


BULL
GENERAL  ELECTRIC
GE 55 Codetabelle

ANSCHLUSSCODES FÜR RANDEINHEITEN						
Kanal	Ohne Halt auf d. Zeich.	Mit Halt auf dem Zeichen	Nummer des Registers	Funktion		
normal	1	95	92	10	Puffer → Zentralspeicher	
		9D	9A	12	Freigabe num. Tastatur	
		96	93	06	Kartenleser	
		97	94	R	Freigabe α-Tastatur	
	2					
	3					
	schnell	1	9D	9A	10	Zentralspeicher → Puffer
			9D	9A	11	Freigabe Sichtanzeige
		2	AD	AA	09	Kartenstanzer
			AF	AC	12	Test
		3	CD	CA	08	Drucker MB 50
		schnell	0	10		R
11				R	- Test	
1	14			R	2. Trommel - Lesen	
	15			R	- Test	
2	18			R	Ext. → Zentralspeicher	
	19			R	Test	
3	1D			08	Drucker I41 Test	
0	12			R	1. Trommel - Schreiben	
	13			R	- Auswahl d. Spur	
1	16			R	2. Trommel - Schreiben	
	17			R	- Auswahl d. Spur	
2	1A			R	Zentralspeicher → Ext.	
	1B		R	Berechnen Schlüssel		
3	1E		08	Drucker I41 drucken		

Ref.-Nr.: 23.57.001

DISPLAY Kin falsche Karte lesen

9D 10 9D 12 95 10 ✓ KUB

9D 10 9A 12 95 10 ✓ KUB

9D 10 9D 11 92 10 ✓ CLB

9D 10 9D 11 95 10 ✓ CLB

hält an,
weiter lernen nach ↑

KKFBBV 151277

17/12/77

Beispiele:

A Kartenleser

IOÇ 53 | 01 | 96 | 06 Reg. 6 = 00 | 00 | 00 | 22 | 56

Zone 6 vorher : 20 | 20 | 20 20 | 20 | F4

nachher : 32 | 31 | 30 38 | 34 | F4
2257 2337

B Pufferspeicher

a) Freigabe numerische Tastatur

IOÇ 53 | 01 | 9A | 12 Hilfsschl. Adresse - 1 des Codes

Reg. 12 0 F | F 4 | 0 0 | 0 1 | 5 5

b) Übertrag Pufferspeicher an Zentralspeicher

IOÇ 53 | 01 | 92 | 10 Reg. 10 00 | 00 | 00 | 23 | 47

Zentralspeicher

Pufferspeicher

vorher : 20 | 20 | 20 | 20 | 20 | 20 | F4

104:

nachher : 20 | 20 | 31 | 30 | 34 | 3A | F4
2348

c) Übertrag Zentralspeicher an Pufferspeicher

IOÇ 53 | 01 | 9A | 10 Reg. 10 00 | 00 | 00 | 23 | 47

Zentralspeicher

Pufferspeicher

vorher : 20 | 20 | 20 | 39 | 38 | 32 | F4

nachher : 20 | 20 | 20 | 39 | 38 | 32 | F4
2348

982

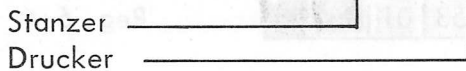
d) Freigabe Sichtanzeige Hilfsschlüssel Adresse - 1

IOÇ 53 | 01 | 9A | 11 Reg. 11 0E | F4 | 00 | 01 | 50

0151

C Stanzen einer Karte und gleichzeitiges Drucken

Befehl der Operationen: IØC |53|02|AA|09|CA|08|



Test des Stanzers IØC |53|01|AC|12|

Reg. 8 Reg. 9 Reg. 12

|00|00|00|20|37| |00|00|00|21|74| |0F|F4|00|01|55|

Zone 9 |41|42|...|39|0D|F4|38|F4| Stanzen von 49 Zeichen

2175 2223 | Schlüssel "Kartenauswurf"

Zone 8 |44|20|32|...|38|0D|0C|F4|20| Drucken von 64 Stellen

2038 2097 | |
 | Schlüssel "Sprung"
 | Schlüssel "Wagenrücklauf"

Register und Zonen sind nach der Operation unverändert.

