



Informations  
Systeme

Serie GE-55

BIBLIOTHEK

# SYSTEMBESCHREIBUNG

**BULL**  

---

**GENERAL  ELECTRIC**

Übersetzung der Informationen:

Ref.-Nr.: 23.20.001 F

Ref.-Nr.: 23.22.000 F

Ref.-Nr.: 23.20.001 F


## Systembeschreibung

November 1967

Neuauflage Juli 1968/Rev. Oktober 1969

**BULL**

**GENERAL  ELECTRIC**

GENERAL  ELECTRIC ist das eingetragene Warenzeichen der GENERAL-ELECTRIC-Company, USA

Ref.-Nr.: 23.20.001 D1



Übersetzung der Informationen:

Ref.-Nr.: 23.20.001 F

Ref.-Nr.: 23.29.006 F

Ref.-Nr.: 23.20.003 F3

November 1967

Neuauflage Juli 1968/Rev. Oktober 1969

Diese Veröffentlichung ist als rein informativ anzusehen. Für den Fall falscher Angaben entbindet sich BULL GENERAL ELECTRIC GMBH jeder Verbindlichkeit, da möglicherweise nicht alle laufenden Verbesserungen dieser Information den Empfänger erreichen können.

© Das Nachdrucken oder Kopieren dieser Veröffentlichung, auch auszugsweise, ist mit Genehmigung der BULL GENERAL ELECTRIC GMBH, Köln, gestattet.

Datum - Erstausgabe  
Datum - Updating

Seiten - Erstausgabe  
Seiten - Updating

Gesamtanzahl  
Seiten

Datum - Erstausgabe Datum - Updating	Seiten - Erstausgabe Seiten - Updating	Gesamtanzahl Seiten
Updating <b>1</b> Oktober 69	Ergänzung zur "GE-55 Systembeschreibung" Deckblatt, Inhaltsverzeichnis: I - II, Seiten: 70/1 - 70/17	21
<b>2</b>		
<b>3</b>		
<b>4</b>		
<b>5</b>		

Wir empfehlen Ihnen, zu Kontrollzwecken dieses Blatt nach der Titelseite  
in Ihr Handbuch einzuordnen.



Inhaltsverzeichnis

	<u>Seite</u>
1.	<u>Einführung</u> 1
1.1.	Einsatzmöglichkeiten 1
1.2.	Aufbau 1
1.3.	Die Zentraleinheit 1
1.4.	Die Software des GE-55 2
2.	<u>Die Hardware des GE-55</u> 3
2.1.	Die Zentraleinheit 3
2.1.1.	Allgemeine Beschreibung 3
2.1.2.	Die Informationen im Zentralspeicher 6
2.1.3.	Einteilung des Zentralspeichers 9
2.1.4.	Die Adressierung im Zentralspeicher 12
2.1.5.	Der Informationsfluß 18
2.1.6.	Der Programmablauf 22
2.1.7.	Bedienung der Maschine 25
2.2.	Die numerische Tastatur und die Leuchtanzeige 33
2.2.1.	Beschreibung 33
2.2.2.	Anwendung 33
2.3.	Die alphanumerische Tastatur 37
2.3.1.	Darstellung 37
2.4.	Der Sichtkartenleser 41
2.4.1.	Darstellung 41
2.4.2.	Funktionieren 41
2.4.3.	Durchführung 42
2.4.4.	Das Lesen von Talonkarten 44
2.5.	Der Kartenstanzer PS 40 45
2.5.1.	Darstellung 45
2.5.2.	Funktionieren 46
2.5.3.	Durchführung 48
2.6.	Der Drucker MB 50 53
2.6.1.	Beschreibung 53
2.6.2.	Arbeitsweise 54
2.6.3.	Bedienungselemente 56
2.7.	Der Drucker I 41 60
2.7.1.	Beschreibung 60
2.7.4.	Programmierung 61
2.8.	Die Magnettrommel 63
2.8.1.	Beschreibung 63
2.8.2.	Arbeitsweise 64
2.8.3.	Auswertung 66
2.9.	Lochstreifenleser/-stanzer 70/1
2.10.	Die Leitungssteuereinheit DATANET 51 70/11

	<u>Seite</u>
3. <u>Die Basisbefehle des GE-55</u>	71
3.1. Einführung	71
3.1.1. Darstellung der Grundsprache	71
3.1.2. Befehlsformat	73
3.1.3. Darstellung der Befehle	76
3.2. Die Befehle zur Programmverknüpfung	79
3.2.1. Adressierung der Sprünge	79
3.2.2. Sprungbefehle	80
3.3. Registeroperationen	83
3.3.1. Allgemeines	83
3.3.2. Rechenbefehle und algebraischer Vergleich	85
3.3.3. Versetzungs- und Rundungsbefehle	91
3.3.4. Übertragungsbefehle	96
3.4. Operationen auf Zeichenbasis	102
3.4.1. Allgemeines	102
3.4.2. Logische Befehle	103
3.4.3. Übertragungsbefehle im Normalspeicherbereich	106
3.5. Mehrfachüberträge	113
3.5.1. Darstellung	113
3.5.2. Allgemeiner Aufbau und Darstellung	113
3.5.4. Mehrfachüberträge (ohne Einsetzen eines Zeichens)	117
3.5.5. Überträge mit Einsetzen eines Zeichens	120
3.6. Ein-/Ausgabeoperationen	122
3.6.0. Die Zone IOC	122
3.6.1. Rolle und Aufgabe des Ein-/Ausgabebefehls	123
3.6.2. Organisation einer Ein-/Ausgabeoperation	124
3.6.3. Simultanbefehle mehrerer Operationen	126
3.6.4. Verhalten bei Störungen	128
3.6.5. Besonderheiten des Ein-/Ausgabebefehls	129
3.6.6. Die Befehle	133
3.6.6. Anschlußcodes für Randeinheiten	134
3.6.7. Die vertikale Simultaneität	138
3.7. Die Umschlüsselungsbefehle	140
3.7.1. Allgemeines	140
3.7.2. Die Umschlüsselungsbefehle	143
3.8. Die Hilfsbefehle	144
3.8.1. Kernspeicherausdruck	144
3.8.2. Laden des Kernspeichers	146
3.8.3. Adressierung der Sprungstufen	147
3.8.4. Fakultatives Halt	148
3.9. Die Pseudobefehle	149
3.9.1. Organisation eines Programmes	150
3.9.2. Erstellung von Konstanten	154

	<u>Seite</u>
4. <u>Simultaneität und Multiprogramming beim GE-55</u>	157
4.0. Vorbemerkungen	157
4.1. Allgemeine Einführung in die Simultaneität	157
4.2. Die Simultaneität der Ein-/Ausgaben des GE-55	162
4.2.1. Darstellung	162
4.2.2. Anwendung	162
4.2.3. Ablauf der Organisationen	163
4.2.4. I 41 und langsame Randeinheiten	166
4.3. Multiprogramming beim GE-55	168
4.3.1. Darstellung	168
4.3.2. Anwendungsweise	169
4.3.3. Die Befehle	173
4.3.4. Beispiele	176
5. <u>Die Bedienung des GE-55</u>	179
5.1. Einführung	179
5.2. Die Software des GE-55	179
5.2.1. Lochkartenanlage GE-55 (ohne Trommel)	179
5.2.2. GE-55 mit Trommel	180
5.2.3. Bibliothek der Service- und Standardverarbeitungsprogramme	184
5.3. Erstellen eines Programms	186
5.3.1. Verfahren	186
5.3.2. Erstellen der Unterprogramme	187
5.3.3. Erstellen und Test von Maschinenprogrammen	195
5.4. Betrieb der Programme	198
6. <u>Tabellen</u>	199
6.1. Die Codes	199
6.1.1. Die externen Codes	199
6.1.2. Der (interne) ISØ - Code	202
6.2. Befehlskatalog	203
6.2.1. Ablaufzeiten der Befehle	203
6.2.2. Befehlskatalog in alphabetischer Ordnung	206
6.2.3. Befehlskatalog in hexadezimaler Reihenfolge	208



	<u>Seite</u>
7. <u>Die Software des GE-55</u>	209
7.1. Die Basis-Software	209
7.1.1. Aufbau der Basis-Software-Programme	210
7.2. Die Basis-Software-Programme	211
7.2.1. Die Programmkartenkontrolle (PKK)	211
7.2.2. Das Listprogramm (PK-Listen)	216
7.2.3. Das Verdichtungsprogramm (MEFE)	219
7.2.4. Das Ladeprogramm (ICARE)	221
7.2.5. Das Löschmodprogramm	225
7.2.6. Das Unterbrechungsprogramm (PGI)	226
7.3. Anwendung der Basis-Software-Programme	234
7.3.1. Das Laden eines Arbeitsprogrammes	234
7.3.2. Die Programmkartenkontrolle	236
7.3.3. Das Listen eines Arbeitsprogrammes	237
7.3.4. Das Verdichten eines Programmes	240
7.4. Wiederanlauf nach Unterbrechung bei Kartenstau auf dem Leser	241
7.5. Wiederanlauf nach Unterbrechung beim Stanzen in den Software-Programmen	242
8. <u>Programmunterlagen</u>	243
8.1. Allgemeines	243
8.2. Die verwendeten Formulare	243
8.2.1. Block- und Flußdiagramm	243
8.2.2. Karteneinteilung	243
8.2.3. Druckbild	243
8.2.4. Programmliste	243
8.2.5. Kernspeicherausdruck	244
8.2.6. Bedienungsanweisung	244
8.2.7. Kernspeicherbelegung	244
8.2.8. Programmformular	244
8.2.9. Allgemeine Beschreibung	244
8.2.10. Symbole und Register	245
8.3. Testunterlagen	245
8.3.1. Probekarten und 80/80-Liste	245
8.3.2. Druckformular und Summenkarten	245
8.4. Formularmuster	246-254
Programmliste des GE-55	255
Kernspeicherausdruck	256
GE-55 Befehle	257
Programmkarten	258

1. Einführung

1.1. Einsatzmöglichkeiten

Nach intensiven Marktanalysen wurde von BULL GENERAL ELECTRIC ein Computer entwickelt, der durch seine Universalität besticht:

- er ist ein Datenverarbeitungssystem, dessen logische Konzeption ein klares Abbild moderner Computer ist;
- er ist darüber hinaus aber auch ein Datenverarbeitungssystem, das direkte Dateneingaben über Tastaturen in den Verarbeitungsprozess erlaubt;
- er ist bei alledem ein echtes Datenverarbeitungssystem, weil er wachsen und größere interne und externe Speicherkapazität erhalten kann.

Unter diesen Gesichtspunkten ist der GE-55 die ideale Maschine für solche kleineren und mittleren Unternehmen, die aus preislichen und organisatorischen Gründen bisher den Schritt in die elektronische Datenverarbeitung nicht tun konnten. Er findet aber auch dort seine Einsatzmöglichkeiten, wo große und größte Unternehmen für die Lösung von Teilproblemen dezentrale Datenverarbeitung bevorzugen.

1.2. Aufbau

Der GE-55 besteht in seinem Grundmodell aus Zentraleinheit, Kartenleser, Kartenstanzer, Drucker, numerischer Tastatur mit Leuchtanzeige und alpha-numerischer Tastatur. Alle Elemente sind zu einer schreibtischgroßen Einheit zusammengefaßt und liegen im Griffbereich der Bedienungskraft. Die Maschine benötigt keine Klimatisierung, keinen doppelten Boden und läßt den Netzanschluß über eine normale Schukodose (220V, 10A) zu.

### 1.3. Die Zentraleinheit

Die Zentraleinheit des GE-55 stellt über 4 Kanäle - 3 normale (KN), für langsame Randeinheiten und 1 Hochleistungskanal (KR) für externe Speichereinheiten und für den schnellen Drucker - die Verbindung der Ein-/Ausgabe- und Speicherperipherie zum Zentralspeicher her. Drei zusätzliche Hochleistungskanäle lassen sich beim Benutzer einbauen.

Der Zentralspeicher ist als Kernspeicher ausgelegt und hat eine Kapazität von 2500, 5000 oder 10000 Bytes. Die Zykluszeit je Byte beträgt 7.9 Mikrosekunden. Interner Code ist der ISO-Code. Der Kernspeicher dient als Daten- und Programmspeicher. Ein Byte, 8 Datenbits und ein Prüfbit, speichert ein alphabetisches Zeichen oder ein oder zwei numerische Zeichen. Die Rechenoperationen erfolgen in gepackter Darstellung. Das Programm speichern im Kernspeicher belegt je Befehl 1 - 14 Bytes.

Der Kernspeicher enthält 100 je 5 Bytes große Register für die Durchführung von arithmetischen und logischen Operationen. Je zwei Register können zu einem Doppelregister zusammengefaßt werden. Werden in einem Programm nicht alle Register benötigt, können sie frei für Daten - oder Programmspeicherung verwendet werden.

Den Analysen der Befehle und ihrer Ausführung dient ein Festspeicher mit einer Kapazität von 1024 Worten zu je 36 Bits.

### 1.4. Die Software des GE-55

Die Software setzt sich wie folgt zusammen:

- Das Programmiersystem zur Vereinfachung der Programmierung enthält im wesentlichen:
  - Symbolsprachen zur eigentlichen Programmierung, von denen die Grundsprache in diesem Handbuch beschrieben wird. Spezielle Sprachen dienen zur Anwendung von externen Speichern.
  - Assembler zur Umwandlung der symbolisch geschriebenen Programme. Es assembliert die verschiedenen Teile und stellt die notwendigen Verbindungen her.
- Das Anwendungssystem besteht aus einer Reihe von Standardprogrammen zur Durchführung der einzelnen Arbeiten auf dem Rechner.

Diese Systeme laden die durchzuführenden Programme in den Speicher, starten sie und steuern ihre Durchführung.



2. Die Hardware des GE-55

2.1. Die Zentraleinheit

2.1.1. Allgemeine Beschreibung

Die Zentraleinheit besteht aus:

- einem Kernspeicher
- einem Festspeicher
- Steuerungsorganen
- Anschlußkanälen für die Randeinheiten
- Internen Stromkreisen.

2.1.1.1. Der Zentralspeicher

Der Zentralspeicher ist ein Magnetkernspeicher. Auf allen Bytes können Informationen gelesen oder geschrieben werden. Beim Schreiben wird die vorher an dieser Stelle gespeicherte Information gelöscht.

Die Kapazität des Speichers und seine Grundzyklen werden in Abschnitt 1.3. beschrieben.

Seine Organisation wird im einzelnen in den Abschnitten 2.1.2. ff erläutert.

2.1.1.2. Der Festspeicher

Der Festspeicher ist das Hauptsteuerorgan des gesamten GE-55

Durch ihn wird das Arbeitsprogramm geladen und alle programmierten Befehle durchgeführt: Dazu analysiert er Befehl für Befehl, knüpft sie in der gewünschten Reihenfolge aneinander und löst sie zur Durchführung ihrer Aufgabe auf. Er steuert die Ein-/Ausgabe der Randeinheiten und führt den Wechsel der Programmserien oder der Programme durch. Das letztere kann entweder auf Grund einer Programmunterbrechung (s. 2.1.6.) oder durch die Multiprogrammierung (s.4.) erfolgen. Diese beiden Funktionen geben dem Festspeicher den Charakter einer Supervisors.

Um alle diese Aufgaben auszuführen, ist der Festspeicher ausgestattet mit:

- Mikroprogrammen zur Steuerung und Überwachung
- Mikroprogrammen zur Befehlsdurchführung

außerdem enthält er:

- arithmetische Rechentafeln
- Umschlüsselungstabellen für externe Codes.

Um die notwendigen Verbindungen mit dem Arbeitsprogramm herzustellen, ist in der Zentraleinheit eine Kommunikationszone dafür vorgesehen.

Die Mikroprogramme sind feste Programme zur Regelung der Elementarfunktionen des Rechners. In gleicher Weise, in der ein Arbeitsprogramm sich aus einer Folge von Befehlen zusammensetzt, setzt sich ein Befehl aus einer Reihe von Elementarfunktionen zusammen, die auch "Mikrofunktionen" genannt werden und die in einer bestimmten Ordnung aneinandergereiht werden. So besteht zum Beispiel ein Befehl für den Übertrag einer Anzahl von Zeichen von Adresse A nach Adresse B aus: Analyse des Operationstyps, Übernahme der ergänzten Parameter (Zahl der Stellen, Adressen) in Rechenregister, Übertragung der Zeichen nacheinander unter Fortrechnen der Adressen der abgebenden und empfangenden Zone, Abzählen der Zahl der Stellen und endlich - nach Beendigung des Übertrags - dem Weiterzählen des Programmadreßregisters, um den folgenden Befehl zu übernehmen.

Bis zur Gegenwart werden alle Elementarfunktionen durch ein Steuersystem aus Dioden oder Transistoren bestimmt, das aufgrund seiner "aktiven" Natur sehr störanfällig ist. Beim GE-55 werden diese Elementarfunktionen durch Mikroprogramme gesteuert, die im Festspeicher festgelegt sind. Aufgrund dieser neuen Technologie ist auch die Verdrahtung übersichtlicher als bei herkömmlichen Stromkreisen. Da sich auch die verschiedenen Befehle nur aus einer Kombination von Elementarfunktionen zusammensetzen, konnte auch die verdrahtete Logik relativ klein gehalten werden. Das ergibt eine im Vergleich zu ihrem Befehlskatalog sehr kleine Zentraleinheit.

DER INHALT DES FESTSPEICHERS LIEGT EIN FÜR ALLEMAL FEST AUFGRUND SEINER KONSTRUKTION. ER KANN VOM PROGRAMMIERER NICHT ANGERUFEN WERDEN.

#### 2.1.1.3. Steuerungsorgane

Die Steuerungsorgane sind Elemente, die zur Durchführung der vom Festspeicher abgerufenen Mikrofunktionen dienen: Vergleich zwischen 2 Bytes, Bit-Analyse, binäres und dezimales Weiter- bzw. Herunterzählen etc.

Fest Register sind zur Durchführung der Elementarfunktionen zugeordnet: Adreßregister, Pufferregister, Rechenregister etc.

#### 2.1.1.4. Die Anschlußkanäle

Die Randeinheiten sind über Kanäle mit der Zentraleinheit verbunden. Der GAMMA 55 hat 2 Arten von Kanälen: die "normalen" (N) für die langsamen Randeinheiten (Tastaturen etc.) und "schnellen" (R) für Einheiten mit größerer Leistung (z.B. Magnettrommel).

KANÄLE		EINHEITEN
NORMALE	1	KARTENLESER NUMERISCHE TASTATUR ALPHANUM. TASTATUR
	2	KARTENSTANZER
	3	DRUCKER MB 50
SCHNELLE	0	MAGNETTROMMEL 1
	1	MAGNETTROMMEL 2
	2	
	3	DRUCKER 141

Wenn mehrere Randeinheiten an einen Kanal angeschlossen sind (z.B. Kanal N1) kann jeweils nur eine mit der Zentraleinheit in Verbindung stehen.

Dasselbe gilt auch für die Kanäle, d.h. es hat immer nur ein Kanal Zugriff zur Zentraleinheit. Jedoch die Durchsatzrate der langsamen Kanäle ist so gering, daß zwischen dem Übertrag von zwei Zeichen es der Zentraleinheit möglich ist, sich mit anderen Kanälen zu verbinden und die Überträge dank des sehr schnellen Rhythmus des Zentralspeichers zu verschachteln, wodurch mehrere Randeinheiten über die verschiedenen Kanäle echt simultan arbeiten können.

#### Die internen Verbindungen

Diese Stromkreise haben die Aufgabe, die notwendigen Verbindungen zwischen den verschiedenen Teilen der Zentraleinheit herzustellen.







Um die interne Darstellung zu vereinfachen, ist das Byte in zwei fiktive Vierergruppen geteilt worden. Diese vier Bits werden HALBBYTE genannt. In Binärdarstellung kann also jedes Halbbyte einen Wert von 0 bis 15 darstellen. Ein Halbbyte wird "dezimal" genannt, wenn es nur die Werte von 0 - 9 enthält und "hexadezimal", wenn es alle Wertigkeiten enthalten kann. Durch Vereinbarung werden die Werte von 0 - 9 durch die jeweilige Ziffer und die Werte 10 - 15 durch die Buchstaben A - F ausgedrückt.

A = 10 ; B = 11 ; C = 12 ; D = 13 ; E = 14 ; F = 15 .

Diese sogenannte hexadezimale Darstellung wird von jetzt ab in diesem Handbuch benutzt.

Zur Adressierung wird das Byte der Stelle im Kernspeicher gleichgesetzt. Die Stellen werden von links nach rechts in aufsteigender Reihenfolge ihrer Nummern (oder Adressen) dargestellt. Die Informationen werden folgendermaßen geschrieben:

Höchste Wertstelle links, niedrigste Wertstelle rechts.

#### 2.1.2.2. Form der Speicherung

##### a) alphanumerische Information

Im Speicher wird die alphanumerische Information im internen ISO-Code (International Standard Organisation) dargestellt. Jedes Zeichen ist auf 7 Bits verschlüsselt, das 8. Bit ist immer gleich 0. Es belegt ein Byte. Es wird in hexadezimaler Form mit 2 Ziffern oder Buchstaben geschrieben.

Beispiel:

<u>Zeichen</u>	<u>Binärschlüssel</u>	<u>hexadezimal</u>
8	0, 0, 1, 1, 1, 0, 0, 0	3 8
M	0, 1, 0, 0, 1, 1, 0, 1	4 D

linkes            rechtes  
Halbbyte

Der interne ISO-Code wird in Form einer Tabelle in Abschnitt 6.1.2 wiedergegeben. Die Werte des linken Halbbytes werden im Kopf (= Spalte) und die des rechten Halbbytes in der Senkrechten rechts (= Zeile) abgelesen.

Das linke Halbbyte bestimmt die Serien des internen Codes:

- 0 und 1 = Hilfsschlüssel des GE-55
- 2 bis 5 = 64 druckbare Zeichen: 10 Ziffern, 26 Buchstaben und 18 Sonderzeichen einschließlich Leerzeichen.
- 6 und 7 = wird z.Zt. noch nicht benutzt.

Zu beachten ist, daß das Zeichen 00 im Kernspeicher von den Ein-/Ausgabeeinheiten nicht erkannt wird. Insbesondere darf es nicht benutzt werden, um eine Leerstelle, z.B. beim Drucken, zu bekommen.

b) numerische Informationen

Die Zahlen können in zwei Arten dargestellt werden, nämlich gepackt und ungepackt. Bei gepackter Darstellung können sie entweder absoluten Wert haben oder algebraischen.

Ungepackt

Die Zahlen stehen im Zentralspeicher wie die alphanumerischen Zeichen, nämlich im ISO-Code. Sie belegen die 10 Ziffern dieses Codes und werden hexadezimal so dargestellt: 30, 31...39. Ungepackt ist eine Zahl immer in absoluter Größe ausgedrückt. Ihr Vorzeichen steht nicht auf einer bestimmten Stelle, sondern wird durch ein beliebiges Zeichen dargestellt.

Gepackt

Zahlen können auch in gepackter Form ausgedrückt werden.

Da nur das rechte Halbbyte der Ziffern eine Unterscheidung bietet, genügt es also diese rechten Halbbytes herauszuziehen und sie fortlaufend, je 2 auf einem Byte, aneinanderzusetzen. Die Halbbytes einer gepackten Zahl stellen also auch die Dezimalstellen dieser Zahl dar. Steht eine Zahl im absoluten Wert und hat sie eine ungerade Anzahl Stellen, wird das linke Halbbyte auf Null gesetzt.

Beispiel: 34.962

ungepackt

gepackt

,20,33,34,39,36,32,20,

,03,49,62,

Die Anzahl der von einer gepackten Zahl belegten Bytes ist gleich dem Quotient  $\frac{N + 1}{2}$  (N = Zahl der ungepackten Stellen).

### Zahlen im algebraischen Wert

Stellt eine Zahl einen Rechenfaktor dar, steht sie gepackt in einem algebraischen Wert. Das heißt, daß sie links durch ein oder zwei Halbbytes ergänzt wird, die das Vorzeichen bedeuten und sie begleiten. Sie belegt immer volle Bytes in folgender Weise:

- ist die Zahl positiv, genügt es, sie in ihrem absoluten Wert auszudrücken, der links durch eine oder zwei Nullen ergänzt wird.
- ist die Zahl negativ, wird sie ersetzt durch ihr Komplement  $10^{n+1}$ , wenn N ungerade ist, oder zu  $10^{n+2}$ , wenn N gerade ist. Das linke Halbbyte ist immer gleich 9.

N bedeutet die Zahl der Wertstellen. Die gepackte Zahl belegt im algebraischen Wert  $\frac{N+2}{2}$  Bytes.

#### Beispiele:

+ 19 562	<u>01,95,62</u>
- 19 562	<u>98,04,38</u>
+ 8 439	<u>00,84,39</u>
- 8 439	<u>99,15,61</u>

Das 1. linke Halbbyte wird "Vorzeichenstelle" genannt. Es ist immer 0 oder 9, je nachdem die Zahl positiv oder negativ ist.

#### c) Programmlinien

Die Programmlinien stellen unter einer verschlüsselten Form die Befehle an die Maschine dar. Sie werden in hexadezimaler Schreibweise dargestellt und benutzen die 16 möglichen Kombinationen eines Halbbytes (0 - 9 und A - F).

### 2.1.3. Einteilung des Zentralspeichers

Die ersten 85 Stellen sind für Hardware vorgesehen. Dort speichert der Rechner die notwendigen Informationen zur Steuerung der Programmabwicklung. Danach folgt eine Registerzone, die aus spezialisierten und numerischen Registern besteht. Der Rest des Speichers ist der Normal-speicherbereich.

### 2.1.3.1. Kommunikationszonen

Es gibt vier reservierte Zonen:

- 00 - 30 Logische Zone (zur Abwicklung der Befehle)
- 31 - 59 IØC-Zone (Input Output Control)  
In dieser Zone erfolgt die Überwachung der Ein- und Ausgabeoperationen mit den peripheren Einheiten.
- 60 - 69 RFA - Zone (Registers Fields Adresses).  
Diese Zone enthält die Adressen der Registerzonen der verschiedenen Programme, die sich im Kernspeicher befinden (für Multiprogramming).
- 70 - 84 PRC - Zone (Programs Control)  
Diese Zone wird benutzt, um den Ablauf der verschiedenen Programme bei Multiprogramming, die sich gleichzeitig im Kernspeicher befinden, zu überwachen.

Nur in einem ganz bestimmten Fall, der nur für die RFA-Zone gilt, und weiter unten erklärt wird, darf der Programmierer sich dieser Zonen bedienen. Jede Veränderung ihres Inhalts, z.B. durch Programmierfehler, würde den Ablauf des Programms stören.

### 2.1.3.2. Die spezialisierten Register

Diese Register, die sich an den Stellen 85 - 95 des Speichers befinden, können in zwei Arten unterteilt werden: die Testregister und die Programmregister. Der Programmierer hat Zugriff auf die verschiedenen Register und kann entweder deren Inhalt analysieren oder verändern.

#### Die Testregister

Diese Register geben Auskunft über (oder für) die Abwicklung gewisser Befehle.

Vorzeichenregister (0085) - **VZ** - . Dieses Register dient zur Erkennung der algebraischen Zahlen (s. 3.5.3.) bei den Mehrfachüberträgen. Es enthält das Vorzeichen der Zahl (Operanden oder Resultate) in hexadezimaler, ungepackter Form, und zwar:

20 oder 00, wenn die Zahl positiv oder Null,  
2D, wenn die Zahl negativ ist.

