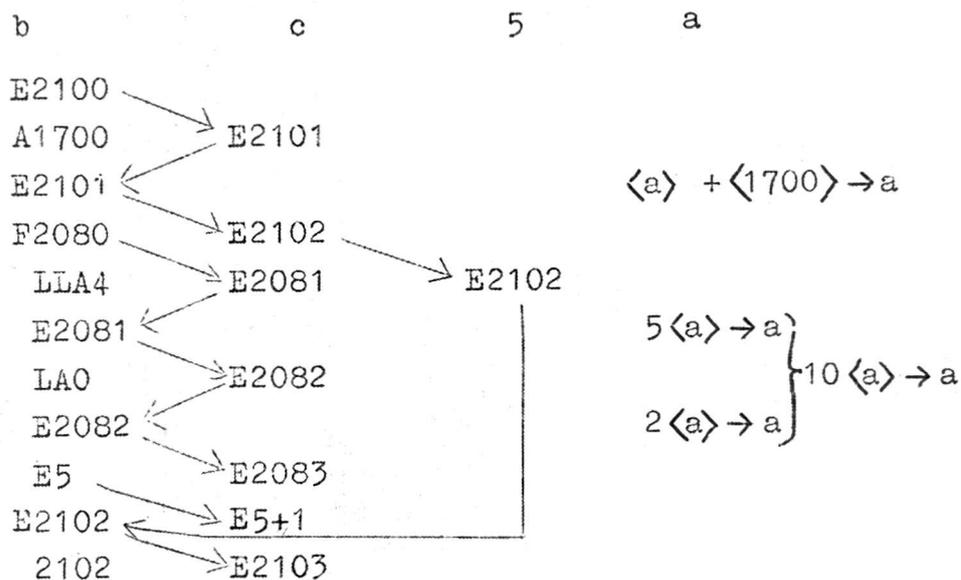
$$\langle m \rangle \rightarrow b \quad E_m + 1' \rightarrow c \quad \langle c \rangle \rightarrow 5$$

Durch diesen Befehl können Unterprogramme aufgerufen werden mit automatischer Rückkehradressennotierung. Durch einen Befehl E 5 am Ende des Unterprogrammes erfolgt dann automatisch der Rücksprung an die Stelle des Oberprogrammes, von wo das Unterprogramm aufgerufen wurde.

Ein Beispiel soll das näher erläutern:

<u>Adr.</u>	<u>Inhalt</u>	<u>Adr.</u>	<u>Inhalt</u>
2100	A 1700	2080	LLA 4
2101	F 2080	2081	LAO
2102	.....	2082	E 5
2103	.....		

} 10  $\langle a \rangle \rightarrow a$



Bei mehrfacher Ineinanderschachtelung von Unterprogrammen muss man die Rückkehradressen aus Speicher 5 umspeichern.

		500	NA5	630	
		:	NU527	:	
		:	:	:	
		:	:	:	
700	F 500	:	F630	:	
701		:	:	:	
		:	:	:	
		:	:	:	
		527	(E701)	672	E5

C Das Zeichen C bewirkt, dass die Stellen 20 bis 38 aus dem Befehlsregister dem Rechenwerk zugeführt werden. Damit kann man positive oder negative (siehe unten) Konstante bis  $2^{19} - 1 = 524.287$  direkt aus dem Befehlsregister holen.

CAn	$\langle a \rangle + n' \rightarrow a$	} $\langle C \rangle \rightarrow b$
CSn	$\langle a \rangle - n' \rightarrow a$	
NCAn	$n' \rightarrow a$	
NCSn	$- n' \rightarrow a$	
CIn	$\langle a \rangle \wedge n' \rightarrow a$	
CISn	$\langle a \rangle \wedge (-n') \rightarrow a$	

mit  $0 \leq n < 2^{19} (=524.288)$

Für die Intersektionsbefehle als Beispiel:

- CI 8191 bewirkt Herausschneiden der letzten 13 Stellen, d. h. die Trommeladressenstellen bleiben im Akkumulator erhalten, wenn dort ein Befehl steht.
- CIS8192 Herausschneiden aller Stellen ausser den Trommeladressenstellen.
- CI 31+0 Heraustrennen nur der Schnellspeicheradressenstellen (vergl. das unter K Gesagte über zwei Adressen).

