

Es erlaubt auf einfache Art, eine falsch liegende Karte einzuordnen. Diese Rubrik ist auf dem Programm nicht festgelegt, damit der Programmierer die Zahl der Spalten reservieren kann, die dafür unbedingt nötig sind. Er benutzt sie, als ob es sich um einen Kommentar handelte.

Art der Zeilen

Art der Zeile	Auszufüllende Rubriken					
	K A 2/3	Zahl Bytes 4/5 D	Ord.N 6-9 D	Symb.OT 10-16 H oder AN	Befehle 17-44 H oder AN	Kommen- tare und Kennz. AN
Elementar-Befehle	12	ja	ja	ja (a)	ja (a) hexad.OT wahlweise (b)	evtl. Sp.46-80 (d)
Pseudo-Befehle BEGIN END LEVEL NØP HEXAD AN	15 17 14 12 32 31	nein 01 02 01 ja ja	} ja (c) (c)	} ja (a)	ja gem. Befehl (a)	d
Kommentar	12	nein	ja	nein	nein	ja von Sp.34 an (d)

Bemerkungen:

- Linksbündig nicht benutzte Spalten bleiben rechts frei
- Die Spalten 17 und 18 bleiben frei, wenn der hexadezimale O.T. nicht angegeben wird.
- Enthält eine Rubrik einen Parameter, ist die Adresse dieses Zeichens in Spalte 17 angegeben.
- Beliebige Form

Abkürzungen:

- D = dezimal
- AN = alphanumerisch
- H = hexadezimal

### Ratschläge für die Ausarbeitung von Programmen

Um ein Programm auszuarbeiten, muß der Programmierer die Programmiersprache beherrschen. Unklarheiten können mit der vorliegenden Unterlage oder mit der Unterlage über die Programmiersprache für die Magnettrommel beseitigt werden. Die Ausarbeitung wird enorm vereinfacht, wenn der Programmierer sich die Mühe macht, sich ein zusammenfassendes Merkblatt anzulegen, das seinen Bedürfnissen angepaßt ist und das die Formate und besonderen Regeln für die Kodierung in eine gedrängte und schematisierte Form bringt.

Er muß außerdem die Normen kennen, die für die Benutzung der Software erforderlich sind, z. B., Anrufsnormen der Unterprogramme, maximale Kapazität der Segmente etc.

Im allgemeinen enthält ein Programm einen ersten Teil, dessen Aufgabe es ist, die Zentraleinheit zu "initialisieren".

Diese Initialisierung besteht darin, vor der Arbeit die benötigten Informationen in den Kernspeicher einzulesen und evtl. noch erforderliche Programmteile anzurufen

Zur Erstellung eines Programms ist in "umgekehrter" Reihenfolge vorzugehen, d. h. am besten wird zunächst der Programmablauf fixiert. Dabei werden Anmerkungen vorgenommen, die Reservierungen von Feldern bzw. das Anlegen von Konstanten betreffen. Ist der Programmablauf festgelegt, können dann aufgrund der besseren Übersicht die entsprechenden Definitionen einfach und leicht erstellt werden.

### Ratschläge für die Ausarbeitung

- In jedem Programm wird die erste Linie für den Pseude-Befehl "BEGIN" reserviert. Die in jedem Befehl enthaltene Adresse kann im Augenblick des Durchlaufs durch die Maschine festgelegt werden.
- Die großen Unterteilungen des Programms (Programme, Segmente, Unterprogramme, wichtige Sektionen, Vorprogramme) werden auf getrennte Formulare geschrieben.
- Jedem Programmteil ist eine Serie von laufenden Nummern zuzuordnen, wobei darauf zu achten ist, daß möglichst weite Abstände zwischen den einzelnen Nummern bestehen bleiben. Dadurch wird vermieden, daß die gesamte Numerierung neu vorgenommen werden muß, wenn eventuelle Modifikationen in bereits bestehende Serien eingefügt werden.

- Die Programme sind mit Hilfe zahlreicher Kommentare zu erläutern.
- Der Pseudo-Befehl "END" am Ende eines jeden Programms darf nicht vergessen werden.

#### Technische Ratschläge

- Übertragungen von Operanden von Rechenoperationen:  
Vorzugsweise sind Mehrfachbefehle zu verwenden. Mehr zusammenfassende Funktionen, weniger platzraubende Befehle, automatische Verarbeitung der in den Ziffern enthaltenen Vorzeichen, richtiges Löschen der Register.
- Löschen der Register:  
Um ein Einfach- oder Doppelregister zu löschen, wird eine Registeroperation verwendet; um mehr als 4 Register zu löschen, benutze man den Befehl "INC", der leistungsfähiger ist.
- Initialisationen:  
Man benutze vorzugsweise die Pseudo-Befehle der Konstantenerstellung, was den Vorteil hat, daß sie während des Programmladens ausgeführt werden und dann nicht mehr vorhanden sind. Dadurch wird eine unnötige Speicherbelegung vermieden.