D Trommel

a) Lesen von einer Trommel, die an Kanal R0 angeschlossen ist, evtl. unabhängig von anderen.

- Auswahl der zu lesenden Bahn IØC 5301 1318
- Test der Auswahloperation IØC 5301 1105
- Lesebefehl IØC 5301 1005
- Test 2 der Leseoperation IØC 5301 1105

Register 18 |00|00|00|06|00|

Register 05 |00|00|00|13|00|

Nummer der Bahn |30|32|34|F4|

0601

Für die Trommel belegte Zone:

vorher : |00|44|52|...|54|52|30|48|43|F4|32|

1300 1533

nachher: |30|33|34|...|37|30|F4|48|43|F4|32|

1300 1533

Daten von Bahn 24 übertragen
(233 Zeichen + Trennmarke F4)

b) Die Benutzung von zwei Trommeln als logische Einheit

Die Programmierung erfolgt hier mit dem $\text{I}\Phi\text{IC}$ -Befehl. Die Anschlußcodes der Trommel, auf die man schreiben will, werden vor dem entsprechenden Trommelbefehl in den Registern 14, 15 und 16 gespeichert.

In dem folgenden Beispiel ist gerade die Trommel an Kanal R1 in Arbeit.

- Auswahl der Schreibbahn $\text{I}\Phi\text{IC}$ 54 01 14 19
- Test 1 der Auswahloperation $\text{I}\Phi\text{IC}$ 54 01 16 18
- Schreibbefehl $\text{I}\Phi\text{IC}$ 54 01 15 10
- Test 2 des Schreibbefehls $\text{I}\Phi\text{IC}$ 54 01 16 18

Reg. 14	Reg. 15	Reg. 16
<u>00 00 00 00 17</u>	<u>00 00 00 00 16</u>	<u>00 00 00 00 15</u>
Reg. 18	Reg. 19	Nr.d. Bahn
<u>00 00 00 12 00</u>	<u>00 00 00 06 00</u>	<u>31 32 36 F4</u>
		0601

Für die Trommel belegte Zone:

vorher: 00|32|48 .. 42|46|F4|38|35
1300

Daten auf Bahn 126 geschrieben
(157 Zeichen + Trennmarke F4)

nachher: 30|32|48 .. 42|46|F4|38|35
1300 1457

*Einzel ohne Halt
Doppel mit Halt*

3.6.7. Die vertikale Simultaneität

Es handelt sich hierbei um eine Simultaneität zwischen der Arbeit einer oder mehrerer Randeinheiten auf der einen und Bearbeitungsbefehlen auf der anderen Seite innerhalb desselben Programms. Hierfür werden die IØC-Befehle in 3 Kategorien aufgeteilt.

3.6.7.1. Gleitende IØC-Befehle

Das "N" in der Mitte dieses Befehls, das die Anzahl der zu verbindenden Randeinheiten angibt, wird in der Form 1N geschrieben.

Hier gibt es wiederum 3 Fälle:

- Der IØC spricht eine langsame Randeinheit mit Halt auf dem Zeichen an. Es kommt zu einer Simultaneität Randeinheit - Bearbeitung. Der Befehl, der diesem IØC folgt, wird sofort nach dem Start der Randeinheit durchgeführt.
- Es handelt sich um eine langsame Randeinheit ohne Halt auf dem Zeichen. In diesem Fall wird der Befehl, der dem IØC folgt, erst durchgeführt, wenn dieser beendet ist.

Beispiel: 53 11 AA 09 (Stanzen)
 Programmfolge
 53 11 96 06 (Lesen)

Das Stanzen wird gestartet, und das Programm läuft simultan hierzu ab bis zum Auftreten des Lesebefehls. Das Lesen erfolgt simultan mit dem Stanzen, wenn dieses noch nicht beendet ist. Der Befehl, der dem Lesebefehl folgt, wird jedoch erst ausgeführt, wenn das Lesen beendet ist, allerdings auch dann, wenn das Stanzen noch nicht beendet ist.