C in Verbindung mit U bildet unwirksame Befehle (vergl. Zählbefehle). Zu den oben angegebenen Befehlen kann man auch gleichzeitig Verschiebungsbefehlszeichen setzen.

z. B. RCAn  $1/2 \langle a \rangle + n' \rightarrow a$

Wenn C und V gleichzeitig in einem Befehl vorkommen, so ist zu beachten, dass V zunächst seine Funktion-Verkoppelung von Akkumulator und Speicher 3 bewirkt, dann aber als  $2^{18}$  (die Stelle V = Stelle 20. Stelle 20 bis 38 sind bei C-Befehlen wirksam), dem Rechenwerk mitzugeführt wird.

z. B. LVCA1'

$(\alpha_2, \alpha_3 \dots \alpha_{38} \delta_1) + 2^{18} + 1 \rightarrow a$

$(\delta_2, \delta_3 \dots \delta_{38} 0) \rightarrow 3$

Es wird zu dem verschobenen Akkumulatorinhalt, wobei in die unterste Stelle die oberste Stelle von 3 gekommen ist.

nicht nur: 1/, sondern  $2^{18} + 1'$  zuaddiert.

C in Verbindung mit E gibt eine Möglichkeit, Kontrollen in den Programmablauf einzubauen.

Wenn in eine Zelle der Befehl CEn eingespeichert wird, so stoppt die Maschine, sofern der Schalter "bedingter Stop" auf "ein" steht. \*

Steht der Schalter "bedingter Stop" dagegen auf "aus", so wird sofort der Befehl aus der Zelle  $n + 1$  geholt.

CEn	Bed. Stop	ein	
1. Schritt	$E_n + 1 \rightarrow c$	$n \rightarrow b$	Stop
2. Schritt	nach Start:	$\langle c \rangle \rightarrow b$	
	Bed. Stop	aus	
1. Schritt	$E_n + 1 \rightarrow c$	$n \rightarrow b$	unerfüllte Bedingung
2. Schritt		$\langle c \rangle \rightarrow b$	

\* Schalter "bedingter Stop":

auf "ein" bewirkt, dass die Maschine stoppt, sobald etwas ins Befehlsregister kommt, was kein Befehl ist.  
 auf "aus": Unerfüllte Bedingung, wenn etwas ins Befehlsregister gelangt, was kein Befehl ist.

K Das interne Zeichen K ist ein Kennzeichen, dass nur die Schnellspeicheraadresse eines Befehls wirksam sein soll. Die Trommeladressenstellen bei Befehlen mit dem Zeichen K können für Zählzwecke benutzt werden. (vergl. Wiederholungsbefehle).

Bei Befehlen, die nur eine Adresse  $<32$  haben, wird vom Leseprogramm automatisch ein K hinzugesetzt und die Adresse an die richtige Stelle geschoben.

Sind zwei Adressen vorhanden und kein K im Befehl, (wenn die Adressen mit + verbunden sind, setzt das Leseprogramm die erste als Schnellspeicheradresse, die zweite als Trommeladresse ein, ohne ein K zuzuführen), so werden beide Speicher gleichzeitig aufgerufen. Bei A, E, I werden beide Speicherinhalte disjunktiv überlagert dem Addierwerk bzw. Befehlsregister zugeführt, bei U wird in beide Speicher gleichzeitig gespeichert.

Wenn ein Befehl nur eine Trommeladresse enthält, so wird stets gleichzeitig der Schnellspeicher 0 mitaufgerufen, der aber 0 liefert, bzw. bei Speicherungen nichts aufnimmt.