#### Kontrolle

Bevor die Lochung auf der Karte vorgenommen ist, empfiehlt es sich, die folgenden Punkte zu prüfen, die Gelegenheit zu häufigen Fehlern bieten:

- Symbolische Kennzeichnungen  
In Sprungbefehlen geht "F1" voran, in den Pseudo-Befehlen LEVEL "F2".
- Pseudo-Befehle  
Kein hexadezimaler Operations-Typ. Die Parameter beginnen in Spalte 17.
- Verschiedene Übertragungen  
Mehrfach-Übertragungen von 3 Ziffern, denen der Buchstabe A vorangeht.
- Befehl INC  
Die zuerst angegebene Adresse als die andere
- Multiplikation  
Der Multiplikand, der weniger als 10 Ziffern enthält, ist in doppelter Länge zu übertragen, um das linke Register zu löschen.

- Verschieben  
in einem einfachen Register, wobei das vorhergehende (linke) mit benutzt wird.
- Löschen der Register  
Einsetzen von 0.

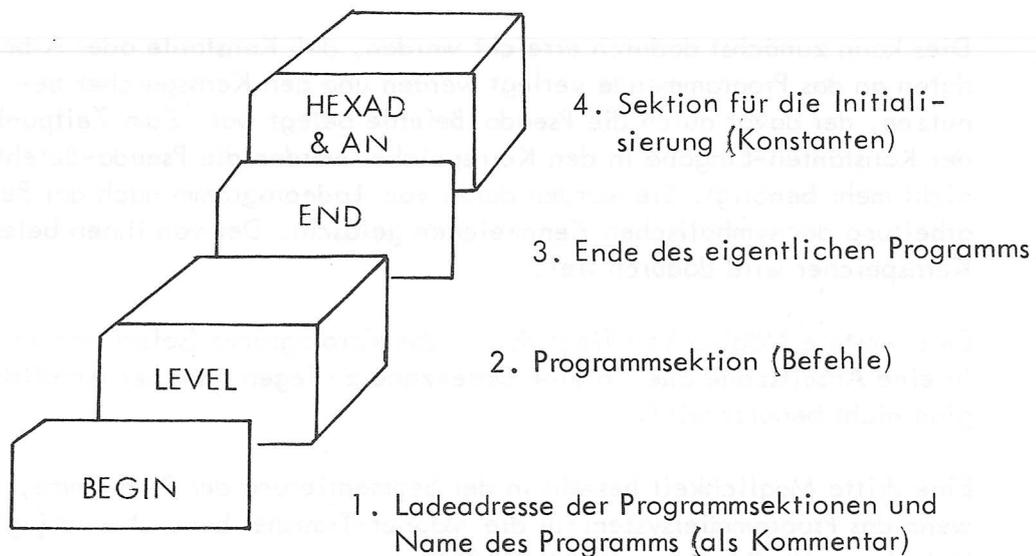
### 5.3.3. Erstellen und Test von Maschinenprogrammen

Die einmal geschriebenen Programme sind auf Programmkarten abgelocht, deren Bild dem Programmformular entspricht. Die gelochten Karten werden im Verhältnis von einer Karte pro Zeile interpretiert und kontrolliert. Danach kann der Programmierer die Aufstellung und das Testen des Maschinenprogramms vornehmen.

#### 5.3.3.1. Kontrolle, Übersetzung, Assemblierung der Sourceprogramme

Die erste Operation in der Maschine besteht aus der Kontrolle des Programmkarten-Pakets, wobei der formelle und logische Aufbau der durch den Programmierer geschriebenen Befehle überprüft und mittels der Standardprogramme, die in dem Programmiersystem enthalten sind.

Die zu überprüfenden Sourceprogramme werden auf folgende Weise zusammengesetzt:



Enthält ein Programm mehrere Einzelprogramme, so wird jedes einzelne von ihnen wie oben gezeigt, zusammengesetzt. Danach werden sie getrennt kontrolliert, und zwar in der Reihenfolge, in der sie in den Kernspeicher geladen werden. Vor jedem Durchlauf muß der Programmierer die Anfangsadresse des Einzelprogramms in die Karte BEGIN lochen. Es ist dabei darauf zu achten, daß evtl. reservierte Zonen bzw. Zonen für vorhergehende Programme frei bleiben.