Vergleichsregister (0086) - **VG** - . Dieses Register enthält das Vergleichsergebnis nach einem durchgeführten Vergleich. Wird Operand 1 mit Operand 2 verglichen, so enthält dieses Register:

3C bei  $1 < 2$   
3D bei  $1 = 2$   
3E bei  $1 > 2$

Das Ergebnis bleibt bis zum nächsten Vergleich erhalten. Es wird durch die drei Vergleichsbefehle CMD, CMDD und CMC verändert.

Register Kapazitätsüberschreitung (0091) - **Kap** -. Dieses Register wird durch die Rechenbefehle geladen, wenn das Ergebnis größer als vorgesehen ist (s. 3.3.1.). Es enthält entweder das Zeichen AA (nach Division) oder das höchste Halbbyte des Ergebnisses (bei den anderen Rechenoperationen). Dieses Ergebnis bleibt bis zur nächsten Kapazitätsüberschreitung erhalten.

#### Die Programmregister

Diese Register dienen dem richtigen Ablauf der Programmlinien. Sie enthalten die Adresse eines Befehls in gepackter Form (4 Dezimalziffern auf zwei Bytes).

Programmadressregister (0089/90) - **PAR** -. In jedem Augenblick enthält dieses Register die Adresse des nächsten Befehls, der ablaufen soll. Bei jedem Befehlsschritt wird der Inhalt des Registers verändert.

Rücksprungadreßregister (0087/88) - **ANR** -. Nach jedem Sprung wird die Adresse des nächsten Befehls, der abgelaufen wäre, wenn kein Sprung erfolgt wäre, an dieser Stelle gespeichert. Das Register wird durch die Sprungbefehle geladen.

Unterbrechungsprogrammadressregister (0094/95) - **AFP** - speichert die Anfangsadresse eines Unterbrechungsprogramms (PGI), das am Anfang der Arbeit in den Kernspeicher geladen wurde.

Register der Rücksprungadresse nach Unterbrechungsprogramm (0092/93) - **ANU** - speichert die Adresse des nächsten Befehls, der abgelaufen wäre, wenn nicht in das Unterbrechungsprogramm gesprungen worden wäre. Die Adresse wird durch das System der "Programmunterbrechung" geladen.

#### 2.1.3.3. Die numerischen Register

Die numerischen Register sind Zonen von 5 Bytes, in denen obligatorisch gerechnet wird. Außerdem werden die algebraischen Vergleiche und die Versetzungen darin vorgenommen. 100 Register, von 00 - 99 numeriert, stehen dem Programmierer zur Verfügung (Stellen 0096 - 0595 einschl.).

Zahlen werden in algebraischer Form gepackt gespeichert. Ein Register enthält 9 Ziffern und ein Vorzeichen. Zwei nebeneinanderliegende Register können zu einem Doppelregister zusammengeschaltet werden. Dieses Doppelregister wird mit der Nummer des rechten Registers bezeichnet (höhere Nummer). In diesem Fall kann man bis zu 19 Ziffern und ein Vorzeichen verwerten.



Die Operationen, welche die Register benutzen (s.3.3.), erfolgen immer in fixer Länge, d.h. je nach gewählter Kapazität auf 5 oder 10 Bytes. Darum wird, wenn eine Zahl nicht 9 oder 19 Stellen erreicht, ihr Vorzeichen so oft wiederholt, wie die Kapazität des Registers es erlaubt.

Außer ihrer Verwendung bei der Verarbeitung von Werten, dienen die Register 0 - 9 auch zum Speichern von speziellen Adressen (s.2.1.4.). Diese Adressen, aus 4 Dezimalziffern bestehend, sind in gepackter Form auf den zwei rechten Bytes des Registers gespeichert. Die restlichen drei Bytes können als Normalspeicher benutzt werden (s. 2.1.3.4.). In Abschnitt 3.6.5 sind Beispiele dieser Benutzung der numerischen Register gegeben.

Anmerkung:

Ein Programm benutzt nicht immer alle 100 Register. Die nicht benutzten Register können als Normalspeicher Verwendung finden.

2.1.3.4. Normalspeicher

Von Stelle 0596 an oder bereits vorher, wenn die numerischen Register nicht benutzt wurden, ist der Speicher nicht spezialisiert. Die Einteilung dieses Speicherbereichs wird vom Programmierer bestimmt, und zwar nach Methoden, die in Abschnitt 2.1.4. beschrieben werden.

Im allgemeinen wird der Speicher enthalten:

- das durchzuführende Programm
- die zu verarbeitenden Daten und die auszugebenden Ergebnisse
- in gewissen Fällen Zusatzinformationen (Konstanten, Tabellen, Zwischenergebnisse etc.).

Alle in 2.1.2.2 angegebenen Arten der Einspeicherung sind im Normalspeicher möglich. Die Informationen werden in wirklicher Länge durch Befehle, die auf Zeichenbasis arbeiten, verarbeitet (s.3.4. - 3.7.). Hauptsächlich werden in diesem Speicherbereich die Ein- und Ausgaben der Informationen, ihre Umschlüsselung, logische Operationen und die Bereitstellung der Daten durchgeführt.

2.1.4. Die Adressierung im Zentralspeicher

2.1.4.1. Prinzipien

Um eine zusammenhängende Gruppe von Bytes anzusprechen, z.B. durch einen Programmbefehl, muß die Zentraleinheit ihren genauen Platz im Kernspeicher kennen.

Der Anfang einer Gruppe wird im allgemeinen durch die Adresse des ersten Bytes von rechts oder links, je nach Ablafrichtung des Befehls, angegeben.

Um das Ende einer Gruppe anzuzeigen, gibt es zwei Möglichkeiten:

1. Die Maschine kennt die Zahl der zu bearbeitenden Bytes.

Es handelt sich hier um spezialisierte Zonen von fester Länge, die durch bestimmte Befehle angesprochen werden (numerische Register beispielsweise). Durch die Analyse des Operationstyp des Befehls erkennt die Zentraleinheit welcher Art die Zone ist und kann daraus die Länge ableiten.

Die Anfangsadresse der Bytegruppe wird durch den Programmierer festgelegt und reicht aus, die Einspeicherung exakt zu beenden.

2. Die Maschine kennt nicht die Zahl der zu bearbeitenden Bytes.

Hierbei handelt es sich um Gruppen von beliebiger Anzahl Bytes, die Normalspeicherzonen bilden.

Außer der Anfangsadresse hat der Programmierer auch noch die Zahl der zu verarbeitenden Bytes zu bestimmen. Zwei Methoden ergeben das gewünschte Resultat:

- a) Die erste besteht darin, die Anzahl der Bytes im Befehl selbst festzulegen. Dieser kann dadurch die bereits bearbeiteten Bytes abzählen bis Gleichheit mit der angegebenen Zahl erreicht wird.

Diese Art wird bei den Befehlen angewandt die auf Zeichenbasis arbeiten. Allgemein werden nur kleinere Gruppen davon betroffen (logische Operationen, Speicherüberträge etc.)

- b) Die zweite Methode besteht darin, die Gruppe von Bytes durch ein Spezialzeichen zu begrenzen, der sogenannten "Trennmarke". Diese wird während des Ablaufs des Befehls festgestellt und beendet ihn.

Diese Art wird bei den Befehlen angewandt, die komplette Zonen bearbeiten. Diese Zonen sind im allgemeinen umfangreicher (Ein/Ausgabeoperationen, Umschlüsselungen etc.).

#### 2.1.4.2. Die Adressierung der Speicherzonen

Im Abschnitt 2.1.1.1 haben wir gesehen, daß jede Stelle des Zentral-speichers einzeln adressierbar ist. Sie wird darum von 0 bis 2499, 4999 oder 9999 numeriert. Die Nummer eines Bytes ist gleichzeitig seine Adresse.

Im Speicher stehen alle Adressen in fester Länge auf 2 Byte gepackt (vier Dezimalziffern).

Beispiel: Adresse 89 : 00 89

Die Halbbytes einer Stelle sind nicht getrennt adressierbar.

Zur Organisation der Arbeit teilt der Programmierer den Zentralspeicher in Zonen ein, deren Zahl, Bedeutung und Belegung von der jeweiligen Arbeit abhängen. Im allgemeinen müssen, außer den reservierten Zonen am Anfang des Speichers (s.2.1.3.1.) und der für das Programm bestimmten Zone, folgende Zonen vorgesehen werden:

- eine Registerzone für die Rechenoperationen
- eine oder mehrere Ein/Ausgabezonen für den Austausch mit den Rand-einheiten.
- eine oder mehrere Arbeitsfelder zum Zwischenspeichern von vorläufigen Ergebnissen.
- eine oder mehrere Speicherzonen, um permanente Daten am Anfang der Arbeit einzuspeichern (Konstanten).

Um diesen Erfordernissen Rechnung zu tragen, wird der Zentralspeicher des GE-55 in mehrere Zonen mit einer Länge von 1 - 999 Stellen eingeteilt. Unter diesen Zonen hat man nun die spezialisierten Registerzonen und die Normalspeicherzonen zu unterscheiden.

Die Registerzone (s.2.1.3.)

Die Registerzone enthält:

- an den Stellen 85 - 95 Spezialregister
- an den Stellen 96 - 595 numerische Register

Die Registerzone kann verkleinert werden, wenn das Programm nicht alle 100 numerischen Register benötigt.

Adressierung der Spezialregister:

Man greift auf diese Register in der gleichen Weise zu wie auf eine Normalspeicherzone.

Adressierung der numerischen Register:

Die numerischen Register sind Zonen von fixer Länge (standardmäßig 5 Bytes), die ganz bestimmte Plätze im Zentralspeicher belegen.

Um auf eins von ihnen zuzugreifen, gibt der Programmierer lediglich die Nummer des Registers an (00 - 99). Die Zentraleinheit rechnet nun

automatisch die Adresse A der rechten Stelle nach folgender Formel aus:

$$A = 5 R + 100$$

Beispiel: Das Register 00 belegt die Stellen 96 - 100.  
Seine reelle Adresse ist: 0100.

Werden Befehle verwendet, die Doppelregister ansprechen, wird die Nummer des rechten Registers angegeben.

Beispiel: Das Doppelregister 17 besteht aus den beiden Einfachregistern 16 und 17 und belegt die Stellen 176 bis 185. Seine reelle Adresse ist: 0185.

Solange man die Register in der eben besprochenen Weise anwendet, ist es nicht möglich auf eine einzelne Stelle eines numerischen Registers zuzugreifen. Das ist erst möglich, wenn man die Registerzone als Normalspeicherbereich ansieht.

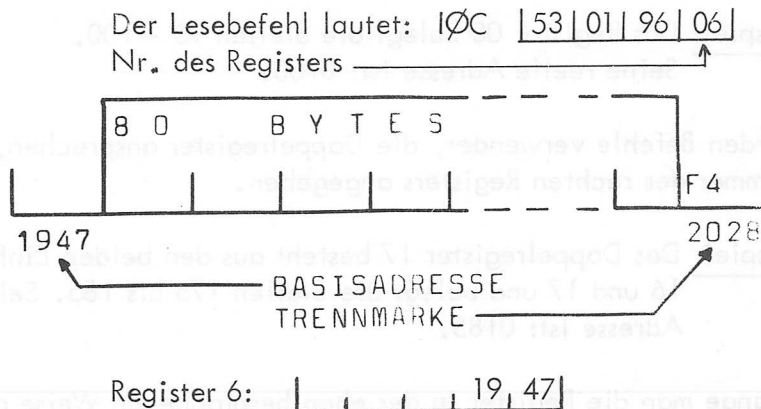
#### Die Normalspeicherzonen (s.2.1.3.4.)

Die Einteilung, Belegung und Anwendung der Normalspeicherzone liegt im Belieben des Programmierers.

Die Zentraleinheit lokalisiert eine Normalspeicherzone, indem sie ein numerisches Register untersucht, in dem die Anfangsadresse dieser Zone gespeichert ist. Diese Adresse, auch Basisadresse genannt, ist genau genommen die Adresse des Bytes, das links vor der Zone liegt.

Der Programmierer belegt für jede angelegte Zone ein numerisches Register und benutzt dessen Nummer in den Befehlen, die diese Zone ansprechen. Die Basisadressen werden also in einem Programm nicht erwähnt. Während des Programmladens werden sie in den Kernspeicher wie Konstanten eingegeben. Bei Ausführung des Programms werden die Basisadressen dann aufgrund der in den Befehlen erwähnten Registernummern eingesetzt.

Beispiel: Die Zone für den Kartenleser ist 80 Stellen groß und belegt z. B. die Stellen 1948 bis 2027; die Basisadresse (1947) wird in dem Register 6 gespeichert.



Wenn eine Zone von einem Befehl angesprochen wird, der komplette Zonen bearbeitet (Ein/Ausgabebefehl, Umschlüsselungsbefehle; s. 3.6. und 3.7.) muß sie durch eine Trennmarke F4 rechts von dem letzten benutzten Byte (s. obiges Beispiel) begrenzt werden. Diese Befehle können als Basisregister ein beliebiges Register von 00 - 99 bearbeiten.

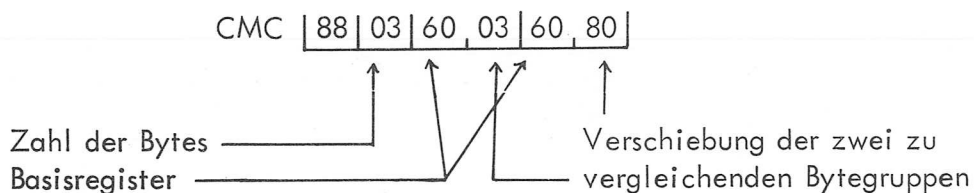
Die anderen Befehle (Bearbeitungen und Überträge innerhalb des Speichers), die auf Zeichenbasis arbeiten, können dagegen nur die Register 0 - 9 verwenden (s.3.4.). Infolgedessen kann auch eine Ein/Ausgabezone, deren Inhalt später weiterverarbeitet werden soll, nur durch die Register 0 - 9 definiert werden.

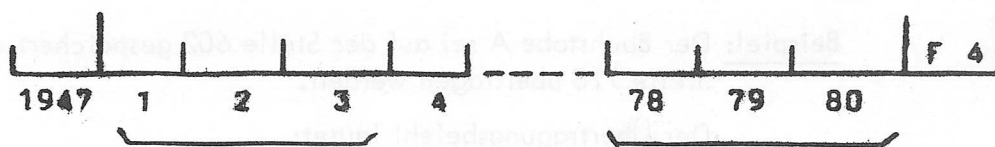
Um auf ein Byte einer Normalspeicherzone zuzugreifen, gibt der Programmierer das Basisregister dieser Zone und die "Verschiebung" an, d.h. die Stelle in Bezug auf den Anfang der Zone. Zu diesem Zweck werden die Bytes einer Zone von 1 bis n von links an nummeriert. Um auf mehrere zusammenhängende Bytes der gleichen Zone zuzugreifen, wird die Verschiebung (=Stelle) des rechten Bytes und die Zahl der Bytes der betr. Gruppe angegeben.

Um die reelle Adresse zu erhalten, addiert die Zentraleinheit die Basisadresse und die Verschiebung.

Beispiel: Die ersten 3 Bytes der Lesezone sollen mit denen der letzten Bytes der gleichen Zone verglichen werden.

Der Vergleichsbefehl lautet:





Die spezialisierten Zonen am Anfang des Speichers.

Der Programmierer kann die ersten 999 Bytes des Zentralspeichers vergleichen mit einer Normalspeicherzone, deren Basisadresse "Null" ist. Die Verschiebungen entsprechen in diesem Fall den reellen Adressen. Das dafür benutzte Basisregister ist das Register 00 (s. Beispiel in 2.1.4.3.).

Auf diese Weise hat man mit Befehlen, die auf Zeichenbasis arbeiten, Zugriff auf die spezialisierten Register sowie auf die Zahlen, die sich in den numerischen Registern befinden, in reeller Länge (natürlich nur in vollen Bytes).

#### 2.1.4.3. Die Adressierung der Informationen

Der vorherige Abschnitt hat die verschiedenen adressierbaren Einheiten des Zentralspeichers dargestellt, d.h. quasi die "Behälter" der Informationen: Byte, Register, Zone.

Es ist jetzt zu zeigen, in welcher Weise man auf Informationen zugreift, d.h. auf den "Inhalt" dieser Speichereinheiten.

Vorab ist es jedoch nützlich, sich nochmals die Struktur der Information im Speicher und die der verschiedenen Einheiten mit denen man arbeitet, ins Gedächtnis zurückzurufen. Obwohl verschlüsselt, behält die vom Rechner verarbeitete Information doch ihre ursprüngliche Struktur. Sie setzt sich, genau wie ein Schriftstück, aus "Wörtern" zusammen, die wiederum aus einem oder mehreren "Zeichen" bestehen, die dem Zeichenvorrat des Druckers entnommen sind.

#### Adressierung der Zeichen

Im Abschnitt 2.1.2 haben wir gesehen, daß ein Byte entweder 1 Zeichen im ISO-Code oder 2 Zeichen (meistens Ziffern) gepackt im internen Code, enthalten kann.

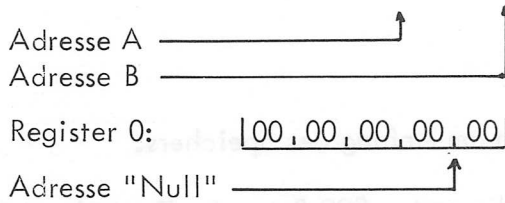
Belegt ein Zeichen nur ein Byte, ist seine Adresse gleich dem Byte, das dieses Zeichen enthält.



Beispiel: Der Buchstabe A sei auf der Stelle 602 gespeichert und soll nach Stelle 718 übertragen werden.

Der Übertragungsbefehl lautet:

MVC |80,01|06,02|07,18|



Die Halbbytes einer Stelle sind nicht adressierbar. Belegen zwei Zeichen dasselbe Byte, kann man nicht direkt auf eines von ihnen zugreifen. Man muß vorher mit dem entsprechenden Befehl diese zwei Zeichen auf zwei Bytes entpackt übertragen.

Diese Arbeitsweise und die Einschränkungen des Abschnitt 2.1.4.2 gelten für alle Bytes, die Zeichen enthalten.

#### Adressierung der Worte

Definitionsgemäß ist ein " Wort" eine Reihe von Zeichen. Die Zahl ist variabel, aber bekannt. Auf ein Wort von gegebener Länge kann man zugreifen, wenn es sich in einer Normalspeicherzone oder einer vergleichbaren Zone (Registerzone als Normalspeicher) befindet. Der Zugriff erfolgt in der gleichen Weise, wie es im vorigen Abschnitt für eine Gruppe von Bytes beschrieben wurde; eine Gruppe von Bytes enthält das in Betracht kommende Wort.

Man kann dagegen nicht in reeller Länge auf ein Wort zugreifen, das sich in einem numerischen Register befindet, indem man einen Befehl verwendet, der eine Registeroperation beinhaltet (s.3.3). Diese Befehle arbeiten in fixer Länge, d.h. mit dem gesamten Inhalt des Registers.

#### 2.1.5. Der Informationsfluß

##### 2.1.5.1. Die verschiedenen Arten des Informationsflusses

Der Informationsfluß im GE-55 kann verschiedener Art sein, je nach Natur der abgebenden oder empfangenden Organe. Dieser Fluß erfolgt durch eine große Anzahl von Übertragungsbefehlen, wovon gewisse, stark symbolisiert, die Programmierung vereinfachen und die Leistung dieser Organe optimieren.

Um einen Überblick über die verschiedenen Arten des Informationsflusses zu geben, soll ein bestimmtes Problem untersucht werden, z.B. ein Problem, das den Kartenleser, den Kartenstanzer und den Drucker benutzt.

Um dieses Problem zu lösen und unter Berücksichtigung dessen, was in den vorhergehenden Abschnitten gesagt wurde, belegt man 5 Zonen im Zentral-  
speicher:

- eine Registerzone für das Rechnen und die Adressierung der Zonen und
- vier Zonen im Normalspeicherbereich, nämlich eine Zone für den Kartenleser (Eingabezone) zwei Zonen für den Kartenstanzer und den Drucker (Ausgabezone)  
eine Zone zur Zwischenspeicherung und zur Bearbeitung.

Der Ablauf kann folgendermaßen schematisiert werden:

1. Den Inhalt einer Karte, die vom Kartenleser gelesen wurde, in die Eingabezone des Kernspeichers übertragen. Je nach Art werden die eingelesenen Angaben einer oder mehreren Verarbeitungen unterworfen:
  - . logische Analyse in der Eingabezone selbst (Klartext, Kennzeichen, Angaben zur Berechnung etc.)
  - . Bereitstellung in einer oder beiden Ausgabezonen (Klartext, Kennzeichen, Rechenresultate etc.)
2. Zwischenspeicherung in einer Bearbeitungszone für nachträgliche oder mehrmalige Benutzung (Klartexte, die auf mehrere Blätter übertragen werden sollen etc.)
3. Berechnungen in den numerischen Registern (numerische Daten).
4. Das Resultat der Berechnungen, das sich in den Registern befindet, wird dann in die eine oder andere oder beide Ausgabezonen übertragen. In gewissen Fällen kann auch hier ein Übertrag in eine Bearbeitungszone zur weiteren Verwendung vorgenommen werden.
5. Wenn eine Anzahl Informationen, die einer Karte oder einer Druckzeile entsprechen, sich in den Ausgabezonen befinden, werden sie an den Kartenstanzer oder den Drucker übertragen.

Die Informationen müssen also folgendermaßen übertragen werden:

- von einer Randeinheit (Leser) zu einer Eingabezone;
- von einer Eingabezone nach ein oder zwei Ausgabezonen;
- von einer Eingabezone nach einer Bearbeitungszone;
- von einer Eingabezone nach einem Register;
- von einem numerischen Register nach ein oder zwei Ausgabezonen;

- von einer Bearbeitungszone nach einer Ausgabezone;
- von einer Ausgabezone nach einer Randeinheit (Stanzer, Drucker).

Ergänzend zu diesem Informationsfluß erfolgen noch zusätzliche interne Überträge, entweder von den Bearbeitungszone oder den Registern her.

Alle Informationsbewegungen erfolgen Zeichen für Zeichen. Sie unterscheiden sich in ihrem Ablauf und durch ihre jeweiligen zusätzlichen Funktionen je nach der empfangenden Zone dieser Information. Bei der Durchführung der Überträge müssen in erster Linie die Überträge von der Zentraleinheit zu den Randeinheiten von dem internen Austausch im Kernspeicher unterschieden werden.

#### 2.1.5.2. Die Ein- und Ausgabe in die bzw. aus der Zentraleinheit

Die Informationen von und nach einer Randeinheit (Kartenleser, Stanzer, Drucker) werden in der gleichen Reihenfolge gespeichert, wie sie sich auch auf der Randeinheit befinden. Übertragen wird von links nach rechts in aufsteigender Reihenfolge der Adressen der Zone, die der Randeinheit zugeteilt ist. Das Ende des Übertrags wird durch eine Trennmarke angezeigt.

Der Programmierer definiert die anzusprechende Randeinheit durch einen Spezialcode und das Register, in dem sich die Basisadresse der Zone befindet. Die Länge der Überträge braucht nicht angegeben zu werden. Diese Überträge werden durch den Ein-/Ausgabebefehl kontrolliert (s.3.6.).

#### 2.1.5.3. Die Überträge im Innern des Zentralspeichers

Die Informationsworte können, wie bereits gesagt wurde, einzeln übertragen werden. Der Übertrag eines Wortes oder einer Wortgruppe geschieht in absteigender Reihenfolge der Adressen, d.h., von rechts nach links. Die Anzahl der zu übertragenden Zeichen wird im Befehl angegeben.

Für die Adressierung wendet der Programmierer die in Abschnitt 2.1.4.3 gegebenen Regeln an. Es soll aber nochmals betont werden, wie einfach die Ein- und Ausgabezonen adressiert werden können: denn die Informationen in diesen Zonen befinden sich in der gleichen Reihenfolge im Speicher, wie sie sich auch an den Randeinheiten befunden haben. Um ein Wort in der dem Kartenleser zugeteilten Zone zu adressieren, braucht der Programmierer nur das Kartenbild der eingelesenen Karte nachzusehen und anzugeben:

- als Verschiebung die Nummer der rechten Spalte des Wortes in der Karte;
- als Länge die Anzahl der von diesem Wort belegten Spalten.

Dasselbe gilt auch für die Ausgabezonen, indem der Kartenvordruck für den Kartenstanzer oder das zu beschriftende Formular für den Drucker zu Rate gezogen wird.

### Zusatzfunktionen

Bestimmte Übertragungsbefehle sind mit einem Formatwechsel der Informationen verbunden. Das gilt aber ausschließlich für numerische Informationen.

Übertrag einer Zone aus dem Normalspeicherbereich in ein numerisches Register.

In einer Zone des Normalspeicherbereichs, z.B. Eingabezone, steht ein Rechenfaktor in seinem absoluten Wert. Er ist in reeller Länge und in ungepackter Form eingespeichert. Wird dieser Operand in ein numerisches Register übertragen, erfolgt die Einspeicherung dort in fixer Länge und in gepackter, algebraischer Form (s. 2.1.3.3.). Diese Umwandlung erfolgt durch die Maschine unter Beobachtung des Vorzeichenregisters. Der Programmierer muß also vor dem Übertrag, je nach dem Vorzeichen des Wertes, dieses Register entsprechend belegen (s. 2.1.3.2.).

Übertrag eines numerischen Registers in eine Normalspeicherzone.

Im Gegensatz zu dem vorher Gesagten muß ein Wert, der in algebraischer, ungepackter Form in einem numerischen Register steht, beim Übertrag in eine Normalspeicherzone in absoluter Größe umgeformt werden. Die Maschine gibt das Vorzeichen im Vorzeichenregister an. Nur der benutzbare Wert wird übertragen und in dieser Zone in ungepackter Form gespeichert. Nullen werden links durch Leerzeichen ersetzt.

### Einfügen von Zeichen

Es besteht die Möglichkeit, in einen nach der Ausgabezone (Druckzone) übertragenen Wert ein Zeichen einzufügen (Komma, Punkt etc.). Dieser Wert kann sowohl aus dem Normalspeicherbereich wie auch aus einem numerischen Register kommen.

### Die Befehle

Es gibt drei Arten von Übertragungsbefehlen für Überträge innerhalb des Zentralspeichers: Mehrfachübertragungsbefehle, Übertragungsbefehle im Normalspeicherbereich und Registerüberträge.

### Mehrfachübertragungsbefehle (s.3.5.)

Diese Befehle übertragen Informationen von Zone nach Zone, d.h.

- von einer Zone im Normalspeicherbereich (z.B. Eingabezone) nach einer anderen Zone im Normalspeicherbereich (z.B. Ausgabezone);  
oder
- von einer Zone im Normalspeicherbereich nach der Registerzone und umgekehrt.

Die Mehrfachübertragungsbefehle können, wie ihr Name sagt, in demselben Befehl einen Wert (im Normalspeicherbereich) oder ein Resultat (in der Registerzone) in eine oder mehrere empfangende Zonen übertragen. Diese Befehle haben folgende Zusatzfunktionen:

Zur Adressierung der Zonen im Normalspeicherbereich benutzen sie für die abgebenden Zonen die numerischen Register 6 und 7 und für die empfangenden Zonen die Register 8 und 9.