Wenn die Adressen mit - verbunden sind, so wird von der ersten Adresse in der Stellung der Schnellspeicheradresse, die zweite Adresse in der Stellung der Trommelspeicheradresse abgezogen und ein K hinzugesetzt. (vergl. Wiederholungsbefehle). (vergl. auch K in Verbindung mit G).

H Das Zeichen H setzt die letzte Stelle von Speicher 3 disjunktiv auf 1. Wenn die Stelle schon auf 1 war, so geschieht nichts, wenn sie auf 0 war, so wird sie auf 1 gesetzt.

Bei Verschiebung der Inhalte von Akkumulator und Speicher 3 wird das gleichzeitig gegebene H nach der Verschiebung wirksam.

Bei LVH.. wird also in die letzte Stelle von 3, die ja nach der Verschiebung eine 0 enthält, eine 1 eingeführt.

EHm Das H wird mit ins Befehlszählregister übernommen (vergl. Verzweigungen)

Z bewirkt einen Stop der Maschine. Bei bedingten Stop-Befehlen (vergl. Bedingungszeichen) tritt der Stop erst ein, wenn die gestellte Bedingung erfüllt ist. Wenn die Maschine durch das Zeichen Z gestoppt wird, so erfolgt die Ausführung der durch andere Zeichen in den Befehl angegebenen Operationen erst nach Drücken der Starttaste.

### 3.5 Bedingungszeichen

Jeder Befehl kann verschiedenen Bedingungen unterliegen, so dass Befehle abhängig von gewissen Rechenresultaten unterdrückt werden können.

Zwei der Bedingungszeichen beziehen sich auf den Inhalt des Testspeichers 2.

P = Führe den Befehl nur aus, wenn der Inhalt von Speicher 2 positiv ist:

$$\langle 2 \rangle \geq 0$$

Q = Führe den Befehl nur aus, wenn der Inhalt von Speicher 2 negativ ist:

$$\langle 2 \rangle < 0$$

Drei der Bedingungszeichen beziehen sich auf den Inhalt des Akkumulators.

PP = Führe den Befehl nur aus, wenn der Inhalt des Akkumulators positiv ist:

$$\langle a \rangle \geq 0$$

QQ = Führe den Befehl nur aus, wenn der Inhalt des Akkumulators negativ ist:

$$\langle a \rangle < 0$$

PFQQ = Führe den Befehl nur aus, wenn der Inhalt des Akkumulators Null ist:

$$\langle a \rangle = 0$$

Das Bedingungszeichen Y bezieht sich auf die 38. Dualstelle des Speichers 3.

Y = Führe den Befehl nur aus, wenn die letzte Stelle im Speicher 3 eine 1 ist:

$$\langle 3 \rangle_{38} = 1'$$

Die Bedingungszeichen können auch kombiniert werden, so dass die Ausführung eines Befehls unter mehreren Bedingungen steht. z.B.: PPQYE2017 d. h. führe den Sprung zur Zelle 2017 nur dann aus, wenn im Akkumulator etwas Positives, in Speicher 2 etwas Negatives und in der letzten Stelle von Speicher 3 eine 1 steht.

Wird der Sprung ausgeführt, so werden vor der Übernahme aus dem Befehlsregister ins Befehlszählregister die Bedingungszeichen gelöscht.

Falls die gestellte Bedingung bei einem Befehl nicht erfüllt ist, so wird der Befehl nicht ausgeführt und automatisch der nächste Befehl (d. h. der Befehl aus der Zelle nach dem bedingten Befehl) in das Befehlsregister geholt.

#### Adressensubstitution

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

G...m bewirkt vor Ausführung des Befehls..., dass der Inhalt der Trommeladressenstellen der Zelle m als neue Adresse  $\tilde{t}$  für den Befehl..., eingesetzt wird. (m ist also die Adresse, unter der die richtige Adresse zu finden ist.) Im Befehlszählregister wird um 1' weitergezählt, das Zeichen G wird gelöscht.

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

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

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

Als nächster Befehl wird der Inhalt der übernächsten Zelle aufgerufen.