Das Kontrollprogramm erstellt eine Liste der Programmkarten, wobei die tatsächlichen Kernspeicheradressen der Elementarbefehle mit ausgedruckt werden. Evtl. entdeckte Form- oder Kodierungsfehler werden ebenfalls auf der Liste angezeigt.

Der Programmierer korrigiert anhand dieser "Kontroll-Liste" evtl. vorhandene Fehler.

Bemerkung:

Bei der Prüfung des Kernspeicherbedarfs für ein Programm ist darauf zu achten, daß die Pseudo-Befehle LEVEL 2N Bytes belegen (N = Anzahl der Pseudo-Befehle). Diese Belegung erscheint nicht auf der Druckliste des Kontrollprogramms. Die Pseudo-Befehle werden nur beim Laden des Programms mit Hilfe des Ladeprogramms gelöscht. Übersteigt das Ursprungsprogramm bei der Eingabe der Programmkarten die Kernspeicherkapazität, kann das Programmladen nicht fortgesetzt werden.

In diesem Fall ist zu prüfen, ob die Kernspeicherbelegung reduziert werden kann.

Dies kann zunächst dadurch erreicht werden, daß Konstante oder Arbeitsdaten an das Programmende verlegt werden und den Kernspeicher benutzen, der davor durch die Pseudo-Befehle belegt war. Zum Zeitpunkt der Konstanten-Eingabe in den Kernspeicher werden die Pseudo-Befehle nicht mehr benötigt. Sie werden daher vom Ladeprogramm nach der Bearbeitung der symbolischen Kennzeichen gelöscht. Der von ihnen belegte Kernspeicher wird dadurch frei.

Eine weitere Möglichkeit liegt darin, das Vorprogramm (sofern vorhanden) in eine Arbeitszone oder in eine Datenzone zu legen, die bei Arbeitsbeginn nicht benutzt wird.

Eine dritte Möglichkeit besteht in der Segmentierung der Programme, wenn das Programmiersystem für die Magnet-Trommel benutzt wird (vgl. Unterlagen für Programmiersystem TB).

Welche Lösung auch immer angewendet wird, es ist empfehlenswert, den Kernspeicher nie vollständig zu belegen, da dadurch spätere Modifikationen u.U. unmöglich gemacht werden. Es könnte in diesem Fall vorkommen, daß das Programm vollständig neu erstellt werden muß, wenn eine unumgängliche Modifikation vorgenommen werden soll.

Nach der Korrektur wird das Ursprungsprogramm einer erneuten Kontrolle durch das Kontrollprogramm unterworfen. Dabei werden die symbolischen OT's übersetzt und das übersetzte Programm auf Karten ausgegeben. Eine Karte enthält in diesem Fall nur einen Befehl. Dieses übersetzte Programm wird nun für Testläufe innerhalb des Operating-Systems für Karten benutzt. Das Ursprungsprogramm wird archviert. Durch diesen zweiten Kontrolllauf wird überdies eine Kontrollliste des korrigierten Programms erstellt.

Vor Beginn der Testarbeiten wird das Programm innerhalb des Programmsystems TB durch den Assembler bearbeitet. Dabei erfolgt eine Assemblierung des Programms im Hinblick auf die Verarbeitung des Programms von der Trommel her. Das auf diese Art umgewandelte und auf der Trommel gespeicherte Programm kann direkt für Testläufe geladen und gestartet werden.

#### 5.3.3.2. Testläufe des Programms

Nach der ersten Formalkorrektur und der Übersetzung und Assemblierung des Programms können die Testläufe begonnen werden. Durch die Testläufe werden die logischen und technischen Fehler des Programmierers festgestellt. Die Testläufe sollen daher dem tatsächlichen Arbeitsablauf des Programms soweit als möglich entsprechen. Es sind so viele Testläufe durchzuführen wie Programmverzweigungen vorhanden sind, wobei möglichst vollständige Testdaten als Grundlage dienen sollen.

Der Programmtest selbst besteht in einem Start des Programms, wobei das entsprechende Operating-System benutzt wird. Es ist dabei so vorzugehen, als ob das Programm für einen tatsächlichen Arbeitsablauf gestartet würde. Beim Operating-System Lochkarten wird das Programm mit Hilfe des Ladeprogramms, beim Operating-System für Magnet-Trommeln mit Hilfe des Monitors geladen.

Beim Test sind nicht nur die Bearbeitungsfunktionen des Programms zu prüfen, sondern auch die spezifischen Wiederanlaufroutrinen im Programm zu testen. Darüber hinaus sind die Standardprogramme für die Programmunterbrechung (OS Karten) bzw. die Supervisor-Funktionen (OS Trommel) zu überprüfen. Dies wird durch Simulation der verschiedenen Fehlerfälle erreicht.