Diese Zuteilung erlaubt dem Programmierer nur die Verschiebungen der Worte in den Zonen anzugeben, die Basisregister sind bereits automatisch durch den Befehl ausgewählt.

### Die Übertragungsbefehle im Normalspeicherbereich (s. 3.4.3.)

Sie erlauben eine Information innerhalb einer oder zwischen zwei Zonen zu übertragen, wenn der Mehrfachübertragungsbefehl nicht benutzt werden kann. Während dieses Übertrags wird die Information nicht umgeformt. Deshalb ist eine besondere Anwendung dieser Befehle der Übertrag einer Zahl in algebraischer, gepackter Form in eine Zone des Normalspeicherbereichs. Die Registerzone wird als Normalspeicherzone betrachtet (s.2.1.4.).

Um dieselbe Zahl vom Normalspeicherbereich nach einem Register zu übertragen, wird ein Spezialbefehl gebraucht. Mit 2 Spezialbefehlen kann man packen oder entpacken, ohne den Wert der Zahl zu verändern.

### Die Registerüberträge (s.3.3.)

Mit ihnen erfolgt der Übertrag einer Zahl von einem Register in ein anderes, von einem Einfachregister in ein Doppelregister und der Austausch des Inhalts zweier Register.

## 2.1.6. Der Programmablauf

Der Programmablauf wird durch das Programmadressregister (PAR) bestimmt. Nach dem Ende eines jeden Befehls untersucht die Zentraleinheit dieses Register, um die Adresse des nächsten Befehls zu übernehmen. Am Anfang der Arbeit befindet sich die Anfangsadresse des Programms in diesem Speicher. Diese Adresse wird am Ende des Programmladens in das Register übernommen.

Die Zentraleinheit läßt die Befehle in der Reihenfolge ihrer Ein-  
speicherung im Zentralspeicher ablaufen, d.h. in aufsteigender Reihen-  
folge ihrer Adressen. Jedesmal, wenn ein Befehl übernommen wurde,  
wird seine Länge dem Inhalt des PAR zuaddiert und ergibt so die Adresse  
des nächsten Befehls.

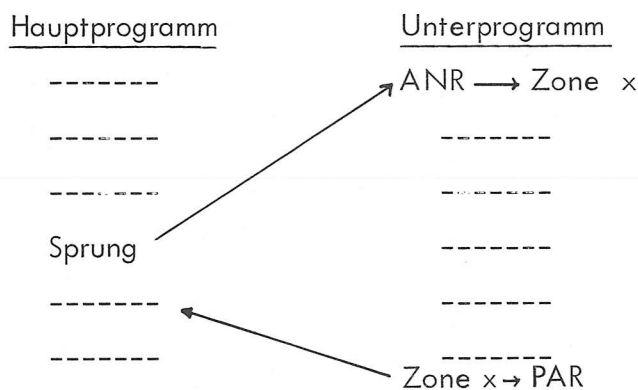
Dieser Ablauf kann in verschiedenen Fällen unterbrochen werden (Sprung  
an den Anfang einer Programmsequenz, von einer Serie zu einer anderen  
etc. oder auch während der Normalarbeit - siehe weiter unten: Pro-  
grammunterbrechung). Im ersten Fall wird die Adresse des Befehls  
auf den man springen will, in das PAR übernommen. Durch einen Sprung-  
befehl wird diese Sprungadresse automatisch übernommen. Sie kann auch  
durch einen Übertragungsbefehl gegeben werden, wenn sie vorher als  
normale Angabe gespeichert wurde. Ist diese Adresse übernommen worden,  
so springt das Programm auf diese angegebene Adresse und der normale  
Programmablauf setzt wieder ein.

#### Einfügen einer Serie in eine andere Serie

Man kann durch einen Sprungbefehl beispielsweise von einer Serie S1 in  
eine andere Serie S2 springen. Dieser Sprungbefehl bewirkt den Übertrag  
der Adresse des nächstfolgenden Befehls (Rücksprungadresse) von dem  
PAR in das Rücksprungadressregister (ANR). Wenn man am Ende der  
Serie S2 wieder in die Serie S1 zurück will, und zwar an den Punkt, von  
dem aus man gesprungen ist, genügt es, wenn man einen Übertrag des  
ANR zum PAR macht. Da das ANR bei jedem Sprung auf dem neuesten  
Stand gehalten wird, kann man durch diese Methode eine gleiche Serie  
(oder Unterprogramm) aus einem Hauptprogramm von verschiedenen  
LEVEL her ansprechen.

#### Anmerkung:

Wenn das Unterprogramm selbst auch Sprungbefehle benutzt, muß der  
Programmierer am Anfang des Unterprogramms den Inhalt des ANR in eine  
spezielle Zone übertragen, um damit die Rücksprungadresse des Haupt-  
programms zu erhalten, denn alle Sprungbefehle ändern den Inhalt dieses  
Registers. Folgende schematisch dargestellte Organisation könnte syste-  
matisch angewendet werden:





### Aufbau der Sprungbefehle (s. 3.2.2.)

Die Sprungbefehle ermöglichen es, den Sprung von einer Serie zur anderen von der An- oder Abwesenheit einer Bedingung abhängig zu machen.

Diese Abhängigkeit könnte unter der Form eines Zeichens der Inhalt eines Spezialregisters sein (Vergleichsauswertung, Resultate aus Rechenoperationen etc.) oder auch eine andere beliebige Stelle des Normalspeicherbereichs (Schlüssel, Kartenart, Resultat einer logischen Operation etc.).

### Programmunterbrechung

Das Hauptprogramm kann unterbrochen werden, um ein Testprogramm ablaufen zu lassen. Diese Unterbrechung geschieht wie ein Programmsprung:

- der Inhalt des PAR (= Rücksprungadresse des unterbrochenen Programms) wird in das ANR übertragen.
- Übertrag des AFP in das PAR. Dieses Register enthält die Anfangsadresse des Testprogramms (Unterbrechungsprogramm). Die Unterbrechung kann durch die Bedienungskraft mit Hilfe einer Spezialtaste angerufen werden. Sie ermöglicht es, ein Programm anzurufen, das eine Operation noch einmal vorne beginnen läßt. Wenn z.B. das Stanzen einer Karte unterbrochen wurde, kann dieses Programm das erneute Stanzen einer Karte versuchen, solange sich die Daten noch in der Stanzzone befinden.

Die Programmunterbrechung kann auch automatisch erfolgen als Folge von bestimmten schwerwiegenden Fehlern, die das Fortsetzen der Arbeit verbieten (s.4.3.2.2.).



### 2.1.7. Bedienung der Maschine

Die Tasten und Lampen, die den Betriebszustand aller Einheiten des GE-55 kontrollieren, sind in einer Reihe nebeneinander angeordnet und befinden sich am oberen Teil der Zentraleinheit, direkt im Blickfeld der Maschinenbedienung. Auf den Randeinheiten befinden sich nur die Tasten, die zu einem speziellen Eingriff an diesen Randeinheiten dienen.

An der Zentraleinheit befindet sich:

- in der Mitte eine Sichtanzeige, an der die Eingabe über die numerische Tastatur durch Klarschrift kontrolliert werden kann oder an der Informationen aus dem Zentralspeicher in Klarschrift sichtbar gemacht werden können. Die Arbeitsweise dieser Anzeigevorrichtung wird in Abschnitt 2.2 erklärt.
- rechts von der Sichtanzeige die Tasten zur Bedienung der Maschine. Sie sind mit Lampen versehen.
- links die Lampen der Störungsanzeige.

#### 2.1.7.1. Die Drucktasten

**ON**

(power on) schaltet die Stromzufuhr zur Maschine ein. Die Maschine ist nach Betätigen dieser Taste nicht sofort betriebsbereit, sondern es muß abgewartet werden, bis sie eine bestimmte Temperatur erreicht hat. Diese wird durch Aufleuchten der Lampen LOAD, PRS, RUN und PRR angezeigt. Erst wenn diese vier Lampen brennen, kann mit dem Laden des Programms begonnen werden. Diese Anlaufzeit kann max. 5 Minuten dauern.

**OFF**

(power off) schaltet die Stromzufuhr zur Maschine ab. Diese Taste dient zum Abschalten der gesamten Stromzufuhr zur Maschine. Sie soll während der Arbeit so wenig wie möglich benutzt werden, da mit ihr auch die interne Klimatisierung abgeschaltet wird. Die bestmögliche Ausnutzung der elektronischen Stromkreise ist aber nur bei möglichst geringen Temperaturschwankungen gegeben. Bei schwerwiegenden Störungen an den Randeinheiten soll sie aber benutzt werden, um diese restlos abzuschalten. Das gilt besonders bei manuellen Eingriffen (z.B. Entfernen eines Kartenstaus am Stanzer).

**LOAD**

(program loading)  
Mit dieser Taste wird ein Programm in Maschinensprache in den Kernspeicher geladen. Die Programmkarten werden in den Kartenleser eingelegt und mit dieser Taste vom Kartenleser gelesen und in den Kernspeicher übertragen. Nach erfolgter Eingabe wird das Programm gestartet.

Die Adresse des Befehls, der als erster ablaufen soll, wird durch die Programmstartkarte gegeben.

Diese Taste hat die gleiche Wirkung wie der Befehl LOAD, der in Abschnitt 3.8.2. beschrieben wird. Im gleichen Abschnitt wird auch der Aufbau der Programmkarten erklärt.

Die Taste LOAD annulliert die Paarigkeitskontrolle. Man kann ein durch PAR angehaltenes Programm damit erneut laden.

SSS

(single step stop) Doppelfunktionstaste.

Wenn der Schalter eingelegt ist (Lampe brennt), erfolgt ein schrittweiser Programmablauf. Das Programm wird vor jedem Befehl angehalten. Jeder folgende Befehl läuft erst nach Druck auf die Taste START ab. Das Programm läuft erst wieder automatisch weiter, wenn die Taste SSS ausgerastet wird (Lampe erlischt) und ein letztmaliger Druck auf die Taste START erfolgt ist.

Diese Taste dient in erster Linie den Maschinenteknikern zur Kontrolle der Stromkreise der Maschine. Im normalen Arbeitsablauf wird sie nur gebraucht, um das Programm direkt anzuhalten, wenn eine besondere Manipulation gemacht werden soll, z.B. Kernspeicherausdruck.

KHLT

(key halt) Doppelfunktionstaste

Ist dieser Schalter eingelegt (Lampe brennt), hält das Programm an bestimmten Stellen an, die durch den Befehl KHLT bezeichnet sind (s. 3.8.4.).

Jedesmal, wenn das Programm auf diesen Befehl trifft, hält es (bei eingelegter Drucktaste) an. Durch die Taste START erfolgt automatischer Weiterlauf bis zum nächsten Befehl KHLT. Das automatische Weiterlaufen erfolgt erst wieder, wenn die Taste KHLT ausgerastet (Lampe aus) wurde und ein letztmaliger Druck auf START erfolgte.

INH

(inhibit) Doppelfunktionstaste

Bei eingelegter Taste (Lampe an) wird die gesamte Arbeit der Zentraleinheit (und des Stundenzählers) vor Übernahme des nächsten Befehls gestoppt. Keine Funktion der Zentraleinheit kann angesprochen werden, solange diese Taste eingelegt ist. Um ein Programm wieder anlaufen zu lassen, muß die Taste ausgerastet (Lampe aus) und Taste START gedrückt werden.

Diese Taste soll der Taste OFF vorgezogen werden, um eine Arbeitsunterbrechung zu erreichen. Die Klimaanlage der Maschine wird durch sie nicht abgeschaltet.

**STOP**

(stop timing)

Mit dieser Taste kann die Arbeit der Zentraleinheit unmittelbar, d.h. sogar mitten in einem Befehlsablauf, unterbrochen werden. Das Programm kann wieder gestartet werden

- entweder durch Druck auf die Taste START, wenn man den unterbrochenen Befehl zu Ende führen will;
- oder durch Druck der Taste RUN am Anfang des nächsten Befehls, wenn man den unterbrochenen Befehl nicht fortsetzen will und es sich nicht um einen Ein/Ausgabebefehl handelt.

Die Taste STOP kann eine Einheit, die nicht auf der Stelle halten kann (Kartenleser, Trommel) im ersten Fall nicht sofort unterbrechen. Diese Randeinheit führt die angefangene Operation zu Ende (Lesen der Karte oder der Trommelbahn beispielsweise), obwohl die Verbindung zur Zentraleinheit unterbrochen ist. Eine Anzahl von Zeichen geht dadurch verloren. Erfolgt aber kein simultanes Arbeiten dieser Einheit mit solchen, die auf der Stelle halten können (Drucker und Stanzer), können diese letzteren auf der Stelle anhalten. Sie beginnen ihre Arbeit wieder mit der Zentraleinheit.

**START**

(start timing) startet die Arbeit der Zentraleinheit und des Programms an der Stelle, wo eine Unterbrechung erfolgte, d.h. entweder vor oder während der Durchführung eines Befehls.

**RUN**

(start run) startet den Programmablauf am Anfang des Befehls, dessen Adresse sich im PAR befindet. Unter Umständen wird die Wirkung der Paarigkeitskontrolle vorher gelöscht.

Die Taste RUN kann dazu benutzt werden, um ein Programm mit dem nächsten Befehl zu starten, wenn es während des Ablaufs eines Befehls mit der Taste STOP oder infolge einer Störung (vorbehaltlich der Folgen eines Verzichts auf diesen Befehl: Verlust von Informationen etc.) angehalten wurde. Diese Taste dient nicht dazu, an das Ende eines Ein/Ausgabebefehls zu springen. In diesem Fall wird zwar der folgende Befehl richtig durchgeführt, die Zentraleinheit nimmt jedoch danach die unterbrochene Ein/Ausgabeoperation wieder auf und beendet sie.

Mit der Taste RUN kann nur nach Betätigen der Tasten SSS, KHLT oder INH ein Programm gestartet werden.

PRS

(print store)

Die Taste PRS ruft ein fest eingespeichertes Unterprogramm an. Das Unterprogramm ermöglicht einen teilweisen Kernspeicherausdruck. Die Anfangsadresse dieses Ausdruckes wird über die numerische Tastatur eingegeben. Derselbe Effekt wird erreicht, wenn im Programm der Befehl PRSTO aufgefunden wird. (Siehe Abschnitt 3.8.1.)

In beiden Fällen werden automatisch die ersten 200 Stellen des Kernspeichers ausgedruckt. Zum Ausdruck des Kernspeichers werden nacheinander die Tasten SSS - KHT - PRS - RUN betätigt.

Erfolgt der Kernspeicherausdruck, um eine Programmblockierung oder eine Störung im Kernspeicher (Paarigkeitskontrolle) festzustellen, sind folgende Regeln anzuwenden:

Im ersten Fall muß das Programm durch die Taste SSS blockiert werden, ehe der Ausdruck beginnt.

Im zweiten Fall genügt es, nur einen Kernspeicherausdruck zu befehlen. Handelt es sich um einen Schlüsselfehler, hält der Drucker an dieser Stelle an, da er es nicht drucken kann.

Wird ein Kernspeicherausdruck während des Programmablaufs gewünscht (Probeausdruck beim Testen), so muß, damit die Arbeit nicht beeinträchtigt wird:

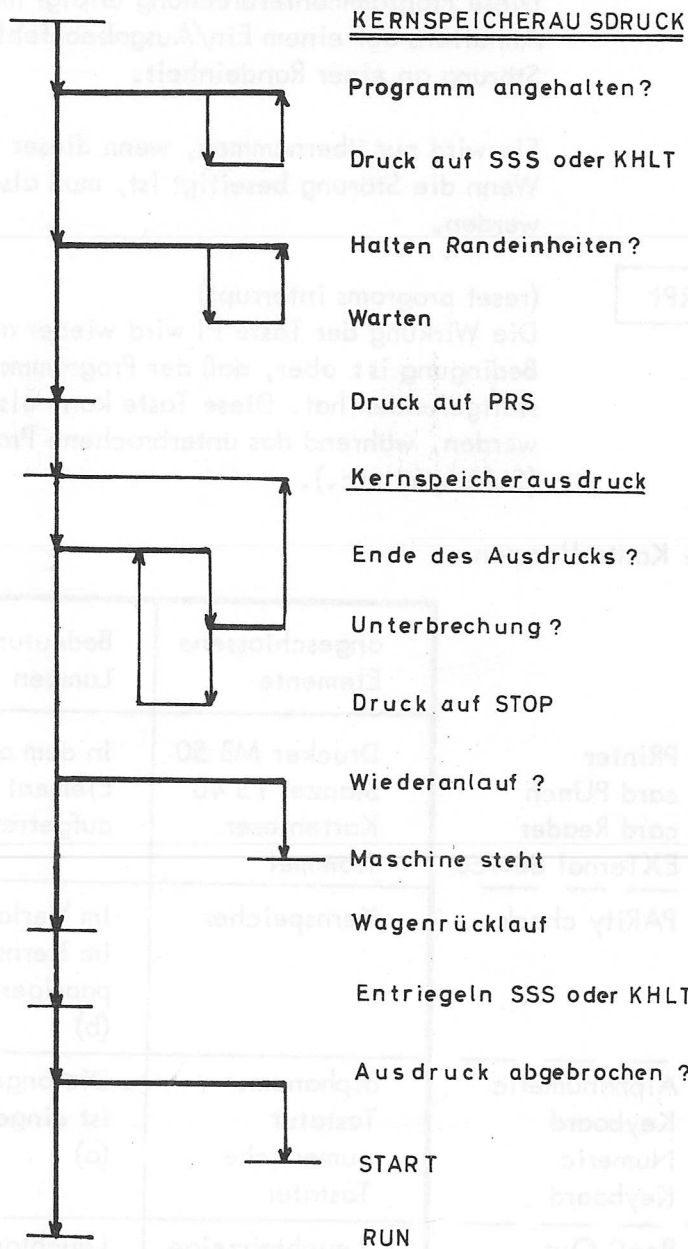
- das Programm am Anfang eines Befehls durch SSS oder an bestimmten Stellen durch den Befehl KHLT und die entsprechende Drucktaste angehalten werden.
- geprüft werden, ob die Operationen der Randeinheiten fertig sind (Einheiten angehalten und zugehörige Lampen aus; s. 2.1.7.2.)

Wird durch das normale Programm das Drucken einer Liste vorgesehen, so muß am Ende des Kernspeicherausdrucks der Schreibkopf auf Stelle 1 und das Formular gewechselt werden, bevor das Programm wieder gestartet wird.

Auf alle Fälle kann der Ausdruck durch die Taste STOP angehalten werden, wenn genügend Informationen ausgedruckt sind.

→ haus.  
5.2.3  
(11.11.68)

Die hier beschriebenen Verfahren sind im folgenden Organigramm schematisch wiedergegeben:



PRR

Diese Taste ist ohne Funktion

**PI**

(programm interrupt)

Diese Taste hält das gerade ablaufende Programm vor dem nächsten Befehl an, blockiert dieses Programm und startet zur gleichen Zeit ein Programm, das sich an der im Register AFP angegebenen Stelle befindet (siehe Programmunterbrechung in Abschnitt 2.1.6.).

Diese Programmunterbrechung erfolgt meistens während des Anhaltens auf einem Ein/Ausgabebefehl durch eine Störung an einer Randeinheit.

Sie wird nur übernommen, wenn dieser Befehl beendet ist. Wenn die Störung beseitigt ist, muß also neu gestartet werden.

**RPI**

(reset programs interrupt)

Die Wirkung der Taste PI wird wieder aufgehoben. Bedingung ist aber, daß der Programmaustausch noch nicht stattgefunden hat. Diese Taste kann also nur benutzt werden, während das unterbrochene Programm anhält (Störungen etc.).

2.1.7.2. Die Kontrolllampen

- PR = PRinter
- PU = card PUnch
- R = card Reader
- EXT = EXTERNAL device
- PAR = PARity check

- A = Alphanumeric Keyboard
- N = Numeric Keyboard
- RDO = ReaD Out

angeschlossene Elemente	Bedeutung der brennenden Lampen
Drucker MB 50 Stanzer PS 40 Kartenleser Trommel	In dem angeschlossenen Element ist ein Fehler aufgetreten (a)
Kernspeicher	Im Verlauf einer Operation im Kernspeicher ist ein paariges Byte aufgetreten (b)
alphanum. Tastatur numerische Tastatur	Die angeschlossene Tastatur ist eingeschaltet (a)
Leuchtanzeige	Leuchtanzeige wahrnehmen (a)

(a) Siehe Kapitel über die Randeinheiten.  
 (b) Paarigkeitskontrolle



Ein aufgetretenes paariges Byte während einer Operation im Kernspeicher hält unmittelbar die Funktionsweise der Zentraleinheit und den ablaufenden Befehl an. Es kennzeichnet einen Lesefehler der Bits. Der Platz des fehlerhaften Zeichens und infolgedessen auch die Information, in der es aufgetreten ist, ist unbekannt.

Darum muß man:

- einen Kernspeicherausdruck machen, um die Adresse des fehlerhaften Zeichens zu finden,
- den aufgetretenen Fehler einem Techniker melden,
- die Arbeit fortsetzen oder den fehlerhaften Teil der Arbeit neu durchführen.

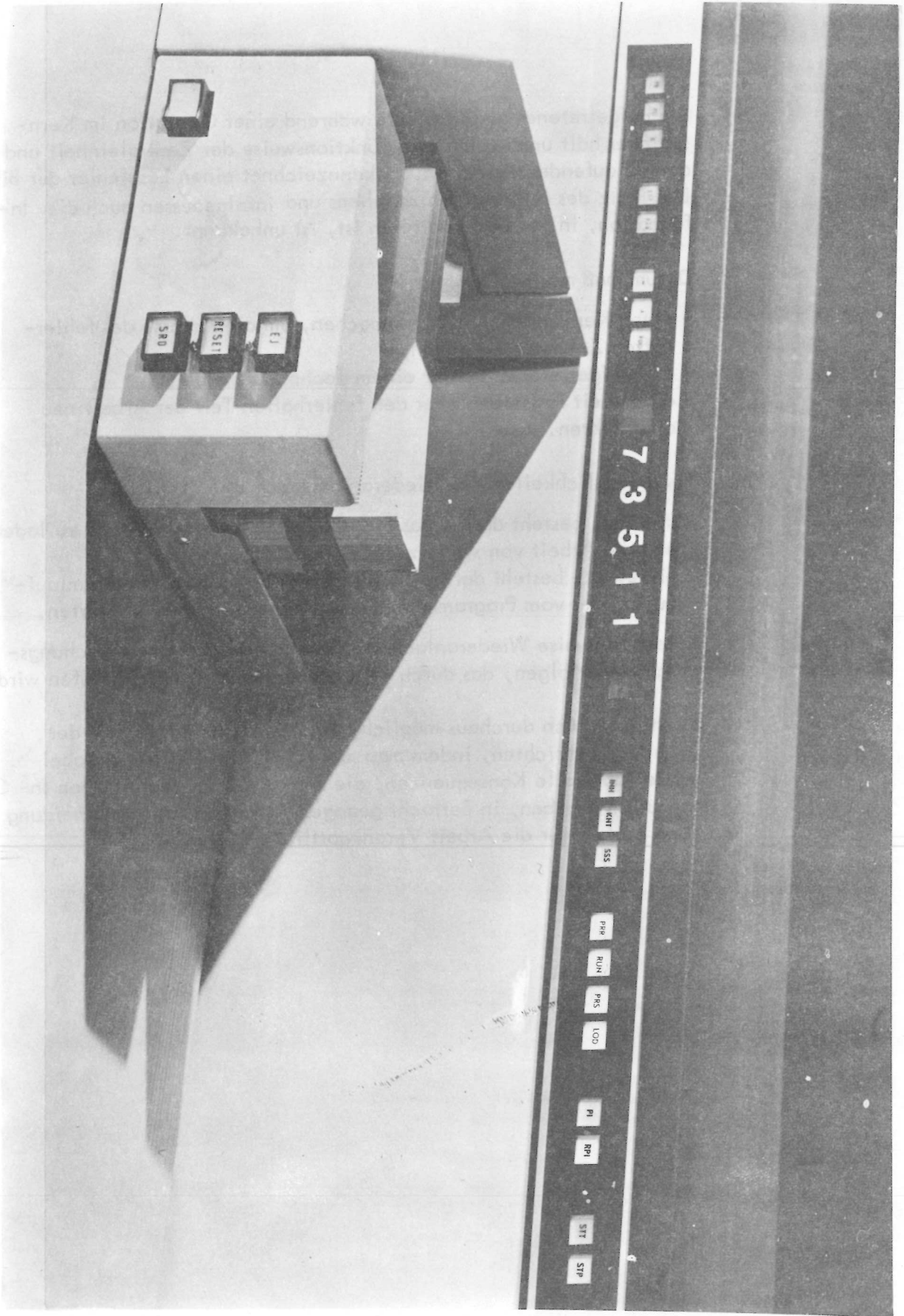
Zwei Möglichkeiten des Wiederanlaufs gibt es:

- die erste besteht darin, das Programm mit Taste LOAD neu zu laden und die Arbeit von vorne zu beginnen.
- die zweite besteht darin, das Programm von einem Wiederanlaufpunkt, der vom Programmierer festgesetzt wird, neu zu starten.

Dieser teilweise Wiederanlauf kann auch durch das Unterbrechungsprogramm erfolgen, das durch Programmunterbrechung angerufen wird.

Es ist aber auch durchaus möglich, auf den fehlerhaften Teil der Arbeit zu verzichten, indem man mit Taste RUN fortfährt. Dabei müssen aber alle Konsequenzen, die sich aus einer fehlerhaften Information ergeben, in Betracht gezogen werden. Diese Entscheidung kann nur der für die Arbeit Verantwortliche treffen.

Ansicht der Bedienungstasten und Kontrolllampen des GE-55



GE-55

Ref.-Nr.: 23:20.001 D

Juli 1968

## 2.2. Die numerische Tastatur und die Leuchtanzeige

### 2.2.1. Beschreibung

Die numerische Tastatur ermöglicht die direkte Eingabe quantitativer Daten in den Kernspeicher ohne Zwischenstation (z.B. Karten etc.).

Folgende Zeichen können eingegeben werden:

- die Ziffern 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- das Leerzeichen
- zwei spezielle Zeichen ":" und ";", die benutzt werden können, um irgendetwas anzuzeigen, z.B. Vorzeichen einer Zahl.

Die über die Tastatur eingegebenen Zeichen werden in einem 6-stelligen Pufferspeicher gespeichert, dessen Inhalt auf der Leuchtanzeige sichtbar ist, die sich am Schrank der Zentraleinheit befindet.

Die Daten, max. 6 Zeichen, werden in reeller Länge eingegeben, und zwar beginnend mit der höchsten Wertstelle; gemäß dem Anschlag erscheinen die Zeichen auf der Leuchtanzeige und verschieben sich von rechts nach links. Die Bedienung kann dadurch ihre Arbeit überprüfen; wenn sie einen Fehler feststellt, löscht sie den Inhalt des Pufferspeichers und beginnt von neuem; wenn die Eingabe in Ordnung ist, gibt sie den Übertrag in den Kernspeicher frei.

Der Pufferspeicher und die Leuchtanzeige können auch gebraucht werden, um der Bedienung eine Information anzuzeigen: ein Ergebnis, das zu notieren oder zu überprüfen ist, einen Schlüssel, aufgrund dessen etwas durchgeführt wird etc.. Die im Kernspeicher zu verarbeitende Information kann, wie oben bereits angegeben, nur aus max. 6 Zeichen bestehen.