GK...s+t Bei diesem Befehl wirkt die Schnellspeicherzelle s als Indexregister mit konstantbleibendem Inhalt. Vor der Ausführung des Befehls... wird der Inhalt der Schnellspeicherzelle s zu t hinzuaddiert; von dieser Summe werden die letzten 15 Stellen (Trommeladressenstellen) als neue Adresse  $\tilde{t}$  eingesetzt. G, K und die Schnellspeicheradresse s werden gelöscht und im Befehlszählregister wird um 1' weitergezählt.

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

1. Schritt  $(\langle s \rangle + t) \xrightarrow{t} \tilde{t}$      $\langle c \rangle + 1' \rightarrow c$     G, K, s löschen  
 $\langle s \rangle \rightarrow s$

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

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

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

$\langle s \rangle + t \rightarrow s$   
1. Schritt  $(\langle s \rangle + t) \xrightarrow{t} \tilde{t}$      $\langle c \rangle + 1' \rightarrow c$     C, G, K, s löschen

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

Der Inhalt der übernächsten Zelle wird als nächster Befehl aufgerufen.

Ist bei den oben beschriebenen Fällen der Befehl... ein Sprungbefehl, so ist zu beachten, dass ins Befehlszählregister nur  $E \tilde{t} + 1'$  übernommen wird; evtl. Zusätze wie L, R, H aber nicht (vergl. 3.3). Das kann man auch ausnutzen, wenn etwa beim Sprung noch einmal linksverschoben werden soll. Man ruft dann den Schnellspeicher 0 als Indexregister auf, der ja nichts liefert.

GKLEO+t    2 <a> → a    <t> → b    Et + 1' → c

Die einzelnen Schritte ausführlicher:

1. Schritt <0> + t = t ⇒  $\tilde{t}$     <c> + 1' → c    GK löschen

2. Schritt LE $\tilde{t}$  ausführen:

$\langle \tilde{t} \rangle \rightarrow b$     2<a> → a    E $\tilde{t}$  + 1' → c

Hier ist L nur einmal wirksam, da es im Steuerregister erhalten bleibt, während der Addition <0> + t. Im Befehlsregister werden dagegen alle Stellen ausser dem Befehlskennzeichen und den Trommeladressenstellen vor Übernahme ins Befehlszählregister gelöscht.

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

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

1. Schritt <s> + t → s    <c> + 1' → c    G, s löschen

2. Schritt CU $\tilde{t}$  wirkungslos

Der Akkumulatorinhalt bleibt erhalten.

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

1. Schritt <s> + t → s  
 $\langle \langle s \rangle + t \rangle_t \Rightarrow \tilde{t}$     <c> + 1' → c    G, s löschen

2. Schritt CA $\tilde{t}$  ausführen

Zu den Zählbefehlen können weitere Zusätze gemacht werden, wie z. B. N, RV oder ähnliche.

CGRVU s+t    <s> + 1 → s     $\frac{1}{2} (4, 3) \rightarrow (4, 3)$

<c> + 1' → b

Die Ausführung der G-Befehle kann selbstverständlich auch unter Bedingungen gestellt werden. Bei den Befehlen, die auch C enthalten, bei denen ein Index- oder -24-

Zählregisterinhalt verändert worden soll, erfolgt die Änderung nur bei erfüllter Bedingung. Das Weiterzählen im Befehlszählregister um 1' erfolgt auch dann, wenn die Bedingung nicht erfüllt ist.

In jedem Fall wird also nach einem G-Befehl eine Zelle übersprungen.

3.7 Hier sei nochmals zusammengestellt, welche Befehlszeichen vor der Übernahme aus dem Befehlsregister ins Befehlszählregister gelöscht werden:

1. alle Bedingungszeichen, N, F und G
2. bei C-Befehlen: der Operationsteil, ausser der Befehlskennzeichenstelle und der Stelle V (Stelle 2 bis 19)\*
3. bei G-Befehlen im 2. Schritt: alle Stellen ausser der Befehlskennzeichenstelle und den Trommeladressenstellen (Stellen 2 bis 25)

\* Die erste Stelle wird auf dem Weg zum Addierwerk noch abgeschnitten.