Beispiel: 53 11 AD 09 (Stanzen)

Es handelt sich um einen gleitenden IØC mit einem Verbindungscode ohne Halt auf dem Zeichen. Der folgende Befehl wird erst ausgeführt, wenn das Stanzen beendet ist.

- Der IØC spricht eine schnelle Randeinheit an. Der folgende Befehl wird erst durchgeführt, wenn der IØC beendet ist.

Im Multiprogramming löst der gleitende IØC weder einen Programmwechsel (Bit 2 im PRC) noch einen Programmhalt (PO im PRC) aus. Ebenso wird dieses Programm nicht wieder "gestartet" (einsetzen der Programmnummer in die PRC-Zone), wenn alle durch den IØC verbundenen Randeinheiten ihre Arbeit beendet haben.

3.6.7.2 Blockierende IØC-Befehle

Die Anzahl der zu verbindenden Randeinheiten wird in der Form ON ausgedrückt.

Erkennen die Schaltkreise des GE-55 einen solchen Befehl, wird ein Programmhalt ausgelöst (PO im PRC). Das Programm wird wieder gestartet, wenn alle durch dieses Programm verbundenen Randeinheiten ihre Arbeit beendet haben (Trennmarke am Ende einer jeden Zone erkannt). Es ist zu beachten, daß es sich hierbei um sämtliche Randeinheiten handelt, verbunden durch:

- den blockierenden Befehl selbst und evtl. durch einen oder mehrere IØC, die vorher vom Programm durchlaufen wurden (gleitende IØC).

Beispiel: 53 11 AA 09
 Programmfolge
 53 01 96 06

Der Befehl, der dem Lesebefehl folgt, wird erst ausgeführt, wenn sowohl das Lesen als auch das Stanzen beendet ist.

3.6.7.3. Wartender IØC-Befehl

Dieser Befehl ermöglicht es, die Verarbeitung, die simultan zur Durchführung eines gleitenden IØC-Befehls abläuft, zu unterbrechen, ohne daß durch ihn neue Randeinheiten verbunden werden.

Die Anzahl "N" der zu verbindenden Randeinheiten ist 0: 53 00

Er hat die gleichen Charakteristiken wie der blockierende IØC. Wird dieser Befehl zu einem Zeitpunkt durch die Schaltkreise des GE-55 erkannt, zu dem bereits alle durch das Programm verbundenen Randeinheiten ihre Arbeit beendet haben, verhält er sich wie ein Befehl NØP.

Beispiel: 53 12 AA 09 CA 08
 simultane Verarbeitung
 53 00
 keine simultane Verarbeitung

Beispiel: Spezieller Fall des Sichtkartenlesers
 53 11 93 06
 simultane Verarbeitung in der Wartezeit von 110 Millisekunden
 53 00

110 Millisekunden benötigt eine Karte, um vor die Lesezellen zu gelangen. Es ist möglich, in dieser Zeit Verarbeitungsprogramm ablaufen zu lassen.

3.7. Die Umschlüsselungsbefehle

3.7.1. Allgemeines

Einige Randeinheiten können den internen ISØ-Code nicht verarbeiten. Sie erhalten oder geben die Informationen in einem Zwischencode aus.

Der Kartenleser und der Stanzer PS 40 haben zwei Zwischencodes, die von den beiden Lochcodes abhängig sind und zwar:

- den Zwischencode T 121
- den Zwischencode H 14.012

Die Umschlüsselungsbefehle bewirken eine Umwandlung des Zwischencodes in den internen Code und umgekehrt.

Diese Umschlüsselungen finden im Kernspeicher statt und zwar in den Ein- oder Ausgabezonen und gelten jeweils für die gesamte angerufene Zone.

Bearbeitung der Überlochungen im Code T 121 und H 14.012

Die Kombination einer Überlochung mit einer numerischen Lochung kann weder vom Code T 121 noch vom Code H 14.012 in den internen ISØ-Code übersetzt werden:

- da einige Kombinationen im ISØ-Code nicht existieren.
Beispiel: im Code T 121: Ziffern von 1 - 6 mit 11 oder 12
- da die Kombinationen, die ein Zeichen geben, nach der Übersetzung nicht mehr in ihrem ursprünglichen binären Aufbau als numerisches Zeichen und Überloch auseinandergeführt werden können.
Beispiel: im Code T 121: Ziffern 9, 8, 7 mit 11 oder 12
im Code H 14.012: alle Kombinationen.