Die Korrektur der beim Test erkannten Fehler wird im Ursprungsprogramm vorgenommen, um es auf dem neuesten Stand zu halten. Dadurch wird auch vorausgesetzt, daß nach den Korrekturen sämtliche bis zu diesem Punkt beschriebenen Arbeiten wiederholt werden müssen. Um eine zu häufige Wiederholung dieser Arbeitsgänge zu vermeiden, empfiehlt es sich, mehrere Testläufe auf einmal durchzuführen und erst dann die erforderlichen Korrekturen im Ursprungsprogramm vorzunehmen.

#### 5.3.3.3. Arbeitsaufnahme mit dem Programm

Nach der Durchführung aller Testläufe und Korrekturen erfolgt eine letztmalige Redaktion des Programms für die endgültige Arbeitsaufnahme. Zu diesem Zweck wird das Programm in den Kernspeicher geladen und mit Hilfe des Programms "auf Programmkarten verdichten" in verdichtete Programmkarten gestanzt. Eine verdichtete Programmkarte enthält in der Regel mehrere Programmbefehle.

#### 5.4. Betrieb der Programme

In diesen Abschnitten wurden nur die Arbeitsgänge innerhalb der einzelnen Operating-Systeme kurz schematisiert. Eine genaue Beschreibung der Operating-Systeme befinden sich im Abschnitt 5.2.

6. Tabellen

6.1. Die Codes

Für jeden Code steht der gleiche Zeichenvorrat zur Verfügung. Das erleichtert den Übergang von einem zum anderen Code. Die verwendeten Zeichen entsprechen den Zeichen des Druckers.

6.1.1. Die externen Codes

6.1.1.1. Die Lochcodes

Diese Codes werden nicht automatisch in den internen Code (oder umgekehrt) umgeschlüsselt. Für jeden gibt es einen speziellen Zwischencode.

Die Tafeln auf den folgenden zwei Seiten zeigen für jeden Code:

- das dargestellte Symbol
- den Zwischencode
- den Lochcode
- die hexadezimale Darstellung im ISO-Code.

# H 14.012

NCR

A

Zeichen	Zwischen code	Lochcode			ISO - code - IEX	Zeichen	Zwischen code	Lochcode			ISO - code - VEX
⌘	20				20	P	57	11	7		50
0	30		0		30	Q	58	11	8		51
1	31		1		31	R	59	11	9		52
2	32		2		32	S	22	0	2		53
3	33		3		33	T	23	0	3		54
4	34		4		34	U	24	0	4		55
5	35		5		35	V	25	0	5		56
6	36		6		36	W	26	0	6		57
7	37		7		37	X	27	0	7		58
8	38		8		38	Y	28	0	8		59
9	39		9		39	Z	29	0	9		5A
.	4B	12	8	3	2E	\$	5B	11	8	3	24
/	21	0	1		2F	] <	4C	12	8	4	5D
#	3B		8	3	23	@	3C		8	4	40
%	2C	0	8	4	25	&	40	12			26
-	50	11			2D	,	2B	0	8	3	2C
*	5C	11	8	4	2A	' ^	5F	11	8	7	27
A	41	12	1		41	( (	4D	12	8	5	28
B	42	12	2		42	< +	4E	12	8	6	3C
C	43	12	3		43	\ !	4F	12	8	7	5C
D	44	12	4		44	↑	70	11	0		5E
E	45	12	5		45	;	5E	11	8	6	3B
F	46	12	6		46	=	2D	0	8	5	3D
G	47	12	7		47	" >	2E	0	8	6	22
H	48	12	8		48	! ?	2F	0	8	7	21
I	49	12	9		49	> =	3E		8	6	3E
J	51	11	1		4A	? "	3F		8	7	3F
K	52	11	2		4B	← \	2A	0	8	2	5F
L	53	11	3		4C	: '	3D		8	5	3A
M	54	11	4		4D	+	60	12	0		2B
N	55	11	5		4E	[ :	3A		8	2	5B
O	56	11	6		4F	) )	5D	11	8	5	29

11-2-8  
 ✓ 4-8  
 ✓ 12  
 5-8  
 ✓ 12-5-8  
 12-4-8  
 0-2-8  
 6-8  
 7-8  
 12-7-8  
 0-6-8  
 0-7-8  
 2-8  
 12-6-8  
 12-2-8  
 ✓ 11-5-8  
 5E  
 5F  
 11-7-8  
 0-5-8