4. ZÄHLUNGEN UND VERZWEIGUNGEN

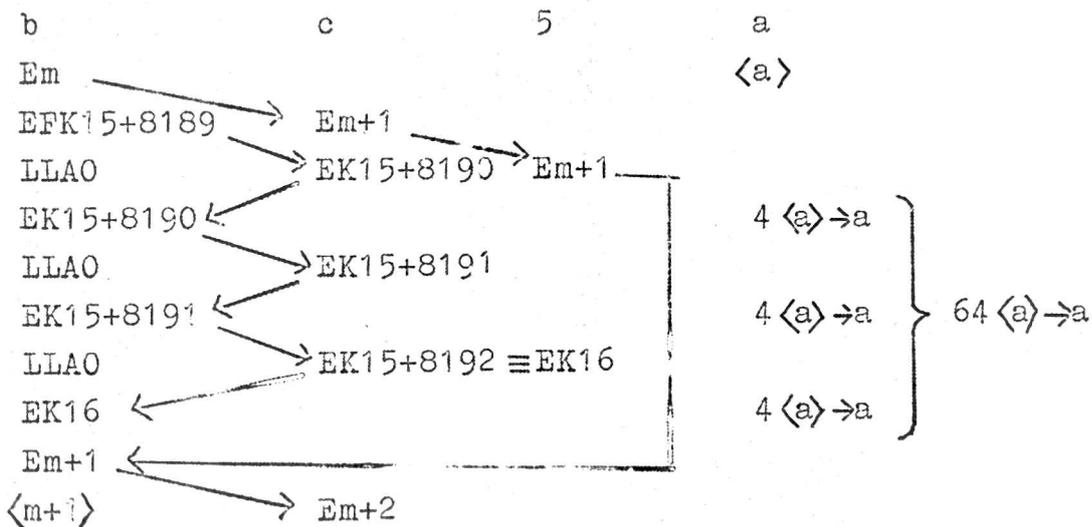
4.1 Wiederholungsbefehle

Sind in einem Befehl zwei Adressen vorhanden und mit einem - Zeichen verbunden, so setzt das Leseprogramm dem Operationsteil ein K zu, und zieht von der als Schnellspeicheradresse eingesetzten ersten Adresse die zweite in der Stellung der Trommelspeicheradresse ab.

Da jetzt ein K im Operationsteil vorhanden ist, wird nur die Schnellspeicheradresse aufgerufen und die Trommeladresse kann zu Zählzwecken benutzt werden.

Ein Beispiel wird das näher erläutern:

Im Schnellspeicher 15 stehe der Befehl LLAO. Wird nun im Programm der Befehl F 16 - 5 gegeben, so erfolgt der Programmablauf. (F 16 - 5 ist in der Maschine EFK 15 + 8189)



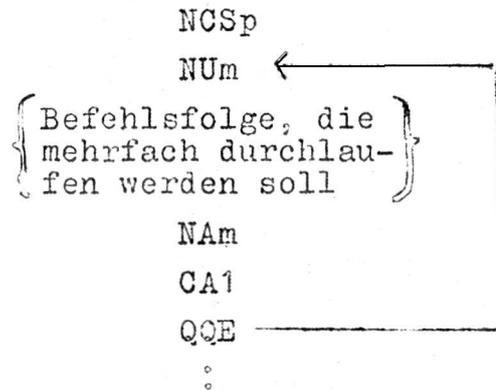
Es wird also durch den Befehl F 16 - t der Inhalt der Schnellspeicherzelle 15 t-mal aufgerufen und anschliessend automatisch unter der Adresse 16 der Inhalt des Rückkehradressenspeichers 5 der Rücksprungbefehl ins Programm bewirkt.