Dagegen können Überlochungen und die numerischen Werte im Zwischen-code leicht unterschieden und getrennt werden; man stellt also vor der Übersetzung in den internen ISØ-Code den ursprünglichen binären Aufbau des numerischen Wertes her, indem man die Bits, die die Überlochung darstellen, eliminiert. Wenn das Überloch für die weitere Bearbeitung noch benötigt wird, werden diese eliminierten Bits in eine vorläufige Bearbeitungszone übertragen.

Im umgekehrten Fall, wenn man in ein und dieselbe Spalte einer Karte ein Überloch und numerische Werte stanzen will, müssen die binären Kombinationen im Zwischencode hergestellt werden; diese Operation wird ausgeführt, nachdem die numerischen Werte, die getrennt im ISØ-Code dargestellt waren, in den Zwischencode übersetzt wurden.

Diese Bearbeitung wird meistens durch logisches UND und logisches ODER je nach den Gegebenheiten durchgeführt.

Zwischencode T 121

Wenn man die Bits von rechts nach links mit 0-7 numeriert, werden die Lochungen 11, 12 und 0 bis 9 wie folgt dargestellt:

Lochungen	Bits	hexadezimaler Code
11	7 und 3	88
12	7	80
0 bis 9	6, 5, 4 und 2, 1, 0	variable

Die Kombination von zwei Lochungen, numerisch und Überloch 11 oder 12, ergibt sich aus der Zusammensetzung der oben angegebenen Bits.

Die Trennung der numerischen Werte von den Überlochungen erfolgt durch Löschen der Bits 7 und 3 mittels logischer Addition folgender hexadezimaler Schlüssel:

- 88 isoliert die Lochung "11";
- 80 isoliert die Lochung "12";
- 77 isoliert die numerischen Lochungen.

Das Einsetzen einer Überlochung über einen numerischen Wert erfolgt dementsprechend durch Belegung der Bits 7 und 3 und zwar wird mittels des logischen ODER mit den hexadezimalen Schlüsseln:

- 88 Wert "11" eingesetzt,
- 80 Wert "12" eingesetzt.

Zwischencode H 14.012

Die Lochungen 0 bis 9 ergeben mit oder ohne Lochungen 11 und 12 kombiniert folgende hexadezimale Schlüssel:

Lochungen	0	1	2	3	4	5	6	7	8	9	nichts
nichts	30	31	32	33	34	35	36	37	38	39	
11	70	51	52	53	54	55	56	57	58	59	50
12	60	41	42	43	44	45	46	47	48	49	40

Die sich ergebenden Schlüssel auf zwei Lochungen, numerisch und Überloch 11 oder 12 (Zeile 2 und 3 der Tafel) zeigen, daß:

a) die Trennung der Überlochung durch Isolierung der Bits 4 und 6 mittels einer logischen Addition mit folgenden hexadezimalen Schlüsseln erlangt werden kann:

- 50 isoliert die Lochung "11",
- 40 isoliert die Lochung "12".

b) die entsprechenden Schlüssel für numerische Lochungen allein (Zeile 1 der Tafel) durch Löschen des linken Halbbytes des kombinierten Schlüssels und durch Belegung der Bits 4 und 5 mittels folgender Befehle, erlangt werden können:

- logisches UND mit dem hexadezimalen Schlüssel OF,
- logisches ODER mit dem hexadezimalen Schlüssel 30.

Das Einsetzen einer Überlochung über einen numerischen Wert im Code H 14.012 (Zeile 1 der Tafel) erfordert das Löschen des linken Halbbytes des numerischen Wertes und die entsprechende Belegung der Bits 4 und 6 mittels folgender Befehle:

- logisches UND mit dem hexadezimalen Schlüssel OF,
- logisches ODER mit dem hexadezimalen Schlüssel:
 - 50 zum Einsetzen des Wertes "11" (Zeile 2 der Tafel),
 - 40 zum Einsetzen des Wertes "12" (Zeile 3 der Tafel).