GE-55

# T 121

Zeichen	Zwischen code	Lochcode			ISO-code	Zeichen	Zwischen code	Lochcode			ISO-code
⌘	70				20	P	55	8		4	50
0	71			0	30	Q	56	8		5	51
1	72			1	31	R	57	8		6	52
2	73			2	32	S	E8		9	11	53
3	74			3	33	T	61		9	0	54
4	75			4	34	U	62		9	1	55
5	76			5	35	V	63		9	2	56
6	77			6	36	W	64		9	3	57
7	30			7	37	X	65		9	4	58
8	50			8	38	Y	66		9	5	59
9	60			9	39	Z	67		9	6	5A
.	F8			11	2E	\$	A0	7	9	12	24
/	20	7	9		2F	J	45	8	9	4	5D
#	21	7	9	0	23	@	47	8	9	6	40
%	25	7	9	4	25	&	26	7	9	5	26
-	43	8	9	2	2D	,	C8	8	9	11	2C
*	F0			12	2A	'	E0		9	12	27
A	B8	7		11	41	(	41	8	9	0	28
B	31	7		0	42	<	F2		1	12	3C
C	32	7		1	43	\	A8	7	9	11	5C
D	33	7		2	44	↑	F9		0	11	5E
E	34	7		3	45	;	27	7	9	6	3B
F	35	7		4	46	=	40	8	9		3D
G	36	7		5	47	"	24	7	9	3	22
H	37	7		6	48	!	F1		0	12	21
I	D0	8		12	49	>	C0	8	9	12	3E
J	D8	8		11	4A	?	46	8	9	5	3F
K	51	8		0	4B	←	23	7	9	2	5F
L	52	8		1	4C	:	22	7	9	1	3A
M	53	8		2	4D	+	44	8	9	3	2B
N	54	8		3	4E	[	FA		1	11	5B
O	B0	7		12	4F	)	42	8	9	1	29

6.1.2. Der (interne) ISO-Code

Die folgende Tafel gibt eine Übersicht über den ISO-Code.

Die mit Spalte bezeichneten Ziffern von 0 - 7 geben die Bitbelegung des linken Halbbytes an und die mit Zeile bezeichneten Ziffern von 0 - F die Belegung des rechten Halbbytes

Beispiel: Buchstabe G = 47 im ISO-Code, Spalte 4 und Zeile 7 ergeben im Schnittpunkt den Buchstaben G

SPALTE	0	1	2	3	4	5	6	7	ISO CODE
Binär darstell.	0 0 0 0	0 0 0 1	0 1 0 0	0 1 0 1	1 0 0 0	1 0 0 1	1 1 1 0	1 1 1 1	ZEILE
0000			⌘	0	@	P			0
0001			!	1	A	Q			1
0010			"	2	B	R			2
0011			#	3	C	S			3
0100			\$	4	D	T			4
0101			%	5	E	U			5
0110			&	6	F	V			6
0111			'	7	G	W			7
1000			(	8	H	X			8
1001			)	9	I	Y			9
1010	Z		*	:	J	Z			A
1011	WRZ		+	;	K	[			B
1100	S		,	<	L	\			C
1101	WR		-	=	M	]			D
1110	V		.	>	N	↑			E
1111	N		/	?	O	←			F

6.2. Befehlskatalog

6.2.1. Ablaufzeiten der Befehle (in Millisekunden)

SPRÜNGE (3.2.)		
	durchgeführt	nicht durchgeführt
JRT	0.8	-
JIERT	1.5	1.3
JIURT	1.5	1.3
NOP	-	0.35
REGISTEROPERATIONEN (3.3)		
Arithmetische Operationen (3.3.2.)		
ADD	2.0	
ADDD	2.7	
SBD	2.0	
SBDD	2.7	
MPD	14.42 + 6.28 Z + 2.42 wenn $m > 0$ und $M < 0$ + 1.60 " $m < 0$ " $M > 0$ Z = Zahl der Wertstellen des Multiplikators m = Multiplikator; M = Multiplikand	
DVD	133,6 + (12,4 X 9)  9 = Anzahl der Ziffern des Quotienten	
Vergleichsbefehle (3.3.2)		
	gleich	ungleich
CMD	1.7	2.0
CMDD	2.0	2.6
Versetzen und Rundung (3.3.3)		
SLD	1.0 + 0.9 V    V = Zahl der	
SRD	1.0 + 0.9 V    Versetzungen	
RØUND	5.0	