### 2.2.2. Anwendung

#### 2.2.2.1. Bedienungstasten und Kontrolllampen

Außer den oben angegebenen Tasten und der Leuchtanzeige hat die Bedienungskraft die Möglichkeit, den Arbeitsablauf über 4 auf der Tastatur angebrachten Bedienungstasten und zwei am Schrank der Zentraleinheit befindlichen Kontrolllampen zu kontrollieren.

Auf der Tastatur:

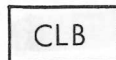
MVB

(MoVe Buffer) bewirkt den Übertrag des Inhaltes des Pufferspeichers in den Kernspeicher;

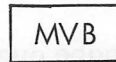
SRD

(Single card Read) dient dem schrittweisen Lesen der Karten am Kartenleser (siehe 2.4.).

rechts von der Tastatur:

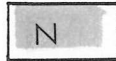


(CLear Buffer) dient dem Löschen des Pufferspeichers;

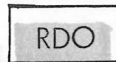


siehe vorstehend.

Die Kontrolllampen



(Numeric Keyboard) zeigt an, daß das Programm eine Eingabe über die Tastatur wünscht



(ReaD Out) zeigt an, daß eine durch das Programm bearbeitete Information in den Pufferspeicher übertragen worden ist.

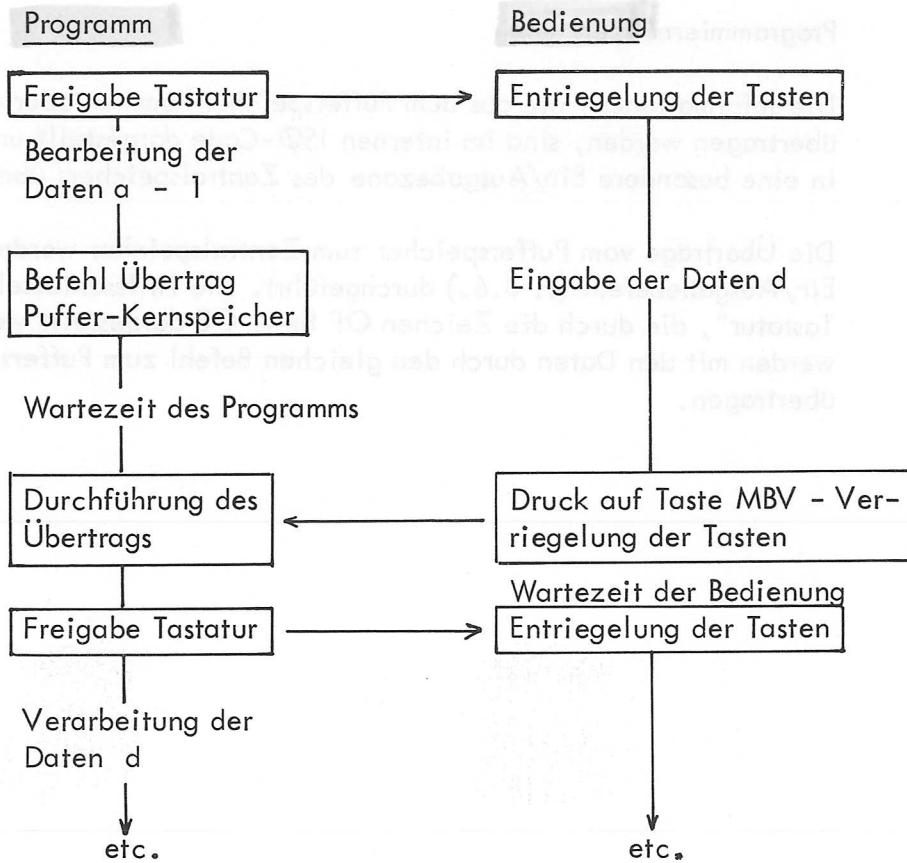
#### 2.2.2.2. Eingabe der Daten

Die Eingabe der Daten wird durch das Programm kontrolliert. Die numerische Tastatur ist normalerweise blockiert; sie bleibt solange blockiert, bis durch Programm ein Befehl erfolgt, der für einen Pufferübertrag einen Schlüssel "Freigabe-Tastatur" enthält. Dieser Schlüssel stellt die Verbindung zwischen dem Pufferspeicher und der numerischen Tastatur her, entblockiert die Tastatur und schaltet die Kontrolllampe N ein.

Die Maschinenbedienung gibt nun die Daten ein. Während des Eintastens ist der Pufferspeicher vom Kernspeicher isoliert, und das Programm fährt in seiner Bearbeitung fort; wenn jedoch ein weiterer Befehl erkannt wird, der den Pufferspeicher anschließt, besonders der, der den Datenübertrag in den Kernspeicher beinhaltet, bleibt das Programm auf diesen Befehl stehen, bis das Ende der Eingabe angezeigt wird.

Am Ende der Arbeit drückt man auf die Drucktaste MVB. Dadurch wird die Kontrolllampe N ausgelöscht, die Tastatur blockiert und die Verbindung zwischen dem Pufferspeicher und dem Kernspeicher hergestellt; der Übertrag der eingegebenen Daten kann nun erfolgen und das Programm wird an der Stelle fortgesetzt, an der es stehengeblieben ist.

Diese Organisation ermöglicht eine Simultaneität zwischen der Dateneingabe über Tastatur und deren interner Bearbeitung, wie es das nachfolgende Diagramm zeigt:



### 2.2.2.3. Sichtbarmachen einer Information

Um der Maschinenbedienung eine im Zentralspeicher erarbeitete Information mitzuteilen, genügt es, diese in den Pufferspeicher zu übertragen. Nach erfolgtem Übertrag kann die Information an der Sichtanzeige abgelesen werden.

Um ihr die Möglichkeit zu geben, vom Inhalt der Information Kenntnis zu nehmen, wird ein Hilfsschlüssel "Sichtanzeige" in den Puffer eingesetzt.

Dieser bewirkt, daß die Lampe RDO aufleuchtet und keine neue Einspeicherung in den Pufferspeicher erfolgen kann. Das bedeutet, daß das Programm anhält, wenn es einen solchen Befehl antrifft.

Die Bedienungskraft zeigt die Kenntnisnahme durch Druck auf die Taste CLB an, die den Inhalt des Pufferspeichers löscht und eine erneute Benutzung desselben ermöglicht.

#### 2.2.2.4. Programmiernormen

Die Informationen, die aus dem Pufferspeicher kommen oder an ihn übertragen werden, sind im internen ISØ-Code dargestellt und werden in eine besondere Ein/Ausgabezone des Zentralspeichers übertragen.

Die Überträge vom Pufferspeicher zum Zentralspeicher werden mit dem Ein/Ausgabebefehl (s. 3.6.) durchgeführt. Die Hilfsschlüssel "Freigabe Tastatur", die durch die Zeichen OF bzw. OE dargestellt werden, werden mit den Daten durch den gleichen Befehl zum Pufferspeicher übertragen.



2.3. Die alphanumerische Tastatur

2.3.1. Darstellung

Die Zusatzeinrichtung der alphanumerischen Tastatur ermöglicht es, veränderliche Texte in den Zentralspeicher einzugeben. Sie enthält 45 Zeichentasten und eine Leertaste.

Die Zeichentasten tragen die 63 Symbole (Buchstaben, Ziffern und Sonderzeichen) die in den internen ISO-Code umgeschlüsselt sein können (§.6.1.2.). Sie entsprechen den Schreibmaschinentastaturen.

Die eingegebenen Texte können eine beliebige Länge haben. Sie werden ausschließlich durch die zur Verfügung stehende Kapazität des Zentralspeichers begrenzt.

2.3.2. Anwendung

2.3.2.1. Eingabe eines Textes (oder einer anderen Information)

Die Eingabe eines Textes wird durch das Programm gesteuert. Die Tastatur ist normalerweise verriegelt. Sobald aber das Programm eine Eingabe verlangt, wird sie entriegelt und die Lampe A leuchtet auf.

Die so aufmerksam gemachte Bedienungskraft setzt nun den Text zusammen, indem sie mit der 1. Stelle von links anfängt. Bei jedem Anschlag werden die Zeichen unmittelbar von links nach rechts in eine bestimmte Zone des Zentralspeichers übertragen. Während der Anschläge "erstarrt" das Programm auf dem entsprechenden Befehl.

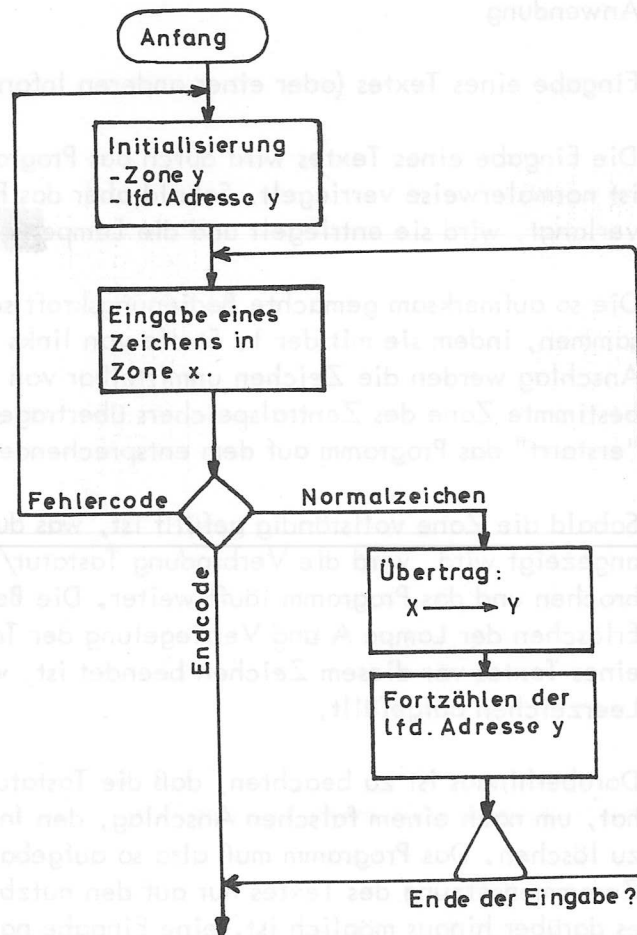
Sobald die Zone vollständig gefüllt ist, was durch eine Trennmarke angezeigt wird, wird die Verbindung Tastatur/Zentralspeicher unterbrochen und das Programm läuft weiter. Die Bedienung merkt das durch Erlöschen der Lampe A und Verriegelung der Tasten. Wenn die Eingabe eines Textes vor diesem Zeichen beendet ist, wird die Zone durch Leerzeichen aufgefüllt.

Darüberhinaus ist zu beachten, daß die Tastatur keine Spezialtasten hat, um nach einem falschen Anschlag, den Inhalt der Eingabezone zu löschen. Das Programm muß also so aufgebaut sein, daß es die Zusammensetzung des Textes nur auf den nutzbaren Teil begrenzt und es darüber hinaus möglich ist, eine Eingabe nach einem Fehler neu zu beginnen.

Hierzu genügt es, zwei Zeichen der Tastatur als Hilfsschlüssel zu verwenden mit folgender Bedeutung: Ende Eingabe und Fehler.

Die eingetasteten Zeichen werden nacheinander in eine Zwischenzone von 2 Stellen eingegeben, von denen die rechte Stelle durch eine Trennmarke F4 belegt ist (Zone x). Dort werden sie analysiert: handelt es sich um normale Zeichen, werden sie von links nach rechts in eine zweite Zone (Zone y) umgespeichert, deren Kapazität durch den größtmöglichen Text, der einzugeben ist, bestimmt wird; handelt es sich aber um einen Hilfsschlüssel, wird der Übertrag unterbrochen und es läuft die entsprechende Funktion ab (Löschen oder Ausgabe).

Diese Organisation ist im folgenden Organigramm schematisch wiedergegeben:



Diese Hilfsschlüssel können auch für andere Steuerungen benutzt werden.

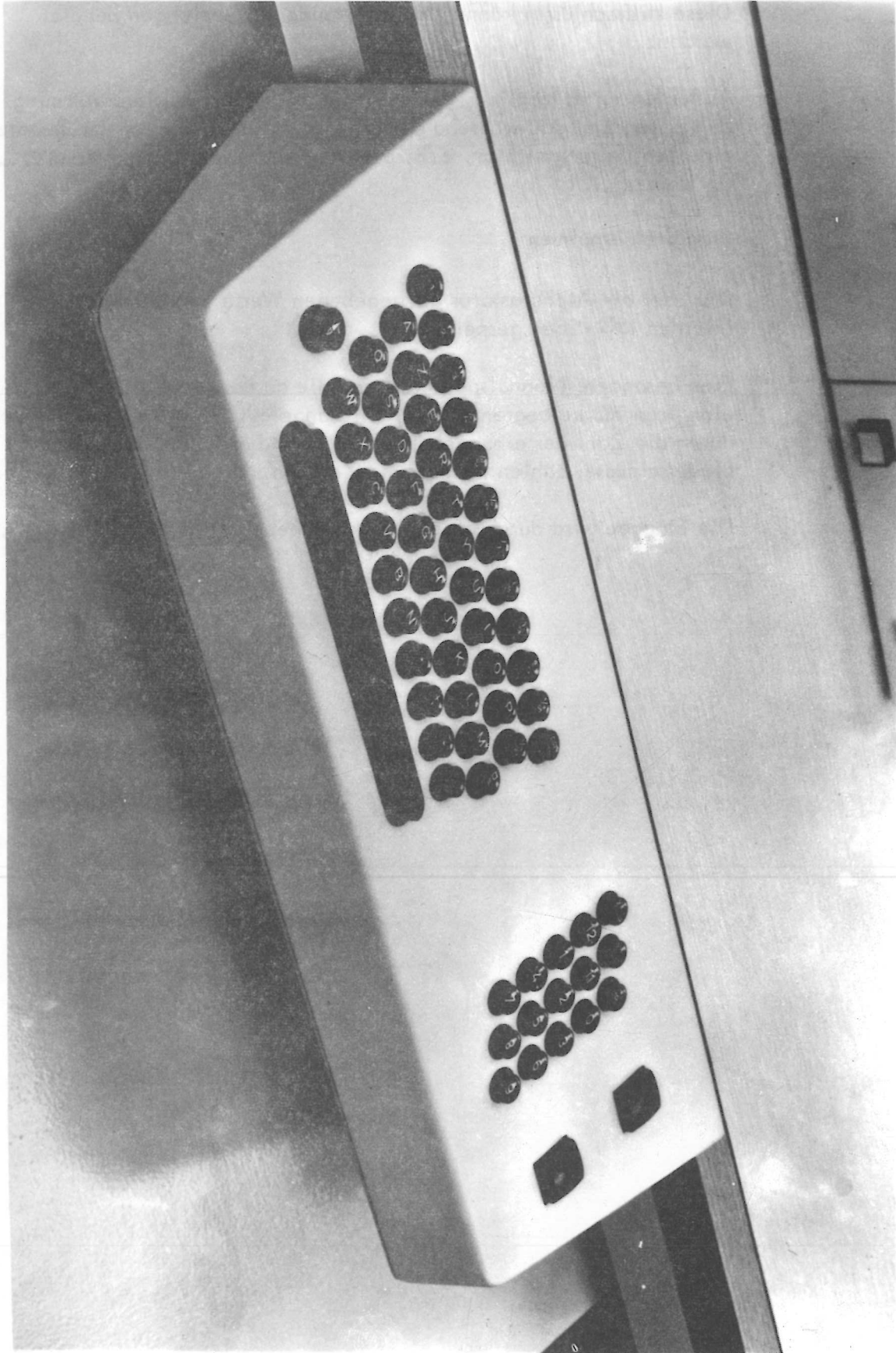
Außerdem ist es möglich, die Besonderheiten des Multiprogramming so zu verwenden, daß während der Eingabe eines Textes auf der Tastatur eine Bearbeitung erfolgt, z.B. der Druck am Ende einer Liste etc.... (s. Kap. 4).

#### 2.3.2.3. Programmnormen

Die über die Alphatastatur eingegebenen Werte werden direkt im internen ISO-Code gespeichert.

Eine besondere Normalspeicherzone, die an der rechten Seite durch eine Trennmarke begrenzt wird, wird angelegt. Ihre Länge wird bestimmt durch die Zahl der einzugebenden Zeichen bei jedem Befehl. Die Steuerschlüssel zählen als normale Zeichen.

Die Eingabe wird durch einen Ein/Ausgabebefehl angerufen. (s.3.6.).



Die alphanumerische und numerische Tastatur des GE-55

GE-55

Ref.-Nr.: 23.20.001 D

Juli 1968

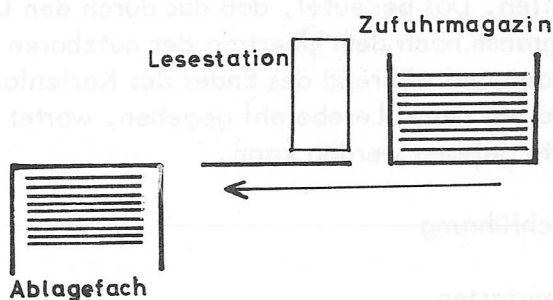
## 2.4. Der Sichtkartenleser

### 2.4.1. Darstellung

Diese Randeinheit ist dazu bestimmt, 80-spaltige Lochkarten entweder im Code T 121 oder H 14.012 zu lesen. Die Karten werden mit einer Maximalgeschwindigkeit von 150 Karten/Min. gelesen. Die in den Spalten gelochten Zeichen werden mit einer photoelektrischen Lese-einrichtung gelesen und seriell in den Zentralspeicher übertragen.

Der Sichtkartenleser enthält:

- ein Zufuhrmagazin für 500 Karten
- eine Lesestation mit 12 Photozellen
- ein Ablagefach für 500 Karten.



Die Karten werden mit Druck nach vorn (Eckenabschnitt oben links) in das Zufuhrmagazin eingelegt. Auf diese Weise ist der obere Teil des Kartenvordrucks und die Lochschriftübersetzung sichtbar. Der obere Teil der Karte ist in einer Höhe von 12 mm frei.

Die Maschinenbedienung kann so die Übereinstimmung zwischen den Karten und einem Beleg feststellen.

Eine mechanische Drucktaste an der vorderen Verkleidung des Zufuhrmagazins ermöglicht es, das Kartenpaket zurückzuschieben, um eine oder mehrere (max. 10) Karten einschieben zu können.

### 2.4.2. Funktionieren

Die Kartenzufuhr ist azyklisch. Jeder Lesebefehl bewirkt, falls die Einheit zur Verfügung steht, den Transport einer Karte, die ohne Zwischenhalt sofort vom Zufuhrmagazin ins Ablagefach transportiert wird. Das Lesen erfolgt kontinuierlich Spalte für Spalte.

Die Reihenfolge der Karten bleibt im Ablagefach erhalten.

Zeitangaben:

Die Durchführung eines Lesebefehls enthält:

- Zufuhr der Karte unter die Lesestation (1. Sp. bereit gelesen zu werden) 110 ms  $\pm$  5
- Das eigentliche Lesen (80 Spalten) 290 ms  $\pm$  15
  - oder 1 Spalte 3,4 - 3,9 ms

Anmerkung:

Die letzten Spalten einer Karte brauchen nicht in den Zentralspeicher übertragen worden zu sein (z.B. ungelochte Spalten). In diesem Fall betrifft die zu berücksichtigende Lesezeit nur die effektiv übertragenen Spalten. Das bedeutet, daß das durch den Lesebefehl "erstartete" Programm nach dem Übertrag der nutzbaren Zeichen wieder gestartet werden und während des Endes des Kartenlaufs weiterarbeiten kann. Wird ein neuer Lesebefehl gegeben, wartet er solange, bis die folgende Karte gelesen werden kann.

2.4.3. Durchführung

2.4.3.1. Steuertasten

Außer der unter 2.4.1. erwähnten mechanischen Drucktaste, stehen der Bedienung zur Steuerung der Funktionen des Lesers drei Leucht-tasten am Leser zur Verfügung:

RES, SRD und EJ; außerdem eine Taste SRD auf der numerischen Tastatur und eine Anzeigelampe R an der Zentraleinheit.

Die Tasten

RES

(RESet) schaltet die automatische Kartenzufuhr ein (Lampe ist erleuchtet).

Diese Taste wird am Anfang der Arbeit und nach einer Störung, die das Anhalten des Lesers hervorgerufen hat, benutzt. Sie bewirkt die automatische Kartenzufuhr unter Steuerung durch das Programm.

SRD

(Single card ReaD) (Lampe leuchtet, wenn Taste ein-gerastet). Bewirkt Transport Karte für Karte.



Die Zufuhr jeder Karte, die durch Programm befohlen wurde, muß durch Druck auf die Taste SRD der numerischen Tastatur freigegeben werden, da das Programm sonst auf dem entsprechenden Befehl "erstarrt", der nicht durchgeführt werden kann.

Diese Art der Kartenzufuhr wird angewendet bei Arbeiten, deren Ablauf von der Bedienungskraft kontrolliert wird, wie z.B. visuelle Kontrolle jeder Karte, Synchronisation des Kartenlesens mit Tastatureingaben, Programmtests etc.

Anmerkung:

Im weiteren Sinne kann die Taste SRD auch dazu dienen, die automatische Kartenzufuhr zu unterbrechen, um Manipulationen vornehmen zu können.

In jedem Fall wird die automatische Zufuhr wieder eingeschaltet durch die Taste RES, nachdem man die Taste SRD durch einen leichten Druck ausgerastet hat.

EJ

(EJect) Bei jedem Tastendruck erfolgt Kartenzufuhr. Diese Kartenzufuhr erfolgt unmittelbar und unabhängig vom Programm. Es erfolgt kein Lesen.

Diese Taste darf nur benutzt werden, wenn das Programm oder der Leser nicht arbeitet, um nicht die Bearbeitung zu stören.

Lampe

R

(card Reader) zeigt folgende Störungen an:

- eine Karte kann aus folgenden Gründen nicht eingezogen werden: Zufuhrmagazin leer, Stau auf der Bahn, Ablagefach voll, Halt durch Taste SRD etc.
- Lesestromkreise gestört.

Diese Störungen werden bei der Karteneingabe entdeckt und bewirken das Halten der Einheit und in Folge davon auch das Halten des Programms auf dem nächsten Lesebefehl. Nach Beseitigung der Störung wird die Einheit durch die Taste RES wieder eingeschaltet.

Das erfordert, daß Karteien oder Programme durch eine Leerkarte abgeschlossen werden, damit eine Störung beim Lesen der letzten Karte (z.B. Lesefehler) der Maschinenbedienung angezeigt werden kann.

#### 2.4.3.2. Programmiernormen

Das Lesen einer Karte wird durch den einen Ein-/Ausgabebefehl (s.3.6.) angerufen. Er arbeitet ohne Halt auf dem Zeichen und kann darum nicht simultan mit einer Programmserie ablaufen (s.Kap. 4).

Die gelesenen Zeichen werden von links nach rechts in eine Normal-speicherzone übertragen, die für den Kartenleser vorgesehen ist und rechts durch eine Trennmarke abgeschlossen ist. Die Stelle dieser Trennmarke bestimmt das Ende des Übertrags, der vor dem Lesen der 80. Spalte liegen kann (s.3.6.).

Die übertragenen Informationen vom Sichtkartenleser werden durch einen Zwischencode dargestellt, der dem Stanzcode entspricht. Vor der Verarbeitung müssen sie in den internen ISO-Code übersetzt werden (s.3.7.).

#### 2.4.4. Das Lesen von Talonkarten

Das Lesen von 51-spaltigen Lochkarten auf dem Sichtkartenleser des GE-55 setzt nur eine mechanische Zusatzeinrichtung im Ablagefach voraus. Es handelt sich um eine Anschlagleiste, die leicht einzusetzen und zu entfernen ist.

51- und 80-spaltige Lochkarten werden mit der gleichen Geschwindigkeit von 150 Karten pro Minute gelesen. Sie können nicht zusammen in einem Paket verarbeitet werden.

Die Codierung der IOC-Befehle entspricht der für 80-spaltige Lochkarten. An die 52. Stelle der Einlesezone muß eine Trennmarke gesetzt werden. Wird die Trennmarke nicht gesetzt, werden die restlichen 29 Stellen mit nicht definierbaren Zeichen ausgefüllt. Darüber hinaus wird die Verarbeitung nicht wieder aufgenommen, solange keine Trennmarke erkannt wird.

## 2.5. Der Kartenstanzer PS 40

### 2.5.1. Darstellung

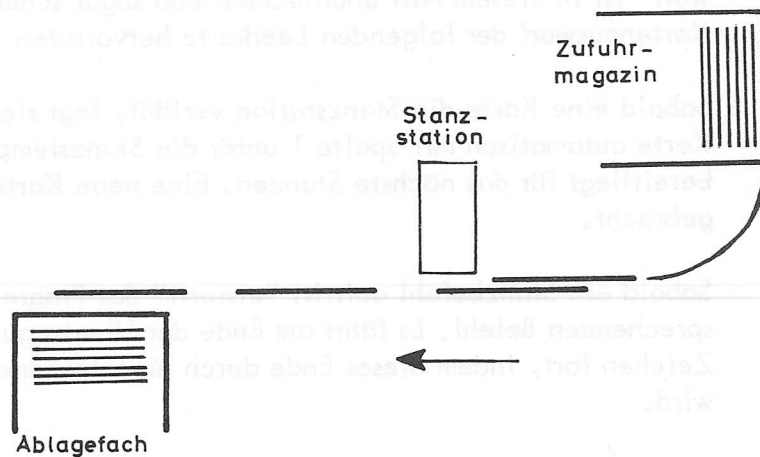
Dieser ist dazu bestimmt, 80-spaltige Lochkarten entweder im Code T 121 oder im Code H 14.012, zu stanzen.

Die Karten werden mit einer Geschwindigkeit von 40 Spalten/Sek. gestanzt. Nichtgelochte Felder werden mit der gleichen Geschwindigkeit überschritten. Außerdem können die Karten am Ende des Stanzvorganges mit einer Geschwindigkeit von 80 Spalten/Sek. ausgeworfen werden. Das wird in Abschnitt 2.5.2 besonders gezeigt.

Das Stanzen wird kontrolliert. Ein Fehler oder ein falsches Stanzen hält sofort die Einheit an. (S. 2.5.3.).

Der Stanzer PS 40 enthält:

- ein Zufuhrmagazin von 500 Karten,
- eine Stanzstation mit 12 Stanzstempeln,
- ein Ablagefach für 500 Karten.



Es ist möglich, eine Karte sofort unter die Stanzstation einzulegen.

Die Karten werden mit der Druckseite nach vorn transportiert und sind vom Ausgang der Stanzstation an sichtbar. Unter der Voraussetzung, daß die Karte übersetzt wurde, kann die Bedienungskraft Kenntnis vom Inhalt der Karte nehmen oder sie prüfen.

Die letzte Karte, die am Ablagefach angelangt, ist von den anderen Karten getrennt und kann leicht herausgenommen werden (Fehlerhafte Karte etc.).

## 2.5.2. Funktionieren

Die Kartenzufuhr und das Weiterrücken der Karte auf der Bahn des Stanzers PS 40 erfolgt Gang für Gang. Drei Gänge sind erforderlich, um eine Karte aus dem Zufuhrmagazin bis unter die Stanzstation zu bringen und zwei weitere, um sie nach dem Stanzen in das Ablagefach zu transportieren (s. Schema im Abschnitt 2.5.1.).