Solche Wiederholungsbefehle können selbstverständlich auch mit allen anderen Schnellspeichern gebildet werden, nur muß dann zusätzlich im Schnellspeicher s (bei einem Befehl Es-t) ein weiterer Sprungbefehl gespeichert werden. Es wird dann der Befehl in der Schnellspeicherzelle (s-1) t-mal ausgeführt und anschliessend der Befehl in s.

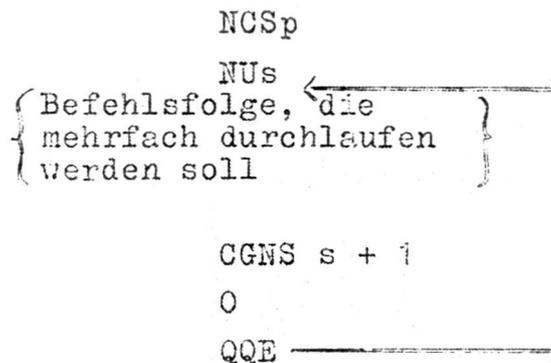
#### 4.2 Abzählen in Schleifen

Durch die Wiederholungsbefehle kann man einzelne Befehle mehrfach hintereinander ausführen. Will man dagegen eine Befehlsfolge mehrfach durchlaufen, so zählt man zweckmässig ab mit Zählbefehlen oder auch durch bloßes Aufaddieren mit C-Befehlen.

Soll z.B. eine Schleife p-mal durchlaufen werden, so braucht man nur zum Anfang -p' in irgendeinen Speicher m zu bringen und am Ende der Schleife zu zählen.

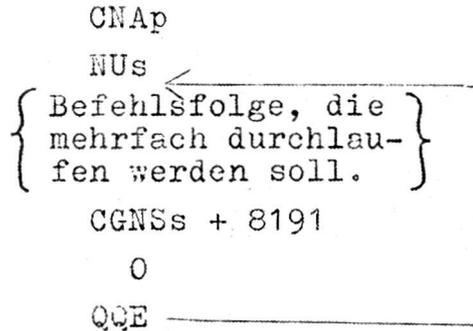


Sobald das Programm p-mal durchlaufen ist, wird der Inhalt von m Null, d. h. positiv und es erfolgt nicht mehr der Sprung an den Anfang der Schleife, sondern es erfolgt weiter ein linearer Programmablauf. Zum Abzählen kann hier ein Schnellspeicher oder ein Trommelspeicher benutzt werden. Auch bei Benutzung der Zählbefehle ist es am zweckmäßigsten bei p-Umläufen -p' in das Zählregister (irgend ein Schnellspeicher) zu bringen und mit einem CGNS s + 1-Befehl weiterzuzählen, der die Trommeladressenstellen negativ in den Akkumulator bringt. Nur dann, wenn der Inhalt des Zählregisters Null geworden ist, (genauer  $\langle s \rangle \equiv 0 \pmod{8192}$ ), wirkt der Akkumulator positiv.



Man kann aber auch von positiven Zahlen zurück auf Null zählen, wobei allerdings nur der Inhalt der Trommeladressenstellen das richtige Zählergebnis enthält. Falls das Zählregister gleichzeitig als Indexregister benutzt wird, schadet dieses

ja nicht, weil nur die Trommeladressenstellen des Indexregisters wirksam werden.

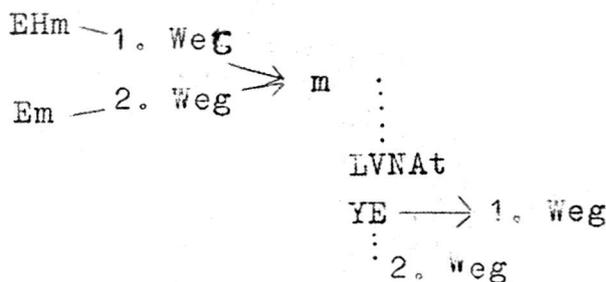


Im Zählregister s wird hierbei um 8191 weitergezählt, was aber für die Trommeladressenstellen eins abziehen bedeutet. ( $8191 \equiv -1 \pmod{8192}$ )

In diesem zweiten Fall wird das richtige Zählergebnis negativ in den Akkumulator gebracht, während im Indexregister das Zählergebnis plus Vielfache von 8192 stehen. Im ersten Fall dagegen steht im Indexregister das tatsächliche Zählergebnis und dem Akkumulator wird nur ein Teil des Zählergebnisses zugeführt, doch ist es für die Abfrage auf Null gleichgültig.