Registerüberträge (3.3.4.)	
MRR	1.5
MDRR	1.7
MSRDR	1.7
EXR	1.6
EXDR	2.0
LRN	1.4 + 0.08 N (= N im Befehl)
OPERATION AUF ZEICHENBASIS (3.4.) Logische Operationen	
CMC	bei Gleichheit : 2.0 + 0.07 N bei Ungleichheit : 2.3 + 0.10 N
NI	1.6
ØI	1.1
Überträge im Normalspeicherbereich (3.4.3.)	
MVC	1.9 + 0.05 N
MVIC	2.2 + 0.05 N
INC	2.0 + 0.20 N
PKH	2.0 + 0.20 N
UPH	2.0 + 0.17 N
MEHRFACHÜBERTRÄGE (3.5.4 und 3.5.5)	
MR <sub>9</sub> <sup>8</sup>	4.7 + 0.19 (a) (b)
MR8G	4.7 + 0.18 N + 0.05 N (a) (b)
MR89	5.6 + 0.23 N (a) (b)
M <sub>7</sub> <sup>6</sup> R89	6.6 + 0.28 N (a)
M <sub>7</sub> <sup>6</sup> R <sub>9</sub> <sup>8</sup>	5.4 + 0.23 N (a)
M <sub>7</sub> <sup>6</sup> R	4.4 + 0.19 N (a)
M <sub>7</sub> <sup>6</sup> 89	3.8 + 0.09 N
M <sub>7</sub> <sup>6</sup> 8G	3.3 + 0.05 N + 0.05 N
M <sub>7</sub> <sup>6</sup> <sub>9</sub> <sup>8</sup>	3.3 + 0.05 N
	(a) zuzüglich : + 1.0, wenn N > 9 (b) " : + 0.4, " negativ

EIN/AUSGABE (3.6)	
ØC ØIC	Übernahme des Befehls: $2.0 + 1. E$ Durchführung: variabel ( $E = \text{Zahl der Einheiten}$ )
UMSCHLÜSSELUNGEN (3.7)	
TRx	$1.1 + 0.10 N$
MULTIPROGRAMMING (5.3.3)	
STØP START PRØ	$1.0 + 0.1 n$ 0.7 bis 1.0 offen: 1.0 geschlossen: $1.0 + 0.1n$
FREE	0.7 bis 1.0 $n = \text{Zahl der Bytes der zu übertragenden Registerzonen.}$

6.2.2. Befehlskatalog in alphabetischer Ordnung

OT	Funktion	Abschn.
ADD	Addition in einfacher Länge	3.3.2.1.
ADDD	Addition in doppelter Länge	3.3.2.1.
CMC	Alphanumerischer Vergleich	3.4.2.1.
CMD	Algebr. Dezimalvergleich (einfacher Länge)	3.3.2.4.
CMDD	Algebr. Dezimalvergleich (doppelter Länge)	3.3.2.4.
DVD	Algebr. dezimale Division	3.3.2.3.
EXDR	Austausch des Inhalts von 2 Doppelregistern	3.3.4.3.
EXR	Austausch des Inhalts von 2 Einfachregistern	3.3.4.3.
FREE	Freigabe eines durch einen geschlossenen Schutz angehaltenen Programms	4.3.3.4.
INC	Einsetzen eines Zeichens	3.4.3.2.
IO̅C	Ein/Ausgabebefehl	3.6.6.
IO̅IC	Indirekter Ein/Ausgabebefehl	3.6.6.
JIERT	Sprung bei Gleichheit	3.2.2.2.
JIURT	Sprung bei Ungleichheit	3.2.2.2.
JRT	Systematischer Sprung	3.2.2.1.
KHLT	Fakultativer Halt	3.8.4.
LEVEL	Sprungniveau	<del>3.2.3.</del>
LØAD	Laden eines Programms	3.8.2.
LRN	Laden eines numerischen Registers	3.3.4.4.
MDRR	Übertrag eines Doppelregister	3.3.4.1.
MPD	Algebr. dezimale Multiplikation	3.3.2.2.
MRR	Übertrag eines Einfachregisters	3.3.4.1.
MSRDR	Übertrag eines Einfachregisters in ein doppeltes	3.3.4.2.
MVC	Übertragungsbefehl	3.4.3.1.
MVIC	Indirekter Übertragungsbefehl	3.4.3.1.
Mxy(y)(y)	Mehrfachübertragungsbefehle	3.5.4.1.
Mx8G	Druckaufbereitungsbefehl	3.5.5.
NI	Logisches UND	3.4.2.2.
NØP	Nullbefehle (keine Operation)	3.2.4.
ØI	Logisches ODER	3.4.2.3.
PKH	Packen	3.4.3.3.
PRØ	Schutz	4.3.3.3.
PRSTØ	Kernspeicherausdruck	3.8.1.
RLA	Adressierung (reell) der Sprünge	3.8.3.
RØUND	Rundungsbefehl	3.3.3.3.
SBD	Subtraktion in einfacher Länge	3.3.2.1.
SBDD	Subtraktion in doppelter Länge	3.3.2.1.
SLD	Verschiebung nach links	3.3.3.2.
SRD	Verschiebung nach rechts	3.3.3.2.
START	Start eines Programms (durch ein anderes)	4.3.3.2.