Der Transport auf der Bahn wird durch das Programm gesteuert. Jeder Stanzbefehl startet einen Gang, in dem die Karte, die sich vor der Stanzstation befindet, diese durchschreitet und auf jeder Spalte anhält. Der Übergang zur folgenden Spalte hängt vom Eintreffen eines Zeichens aus der Zentraleinheit ab. Dieses Zeichen kann entweder ein zu lochendes Zeichen sein, das Zeichen "Leerspalte" oder ein Steuerschlüssel "Auswurf", der anzeigt, daß das Stanzen für diese Karte beendet ist. Sie wird danach ohne Zwischenhalt von der Stanzstation "ausgeworfen". Fehlt eins der oben angeführten Zeichen, bleibt die Karte auf der laufenden Spalte stehen und erwartet den nächsten Übertrag eines Zeichens.

Wurde die Karte bis zur Spalte 80 gelocht, wird sie automatisch ausgeworfen, sobald diese Spalte gelocht wurde. Der Steuerschlüssel "Auswurf" ist in diesem Fall unbrauchbar und sogar schädlich, da er den Kartenauswurf der folgenden Leerkarte hervorrufen würde.

Sobald eine Karte die Stanzstation verläßt, legt sich die folgende Karte automatisch mit Spalte 1 unter die Stanzstempel, so daß sie bereitliegt für das nächste Stanzen. Eine neue Karte wird auf die Bahn gebracht.

Sobald ein Stanzbefehl auftritt "erstarrt" das Programm auf dem entsprechenden Befehl. Es fährt am Ende der Übertragung der zu stanzenden Zeichen fort, indem dieses Ende durch eine Trennmarke festgestellt wird.

Die Karten werden in derselben Reihenfolge abgelegt, in der sie eingeführt wurden.

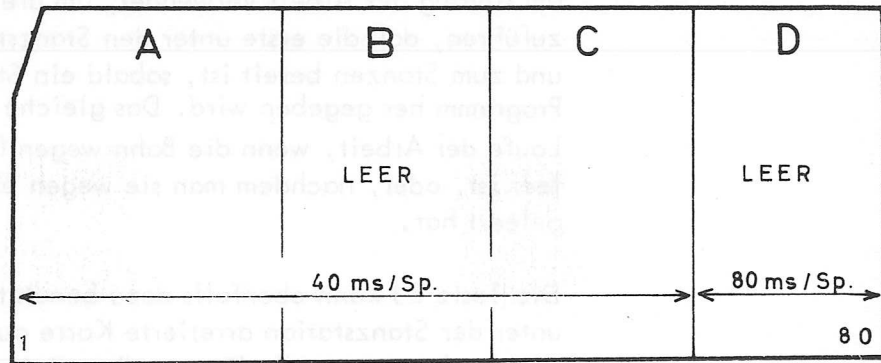
### Zeitangaben:

- |                                     |         |
|-------------------------------------|---------|
| - Lochen ohne Übersetzen, je Spalte | 25 ms   |
| - Lochen mit Übersetzen, je Spalte  | 50 ms   |
| - Auswurf, je Spalte                | 12.5 ms |
| - Zeit zwischen zwei Karten         | 330 ms  |

Die Stanzzeit wird errechnet, indem man die Zahl der überschrittenen Spalten von Spalte 1 an bis zum Kartenauswurf zählt. Sie bestimmt die Durchführungszeit des Befehls.

Die Auswurf- und Wartezeit erlaubt, die zwischen zwei aufeinanderfolgenden Stanzungen zur Verfügung stehende Zeit zu bestimmen.

Beispiel:



2.5.3. Durchführung

2.5.3.1. Steuertasten

Zur Steuerung des Stanzers PS 40 stehen der Bedienungskraft drei Tasten an der Einheit und eine Lampe an der Zentraleinheit zur Verfügung.

Die Tasten

EJ

(EJect) Bei jedem Druck erfolgt eine Kartenzufuhr und ein Kartentransport ohne Stanzen. Die Taste EJ wird am Anfang der Arbeit verwendet, um drei Karten so einzuführen, daß die erste unter den Stanzstempeln liegt und zum Stanzen bereit ist, sobald ein Stanzbefehl vom Programm her gegeben wird. Das gleiche erfolgt im Laufe der Arbeit, wenn die Bahn wegen fehlender Karten leer ist, oder, nachdem man sie wegen einer Störung geleert hat.

Die Taste EJ kann ebenfalls dazu benutzt werden, eine unter der Stanzstation arretierte Karte auszuwerfen und sie abzulegen, um die Karte während des Entstehens herauszunehmen (s. unten).

Die Taste darf nur benutzt werden, wenn der Stanzer nicht arbeitet.

RES

(RESet) setzt die Maschine in Arbeitsbereitschaft. Diese Taste wird benutzt, um den Normalzustand der Einheit nach einigen unten beschriebenen Störungen wiederherzustellen.

PRT

Das Ausrasten der Taste PRT und das Anlegen der Farbrolle an das Schreibrad bewirkt das Beschriften der gestanzten Werte. Bei eingerasteter Taste PRT wird nicht beschriftet.

Lampe

PU

(card PUnch) zeigt folgende Störungen an:

- Fehlen einer Karte unter der Stanzstation
- Volles Ablagefach
- Stanzfehler



### 2.5.3.2. Beseitigung von Störungen

#### Fehlen einer Karte unter der Stanzstation

Dieser Fehler tritt auf, wenn entweder keine Karten im Zufuhrmagazin sind oder eine Stauung aufgetreten war (z.B. Stau am Karteneinlaß). In einem Zufuhrgang wird er entdeckt und beim nächsten Stanzbefehl, der darum nicht durchgeführt werden kann, angezeigt. Er ruft einen Halt des Programms und der Randeinheit hervor.

Nachdem die Ursache des Halts beseitigt ist - in den meisten Fällen dürfte ein Nachfüllen des Magazins genügen - füllt die Bedienungskraft die Bahn mit der Drucktaste EJ. Sobald nun eine Karte vor der Stanzstation liegt, wird der wartende Befehl durchgeführt und das Programm fährt fort.

#### Volles Ablagefach

Diese Störung wird entdeckt und angezeigt unter den gleichen Voraussetzungen wie die vorige. Nachdem das Fach geleert wurde, startet die Bedienung die Arbeit durch einen Druck auf RES.

#### Stanzfehler

Wird eine fehlerhafte Einstellung der Stanzorgane im Laufe des Stanzens einer Spalte der Karte festgestellt, wird die nächste Spalte nicht gestanzt und die Einheit hält an. Gleichzeitig blockiert sie den Stanzbefehl und damit auch das Programm. Daraus ergibt sich, daß ein Fehler, der während des Stanzens der letzten Spalte einer Karte (z.B. Spalte 80) auftritt, erst entdeckt wird, wenn die nächste Karte, die sich in diesem Augenblick noch in Wartestellung befindet, durch das Programm gestanzt werden soll. Die Daten der verstanzten Karten stehen aber nicht mehr zur Verfügung, um ein erneutes Stanzen vornehmen zu können.

Um das zu verhindern, ist es erforderlich, nach jedem Stanzbefehl eine Spezialtestoperation zu programmieren, die weiter keine Aufgabe hat als zu untersuchen, ob die vorhergegangene Operation ohne Störung abgelaufen ist. Ist das nicht der Fall, blockiert das Programm also entweder auf dem Stanzbefehl (Fehler bei einer anderen Spalte als der letzten) oder auf dem Testbefehl (Fehler in der letzten Spalte).

Auf diese Weise stehen die Daten der verstanzten Karte noch in der Stanzzone zur Verfügung und man kann nun ein erneutes Stanzen mit Hilfe des Unterbrechungsprogramms starten.

Hierzu muß man:

- das Unterbrechungsprogramm (durch Druck auf die Taste PI) einschalten
- die Karte, die sich unter der Stanzstation befindet, auswerfen (durch einen Druck auf die Taste EJ).
- den Stanzer durch einen Druck auf die Taste RES starten; damit wird zunächst der blockierte Befehl zu Ende geführt.  
Handelt es sich bei diesem Befehl um den Stanzbefehl selbst, wird eine weitere Lochkarte verstanzt, die ebenfalls unbrauchbar ist; handelt es sich aber um den Testbefehl, ist keine besondere Manipulation nötig.

In beiden Fällen tritt nach Beendigung des Befehles das Unterbrechungsprogramm an die Stelle des Hauptprogrammes und man kann aufgrund der Besonderheiten dieses Programmes eine neue Karte stanzen. Es braucht nicht besonders erwähnt zu werden, daß die eine oder zwei fehlerhaften Karten aus dem Kartenstapel entfernt werden müssen.

### 2.5.3.3. Normen der Programmierung

#### Stanzen

Die Informationen, die gestanzt werden sollen, müssen in einem Zwischencode, der dem Stanzcode entspricht, dargestellt werden. Sie müssen also vor der Ausgabe übersetzt werden (s. 3.7.).

Vorher wurden sie in eine Normalspeicherzone übertragen, die dieser Randeinheit zugeordnet ist und rechts durch eine Trennmarke begrenzt wird. Diese wird nicht übersetzt und bestimmt sowohl das Ende des Übersetzens wie auch des Übertrags an den Stanzer PS 40.

Wird eine Karte nicht bis zur Spalte 80 gestanzt, muß eine Stelle zwischen den letzten zu stanzenen Zeichen und der Trennmarke vorgesehen werden. Diese erhält nach dem Übersetzen den Steuerschlüssel "Auswurf", der durch eins der Zeichen 09, 0B, 0D oder 0F dargestellt wird.

Das Stanzen der Karten wird durch den Ein-/Ausgabebefehl gesteuert. Er wird, wie bereits erwähnt, mit "Halt auf dem Zeichen" durchgeführt. Diese Eigenart wird bei der Multiprogramming benutzt (s. Kap. 4).

#### Test

Obwohl diese Operation keinen Übertrag bewirkt, benötigt sie trotzdem 2 Bytes im Zentralspeicher, von denen das erste irgendein Zeichen und das zweite eine Trennmarke F4 enthält.

Die Testoperation wird durch einen Ein-/Ausgabebefehl unmittelbar nach dem zu testenden Stanzbefehl programmiert.

Anmerkung:

Das Stanzen einer Karte kann nicht in verschiedenen Abschnitten durch mehrere Stanzbefehle erfolgen. Umgekehrt ist es auch nicht möglich, mit einem Befehl, mehrere Karten zu stanzen.

2.3.3.4. Beschriften auf dem Stanzer PS 40

Das Beschriften wird durch eine Zusatzeinrichtung ermöglicht. Die der Stanzung entsprechenden Zeichen werden an den oberen Rand der Lochkarte geschrieben. Die Beschriftung von vorgelochten Karten ist nicht möglich.

Taste

Die Zusatzeinrichtung wird durch die Taste "PRT" am Stanzer eingeschaltet. Die Taste hat zwei mögliche Stellungen:

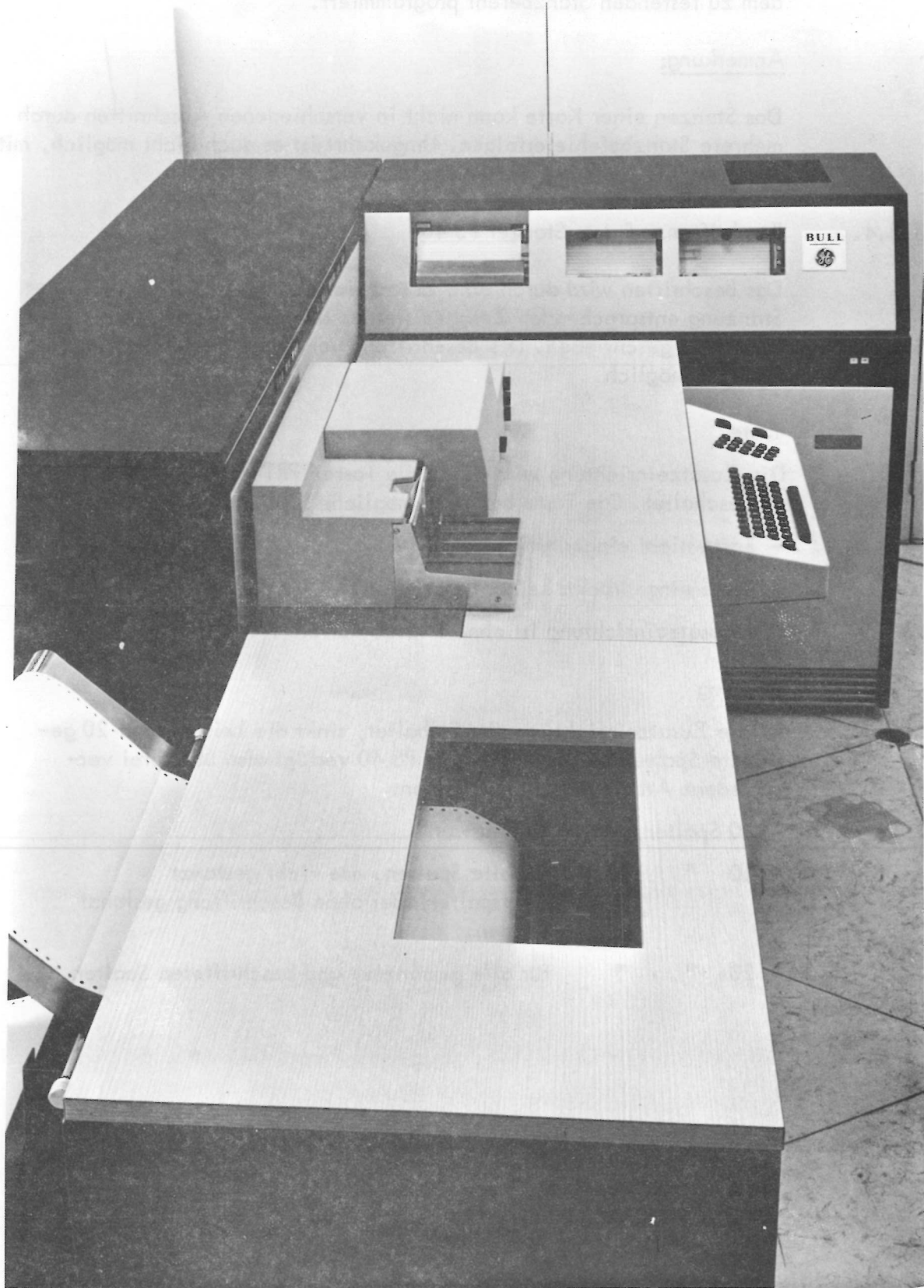
- Taste nicht eingedrückt: Beschriftung
- Taste eingedrückt: keine Beschriftung

Die Zusatzeinrichtung ist ohne Einfluß auf die Programmierung.

Leistung

Ist die Zusatzeinrichtung eingeschaltet, sinkt die Leistung auf 20 gestanzte Spalten pro Sekunde. Der PS 40 verfügt also über drei verschiedene Arbeitsgeschwindigkeiten:

- 80 Spalten/sek. bei Auswurf
- 40 " " für alle Spalten, die nicht gestanzt (Leerspalten) oder ohne Beschriftung gestanzt werden.
- 20 " " für alle gestanzten und beschrifteten Spalten.



GE-55

Ref.-Nr.: 23.20.001 D

Juli 1968

2.6. Der Drucker MB 50

2.6.1. Beschreibung

2.6.1.1. Allgemeine Merkmale

Der Drucker MB 50 druckt 132 Zeichen pro Zeile mit einer Geschwindigkeit von 50 Zeichen in der Sekunde. Die zu druckenden Zeichen bestehen aus 64 Symbolen, die zur Darstellung des internen Codes dienen. Die Breite der Druckformulare kann zwischen 7,62 cm und 38 cm schwanken. Der Druckabstand beträgt horizontal 1/10" (10 Zeichen je Zoll) und vertikal 1/6" (6 Zeilen je Zoll). 1- 6 facher Formularsatz kann verwendet werden.

2.6.1.2. Beschreibung

Der Drucker ist in einem kleinen Schrank an der linken Seite des Arbeitsplatzes untergebracht. In der Abdeckplatte befindet sich ein Glasfenster, um den Druck und den Papiervorschub kontrollieren zu können. Die Papierbahn ist folgendermaßen eingeteilt:

vorne: - die Papiervorschubeinrichtung  
- die Druckstation

hinten: - eine verstellbare Bremsvorrichtung zur Papiereinführung  
- eine Papierablage mit der Vorrichtung zum Feststellen 'Papierende'.

Das Papier wird auf beiden Seiten durch zwei Traktoren geführt, die auf einer horizontalen Achse sitzen. Der linke Traktor ist mit einer Haltevorrichtung versehen, die den Vorschubstreifen aufnimmt. Dieser läuft mit dem Traktor synchron. Außerdem ist hier die Abtastvorrichtung der Lochungen dieses Streifens angebracht. Mit dieser Einrichtung wird das Papier auf einer gewissen Höhe nach einem Sprung angehalten.

Die Traktoren lassen sich seitlich verschieben und zwar unabhängig voneinander. Nach dem Einlegen des Papiers werden die Traktoren verriegelt. Das Papier kann nun manuell gefaßt und z. B. auf die erste zu druckende Zeile eingestellt werden.

Der bewegliche Druckkopf besteht aus einer rotierenden Zeichenwalze und einem Hammer. Eine Kontrolleinrichtung stellt das evtl. Zerreißen des Papiers fest.

Weiter befindet sich dort noch die Transportvorrichtung für das Farbband.



### 2.6.1.3. Papierqualitäten

Werden Formulasätze verwendet, darf das Gesamtgewicht des Satzes einschließlich Kohlepapier nicht schwerer als 350 gr/m<sup>2</sup> sein.

Beispiel: 5 Blatt Papier von 50 gr/m<sup>2</sup> = 250  
4 Blatt Kohlepapier zu 20 gr/m<sup>2</sup> = 80  

---

330 gr/m<sup>2</sup>

### 2.6.2. Arbeitsweise

#### 2.6.2.1. Drucken

Das Drucken einer Zeile wird durch Programm ausgelöst und erfolgt Zeichen für Zeichen je nach Übertrag aus der Zentraleinheit.

Der Druckkopf kann 132 Stellen - von links an numeriert von 1 bis 132 - bedrucken. Er bewegt sich von links nach rechts beginnend bei Stelle 1 und hält an jeder Stelle an. Der Weitertransport zur nächsten Stelle erfolgt nach Ankunft eines Zeichens. Dieses kann ein druckbares Symbol, ein Zeichen für Leerstelle oder auch ein Zeichen für den Wagenrücklauf sein. In diesem Fall läuft der Druckkopf zur Stelle 1 zurück. Erhält er an einer Stelle kein Zeichen, bleibt er bis zum nächsten Übertrag dort stehen. Man kann so eine Zeile in mehreren Abschnitten drucken oder zwei Zeilen mit Papiervorschub ohne Wagenrücklauf. Diese Zeilen ergänzen sich, wodurch ein spürbarer Zeitgewinn entsteht.

Ist eine Zeile gedruckt, wird der Druckkopf normalerweise auf die Stelle 1 zurücktransportiert, um die nächste Zeile zu drucken. Der Rücksprung erfolgt automatisch, wenn die Stelle 132 erreicht und eins der oben erwähnten Zeichen übertragen wurde, oder wenn ein Steuerschlüssel 'Wagenrücksprung' übertragen wurde.

Aus mechanischen Gründen ist es verboten, einen Wagenrücksprung zu befehlen, wenn der Druckkopf sich bereits an Stelle 1 befindet.

#### 2.6.2.2. Papiervorschub

Der Papiervorschub wird durch Steuerschlüssel ausgelöst. Sie werden wie Daten an den Drucker übertragen. Sofort nach dem Übertrag eines solchen Schlüssels erfolgt der Papiervorschub.

Der Papiervorschub kann das Papier um eine oder mehrere Zeilen weitertransportieren oder auf eine bestimmte Zeile springen lassen.



Die Zeilenabstände werden einzeln nacheinander durch den Steuerschlüssel "Zeilenvorschub" gesteuert. Das bedeutet, daß soviel Steuerschlüssel wie Zeilenabstände gewünscht werden, zu übertragen sind.

Es ist möglich, einen Zeilenvorschub mit einem Wagenrücklauf nach Stelle 1 simultan ablaufen zu lassen. Das erfolgt durch einen speziellen Steuerschlüssel. Hingegen ist es, wie bereits gesagt wurde, verboten, hintereinander mehrere Male diesen Schlüssel zu verwenden.

Der Papiersprung wird durch einen Steuerschlüssel "Sprung" ausgelöst. Der Sprung wird durch den Vorschubstreifen auf der gewünschten Zeile angehalten. Der Vorschubstreifen stellt das zu druckende Formular dar. Seine Länge ist gleich oder ein vielfaches der des Blattes. Er ist in Zeilen unterteilt. Lochungen in einem Kanal bestimmen die Zeilen, auf die man springen will. Der Streifen wird zu einem Ring zusammengeklebt und synchron mit dem Papier transportiert. Dadurch hat jedes Blatt ein und dasselbe Sprungprogramm. Die Lochungen durchlaufen ein besonderes Abfühlsystem. Nach dem Starten eines Sprungs wird dieser durch die erste Lochung, die abgeföhlt wird, gestoppt.

#### 2.6.2.3. Auswirkungen auf das Programm

Sobald ein Druckbefehl auftritt, hält das Programm auf dem entsprechenden Befehl während des Übertrags der Zeichen und/oder der Steuerschlüssel an, d.h. während der Ausführung des Druckens und der Hilfsfunktionen (Wagenrücklauf, Papiervorschub etc.). Das Programm läuft beim Übertrag des letzten Zeichens oder Schlüssels, das durch eine darauf folgende Trennmarke angezeigt wird, wieder an. Die letzte durchzuföhrende Operation läuft dann simultan mit dem Wiederanlauf des Programms ab.

#### 2.6.2.4. Zeitangaben

Drucken eines Zeichens oder Leerstelle	20 ms
Wagenrücklauf (maximal)	450 ms
Papiervorschub je Zeile	40 ms
Sprunggeschwindigkeit	25 Zeilen/sec.

Die Druckzeit wird durch die Anzahl der zu druckenden Zeichen von Stelle 1 bis zum Wagenrücklauf bestimmt.

Die Druckzeit hängt von der Anzahl der zu druckenden Stellen ab. Die Maximalzeit entspricht also dem Rücklauf über 132 Stellen. Der Zeilenvorschub erfolgt simultan mit dem Wagenrücklauf und braucht nicht berücksichtigt zu werden.

## 2.6.3. Bedienungselemente

### 2.6.3.1. Steuertasten

Außer den manuellen Einstellelementen, die in Abschnitt 2.6.1 erwähnt wurden und die sich auf der Papiervorschubeinrichtung befinden, stehen der Maschinenbedienung vier Drucktasten an der Vorderseite der Randeinheit und eine Lampe an der Zentraleinheit zur Verfügung.

Die Taste

SCR

(Skip and Carriage Return) bewirkt einen Papiersprung und den Rücklauf des Schreibkopfes zur Stelle 1. Mit dieser Taste kann man vor Beginn der Arbeit den Papiertransport prüfen.

RD

(lower ribbon to ReaD) bewirkt ein Abheben des Farbbandes, um die Lage des Papiers vor dem Schreibkopf prüfen zu können.

RES

(RESet) startet den Drucker nach einem Halt, der durch "Papierende" hervorgerufen wurde.

MUL

ermöglicht, die Anschlagstärke auf 1-3 oder 4-6 Blätter umfassende Formulare einzustellen.

Die Lampe

PR

(PRinter) zeigt "Papierende" an.

Das Papierende wird am Ende des nächsten Papiervorschubs angezeigt und ruft den Halt des Druckers und des Programms beim nächstfolgenden Druckbefehl hervor.

- Benutzt das Programm nur einen Papiersprung bei vollem Blatt, erfolgt der Halt am Anfang des nächsten Blattes. Es genügt also, einen neuen Papierstapel einzulegen und das neue Formular auf die erste Zeile einzustellen, um die Arbeit fortzusetzen. Durch Betätigung der Taste RES startet der Druck wieder.
- Benutzt das Programm dagegen mehrere Sprünge, kann der Halt in der Mitte eines Blattes erfolgen. Man muß also erst das Formular zu Ende schreiben, um das Papier wechseln zu können. Dazu drückt man so oft wie erforderlich die Taste RES. Bei jedem Druck auf die Taste wird bis zum nächsten Sprung geschrieben oder, da die Kontrolle jedesmal anspricht, erfolgt wiederholter Programmhalt. Sobald das Blatt zu Ende gedruckt ist, liegt der vorige Fall vor und man kann jetzt das Papier wechseln.

Anmerkung:

Um die Arbeit der Maschinenbedienung zu erleichtern, kann das Programm so organisiert sein, daß ein fakultativer Halt durch den Befehl KHLT am Blattanfang vorgesehen ist. Sobald nun Papierende angezeigt wird, gibt die Bedienung diesen Befehl durch Betätigen der Taste KHLT an der Zentraleinheit frei. Der Druck wird nun, wie oben beschrieben, wiedergestartet. So wird der Blattwechsel durch Programmblockade angezeigt.

2.6.3.2. Normen der Programmierung

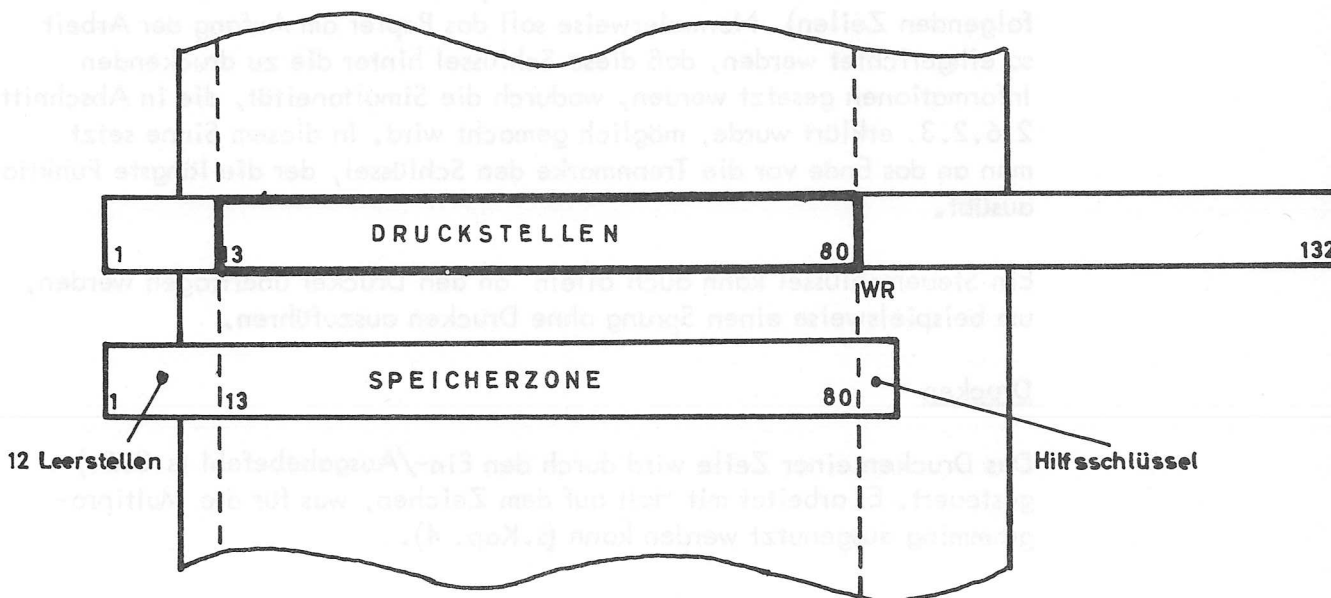
Die Informationen, die auf dem Drucker MB 50 geschrieben werden sollen, müssen im internen ISO-Code dargestellt werden. Sie werden aus einer Normalspeicherzone des Zentralspeichers, die für diese Randeinheit angelegt wurde, übertragen.