Anmerkung:

Es ist nicht möglich, nach dieser Methode in die gleiche Spalte ein Überloch mit einer Ziffer "0" zu stanzen; die "0" wird nicht erkannt (siehe 1. Spalte der Tafel). Um die Ziffer "0" zu erhalten, muß ein zusätzliches logisches ODER mit dem hexadezimalen Schlüssel 20 durchgeführt werden.

Lochungen	0	1	2	3	4	5	6	7	8	9	nicht
12	50	41	42	43	44	45	46	47	48	49	40
11	70	51	52	53	54	55	56	57	58	59	50
nicht	30	31	32	33	34	35	36	37	38	39	30

3.7.2. Die Umschlüsselungsbefehle

TRX

 TRanslate X

X = siehe nachstehende Tafel.

OT	R
----	---

OT = siehe nachstehende Tafel.

R = Basisregister der zu übersetzenden Zone.

Die Zeichen, die in der durch das Basisregister angegebenen Zone enthalten sind, werden nacheinander in aufsteigender Reihenfolge übersetzt. Auf die Basisadresse wird solange "1" addiert, bis die Trenmarke F4 erkannt wird.

Das Resultat steht in der gleichen Zone, jedes übersetzte Zeichen an der gleichen Stelle, an der es vorher stand.

Die Trenmarke wird nie verändert und nie gelöscht.

Die Übersetzungen:

Symbolischer OT	Maschinen OT	durchgeführte Übersetzung
T R 0	50	Zwischencode T 121 in ISØ-Code
T R 1	51	Zwischencode H 14.021 in ISØ-Code
T R A	5A	ISØ-Code in Zwischencode T 121
T R B	5B	ISØ-Code in Zwischencode H 14.012
T R D	5D	141 64 Zeichen

3.8. Die Hilfsbefehle

Die Hilfsbefehle werden im wesentlichen nur bei den Programmen der Software verwendet. Da der Benutzer aber eventuell einige von ihnen gebrauchen muß, werden sie hier beschrieben.

Einige dieser Befehle bestehen nur aus dem Operationstyp.

3.8.1. Kernspeicherausdruck

PRSTØ

PRint STORe

OT

OT = 78 für PRSTØ

Dieser Befehl bewirkt den Kernspeicherausdruck über den Drucker und zwar durch Anruf eines fest eingespeicherten Unterprogrammes (ca. 204 Bytes).

Zu Beginn erfolgt systematisch ein Ausdruck bis Kernspeicherstelle 0200. Anschließend wird die numerische Tastatur freigegeben; durch Eintasten der Anfangs- und Endhunderteradresse werden dann die Zeichen Byte für Byte in aufsteigender Reihenfolge ausgedruckt. Bei Eingabe von z. B. 0306 werden die Kernspeicherstellen 0301 bis 0600 ausgedruckt.

Anmerkung:

Der Inhalt der ersten 25 Stellen des Kernspeichers wird nicht gedruckt.

Druckaufteilung

Der Kernspeicherausdruck beginnt mit einem Wagenrücklauf auf Stelle 1 und einem Zeilenvorschub. Anschließend werden je Zeile 25 Bytes zu je 5 gedruckt. Ein Leerzeichen trennt jede 5er-Gruppe.

Die Bytes werden in hexadezimaler Form ausgegeben. Jedes Byte wird durch zwei Zeichen dargestellt.

Die Adresse des ersten Bytes jeder Zeile ist am Anfang der Zeile angegeben. Sie ist durch eine Leerstelle von der Zeile getrennt.

Am Ende des Ausdruckes bleibt der Schreibkopf am Ende der Zeile stehen. Will man mit einer Arbeit fortfahren muß man den Wagenrücklauf und einen Sprung durch die Taste SCR herbeiführen.