39.13

6.2.2. Befehlskatalog in alphabetischer Ordnung

OT	Seite	Funktion	Abschn.
ADD	85	Addition in einfacher Länge	3.3.2.1.
ADDD	85	Addition in doppelter Länge	3.3.2.1.
CMC	103	Alphanumerischer Vergleich	3.4.2.1.
CMD	90	Algebr. Dezimalvergleich (einfacher Länge)	3.3.2.4.
CMDD	90	Algebr. Dezimalvergleich (doppelter Länge)	3.3.2.4.
DVD	88	Algebr. dezimale Division	3.3.2.3.
EXDR	100	Austausch des Inhalts von 2 Doppelregistern	3.3.4.3.
EXR	100	Austausch des Inhalts von 2 Einfachregistern	3.3.4.3.
FREE	105	Freigabe eines durch einen geschlossenen Schutz angehaltenen Programms	4.3.3.4.
INC	108	Einsetzen eines Zeichens	3.4.3.2.
IØC	133	Ein/Ausgabebefehl	3.6.6.
IØIC	133	Indirekter Ein/Ausgabebefehl	3.6.6.
JIERT	81	Sprung bei Gleichheit	3.2.2.2.
JIURT	81	Sprung bei Ungleichheit	3.2.2.2.
JRT	80	Systematischer Sprung	3.2.2.1.
KHLT	148	Fakultativer Halt	3.8.4.
LEVEL	152	Sprungniveau	<del>3.2.3.</del>
LØAD	146	Laden eines Programms	3.8.2.
LRN	101	Laden eines numerischen Registers	3.3.4.4.
MDRR	91	Übertrag eines Doppelregister	3.3.4.1.
MPD	82	Algebr. dezimale Multiplikation	3.3.2.2.
MRR	90	Übertrag eines Einfachregisters	3.3.4.1.
MSRDR	92	Übertrag eines Einfachregisters in ein doppeltes	3.3.4.2.
MVC	106	Übertragungsbefehl	3.4.3.1.
MVIC	106	Indirekter Übertragungsbefehl	3.4.3.1.
Mxy(y)(y)	102	Mehrfachübertragungsbefehle	3.5.4.1.
Mx8G	116	Druckaufbereitungsbefehl	3.5.5.
NI	104	Logisches UND	3.4.2.2.
NØP	113	Nullbefehle (keine Operation)	<del>3.2.4.</del>
ØI	105	Logisches ODER	3.4.2.3.
PKH	109	Packen	3.4.3.3.
PRØ	114	Schutz	4.3.3.3.
PRSTØ	144	Kernspeicherausdruck	3.8.1.
RLA	112	Adressierung (reell) der Sprünge	3.8.3.
RØUND	85	Rundungsbefehl	3.3.3.3.
SBD	85	Subtraktion in einfacher Länge	3.3.2.1.
SBDD	85	Subtraktion in doppelter Länge	3.3.2.1.
SLD	81	Verschiebung nach links	3.3.3.2.
SRD	82	Verschiebung nach rechts	3.3.3.2.
START	112	Start eines Programms (durch ein anderes)	4.3.3.2.

39.13

39.14

OT	Funktion	Abschn.
STØP 112	Halt eines Programms	4.3.3.1.
TRx 143	Umschlüsselung des Inhalts einer Zone	3.7.2.
UPH 111	Entpacken	3.4.3.4.
BELIN 170		
END 171		
HEYAD 174		
AN 175		
Kardmarke 192		
Heberloch 142		
Teileumschaltung 61		
Acceptability 212		

OT	Funktion	Abschn.
STØP TRx UPH	Halt eines Programms Umschlüsselung des Inhalts einer Zone Entpacken	4.3.3.1. 3.7.2. 3.4.3.4.