Informationsaufbau in der Normalspeicherzone

Die Kapazität dieser Zone wird durch die Maximalzahl der Stellen und Steuerschlüssel bestimmt. Jede vom Druckkopf eingenommene Stelle, beginnend mit Stelle 1, wird in dieser Zone durch ein Byte dargestellt, das entweder ein Druckzeichen oder ein Zeichen 'Leerstelle' (20) enthalten kann.

Wird der Druck einer Zeile vor der Stelle 132 beendet, brauchen die rechts freibleibenden Stellen nicht belegt zu sein. Beginnt der Druck dagegen nicht auf Stelle 1, müssen die nicht benutzten Stellen links vom ersten zu druckenden Zeichen durch Leerzeichen ausgefüllt sein. Das führt zu einem Zeitverlust, der vermieden wird, wenn man die Formulare möglichst weit links beginnen läßt.

Beispiel: Es ist eine Liste zwischen den Druckstellen 13 und 80 zu drucken.



Es gibt eine Ausnahme von der oben gesagten Regel. Aus Abschnitt 2.6.2.1 geht hervor, daß der Druckkopf an einer beliebigen Stelle am Ende einer Operation stehen bleiben kann und daß man später den Druck an dieser Stelle fortsetzen kann. Beim Wiederanlauf muß man diese Zwischenstelle so behandeln, als wäre es die Stelle 1 in der Speicherzone und dementsprechend die zu druckenden Informationen verschieben. Im oben angegebenen Beispiel müßte man, wenn man einen Druck von Stelle 40 an wiederaufnehmen möchte, die an diese Stelle zu druckenden Zeichen in der Ausgabezone an Stelle 1 und nicht an Stelle 13 oder 40 plazieren. Die Informationen in der Normalspeicherzone werden rechts durch eine Trennmarke begrenzt, die das Ende des Übertrags angibt.

#### Die Steuerschlüssel

Zusätzliche Stellen müssen in ausreichender Zahl für die Steuerschlüssel freigehalten werden, die mit den zu druckenden Daten übertragen werden.

Diese Schlüssel werden durch folgende Zeichen dargestellt:

Aufgabe	Zeichen
Wagenrücklauf	OD
Zeilenvorschub	OA
Wagenrücklauf und Zeilenvorschub	OB
Sprung	OC

Sie können irgendeine Stelle in der Normalspeicherzone belegen, entweder am Anfang oder am Ende (Zeilenvorschub vor oder nach dem Druck) oder zwischen zwei zu druckenden Zeichen (Druck von zwei aufeinanderfolgenden Zeilen). Normalerweise soll das Papier am Anfang der Arbeit so eingerichtet werden, daß diese Schlüssel hinter die zu druckenden Informationen gesetzt werden, wodurch die Simultaneität, die in Abschnitt 2.6.2.3. erklärt wurde, möglich gemacht wird. In diesem Sinne setzt man an das Ende vor die Trennmarke den Schlüssel, der die längste Funktion ausübt.

Ein Steuerschlüssel kann auch allein an den Drucker übertragen werden, um beispielsweise einen Sprung ohne Drucken auszuführen.

#### Drucken

Das Drucken einer Zeile wird durch den Ein-/Ausgabebefehl (s.3.6.) gesteuert. Er arbeitet mit Halt auf dem Zeichen, was für die Multiprogramming ausgenutzt werden kann (s.Kap. 4).

Volles Blatt

Der Blattwechsel wird bestimmt durch das Programm, das in diesem Fall die Zeilen zählt, um "Volles Blatt" festzustellen.

Der Drucker 141 druckt 128 Zeichen pro Zeile mit einer Geschwindigkeit von 100 Zeilen in der Minute bei Druck von allen 64 Zeichen oder mit einer Geschwindigkeit von 500 Zeilen in der Minute bei Druck der 48 gebräuchlichsten Zeichen. Der Zeichenabstand beträgt 10 Zeichen pro Zoll. Die Breite der Druckformulare kann zwischen 10,16 cm und 40,60 cm schwanken. Der Zeichenabstand beträgt 6 oder 8 Zeichen pro Zoll. Der 141 kann die zu 6-fache Formularegröße bedrucken.

2.7 Der Drucker I 41

2.7.1. Beschreibung

2.7.1.1. Allgemeine Merkmale

Der Drucker I 41 druckt 128 Zeichen pro Zeile mit einer Geschwindigkeit von 100 Zeilen in der Minute bei Druck von allen 64 Zeichen oder mit einer Geschwindigkeit von 200 Zeilen in der Minute bei Druck der 48 gebräuchlichsten Zeichen. Der Zeichenabstand beträgt 10 Zeichen pro Zoll. Die Breite der Druckformulare kann zwischen 10,16 cm und 40,60 cm schwanken. Der Zeilenabstand beträgt 6 oder 8 Zeilen pro Zoll. Der I 41 kann bis zu 6-fache Formulare drucken.





Die Trennmarke F4 auf der 130. Stelle der Druckzone ist nur beim Drucken aller 64 Zeichen erforderlich. Der Schlüssel auf Stelle 129 wird durch den Druckvorgang nicht zerstört.

Bei Druck der 48 häufigsten Zeichen wird nur ein Druckbefehl mit Anschlußcode 1E gegeben. Bei Druck von 64 Zeichen wird nach dem ersten Druckbefehl ein Umschlüsselungsbefehl TRD 5D R programmiert, der die bereits gedruckten Zeichen in Blanks, die Sonderzeichen in normal druckbare Zeichen und den Funktionsschlüssel von 64 Zeichen in den Funktionsschlüssel von 48 Zeichen umwandelt. Ein Zeilen- bzw. Papier-vorschub ist noch nicht wirksam geworden. Durch einen zweiten programmierten Druckbefehl werden die restlichen Sonderzeichen gedruckt; anschließend wird der Funktionsschlüssel wirksam.

Beispiel für Druck von 48 häufigen Zeichen und Test auf volles Blatt:

INC 82 3D 81 29 81 28  
IOC 53 02 1E 08 1D 08  
JIERT 04 30 80 00 F1 xx (xx= Adresse, wenn volles Blatt)

Beispiel für Druck von 64 Zeichen und Test auf volles Blatt:

INC 82 4D 81 29 81 28  
IOC 53 01 1E 08  
TRD 5D 08  
IOC 53 02 1E 08 1D 08  
JIERT 04 30 80 00 F1 xx (xx= Adresse, wenn volles Blatt)

Simultaneität des I 41:

(s. 4.2.4.)

Funktion	48 Zeichen	64 Zeichen
keine Zwischenzeile	4C	3C
Einmaliger Zeilenvorschub	4D	3D
Doppelter Zeilenvorschub	4E	3E
Dreifacher Zeilenvorschub	4F	3F
Papiervorschub + Stop	51	31
Kanal A		
Papiervorschub + Stop	52	32
Kanal B		

## 2.8. Die Magnettrommel

### 2.8.1. Beschreibung

#### 2.8.1.1. Allgemeine Merkmale

Die Trommel ist ein Zusatzspeicher und dient sowohl zum Speichern von einem oder mehreren Programmen als auch Dateien. Schematisch gesehen ist die Trommel ein rotierender Zylinder, auf dessen Außenseite sich parallel laufende Einspeicherungsbahnen befinden. Sie dreht sich mit einer Geschwindigkeit von 3000 Upm vor einem Lese-Schreibaggregat vorbei.

Die Trommel kann 89.600 Stellen aufnehmen. Diese Stellen sind auf 128 Speicherbahnen zu je 700 Bytes (= Stellen) verteilt. Jede Bahn besitzt noch eine 701. Stelle, deren Anwendung später erklärt wird. Jede Bahn kann in einer beliebigen Reihenfolge angesprochen werden, um einen Informations- oder Programmblock von variabler Länge zu lesen und zu schreiben.

Die Lese/Schreibgeschwindigkeit einer Bahn beträgt 20 ms. Das entspricht einer Trommelumdrehung. Die Zugriffszeit auf eine Bahn schwankt zwischen 0 und 20 ms je nach Stellung des Bahnbeginns zum Lese-Schreibkopf im Augenblick des Ein/Ausgabebefehls. In der Praxis wird sie jedoch bei maximal 5 ms liegen, nämlich der Zeit, die zwischen dem Geben des Befehls und dessen Ausführung liegt; 5 ms, die zur Durchführung anderer Ein-/Ausgabebefehle benutzt werden können oder sogar zum Programmablauf bei Multiprogramming, wurden abgezogen.

Es erfolgt eine Prüfung des Unpaarigkeitsbits bei Überträgen auf oder von der Trommel.

#### 2.8.1.2. Die Informationen auf der Trommel

Die Speichereinheit ist das Byte (8 Informationsbits und ein Unpaarigkeitsbit). Die Informationen werden so auf die Trommel geschrieben, wie sie übertragen werden; ungepackt (Daten im ISO-Code) oder gepackt (algebraische Zahlen, Befehle in Maschinensprache etc.).

Sie werden, wie bereits gesagt wurde, in Blocks zusammengefaßt, und zwar je ein Block pro Bahn. Der Block ist eine Zusammenfassung von Zeichen, die auf einmal übertragen werden können. Der Block steht linksbündig auf der Bahn, und wird rechts durch eine Trennmarke abgeschlossen, die sich bei voller Belegung der Bahn an der 701. Stelle befindet.

## 2.8.2. Arbeitsweise

### 2.8.2.1. Lesen, Schreiben und Kontrolle

Jede Bahn besitzt einen Lese-Schreibkopf, vor dem sie sich ständig dreht. Die Trommel ist also mit 128 LS-Köpfen, die von 0 bis 127 numeriert sind, ausgestattet. Zu einem gegebenen Zeitpunkt ist immer nur ein LS-Kopf über den schnellen Kanal mit der Zentraleinheit in Verbindung.

Die Verbindung zu einem gegebenen Kopf wird durch das Programm gesteuert. Sie besteht einfach darin, durch eine Spezialoperation die Nummer des gewünschten LS-Kopfes (= der Bahn) an die Trommel zu übertragen. Die Nummer wird bei ihrer Ankunft an der Trommel kontrolliert. Wird ein Fehler festgestellt (Paarigkeit, rechtes Halbbyte nicht numerisch, Adresse > 127), kann die Auswahl der Bahn nicht erfolgen und die nächste Operation, normalerweise Lesen oder Schreiben, wird blockiert.

Wird kein Fehler festgestellt, ist die Verbindung hergestellt und wird solange aufrechterhalten, bis eine neue Bahn vom Programm her angerufen wird. Man kann also die eingespeicherte Information auf der ausgewählten Bahn lesen oder eine neue Information dort speichern. Der vom Programm befohlene Übertrag mit der Zentraleinheit erfolgt, ob Lesen oder Schreiben, von der 1. Stelle der Bahn an, ohne anzuhalten, Stelle für Stelle, und endet erst bei Feststellung der Trennmarke, welche die Information beendet. Ist diese Trennmarke nicht vorhanden oder der Informationsblock größer als 700 Zeichen, endet der Übertrag am Ende der Bahn: Das Ergebnis ist eine Kapazitätsüberschreitung, die gespeichert wird.

#### Schreiben

Die im Zentralspeicher am Ende der Information befindliche Trennmarke wird mit übertragen. Die übertragenen Zeichen einschließlich Trennmarke werden bei Ankunft an der Trommel geprüft. Wird eine Paarigkeit festgestellt, wird dieser Fehler vorläufig gespeichert, der Übertrag geht aber weiter. Die Trommel regeneriert aber auf jeden Fall das unpaarige Zeichen, bevor es geschrieben wird.

#### Lesen

Die sich auf der Trommel am Ende der Information befindende Trennmarke wird in den Zentralspeicher übertragen. Die auf der Trommel gelesenen Zeichen einschließlich Trennmarke werden, bevor sie an die Zentraleinheit übertragen werden, geprüft. Wird eine Paarigkeit festgestellt, wird auch dieser Fehler gespeichert, das Lesen geht aber weiter. Ebenfalls regeneriert die Trommel auch hier das Unpaarigkeitsbit, bevor es übertragen wird. Entsteht ein Fehler während des Übertrags, so wird er bei der nächsten Auswertung der Information im Zentralspeicher festgestellt.

### 2.8.2.2. Bearbeitung der Fehlerfälle

Im vorhergehenden Abschnitt wurde gesagt, daß die verschiedenen auftretenden Fehlerfälle keinen Einfluß auf den Ablauf haben, aber die Fehlerfälle oder Kapazitätsüberschreitungen gespeichert werden. Das hat zur Folge, daß die Trommel am Ende der Operation unbenutzbar ist und die nächste Trommeloperation nicht durchgeführt werden kann. Es ist jedoch möglich, das Programm früher anzuhalten, d.h. am Ende der falschen Operation durch eine spezielle Testfunktion. Diese Operation hat nur die Aufgabe zu testen, ob die Trommel frei ist.

### 2.8.2.3. Einwirkungen auf das Programm

Sobald das Programm einen Befehl gibt (Auswahl der Bahn, Lesen, Schreiben oder Test), bleibt es auf diesem Befehl stehen, bis die Operation ausgeführt ist.

Ist die Trommel verfügbar, d.h. daß bei der vorhergehenden Operation kein Fehler aufgetreten ist, so kann der Befehl von der Trommel übernommen werden. Im gegenteiligen Fall wartet der Befehl so lange, bis die Auswirkungen des Fehlers durch die Drucktaste RES (s. nächsten Abschnitt) aufgehoben sind.

Die Test- und Bahnauswahlbefehle werden, wenn die Trommel verfügbar ist, sofort übernommen. Die Lese- und Schreibbefehle dagegen können nur in den 5 ms von der Trommel übernommen werden, die der Ankunft des Bahnbeginns am LS-Kopf vorangehen. Während dieser Wartezeit kann die Zentraleinheit andere Operationen ausführen, so wie es bereits in Abschnitt 2.8.1. beschrieben wurde.

### 2.8.2.4. Ausführungszeiten

Auswahl einer Bahn, im Durchschnitt	10 ms
Lesen oder Schreiben, je Byte incl. TM	28,6 µs

2.8.3. Auswertung

2.8.3.1. Tasten und Lampen

Folgende Fehlerfälle werden an der Zentraleinheit durch die Kontroll-Lampe

**EXT**

(EXTernal device) angezeigt:

- Paarigkeit eines übertragenen Zeichens
- Kapazitätsüberschreitung
- Nummer der Bahn größer 127 oder falsch (nicht numerisch)
- Schreibversuch auf abgeriegelter Bahn (für Software belegte Bahn)

Mit Ausnahme des Erstgenannten sind alle anderen Programmierfehler und werden daher meistens während des Testens auffallen. Um den Fehler einzukreisen und evtl. das Programm weiter zu testen, steht dem Programmierer eine Kontrolleinrichtung an der Trommel zur Verfügung. Diese enthält folgende Lampen und Drucktasten:

- 2 Lampen, welche die Fehloperationen angeben
- 4 Lampen, die die Fehler angeben
- 1 Drucktaste.

Die Lampen

Operation	Auswahl Bahn	Schreiben	Lesen
L + E		X	X
ECR		X	X
Fehler:			
CLE	Paarigkeit in der übertragenen Nummer	Paarigkeit in geschriebenen Zeichen	Paarigkeit in gelesenen Zeichen
DP		TM fehlt oder Block > 700 Bytes	TM fehlt
CAPA	Nummer > 127	Versuch auf eine verriegelte Bahn zu schreiben	
DCB	Ein Zeichen der Nummer ist keine Ziffer		



Drucktaste

RES

(RESet) Hebt die Folge der Kontrolle auf:  
Die Trommel übernimmt die aufgeschobene Operation

Anmerkung:

Bei reeller Arbeit ist es streng untersagt, einen Fehlerfall wie Paarigkeit oder Kapazitätsüberschreitung zu übergehen, da diese mit dem Risiko der Übernahme falscher Daten verbunden sind. Man soll darum, sobald eine Störung auftritt, die Arbeit sofort anhalten, die Art des Fehlers feststellen, diesen Fehler beseitigen und dann erst die Arbeit wieder aufnehmen, so daß man korrekte Angaben bekommt. Das Anhalten kann durch die Testoperation erfolgen, die programmiert ist. Handelt es sich um einen Paarigkeitsfehler, kann man versuchen, die gestörte Operation durch ein Wiederherstellungsprogramm, das durch Programmunterbrechung angerufen wird, wieder anzufangen (s.2.1.6.).

2.8.3.2. Programmiernormen

Arbeiten mit einer Trommel

Auswahl einer Bahn

Die Nummer der auszuwählenden Bahn ist eine Zahl von drei Ziffern, die in ungepackter Form ausgedrückt wird. Nullen links können durch Leerstellen (20) ersetzt werden. Nur die rechten Halbbytes werden von der Trommel berücksichtigt. Rechts wird diese Zahl durch eine Trennmarke F 4 abgeschlossen. Im Zentralspeicher wird sie in einer Normalspeicherzone von 4 Bytes Länge gespeichert und zur Trommel als Daten übertragen.

Lesen

Der Informationsblock (incl. TM F 4) wird im Zentralspeicher in einer der Trommel zugeordneten Normalspeicherzone gespeichert. Dieser Speicherbereich kann noch Trennmarken F 4 von vorhergehenden Operationen enthalten. Diese werden entweder gelöscht oder stehen weiter rechts als der einzulesende Block. Sie brauchen also nicht beachtet zu werden.

Schreiben

Der Informationsblock, der auf die Trommel übertragen werden soll, wird in einem Normalspeicherbereich zusammengestellt und rechts mit einer Trennmarke F4 versehen. Eine Trennmarke in der Mitte des Blocks ist verboten.

Anmerkung:

Der gleiche Speicherbereich kann für alle Überträge von oder nach der Trommel vorgesehen werden.

Test

Durch die Testoperation wird die vorangegangene Operation überprüft. Bei ordnungsgemäß durchgeführter Operation wird an der Stelle, die im Register angegeben ist, das Testzeichen "10" gespeichert. Durch Analyse des Testzeichens kann gegebenenfalls eine falsch durchgeführte Operation wiederholt werden.

Befehle

Alle oben erwähnten Operationen werden durch den Ein/Ausgabebefehl (s.3.6.) ausgeführt.

Der Testbefehl muß hinter jedem Lese- oder Schreibbefehl programmiert werden.

Auf diese Weise bleibt das Programm, wenn ein Fehler während der Überträge aufgetreten sein sollte, auf diesem Testbefehl stehen und alle Informationen (Bahnnummer, Daten) stehen noch zur Verfügung, um die gestörte Operation erneut zu beginnen, wie es in Abschnitt 2.7.3.1 beschrieben wurde.

Arbeiten mit zwei Trommeln

Die oben angegebenen Normen gelten auch bei Anschluß von zwei Trommeln für jede von ihnen, wenn sie im gleichen Programm benutzt werden.

Die folgenden Zonen im Zentralspeicher müssen aber für jede Trommel angelegt werden:

- eine Normalspeicherzone "Nummer der Bahn" zur Auswahl der Bahn auf der gewünschten Trommel.
- n Normalspeicherzonen zur Aufnahme der Daten
- n Register, die die Basisadressen dieser Zonen enthalten
- 1 Byte für die Testoperation.

Anmerkung:

$N = 1$ , wenn Ein- und Ausgabe wechselweise in der gleichen Normalspeicherzone erfolgen.

Durch diese Organisation kann man die zwei Trommeln wie zwei verschiedene Speicher betrachten und sie unabhängig voneinander anrufen, z.B. in Zusammenhang mit dem Multiprogramming (s. Kap.4). Ebenso kann man in ein und demselben Befehl Ein/Ausgabeoperationen für beide Trommeln simultan befehlen. Es ist lediglich zu beachten, daß die betr. Operationen nacheinander in nicht festgelegter Reihenfolge je nach Freigabe der Trommeln ablaufen.

Diese Organisation ist dagegen sehr aufwendig, wenn man die beiden Trommeln nur abwechselnd in einem Programm benutzt, besonders, wenn beide Trommeln nur einen Speicher mit doppelter Kapazität bilden. In diesem Fall genügt es, die oben angeführten Zonen nur einmal anzulegen und sie je nach dem für die eine oder andere Trommel zu gebrauchen. Ebenso können diese Operationen nicht gleichzeitig in ein und demselben Ein/Ausgabebefehl befohlen werden, da sie in einer bestimmten Reihenfolge in Abhängigkeit von dem Vorhandensein der Informationen ablaufen.

Anmerkung:

Diese Operationen können nacheinander durch den gleichen Ein/Ausgabebefehl angerufen werden. Dieser besondere Operationstyp kann nacheinander verschiedene Randeinheiten anrufen (IØIC s.3.6.).

Beispiele zur Verdeutlichung der Anwendung von Ein/Ausgabebefehlen sind in Abschnitt 3.6. aufgeführt.

Durch diese Organisation kann man die zwei Themen wie zwei ver-  
schiedene Aspekte betrachten und sie unabhängig voneinander an-  
zusehen, z.B. in Zusammenhang mit dem Mehrprogramm (s. Kap. 4). Ebenso kann  
man in ein und demselben Bereich Eiz/Ausgabenoperationen für beide  
Themen simultan betrachten. Es ist lediglich zu beachten, daß die bet-  
reffenden Operationen nacheinander in nicht festgelegter Reihenfolge je nach Frei-  
gabe der Themen zu erfolgen.

Diese Organisation ist dagegen sehr aufwendig, wenn man die beiden  
Themen nur oberflächlich in einem Programm behaltet, besonders, wenn  
beide Themen zu einem Zweifach mit doppelter Kapazität führen. In  
diesem Fall genügt es, die oben angeführten Themen nur einzeln anzulegen  
und sie je nach dem für die eine oder andere Themen zu gestalten.  
Ebenso können diese Operationen nicht gleichzeitig in ein und demselben  
Eiz/Ausgabenbereich betrieben werden, da sie in einer bestimmten Reihenfolge  
in Abhängigkeit von dem Vorhandensein der Informationen ablaufen.

Anhang

Dieser Anhang kann nacheinander durch den gleichen Eiz/Ausgaben-  
bereich organisiert werden. Dieser besondere Operationstyp kann nachein-  
ander verschiedene Kombinationen anfordern (s. 3.5.).  
Beispiele zur Veranschaulichung der Anwendung von Eiz/Ausgabenbereich  
sind im Anhang 3.5. beigefügt.

2.9. Lochstreifenleser / -stanzer

2.9.1. Allgemeines

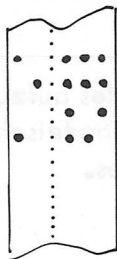
1. Technische Daten

Der Lochstreifenleser/-stanzer besteht aus einem Gerät, in das zwei verschiedene Einheiten (Leser und Stanzer) eingebaut sind; die Schaltkreise und Anschlüsse beider Einheiten sind voneinander unabhängig und benutzen zwei Normkanäle des Systems GE-55. Jede Einheit kann unabhängig von der anderen montiert werden.

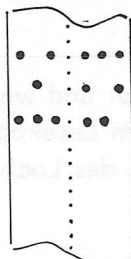
Beide Einheiten verwenden Lochstreifen mit runden Löchern, die den Normen ISO/TC97/SC4/GT3 und ECMA 10 entsprechen.

- Beim Lesen, die Lochstreifen 11/16 Zoll, 7/8 Zoll und 1 Zoll:

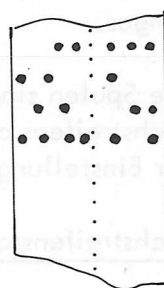
11/16"  
(5 Kanäle)



7/8"  
(6 Kanäle)



1"  
(7 oder 8 Kanäle)

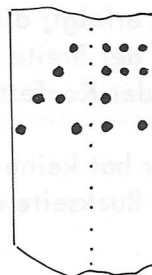


- Beim Stanzen, die Lochstreifen 11/16 Zoll und 1 Zoll:

11/16"  
(5 Kanäle)



1"  
(6 oder 8 Kanäle)



- Lesegeschwindigkeit: 125 Zeichen pro Sekunde
- Stanzgeschwindigkeit: 105 Zeichen pro Sekunde

## 2. Beschreibung

Von aussen gesehen ist der Lochstreifenleser/-stanzer ein niedriger Schrank, der die logischen Schaltkreise enthält. An der Vorderseite sind eine Reihe Tasten angebracht; die für die Steuerung des Lesers benötigten Tasten befinden sich rechts, die des Stanzers links. In der Mitte der Tastenreihe sind zwei Tasten angebracht, die das Ein- und Ausschalten beider Einheiten ermöglichen (s. auch 4.9.3. 1.).

Auf diesem niedrigen Schrank sind rechts der Lochstreifenleser und links der Lochstreifenstanzer montiert.

### - Lochstreifenleser:

Auf der Vorderseite des Lesers befinden sich die Ab- und Aufwickelspulen sowie der Lesekopf; diese drei Elemente sind ausserhalb des Gehäuses angebracht. Die beiden Spulen werden elektrisch angetrieben. Ein mit REV (REVerse) gekennzeichnete Schalter, der neben dem Lesekopf montiert ist, ermöglicht das Steuern des Rückspulvorganges.

Die Spulen sind abnehmbar und weisen die Breite des aufzunehmenden Lochstreifens auf. Eine am Lesekopf angebrachte Rändelschraube dient zur Einstellung der Breite des Lochstreifenlaufweges.