Anmerkung:

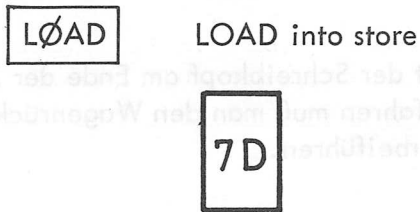
Diese Befehle können beim Test eines Programmes gebraucht werden, um den Inhalt bestimmter Kernspeicherzonen nach bestimmten Bearbeitungsfolgen auszudrucken. Sie werden an der Stelle in das Programm eingeschoben, die vom Programmierer gewählt wurde. Nach dem Test läuft die normale Bearbeitung weiter.

Der Ausdruck des Kernspeichers kann auch mittels der Drucktaste PRS am Bedienungspult hervorgerufen werden. → Ω 28/79

Beispiel:

1026	47515A6B57	4E52
1051	5263616272	7121
1076	6022233132	3538
1101	3500003728	2931

3.8.2. Laden des Kernspeichers



Dieser Befehl lädt den Inhalt einer oder mehrerer Karten in den Kernspeicher, normalerweise ein Programm oder eine Konstante.

Die zu speichernden Informationen sind:

- in der ersten Karte von Spalte 11 bis 80 gelocht,
- in den Folgekarten von Spalte 1 bis 80 gelocht.

Das Ende der Information wird durch das Zeichen "FF" angezeigt, das in der letzten Karte hinter das letzte zu speichernde Zeichen in zwei Spalten gestanzt wird. Die Information wird in hexadezimaler Form ausgedrückt: jede Spalte der Karte stellt ein Halbbyte dar und enthält ein hexadezimaler Zeichen (0 bis F). Zwei aneinandergrenzende Spalten, ungerade und gerade, entsprechen einem Byte.

Die erste Karte enthält unter anderem die notwendigen Angaben für das Laden, und zwar:

- in Spalte 1 und 2 einen Schlüssel, der den benutzten Lochcode angibt:
"00" für Code T 121,
"04" für Code H 14.012.
- in Spalte 3 bis 6 die Speicheradresse der in Spalte 11 und 12 gelesenen Zeichen,
- in Spalte 7 bis 10 die Startadresse.

Das Laden des Kernspeichers erfolgt in aufsteigender Reihenfolge ab der 1. Speicheradresse. Die gelesenen Spalten werden übersetzt und anschließend je zwei auf eine Kernspeicherstelle verdichtet. Das Laden wird beendet, wenn das Zeichen FF erkannt wird, das nicht eingespeichert wird.

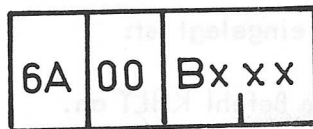
Wenn der Befehl "LOAD" ausgeführt ist, übernimmt das Programm den Befehl, der sich an der oben erwähnten Startadresse befindet, und der in das Programm-Adreß-Register übertragen worden ist.

Anmerkung:

- a) Es können zwei Ladebefehle nacheinander ausgeführt werden unter der Voraussetzung, daß zwischen dem Ende des ersten und dem Anfang des zweiten Befehls 3 ms verstreichen.
- b) Die oben beschriebenen Funktionen des Ladens können auch von der Bedienung mittels einer am Bedienungspult befindlichen Taste "LOAD" durchgeführt werden.

3.8.3. Adressierung der Sprungstufen

R L A	Replace Levels by Adresse
-------	---------------------------



Bxxx = Speicheradresse im Programm

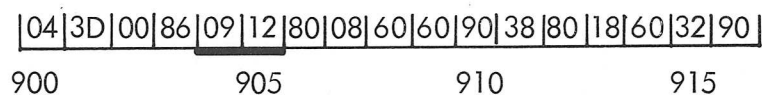
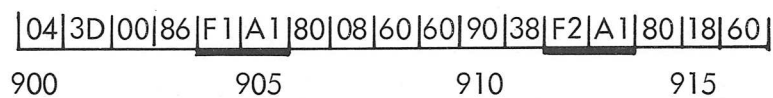
Dieser Befehl ersetzt in einem Programm die symbolischen Sprungadressen (siehe 3.2.) durch ihre wirkliche Adresse.