6.2.3. Befehlskatalog in hexadezimaler Reihenfolge

OT hexad.	L	OT symb.	Abschnitt	OT hexad.	L	OT symb.	Abschnitt
0 0	1	NØP	3.2.4.	6 0	5	LRN	3.3.4.4.
0 2	3	JRT	3.2.2.1.	6 1	4	NI	3.4.2.2.
0 4	6	JIERT	3.2.2.2.	6 3	4	ØI	3.4.2.3.
0 5	6	JIURT	3.2.2.2.	6 A	4	RLA	3.8.3.
1 2	4/5	MR8	3.5.4.1.	7 0	3	SLD	3.3.3.1.
1 4	4/5	MR9	3.5.4.1.	7 1	3	RØUND	3.3.3.3.
1 6	5/7	MR89	3.5.4.1.	7 2	3	SRD	3.3.3.2.
1 E	5/6	MR8G	3.5.5.	7 8	1	PRSTØ	3.8.1.
				7 D	1	LØAD	3.8.2.
2 1	4/5	M6R	3.5.4.1.				
2 2	4/6	M68	3.5.4.1.				
2 3	5/7	M6R8	3.5.4.1.				
2 4	4/6	M69	3.5.4.1.	8 0	6	MVC	3.4.3.1.
2 5	5/7	M6R9	3.5.4.1.	8 2	6	INC	3.4.3.2.
2 6	5/8	M689	3.5.4.1.	8 4	6	PKH	3.4.3.3.
2 7	6/9	M6R89	3.5.4.1.	8 5	6	UPH	3.4.3.4.
2 E	5/7	M68G	3.5.5.	8 8	6	CMC	3.4.2.1.
				8 A	6	MVIC	3.4.3.1.
3 1	4/5	M7R	3.5.4.1.				
3 2	4/6	M78	3.5.4.1.				
3 3	5/7	M7R8	3.5.4.1.	9 1	3	CMD	3.3.2.4.
3 4	4/6	M79	3.5.4.1.	9 2	3	CMDD	3.3.2.4.
3 5	5/7	M7R9	3.5.4.1.				
3 6	5/8	M789	3.5.4.1.				
3 7	6/9	M7R89	3.5.4.1.	A 1	3	ADD	3.3.2.1.
3 E	5/7	M78G	3.5.5.	A 2	3	ADDD	3.3.2.1.
				B 1	3	SBD	3.3.2.1.
4 0	1	STØP	4.3.3.1.	B 2	3	SBDD	3.3.2.1.
4 1	2	START	4.3.3.2.				
4 2	2	PRØ	4.3.3.3.	C 1	3	MPD	3.3.2.2.
4 4	2	FREE	4.3.3.4.				
				D 1	3	DVD	3.3.2.3.
5 0	2	TRØ	3.7.2.				
5 1	2	TRI	3.7.2.	E 1	3	MRR	3.3.4.1.
5 3	4/14	IØC	3.6.6.	E 2	3	MDRR	3.3.4.1.
5 4	4/14	IØIC	3.6.6.	E 3	3	MSRDR	3.3.4.2.
5 7	1	KHLT	3.8.4.	E 5	3	EXR	3.3.4.3.
5 A	2	TRA	3.7.2.	E 6	3	EXDR	3.3.4.3.
5 B	2	TRB	3.7.2.				
				F 2	2	LEVEL	3.2.3.

7. Die Software des GE-55

Wie jeder Computer ist auch der GE-55 mit einer Software ausgestattet. Sie besteht aus einer Reihe von Standardprogrammen, um dem Benutzer der Anlage bei dem Anlauf seiner Arbeitsprogramme zu helfen. Es sind mehrere Gruppen der Software vorhanden, die den einzelnen Konfigurationen der Anlage entsprechend für die verschiedene Methoden angewendet werden müssen: so gibt es zum Beispiel verschiedene Software für den GE-55 mit oder ohne Trommel.

7.1. Die Basissoftware

Die Anwendung der Basissoftware setzt einen GE-55 mit folgender Mindestausstattung voraus:

- einer Zentraleinheit mit:
  - Festspeicher MMS9
  - Zentralspeicher 2500, 5000 oder 10 000 Bytes
  - Pufferspeicher mit Sichtanzeige
- einem Kartenleser
- einer numerischen Tastatur
- einem Stanzer PS 40
- einem Drucker MB 50.

Die Arbeiten auf einer solchen Anlage werden aufgrund von Programmen durchgeführt, die in Karten gelocht und in den Zentralspeicher vor der Anwendung eingegeben wurden.

Die Basissoftware besteht im einzelnen aus folgenden Programmen:

a) zum Einstellen und Testen der Arbeitsprogramme:

- einem Kontrollprogramm
- einem Listprogramm
- einem Verdichtungsprogramm

b) zur Ausführung dieser Arbeitsprogramme:

- einem Ladeprogramm
- einem Löschmodul
- einem Unterbrechungsprogramm.