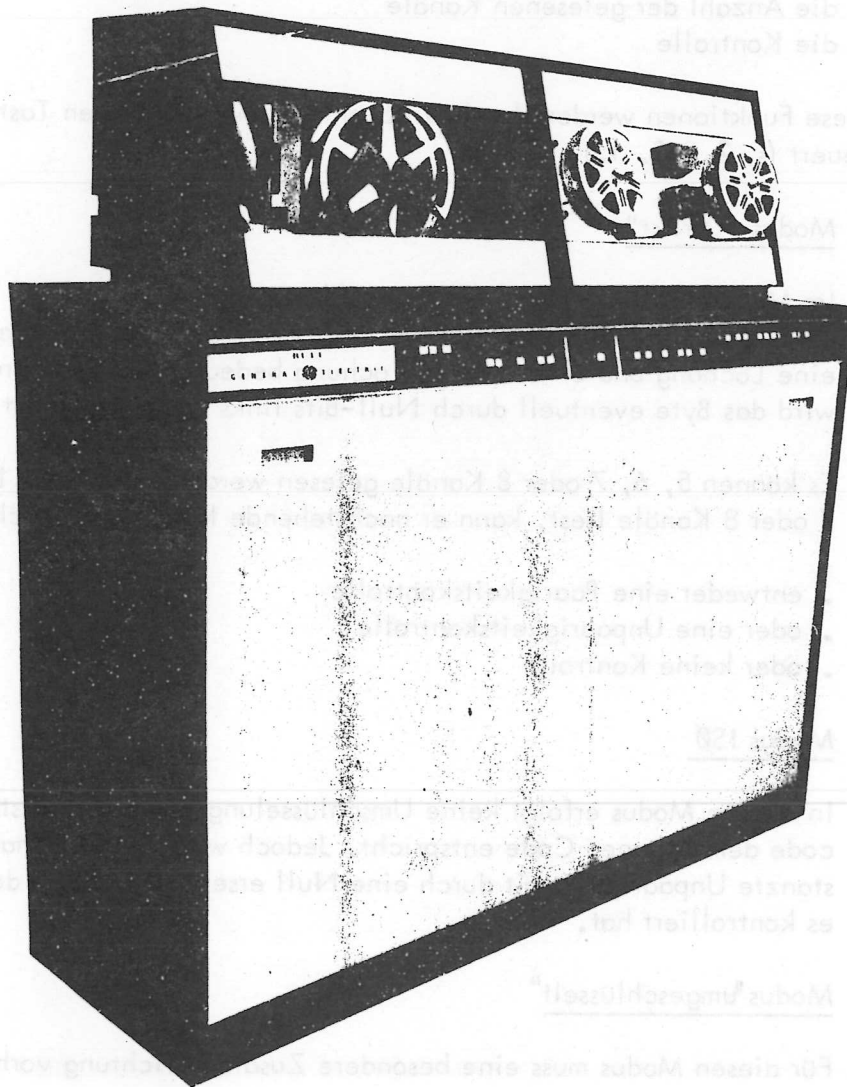
### - Lochstreifenstanzer:

Die Vorderseite des Stanzers besteht aus einer verglasten Tür, hinter der die Abwickelspule für den Leerstreifen montiert ist. Diese Spule läuft frei auf ihrer Achse und ist abnehmbar; sie hat die Breite des aufzunehmenden Lochstreifens.

Links ist eine zweite Tür vorhanden, durch die der Zugang zum Stanzbloc erfolgt; die hier angebrachte Rändelschraube dient zur Einstellung der Breite des Lochstreifenlaufweges. Durch diese Tür kann auch der Konfettkasten zum Entleeren herausgenommen werden.

Der Stanzer hat keine Aufwickelspule; der gestanzte Lochstreifen tritt an der Rückseite des Gerätes aus und fällt in einen Auffangkorb.





## 2.9.2. Arbeitsweise

### 1. Lochstreifenleser

Der Lochstreifenleser kann wahlweise folgende Funktionen ausführen:

- den Lesemodus
- die Anzahl der gelesenen Kanäle
- die Kontrolle

Diese Funktionen werden durch Drücken der entsprechenden Tasten gesteuert (s. 2.9.3. 1.).

#### - Modus "Direkt"

Im Modus "Direkt" wird keine Umschlüsselung vorgenommen; das Zeichen wird dem Kernspeicher als Byte übermittelt, indem eine 1 eine Lochung und eine 0 keine Lochung bedeutet; wenn erforderlich, wird das Byte eventuell durch Null-Bits links komplementiert.

Es können 5, 6, 7 oder 8 Kanäle gelesen werden. Wenn der Leser 6, 7 oder 8 Kanäle liest, kann er nachstehende Kontrollen durchführen:

- entweder eine Paarigkeitskontrolle
- oder eine Unpaarigkeitskontrolle
- oder keine Kontrolle

#### - Modus ISØ

In diesem Modus erfolgt keine Umschlüsselung, da der Lochstreifen-code dem internen Code entspricht. Jedoch wird das in Kanal 8 gestanzte Unpaarigkeitsbit durch eine Null ersetzt, nachdem der Leser es kontrolliert hat.

#### - Modus "umgeschlüsselt"

Für diesen Modus muss eine besondere Zusatzeinrichtung vorhanden sein, die umschlüsselt:

- vom IBM-FRIDEN Code (8 Kanäle) in den internen Code (1)
- oder vom TELEX Code (5 Kanäle) in den internen Code.

Die Lesekontrolle wird dabei automatisch durchgeführt.

- (1) Genau gesagt, führt der Leser erst die Umschlüsselung des IBM-FRIDEN Code in einen Zwischencode durch, der dann - bei Benutzung der Basissprache - erneut vom GE-55 durch den Basisbefehl TR 1 umgeschlüsselt wird.

Jeder Code besitzt eine gewisse Anzahl von "nicht auswertbaren" Zeichen, die vom Leser zwar erkannt, dem Kernspeicher aber nicht übermittelt werden. Es handelt sich :

- im Code ISØ um:

- das Leerzeichen (kein Kanal gestanzt)
- das Löschzeichen (alle Kanäle gestanzt)

- im IBM-FRIDEN Code um:

- das Leerzeichen
- jedes Zeichen, das eine Lochung in Kanal 8 aufweist

- im Lesemodus "Direkt" um:

- das Leerzeichen.

Alle vorhergenannten Zeichen sind Füllzeichen, die beim Lesen einfach unterdrückt werden.

- im TELEX Code um:

- das Leerzeichen, das als Füllzeichen dient
- die Aufrechterhaltungscodes:

Buchstaben - Kanäle 1, 2, 3, 4, 5  
Ziffern - Kanäle 1, 2, 4, 5

Diese beiden Zeichen werden vom Leser ausgewertet, um die Umschließung fortsetzen zu können.

Der Leser verarbeitet Zeichen für Zeichen, d.h. auf jedes Signal der Zentraleinheit hin übermittelt er ihr ein Zeichen. Der Leser kann zwischen zwei aufeinanderfolgenden Zeichen anhalten.

Bei Drücken der Taste SST sucht der Leser das erste Zeichen und speichert es unter dem Leseknopf. Bei Eingang des Signals der Zentraleinheit wird das Zeichen gelesen und dem Speicher des GE-55 übermittelt; dann sucht der Leser sofort das nächste Zeichen und der Vorgang wiederholt sich.

Tritt ein Paarigkeitsfehler auf, wird das Zeichen dem GE-55 nicht übermittelt; der Leser stoppt. Der Operator kann das Zeichen untersuchen, ohne hierbei den Lochstreifen berühren zu müssen:

- entweder durch Drücken der Taste **STOP** und manuelle Handhabung des Lochstreifens,
- oder durch Drücken der Taste **KCK**, dann wird das fehlerhafte Zeichen unverändert dem GE-55 übermittelt. Anschliessend wird der Lesevorgang fortgesetzt.

Bei Lochstreifenende hält der Leser an; er startet erneut, nachdem ein neuer Streifen aufgelegt wurde. Wenn das Ende des Lochstreifens auch Dateiende bedeutet, muss der Bediener erst die Taste **STOP** und dann die Taste **KCK** drücken, damit der Lesevorgang mit Füllzeichen ergänzt wird.

## 2. Lochstreifenstanzer

Die Arbeitsweise des Lochstreifenstanzers entspricht der Arbeitsweise des Lesers; wie beim Leser besteht die Möglichkeit, mehrere Stanzarten durchzuführen.

Jede Stanzart wird durch Drücken der entsprechenden Taste gesteuert (s. 2.9.3. 1.).

### - Modus "Direkt"

Bei diesem Modus ist ein Umschlüsseln nicht zulässig; das Byte wird in den 8 Kanälen des 1 Zoll-Lochstreifens gestanzt, wobei eine Lochung für eine 1 und keine Lochung für eine 0 vorgenommen wird. Will man nur 7, 6 oder 5 Kanäle stanzen, muss das Programm das Zeichen entsprechend aufbereiten.

Wird ein Lochstreifen von 11/16 Zoll verwendet, kann ein Programmierfehler Lochungen ausserhalb oder knapp am Rand des Lochstreifens verursachen; dies wird vom Stanzer nicht überprüft.

### - Modus ISØ

Bei diesem Modus wird keine Zeichenumschlüsselung durchgeführt. Jedoch wird, wenn erforderlich, eine Unpaarigkeitslochung in Kanal 8 hinzugefügt (das entsprechende Bit im Kernspeicher ist immer eine 0).

### - Modus "umgeschlüsselt"

Für diesen Modus muss eine besondere Zusatzeinrichtung vorhanden sein; je nach Art der Zusatzeinrichtung kann nachstehende Umschlüsselung vorgenommen werden:

- vom internen Code in den IBM-FRIDEN-Code (8 Kanäle) (1)
- vom internen Code in den TELEX-Code (5 Kanäle)

Wenn die zum Stanzer übermittelte Kernspeicherzone andere Werte als den internen ISO-Code enthält, kann das Ergebnis der Umschlüsselung völlig falsch sein.

Bei einer Umschlüsselung im TELEX-Code erzeugt der Stanzer selbst die Code "Buchstabe" und "Ziffer", die die ausgegebenen Informationen verständlich machen.

Der Lochstreifenstanzer verarbeitet Zeichen für Zeichen; er stanzt die Zeichen in den Lochstreifen entsprechend ihrer Übermittlung von der Zentraleinheit und kann jederzeit stoppen.

Bei Annäherung an das Streifenende hält der Stanzer an und die Lampe **ECN** leuchtet auf. Der Operator kann dann den Lochstreifen austauschen. Wenn jedoch bekannt ist, dass nur noch ein paar Zeichen zu stanzen sind, kann der Operator die Taste **ECN** drücken, um den Stanzvorgang bis zum Ende des Lochstreifens ausführen zu können.

### 2.9.3. Handhabungen und Funktionen

#### 1. Steuerpult

Das Steuerpult des Lochstreifenlesers/-stanzers ist in drei Teile aufgeteilt:

- In der Mitte der gemeinsame Teil, bestehend aus den Tasten:

<b>ON</b>	Einschalten (Power ON)	schaltet das Gerät ein. Leuchtet die Taste auf (grünes Licht), ist die Stromzufuhr hergestellt.
<b>OFF</b>	Ausschalten (Power OFF)	schaltet das Gerät aus. Wenn die Taste aufleuchtet (rot), ist das Gerät ausgeschaltet, aber die Netzspannung ist noch vorhanden.

- Rechts der "Leserteil", bestehend aus den Tasten:

<b>SST</b>	Start (STarT)	startet den Leser. Ist die Taste erleuchtet, führt der Leser Arbeiten durch.
------------	------------------	--

(1) Genau gesagt führt der Stanzer im Falle des IBM-FRIDEN-Code zuerst eine Umschlüsselung von einem Zwischencode durch, der - bei Verwendung der Basissprache - vorher durch den Umschlüsselungsbefehl TRB erzeugt wurde.

<b>STP</b>	Stopp (SToP)	stoppt den Leser. Wenn die Taste aufleuchtet, ist der Leser gestoppt (die Taste <b>ON</b> ist dabei gedrückt).
<b>KCK</b>	Schlüssel- kontrolle (Key ChecKing)	leuchtet auf, wenn ein Fehler festgestellt wird. Ein Drücken der Taste verursacht die Übermittlung des unter dem Lesekopf vorhandenen Zeichen zum GE-55.
<b>MST</b>	Vorschub um 1 Zeichen (Move SStep)	Mittels dieser Taste kann der Lochstreifen um jeweils 1 Zeichen weitertransportiert werden.
<b>MFD</b>	Dauervorschub (Move Forward)	Solange diese Taste festgehalten wird, läuft der Lochstreifen vorwärts (die Taste rastet nicht ein).

Anmerkung:

Die beiden letztgenannten Tasten können nur dann verwendet werden, wenn der Lochstreifenleser angehalten wurde (**STP** gedrückt).

<b>LDT</b>	Laden Streifen (LoaD Tape)	Schaltet die Ab- und Aufwickelvorrichtung ein und setzt den Leser auf STOP, falls erforderlich. Die Taste rastet ein und leuchtet nach dem Drücken auf.  Ein zweites Drücken der Taste gibt sie frei und unterbricht die Spannungszufuhr zur Ab- und Aufwickelvorrichtung.
------------	-------------------------------	--

<b>ISO</b>	Code ISO	} Diese drei Tasten bestimmen den Lesemodus. Sie werden selbständig gehalten (nur eine kann jeweils gedrückt werden), leuchten jedoch nicht auf.
<b>TRP</b>	Code "Direkt" (TRansParent)	
<b>SPC</b>	Spezialcode (SPecialCode)	
<b>5CR</b>	Lesen 5 Kanäle (5 Channel Read)	} Diese Tasten bestimmen die Anzahl der zu lesenden Kanäle. Sie werden selbständig gehalten (nur eine Taste kann jeweils gedrückt werden) und leuchten nicht auf.
<b>6CR</b>	Lesen 6 Kanäle (6 Channel Read)	
<b>7CR</b>	Lesen 7 Kanäle (7 Channel Read)	
<b>8CR</b>	Lesen 8 Kanäle (8 Channel Read)	



OPC	} (Odd Parity Check) (No Parity Check) (Even Parity Check)	Im Modus "Direkt" bestimmen diese drei Tasten, ob und welche Lesekontrolle durchzuführen ist. Sie werden selbständig gehalten (nur eine Taste kann gedrückt werden) und leuchten nicht auf.
NPC		
EPC		

- Links der "Stanzerteil", bestehend aus den Tasten:

ISO	} Code ISO Code "Direkt" (TRAnsParent) Spezialcode (SPeCial Code)	Diese drei Tasten bestimmen den Stanzmodus. Sie werden selbständig gehalten (nur eine kann gedrückt werden) und leuchten nicht auf.
TRP		
SPC		

STT	Start (STArT)	startet den Stanzer; leuchtet die Taste, führt der Stanzer seine Funktionen aus.
-----	------------------	--

STP	Stop (SToP)	stoppt den Stanzer. Wenn die Taste aufleuchtet, ist der Stanzer gestoppt (die Taste <b>ON</b> ist dabei gedrückt).
-----	----------------	--

ECN	Streifenende Störung (End CaNcelling)	leuchtet bei Annäherung des Streifenendes und bei jeder Störung auf. Das Drücken der Taste startet erneut den Stanzer.
-----	---	--

PSP	Stanzen Antriebs- lochung (Punch SProket)	bewirkt das Stanzen von Leerzeichen (nur die Antriebsspur wird gestanzt).
-----	---	---

RCP	Stanzen Löschen zeichen (Reset Code Punch)	bewirkt das Stanzen von Löszeichen (alle Kanäle werden gestanzt).
-----	--	---

Anmerkung:

Der Stanzvorgang wird solange fortgesetzt, wie die beiden letztgenannten Tasten gedrückt bleiben. Beide werden manuell gehalten und leuchten nicht auf.

## 2. Programmierungsnormen

Die Informationen sind in einer dem jeweiligen Gerät zugeordneten Zone angeordnet oder ihr entnommen. Die Lese- oder Stanzfelder können deutlich getrennt sein oder auch nicht.

Beim Lesen wie beim Stanzen muss die Zone durch eine Trennmarke F4 begrenzt werden. Das Erkennen dieser Trennmarke durch die logischen Schaltkreise des GE-55 begrenzt den Informationsübertrag.

Der Lesebefehl bewirkt das Lesen des Lochstreifens, bis die Zone belegt ist. Der Stanzbefehl verursacht das Stanzen des gesamten Inhaltes der Zone ausser der Trennmarke. Diese Operationen werden wie im Absatz 4.9.2. beschrieben durchgeführt.

## 3. Simultaneität

Da der Lochstreifenleser und -stanzer Einheiten sind, die ein Zeichen nach dem anderen verarbeiten, können ihre Operationen gleichzeitig mit den anderen normalen Einheiten oder mit diesen Einheiten und der Verarbeitung durchgeführt werden, wobei der jeweils verwendete Anschlusscode zu berücksichtigen ist (s. 2.18.).

Das Lesen eines Zeichens belegt die Zentraleinheit während  $600 \mu\text{s}$  im Verhältnis zu den  $8 \text{ ms}$ , die für die Ausführung der gesamten Operation benötigt werden.

Das Stanzen eines Zeichens kann dagegen eine sehr verschiedene Belegungszeit des Kernspeichers erfordern. Der mechanische Teil des Stanzers hat einen Zyklus von  $9,5 \text{ ms}$ ; der Kernspeicher kann daher während einer Zeit belegt werden, die zwischen  $20 \mu\text{s}$  und der gesamten Zykluszeit liegen kann, je nachdem, wann der Anruf erfolgt.

2.10. Die Leitungssteuereinheit DATANET 51

2.10.1. Allgemeines

2.10.1.1. Kenndaten

Das DATANET 51 ist eine Steuereinheit zur synchronen Datenfernübertragung über eine Telefonleitung und an einen Hochleistungskanal der GE - 55 anschließbar.

Seine Aufgabe besteht darin, die von der GE - 55 gelieferten Zeichen auf die Telefonleitung zu übertragen und umgekehrt. Die Zeichen, welche aus 7 Datenbits und einem 8. ungeraden Paarigkeitsbit bestehen, werden mit einer Schrittgeschwindigkeit von 600 bis 2.400 Baud, entsprechend 75 bis 300 Zeichen pro Sekunde, übertragen.

Das DATANET 51 verfügt über eine entsprechende Hardwarelogik, die es ermöglicht, alle erforderlichen Signale zwischen GE - 55 einerseits und der entsprechenden Datenübertragungseinrichtung ( Modem ) andererseits auszutauschen.

Mit der Datenübertragungseinrichtung ist ein normaler Fernsprechapparat verbunden.

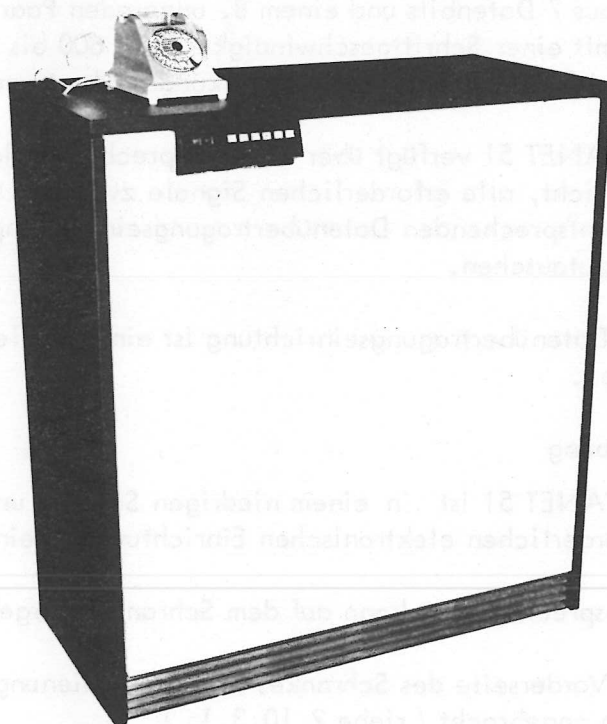
2.10.1.2. Beschreibung

Das DATANET 51 ist in einem niedrigen Schrank untergebracht, der alle erforderlichen elektronischen Einrichtungen beinhaltet.

Der Fernsprechapparat kann auf dem Schrank untergebracht werden.

Auf der Vorderseite des Schrankes sind die Bedienungstasten und Leuchtanzeigen angebracht ( siehe 2.10.3.1. ).

Das DATANET 51 muß in der Nähe der GE-55 installiert werden, damit der Bediener beide Einheiten leicht bedienen kann.



2.10.2. Arbeitsweise

2.10.2.1. Prinzip

Das DATANET 51 besitzt einen Pufferspeicher von 9 Bits ( 8 Informationsbits und 1 Paarigkeitsbit ). Beim Senden wird der Pufferspeicher durch die GE-55 bitparallel gefüllt und bitseriell über einen entsprechenden Modem zur Telefonieitung entleert. Beim Empfangen von Daten ist die Arbeitsweise umgekehrt.

Das DATANET 51 führt bestimmte Kontrollen durch und kann gewisse Steuerzeichen erkennen ( siehe nachstehende Beschreibung ).

2.10.2.2. Verbindungsaufbau und Verbindungsabbau

Die Verbindung DATANET 51 - Telefonieitung kann zwei verschiedene Betriebszustände einnehmen:

"TALK" : Der Fernsprechapparat ist mit der Telefonieitung verbunden, es besteht keine Verbindung zum DATANET 51.

"DATA" : Das DATANET 51 ist über den Modem mit der Telefonieitung verbunden, es besteht keine Verbindung zum Fernsprechapparat.

Der Übergang von "TALK" auf "DATA" erfolgt:

- entweder manuell durch Drücken der Taste TPH ( beim Anschluß an das SWFD-Netz der Deutschen Bundespost muß zusätzlich die Datentaste am Fernsprechapparat gedrückt werden )
- oder automatisch, wenn der verwendete Modem die Schnittstellenleitung M3 bedienen kann.

Anmerkung:

Der für das SWFD-Netz der Deutschen Bundespost bereitgestellte Postmodem D 1200 S besitzt technisch diese Fähigkeit. Zur Zeit ist aber ein automatischer Betrieb mit diesem Modem von Seiten der Deutschen Bundespost noch nicht zulässig.

Der Übergang von "DATA" nach "TALK" erfolgt durch einen entsprechenden Befehl der GE-55.

### 2.10.2.3. Befehle

Wenn das DATANET 51 im Zustand "DATA" ist, kann es 5 verschiedene Befehle ausführen:

- DATA  
Dieser Befehl schaltet den Modem an die Übertragungsleitung an.
- TRANSMIT  
Dieser Befehl veranlaßt die Übertragung eines Zeichens in den Pufferspeicher, die Aussendung des Zeichens über die Leitung mit einer vom Modem bestimmten Schrittgeschwindigkeit, die Übertragung des nächsten Zeichens usw. ...., bis eine Trennmarke F4 auftritt, die nicht übermittelt wird, aber die Operation stoppt.

Die gesendete Message muß ein bestimmtes Format haben.  
Das DATANET 51 führt beim Senden keine Kontrollen durch.

- RECEIVE  
Dieser Befehl setzt das DATANET in den Empfangszustand.  
Die Message muß mit Synchronisationszeichen beginnen (16 hexadezimal), die das DATANET 51 erkennen kann, jedoch nicht in den Kernspeicher überträgt. Anschließend muß ein Steuerzeichen STX ( 02 hexadezimal ), dann die Daten, ein Steuerzeichen ETX ( 03 hexadezimal ) und ein Paarigkeitszeichen folgen. Das DATANET 51 prüft das Format der Message, die es der GE - 55 mit den Zeichen "Anfang" und "Ende" übermittelt, und führt eine doppelte Paarigkeitskontrolle durch ( laterale und longitudinale Paarigkeitsprüfung).

Beim Übertrag zur GE - 55 wird das Paarigkeitszeichen durch eine Trennmarke F4 ersetzt.

- TEST  
Dieser Befehl veranlaßt den Übertrag eines Bytes zur GE - 55.
- PHONE

Dieser Befehl schaltet den Modem von der Übertragungsleitung ab und versetzt das DATANET 51 in den Betriebszustand "TALK". Die automatische Rufannahme wird außer Betrieb gesetzt.

Die Befehle werden wie folgt codiert:



Befehle	Anschlußcode an Kanal 1	Komplement
Data	17 hexadezimal	C2 hexadezimal
Transmit	16 "	entfällt
Receive	14 "	entfällt
Test	15 "	entfällt
Phone	17 "	C1 hexadezimal

### 2.10.3. Bedienung

#### 2.10.3.1. Steuerpult

Das Steuerpult des DATANET 51 weist nachstehende Tasten und Leuchtanzeigen auf:

- Tasten

**ON**

Stromversorgung "Ein"

Im Schalter leuchtet eine Kontroll-Lampe auf, wenn die Einheit eingeschaltet ist.

**OFF**

Stromversorgung "Aus"

Im Schalter leuchtet eine Kontroll-Lampe auf, wenn die Einheit ausgeschaltet ist, aber die Netzspannung noch vorhanden ist.

**TPH**

Telefon

Die Betätigung des Schalters versetzt das DATANET 51 in den Betriebszustand "TALK" (nur, wenn die Kontroll-Lampe MAN leuchtet).

Kontroll-Lampe leuchtet:  
Fernsprechapparat mit der Übertragungsleitung verbunden.

Kontroll-Lampe erloschen:  
DATANET 51 mit der Übertragungsleitung verbunden.

<b>600</b>	600 Bit pro Sekunde	}	Diese beiden Schalter ermöglichen die Auswahl der erforderlichen Schrittgeschwindigkeit für die Datenfernübertragung.
<b>1200</b>	1200 "		

Hinweis:

Statt 600 und 1200 können die Ziffern 1200, 2400 vorhanden sein.

<b>MAN</b>	Manual	}	*)	Manuelle Anschaltung der Übertragungsleitung ( durch Taste TPH )
<b>RIN</b>	Ring Indicator			Automatische Verbindungsannahme

\*) Die entsprechende Kontroll-Lampe leuchtet bei Betätigung auf.

Leuchtanzeigen

<b>DSR</b>	Diese Kontroll-Lampe leuchtet auf, wenn der Modem mit der Übertragungsleitung verbunden ist.
<b>SD</b>	Diese Kontroll-Lampe leuchtet auf, wenn das DATANET 51 Daten sendet
<b>WBR</b>	Diese Kontroll-Lampe leuchtet auf, wenn das DATANET 51 im Empfangszustand ist und keine Daten empfangen werden.
<b>RD</b>	Diese Kontroll-Lampe leuchtet auf, wenn das DATANET 51 Daten empfängt.

2.10.3.2. Programmierhinweise

Die Daten für den Sendevorgang müssen bereitgestellt und können in einem beliebigen Puffer des Kernspeichers empfangen werden; zum Senden und Empfangen können getrennte oder auch gemeinsame Puffer benutzt werden.

Die Steuerzeichen werden beim Senden und Empfangen in den vom Benutzerprogramm vorgegebenen Puffer geschrieben oder aus ihm gelesen.

Es ist nicht möglich, das DATANET 51 einzusetzen, ohne vorher einen geeigneten Dialogverkehr zu vereinbaren, der die Probleme des Anrufens, Quittierens, Wiederholens, usw.....regelt. Eine Hilfe hierfür ist GECOR ( General Electric Communication Routin ).



### 3. Die Basisbefehle des GE-55

#### 3.1. Einführung

Dem Benutzer des GE-55 stehen zur Programmierung der Arbeiten zwei symbolische Programmiersprachen zur Verfügung:

- die "Grundsprache" und
- der "Assembler Trommel", der die Grundsprache bei Verwendung der Magnettrommel ergänzt.

Diese Sprachen werden durch Standardprogramme interpretiert, die nach dem Laden die Programme in eine für den Rechner verständliche Maschinensprache umsetzen. Der "Assembler Trommel" ist in dem Handbuch "Programmiersystem GE-55 - Trommel" beschrieben.

Die "Grundsprache" ist Gegenstand des vorliegenden Kapitels.

Die Programme werden auf Kodierformularen (Best.-Nr. BULL 1009) geschrieben, die gleichzeitig Belege zum Ablocken der Programmkarten sind. Ein Musterformular wird in Kapitel 5 beschrieben, in dem ebenfalls die Richtlinien zur allgemeinen Programmierung gegeben werden.

#### 3.1.1. Darstellung der Grundsprache

##### 3.1.1.1. Allgemeine Definitionen

Diese Sprache heißt "Grundsprache", weil sie erstens für alle Konfigurationen des GE-55 anwendbar ist und zweitens der internen Maschinensprache am nächsten kommt.

Sie besteht aus Elementarbefehlen und symbolischen Pseudobefehlen.