### 4.3 Verzweigungen

Es sollen hier einige Möglichkeiten angedeutet werden, wie man eine Verzweigung eines Programms erreichen kann. Soll z. B. ein Teil des Programmes gemeinsam durchlaufen werden, dann aber wieder getrennte Wege eingeschlagen werden, so kann man das erreichen, indem man einmal mit Em, das andere Mal mit EHm in den Programtteil springt. Vor der Abfragung, ob das letzte Bit von 3 auf 1 oder 0 steht, muss einmal Speicher 3 gelöscht, bzw. nur die letzte Stelle von 3 gelöscht werden. Da das H stets mit übertragen wird, erfolgt ein neues Setzen der letzten Stelle von 3 in dem einen Fall und man kann durch Y die letzte Stelle abfragen.





5. 1 Zugriffszeiten und deren Vermeidung  
=====

Die mittlere Zugriffszeit zu einem beliebigen Trommelspeicher beträgt 5 msec. So geht im allgemeinen bei einem Lese- oder Speicherbefehl eine ganze Trommelumdrehung verloren, da man einmal zu der aufgerufenen Speicherzelle gelangen muß zum anderen aber auch zum Programm zurück, um den nächsten Befehl ins Befehlsregister zu holen. Diese Wartezeiten lassen sich jedoch durch geschickte Verwendung der Speicherzellen bzw. durch Benutzen von Schnellspeichern, die jederzeit ohne Zugriffszeit zur Verfügung stehen, weitgehend vermeiden.

Wie schon erwähnt, sind je 256 Speicherzellen zu einem Block zusammengefasst. Die Anordnung der Zellen in jedem Block ist gleich, nur erhöhen sich die Adressen von Block zu Block um 256.

Jeder Block umfasst 8 Spuren. Die Spuren sind in 32 Sektoren (0 bis 31) eingeteilt, wobei jeder Sektor eine Speicherzelle darstellt.

Die Anordnung der fortlaufenden Numerierung erfolgt im 2-Takt. Die Zellen 0 bis 127 liegen in den geraden Sektoren der Spuren 0 bis 7 (0 bis 15 auf Spur 0, 16 bis 31 auf Spur 1 usw.) Die weiteren Zellen 128 bis 255 liegen dann in den noch frei gelassenen ungeraden Sektoren der Spuren 0 bis 7. So haben wir die folgende Anordnung:

Sektor	Spur	0	1	2
0		0	16	32
1		128	144	.
2		1	17	.
3		129	145	.
4		2	18	
5		130	146	
6		3	19	
7		.	.	
		.	.	

Durch einen Befehl im  $k$ ten-Sektor kann man jede Zelle im  $(k+1)$ ten-Sektor (gleichgültig auf welcher Spur) ohne Wartezeit erreichen und zum Programm zurückkehren. Allerdings geht bei Speicherungen stets eine ganze Trommelumdrehung verloren, da das Schreibsignal nicht so schnell abklingt, dass sofort in der nächsten Wortzeit wieder gelesen werden kann. In der Maschine wird deshalb nach einem Speicherbefehl in eine Trommelzelle für eine Wortzeit der Worttransport gesperrt. Bei den Befehlen, die aus der Trommel etwas herauslesen, (E, A, I) kann aber stets der nächste Sektor ohne Zugriffszeit erreicht werden.

Für die Anordnung aller Speicherzellen der Trommel vergleiche man die Programmierungsanleitung "Freiburger Code" Abschnitt 7.4.