Die reelle Adresse jeder Sprungstufe, die durch einen Pseudobefehl LEVEL angegeben ist, wird ab der Anfangsadresse Bxxx des Programmes im Kernspeicher berechnet. Dann wird sie an die Stelle der symbolischen Adresse gesetzt, und zwar bei allen Sprungbefehlen, die sich im Programm befinden.

Am Ende der Operation sind alle Pseudobefehle herausgefallen, und das Programm ist zur Adresse Bxxx hin verschoben worden. Diese Verschiebung wird bei der Kernspeicherbelegung mit gerechnet.

Beispiel:

R L A 6A | 00 | 08, 00 Basis 0 = 0000



3.8.4. Fakultatives Halt

KHLT

Key Halt

57

Dieser Befehl ermöglicht es, an einer bestimmten Stelle der Bearbeitung ein fakultatives Halt in das Programm einzuschieben, z.B. am Anfang des Druckens auf ein Blatt.

Der Halt wird durch die Bedienung mit der Drucktaste KHLT freigegeben.

Wenn die Drucktaste KHLT eingelegt ist:

- hält das Programm an dem Befehl KHLT an.
- ein Druck auf die Taste START hebt den Halt auf, das Programm fährt mit dem folgenden Befehl fort. Wenn ein neuer KHLT-Befehl erkannt wird, hält das Programm aufs neue an, aber nur, wenn die Taste KHLT eingelegt bleibt.

Wenn die Drucktaste KHLT nicht eingelegt ist:

- ist der KHLT-Befehl ohne Wirkung. Das Programm läuft ohne Halt weiter.

3.9. Die Pseudobefehle

Die Pseudobefehle der Grundsprache dienen teils der Steuerung des Arbeitssystems und beziehen sich auf das Laden eines Programms in den Kernspeicher, teils sind es feste Werte, die an bestimmten Stellen des Zentralspeichers gespeichert werden.

Da sie keine Beziehung zur Maschinensprache haben, entfällt bei ihnen auch der entsprechende Binärwert des Operationstypes. Sie setzen sich lediglich aus dem symbolischen OT und evtl. einigen Parametern oder festen Werten zusammen. Diese werden symbolisch dargestellt, so wie sie in den folgenden Abschnitten beschrieben werden.

Man unterscheidet zwei Arten von Pseudobefehlen:

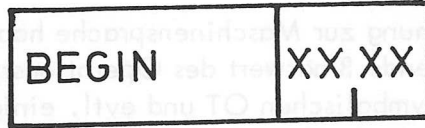
- organisatorische Pseudobefehle, die den Anfang und das Ende eines Programms festlegen sowie die vorläufigen Niveaus (Anfang der Sektionen).
- Pseudobefehle zur Erstellung von "Konstanten" (festen Werten), die feststehende und immer wieder beim Ablauf der Programme benutzten Daten eingeben.

3.9.1. Organisation eines Programmes

3.9.1.1. Programmanfang

BEGIN

BEGINning adress where the program is loading



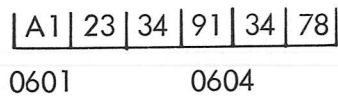
xxxx = absolute Adresse

Dieser Pseudobefehl steht immer am Anfang eines Programmes und gibt dem Ladeprogramm die Adresse im Zentralspeicher an, von der ab die Befehle eingespeichert werden, die diesem Befehl folgen.

Beispiel:

BEGIN	06 01
ADD	A1 23 34
CMD	91 34 78

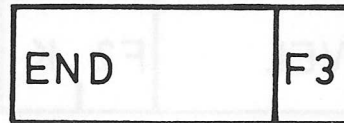
Zentralspeicher



3.9.1.2. Programmende

END

END of program



F3 = fester Wert (Trennmarke)

Dieser Pseudobefehl steht immer am Ende eines Programmes und zeigt an, daß der vorhergehende Befehl der letzte des betreffenden Programmes ist.

Er bewirkt das Einsetzen der Trennmarke F3 an die Stelle, die diesem Befehl folgt.

Beispiel:

IOÇ | 53 | 01 | CA | 08 |

END | F3 |

Zentralspeicher