Jedem Elementarbefehl entspricht ein Befehl in Maschinensprache, der von der Zentraleinheit verstanden wird. Er besteht aus einem symbolischen Schlüssel, der den Operationstyp identifiziert und einer variablen Anzahl Parameter, die

- Ergänzungsschlüssel (z.B. absolute Werte, Längenangaben, Codes der Randeinheiten etc.) oder
- Adressen von Daten oder Zonen sein können.

Einem Pseudobefehl dagegen entspricht kein Maschinenbefehl. Er dient lediglich dazu, Steuerimpulse an das "Programmsystem" zu geben, das nach dem Laden ein Programm in Maschinensprache umwandelt, z.B. Bestimmen der realen Adresse bei einem Programmsprung.

Die Pseudobefehle entsprechen in ihrem Aufbau den Elementarbefehlen und können darum jeden beliebigen Platz in einem vom Programmierer erstellten Programm belegen.

Ein in der Grundsprache geschriebenes Programm kann an beliebiger Stelle im Zentralspeicher stehen, natürlich im Rahmen seiner Gesamtkapazität. So kann man auch, unter den gleichen Bedingungen, Verschiebungen vornehmen, um Änderungen einzusetzen.

Die Grundsprache ermöglicht es auch, Programme in "Sektionen" aufzuteilen, die in gleicher Weise in beliebiger Reihenfolge eingespeichert und verschoben werden können. Eine "Sektion" wird als eine Folge von Befehlen verstanden, die immer nacheinander ablaufen. Um sie zu identifizieren genügt es, die reelle Adresse durch eine Verweisung zu symbolisieren. Durch einen solchen Programmaufbau kann man innerhalb einer Sektion Änderungen vornehmen, die im allgemeinen in einem Hinzufügen oder Wegnehmen von Befehlen bestehen, ohne die nicht betroffenen Sektionen zu berühren.

Um ein Grundprogramm in ein von der Maschine auswertbares Programm zu übertragen, übersetzt das "Programmsystem" die symbolischen Operationstypen in Binärcodes oder Operationstypen in Maschinensprache, teilt jedem Befehl eine Adresse im Kernspeicher zu und setzt aufgrund der symbolischen Verweisungen jeder Sektion die Adresse des ersten Befehls dieser Sektion ein.

#### 3.1.1.2. Die einzelnen Befehlstypen

Die in den folgenden Abschnitten beschriebenen Befehle lassen sich in folgende große Gruppen mit gemeinsamen Merkmalen einteilen:

- (3.2.) Befehle zur Programmverknüpfung: Programmsprünge.
- (3.3.) Registeroperationen: Rechenoperationen, numerische Vergleiche, Verschiebungen und Überträge.
- (3.4.) Operationen auf Zeichenbasis: Logische Operationen und Überträge im Normalspeicherbereich.
- (3.5.) Mehrfachüberträge: Von einer Zone nach mehreren anderen.
- (3.6.) Ein-/Ausgabeoperationen: Überträge zwischen dem Zentralspeicher und den Randeinheiten.
- (3.7.) Umschlüsselungsbefehle: Von externen Codes in den internen Code und umgekehrt.
- (3.8.) Hilfsbefehle: Zum Testen und Laden der Programme.



Eine achte Gruppe von Befehlen, die sich auf das Multiprogramming beziehen, sind in Kapitel 4 beschrieben.

### 3.1.2. Befehlsformat

#### 3.1.2.1. Die Maschinenbefehle

Das Befehlsformat hängt von der Art des Befehles ab. Von ihrer Struktur her kann man zwei Hauptklassen unterscheiden:

- einfache Befehle; Es erfolgt eine Operation, entweder ein Übertrag, eine logische Analyse oder eine Umwandlung. Sie geben eine große Flexibilität bei der Datenverarbeitung.
- komplexe Befehle; Es erfolgen mehrere Operationen, entweder Lesen oder Schreiben auf externe Randeinheiten oder gleichzeitiger Übertrag und Umwandlung. Diese Befehle entsprechen ganz besonders den Erfordernissen bei der Eingabe von Daten und bei der Ausgabe von Resultaten und optimieren die Leistung eines Programmes, indem sie seinen Umfang und die Zeit der Entschlüsselung der Operationstypen verringern. Sie geben eine große Beweglichkeit bei Operationen mit den Randeinheiten und ihrer Simultaneität.

#### Die einfachen Befehle

Die einfachen Befehle erfordern einen oder zwei Operanden. Sie bestehen aus:

- einem Operationstyp
- evtl. einem Ergänzungsschlüssel, der
  - . entweder eine Zahl N ist, die die Länge des (oder der) Operanden angibt
  - . oder ein Wert K, der entweder ein Bezugszeichen oder ein Operand ist.
- einer oder zwei Adressen, die die Operanden bezeichnen (A1 und A2).

An dieser Stelle sei darauf hingewiesen, daß die allgemeinen Adressen A ebenfalls die Nummer R eines numerischen Registers oder die Nummer B eines Basisregisters, ergänzt durch eine Verschiebung xxx, sein können (s. 2.1.4.).

### Die komplexen Befehle

Diese Befehle setzen sich zusammen aus:

- einem Operationstyp
- einem Nebenschlüssel N, der die Anzahl der durchzuführenden Operationen angibt
- den Adressen der entsprechenden Ein-/Ausgabemedien.

Zu diesen Befehlen gehören:

- die Befehle der Mehrfachüberträge, bei denen die Adressen durch die Verschiebungen D (die Basisregister sind einbegriffen) oder durch die Nummer R des numerischen Registers dargestellt werden.
- die Ein-/Ausgabebefehle, bei denen die Randeinheiten durch Spezialcodes C und die Ein-/Ausgabezonen im Zentralspeicher durch die Nummern R der numerischen Register identifiziert werden.

Die speziellen Formate dieser Befehle werden in Abschnitt 3.5 und 3.6 beschrieben.

### Format der Parameter

- . Der Operationstyp ist ein Binärschlüssel und belegt ein Byte.
- . Der Nebenschlüssel N ist eine zweistellige Dezimalzahl in gepackter Form auf einem Byte.
- . Der Nebenschlüssel K ist ein Binärwert oder eine Kombination des ISØ-Codes auf einem Byte.
- . Die Adressen belegen in der Form von zwei-, drei- oder vierstelligen gepackten Dezimalzahlen ein oder zwei Bytes.

Die Art der Adressierung ist dabei gleichgültig:

- absolute Adressierung: 4 Ziffern (2 Bytes) bezeichnen die Nummer eines Bytes.
- Adressierung der Register: 2 Ziffern (1 Byte) bezeichnen die Nummer eines numerischen Registers oder eines Basisregisters, das die Adresse einer Speicherzone enthält.
- Adressierung einer Sektion in einer Speicherzone:
  - . mit ausführlicher Basis: 4 Ziffern (2 Bytes), von denen die 1. die Nummer des Basisregisters und die drei anderen die Verschiebung darstellen.

mit einbezogener Basis: - 2 Ziffern (1 Byte) stellen eine Verschiebung  $\leq 99$  dar,

- 3 Ziffern mit vorangestelltem Buchstaben A (2 Bytes) stellen eine Verschiebung zwischen 100 und 999 dar.

- Der **Anschlußcode C** ist ein Binärkode und belegt 1 Byte.

Die Länge der Befehle ist variabel, aber immer gleich einer Zahl ganzer Bytes. Sie wird bestimmt durch die Analyse des Operationstyps und des Nebenschlüssels bei komplexen Befehlen.

### 3.1.2.2. Die Befehle der Grundsprache

#### Die Elementarbefehle

Die Elementarbefehle, aus denen ein Programm besteht, werden vom Programmierer im Format der Maschinenbefehle geschrieben, die ihnen entsprechen. Wie die Maschinenbefehle besitzt auch der Elementarbefehl einen Operationstyp, evtl. einen Nebenschlüssel und eine oder mehrere Adressen.

- Der **Operationstyp** wird symbolisch durch einen numerischen Schlüssel dargestellt, der sich aus Buchstaben und Ziffern für bestimmte Befehle zusammensetzt.

#### Beispiel:

	<u>OT symb.</u>	<u>Binärkode</u>
- Addition in einfacher Länge	ADD	1010 0001

Der Binärkode des OT in Maschinsprache erscheint bei einem Kernspeicherausdruck oder bei Auflisten eines Programms in Form von zwei hexadezimalen Zeichen nach den Normen, die in Abschnitt 2.1.2.1 beschrieben wurden. (z.B. 1010 0001 = A1)

- Der **Nebenschlüssel N**, dargestellt durch Dezimalziffern, kann eine Zahl von 00 bis 99 sein. Bei bestimmten Befehlen kann dieser Wert jedoch niedriger sein.
- Der **Nebenschlüssel K** und die **Anschlußcodes C** werden dargestellt durch zwei hexadezimale Zeichen (s. 2.1.2.1). Die Tafel des internen Codes in Kapitel 6 zeigt den Zusammenhang zwischen den Binär- und hexadezimalen Kombinationen.

#### Beispiele:

- 1) K = A4; K = 31; K = BC
- 2) Anschlußcode Drucker MB = CA

- Die **Adressen** werden durch Dezimalziffern nach den gleichen Bedingungen wie bei den Maschinenbefehlen dargestellt.

Beispiele:

- absolute Adresse: 4021 bezeichnet das Byte 4021 im Zentralspeicher

- Adresse der Register: 04 bezeichnet das 4. numerische Register.

- Basisadresse: 4021 bezeichnet als Basisregister:  
Register 4 und als Verschiebung 21.  
Enthält Reg. 4 z.B. 2320, so lautet die absolute Adresse: 2341 (2320 + 21).

- Adresse mit ein-  
bezogener Basis: 62 bezeichnet eine Verschiebung von 62 Bytes;  
A 105 eine von 105 Bytes.  
Die einbezogene Basis ist eins der Register 6-9.  
Enthält Reg. 8 z.B. 2114, so ist die absolute Adresse im 1. Fall: 2176 und im 2. Fall: 2219.

Der Anfang der Sektionen eines Programms wird durch ein symbolisches Bezugszeichen gekennzeichnet, das sich aus zwei hexadezimalen Zeichen (\*) zusammensetzt. Dieses Symbol bezeichnet bei Verwendung in einem Pseudobefehl die Adresse des ersten Befehls einer Sektion.

(\*) Anmerkung: Die Kombinationen F1 - FF sind verboten. Sollte der Programmierer eines dieser Zeichen unbedingt verwenden müssen, ist es erforderlich, innerhalb des Programms eine Serie zu programmieren, in der diese Zeichen durch logische Operationen zusammengesetzt werden. Dieser Vorgang ist nicht anwendbar bei den symbolischen Bezugszeichen. Es bleiben jedoch 240 Kombinationen zur freien Verfügung.

Die Pseudobefehle

Die Pseudobefehle werden in der gleichen Weise geschrieben wie die Elementarbefehle und enthalten einen symbolischen Operationstyp und evtl. symbolische Parameter.

3.1.3. Darstellung der Befehle

Die in den folgenden Abschnitten beschriebenen Befehle sind in folgende Gruppen eingeteilt:

- Befehle zur Programmverknüpfung (3.2)
- Registeroperationen (3.3)
- Operationen auf Zeichenbasis (3.4)
- Mehrfachüberträge (3.5)
- Ein/Ausgabeoperationen (3.6)

- Umschlüsselungsbefehle (3.7)
- Hilfsbefehle (3.8)
- Pseudobefehle (3.9)

Für jede Befehlsgruppe enthält ein Einführungsparagraf die gemeinsamen Charakteristika.

Die besonderen Eigenschaften jedes Befehls werden in folgender Weise dargestellt:

- symbolischer Operationstyp und englische Bezeichnung,
- Befehlsformat in Maschinensprache: Operationscode und Ergänzungsparameter,
- Länge des Befehls in Bytes,
- besondere Normen der Anwendung und der Parametrierung im gegebenen Fall,
- Wirkung, Anwendung und Besonderheiten,
- Beispiele.

Falls mehrere Befehle identische Eigenschaften haben, werden sie in einer gemeinsamen Beschreibung zusammengefaßt. Am Anfang des Abschnittes sind in einer Tabelle die verschiedenen symbolischen Operationstypen und die entsprechenden OT im Maschinencode angegeben.

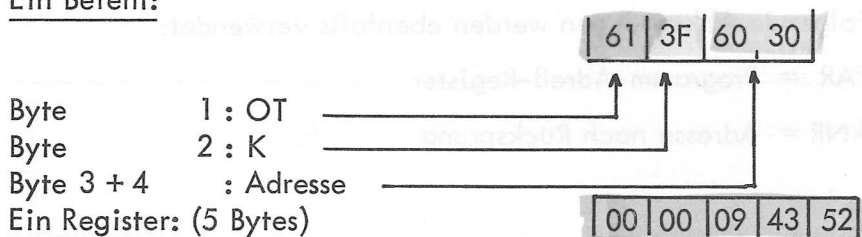
Innerhalb der Beispiele sind die Werte, die im Speicher enthalten sind, wie folgt dargestellt:

- der größte Teil in hexadezimaler Form (1 Zeichen je Byte),
- einige in Binärdarstellung (acht Ziffern (0/1) je Byte),
- einige in Klartext (2 Zeichen je Byte).

Eine Information oder ein Befehl besteht aus einer Folge von Zeichen, die immer in ein Schema gezeichnet sind, das die Speicherstellen (oder Bytes) darstellt, das diesen Wert enthält. Die interessanten Zonen (Register, Information) sind mit Markierungen versehen.

Beispiel:

Ein Befehl:



Am Ende des Handbuches geben Tabellen Auskunft über die Ablaufzeit eines Befehles und verweisen auf die entsprechenden Abschnitte , in denen der Befehl beschrieben wurde.

### Definitionen

Die verschiedenen Elemente eines Befehls werden durch folgende Zeichen dargestellt:

**OT** = Operationstyp

**N** = Nebenschlüssel dezimal, oder zwei gepackte Dezimalziffern.

**K** = Nebenschlüssel hexadezimal (Binärwert) oder zwei gepackte hexadezimale Zeichen.

**C** = Anschlußcode (hexadezimal) einer Randeinheit.

**R** = Nummer eines Registers (2 Dezimalziffern)

**A** = 1 alphanumerisches ungepacktes Zeichen

**xxxx** = absolute Adresse (dezimal)

**Bxxx** = Basisadresse (dezimal) oder:

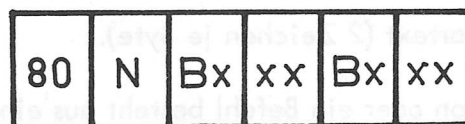
B = Nummer des Basisregisters, das die Basisadresse enthält

xxx = Verschiebung oder Ergänzung der Adresse. Dieser Wert wird zur Basis addiert, um die absolute Adresse zu erhalten.

**D<sub>n</sub>** = Verschiebung (dezimal) in bezug auf eine Basis n.

Die Befehle werden so eingerahmt, daß ihr Format und die Anzahl der Bytes sichtbar werden.

### Beispiel:



Binärcodes des Operationstypes

Folgende Abkürzungen werden ebenfalls verwendet:

PAR = Programm-Adreß-Register

ANR = Adresse nach Rücksprung



### 3.2. Die Befehle zur Programmverknüpfung

#### 3.2.1. Adressierung der Sprünge

Der Programmierer hat die Wahl zwischen zwei Formulierungen: Er kann die Sprünge mit reellen Adressen oder mit Symbolen programmieren.

##### a) reelle Adressen

Der Programmierer prüft die effektive Einspeicherung des Programmes im Zentralspeicher, um die Adresse jedes Sprunges zu bestimmen. Die anzugebende Adresse ist die Stelle, die der Operationstyp des Befehls belegt, mit dem nach dem Sprung fortgefahren werden soll.

##### b) Symbole

Der Programmierer springt aus dem Programm auf Sprungniveaus, die durch den Pseudobefehl LEVEL (Abschnitt 3.9.) dargestellt werden. Jedes Niveau erhält ein besonderes Etikett, das die Sprungadresse ersetzt.

Der Programmierer setzt dieses Symbol in den Sprungbefehl in folgender Weise ein:

F1	K
----	---

F1 = festes Zeichen, das besagt, daß es sich um ein Bezugszeichen handelt.

K = ein Zeichen in hexadezimaler Form, das dieses Bezugszeichen darstellt. Die Kombinationen F1 bis FF sind verboten.

Um von der Maschine verarbeitet werden zu können, werden die Symbole durch reelle Adressen des Sprungniveaus ersetzt. Dieser Austausch findet beim Laden des Programmes in den Kernspeicher durch das Standardladerprogramm statt.

#### Wahl der Methode

Die erste Methode setzt voraus, daß das Programm völlig fertiggestellt ist, damit die reellen Sprungadressen festgelegt werden können. Weiterhin ist die Veränderung eines Programmteiles mit einer Revision auch der Sprungbefehle, die nicht direkt berührt werden, verbunden. Das ist eine ständige Quelle des Ärgers und der Schwierigkeiten.

Die zweite Methode enthält nicht die vorgenannten Unbequemlichkeiten.

Die symbolische Adressierung kann in dem Augenblick festgelegt werden, in dem das Programm entsteht. Der Programmierer setzt die Sprungniveaus in das Flußdiagramm ein und ordnet die Bezugszeichen zu. Nach Maßgabe

des Programmes schiebt er die Pseudobefehle LEVEL ein und vermerkt die Symbole in den Sprungbefehlen.

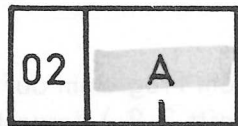
Spätere Veränderungen des Programmes bleiben ohne Einfluß auf die Sprungbefehle, wenn sie nicht gerade selbst benutzt werden. In diesen Fällen werden die neuen reellen Adressen automatisch während des Ladens des veränderten Programmes in den Kernspeicher festgelegt.

3.2.2. Sprungbefehle

3.2.2.1. Unbedingter Sprung

J R T

Jump; save adress for Return



A  
(Sprungadresse)

reelle Adresse  
symbolische Bezugs-  
zeichen

: |xx |xx|

: |F1 |K|

Der Inhalt des Programm-Adress-Registers wird im Register der Adresse bei Rücksprung (ANR) gespeichert (Adresse des dem JRT folgenden Befehls). Die im JRT festgelegte Sprungadresse wird in das PAR übertragen. Das Programm fährt mit dieser Adresse fort.

Beispiele:

Adresse	Programm	PAR	ANR
734	JRT <u>0 2   1 2   8 7</u>	vorher: 0734	0000
737	ADD <u>A 1   3 2   1 6</u>	nachher: 1287	0737

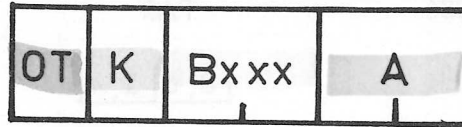
3.2.2.2. Bedingte Sprünge

JIERT

Jump if Immediate octet  
save adresse for ReTurn

Equal to store octet;  
Uniqual

JIURT



OT = 04 JIERT  
05 JIURT

K = Bezugszeichen

Bxxx = Adresse der zu untersuchenden Stellen

A = Sprungadresse { reelle Adresse |xx|xx|  
Etikett des Niveaus |F1|xx|

Der Programmsprung ist abhängig vom Vergleichsergebnis zwischen dem Zeichen K und dem Zeichen, das sich an der Adresse Bxxx befindet.

- bei JIERT wird der Sprung ausgeführt bei Gleichheit zwischen den beiden Zeichen.

- bei JIURT wird der Sprung durchgeführt, wenn Ungleichheit zwischen den Zeichen besteht.

a) Bei Sprungausführung

Der Inhalt des PAR wird im ANR gespeichert (Adresse des Befehles, der auf den Befehl JIERT oder JIURT folgt)

Die Sprungadresse des JIERT oder JIURT wird in das PAR übertragen und das Programm fährt an der Stelle fort, die dieser Adresse entspricht.

b) Bei Nichtausführung des Sprunges

Der Inhalt des PAR bleibt unverändert und das Programm fährt mit dem Befehl, der dem JIERT oder JIURT folgt, fort.

Beispiele:

a) JIERT (Sprung wird ausgeführt)

Adresse 654 

0	4	3	E	0	0	8	6	1	5	0	1
---	---	---	---	---	---	---	---	---	---	---	---

 Basis 0 = 0000

	PAR	ANR								
Stelle 0086 (VG)										
vorher : 3 E	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>0</td><td>6</td><td>5</td><td>4</td></tr></table>	0	6	5	4	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	x	x	x	x
0	6	5	4							
x	x	x	x							
nachher : 3 E	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>1</td><td>5</td><td>0</td><td>1</td></tr></table>	1	5	0	1	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>0</td><td>6</td><td>6</td><td>0</td></tr></table>	0	6	6	0
1	5	0	1							
0	6	6	0							

b) JIURT (Sprung wird nicht ausgeführt)

Adresse 654 

0	5	0	0	0	0	9	1	1	2	1	8
---	---	---	---	---	---	---	---	---	---	---	---

 Basis 0 = 0000

	PAR	ANR								
Stelle 0091 (Kap)										
vorher : 0 0	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>0</td><td>6</td><td>5</td><td>4</td></tr></table>	0	6	5	4	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	x	x	x	x
0	6	5	4							
x	x	x	x							
nachher : 0 0	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>0</td><td>6</td><td>6</td><td>0</td></tr></table>	0	6	6	0	<table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	x	x	x	x
0	6	6	0							
x	x	x	x							

### 3.3. Registeroperationen

#### 3.3.1. Allgemeines

Die Operationen, die in den Registern durchgeführt werden, beziehen sich im wesentlichen auf die Verarbeitung von numerischen Daten, die in gepackter, algebraischer Form dargestellt werden. (s. 2.1.2.2.) Die Verarbeitung erfolgt in einfacher oder doppelter fester Länge (s. 2.1.3.3.).

Die Befehle dieser Gruppe enthalten:

- Rechen- und algebraische Vergleichsbefehle,
- Versetzungs- und Rundungsbefehle,
- Übertragungsbefehle.

Ihre Beschreibung und die angeführten Beispiele beziehen sich nur auf Dezimalzahlen.

Die Benutzung dieser Befehle setzt die folgenden allgemeinen Richtlinien voraus.

#### Löschen der Register

Es besteht kein besonderer Löschbefehl.

Ein Löschen der Register besteht darin, jedes Halbbyte auf Null zu setzen.

Man erreicht dies, indem man das Register mit dem Zeichen für "Nichts" (00) belegt, entweder durch den Befehl "Einsetzen eines Zeichens" -INC- (s. 3.4.3.2.) oder durch den Übertrag des Inhaltes eines anderen Registers, das nichts enthält.

Da die Zeichen "0" (30) und "Leerzeichen" (20) ein Halbbyte ungleich Null enthalten, können diese Zeichen nicht benutzt werden.

#### Versetzung

Die Rechenbefehle und der Vergleich erfordern Operanden, die in den numerischen Registern rechtsbündig stehen. Wird ein Operand nach links verschoben, so daß die rechten Stellen mit Null aufgefüllt werden, wird der Wert so behandelt, als ob er mit  $10^n$  multipliziert worden wäre.  $n$  entspricht der Anzahl der Nullen rechts.

Die Rechenergebnisse stehen immer rechtsbündig in den Registern.

### Kapazitätsüberschreitung

Je nachdem es sich um ein Einfach- oder Doppelregister handelt, kann ein numerisches Register 9 oder 19 Ziffern und ein Vorzeichen (0 oder 9) enthalten.

Es ist Aufgabe des Programmierers darauf zu achten, daß die Rechenoperanden die besonderen Normen jedes Befehls beachten, so daß sie niemals die erlaubte Maximalkapazität überschreiten. Die Kapazität, die man für einen Operanden in Betracht ziehen muß, ist gleich der Maximalzahl von Ziffern, die er enthält, zuzüglich eventueller Nullen rechts im Register.

<u>Beispiel:</u>	<u>Register</u>	<u>zu beachtende Kapazität</u>										
Operand rechtsbündig:	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	0	0	0	0	0	x	x	x	x	x	5 Ziffern
0	0	0	0	0	x	x	x	x	x			
Operand versetzt:	<table border="1"><tr><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	x	x	x	x	x	0	0	0	8 Ziffern
0	0	x	x	x	x	x	0	0	0			

Am Ende der Rechenoperationen prüft die Maschine das erhaltene Ergebnis. Diese Kontrolle besteht darin, den Wert der Vorzeichenstelle des Resultats zu analysieren. Ein Wert ungleich 0 oder 9 zeigt an, daß es sich um eine Kapazitätsüberschreitung handelt und daß dadurch das Resultat unauswertbar ist (falsches Vorzeichen). In diesem Fall wird ein Testzeichen in das Register "Kapazitätsüberschreitung" gesetzt.

Bei der Division wird dieses Testzeichen dargestellt durch AA. Bei den anderen Operationen ist es ein unbestimmbarer Wert; nämlich der Inhalt des äußersten linken Bytes des Resultats. Dieser Inhalt ist immer ungleich 00 und variabel.

Um sich von der Art des Resultats zu überzeugen, geht der Programmierer nach folgender Methode vor:

- Vor der Rechenoperation setzt er das Register "Kap" auf Null (00).
- nach der Operation analysiert er diese Stelle durch einen Vergleich gegen 00, um festzustellen, ob sich der Inhalt geändert hat.

Diese Methode ist unabhängig vom Wert des Testzeichens gültig.

### Nicht dezimale Werte

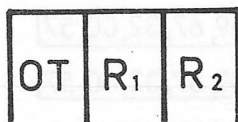
Das Vorhandensein von nicht dezimalen Werten (größer 9) in den Registern wird von den Befehlen nicht festgestellt. Sie laufen ohne einen eventuellen Fehler anzuzeigen ab. Bei den Rechen- und Rundungsbefehlen erhält man falsche Resultate. Bei Versetzungsbefehlen dagegen kann es geduldet werden.



### 3.3.2. Rechenbefehle und algebraischer Vergleich

#### 3.3.2.1. Additionen und Subtraktionen

ADD	ADd Decimal single length	A 1
ADDD	ADd Decimal Double length	A 2
SBD	SuBtract Decimal single length	B 1
SBDD	SuBtract Decimal Double length	B 2



OT : siehe Maschinencode in obenstehender Tabelle

R1 und R2 : Einfach- oder Doppelregister je nach OT

Der Inhalt des Registers 1 wird auf den Inhalt des Registers 2 addiert oder von ihm subtrahiert. Das Resultat steht in Register 2. Wenn Register 1 ein anderes als Register 2 ist, wird sein Inhalt durch die Operation nicht verändert.

Der Befehl wird algebraisch durchgeführt.

Wenn R1 und R2 das gleiche Register bezeichnen, wird der Inhalt dieses Registers verdoppelt oder gelöscht, je nachdem, ob eine Addition oder Subtraktion vorlag.

Die Operation bezieht sich auf Register von

- 5 Bytes (9 Ziffern und Vorzeichen) bei ADD und SBD (einfache Länge)
- 10 Bytes (19 Ziffern und Vorzeichen) bei ADDD und SBDD (dopp. Länge)

Überschreitet das Ergebnis 9 oder 19 Ziffern, liegt Kapazitätsüberschreitung vor. Das Resultat bleibt erhalten, das höchste Byte wird aber in das Register Kap (0091) übertragen.

Beispiele:

A) Addition, einfache Länge: ADD    | A1 | 27 | 35 |

	Reg. Kap.	Reg. 27	Reg. 35
a) vorher:	00	00,76,93,21,69	99,99,83,29,67
nachher:	00	00,76,93,21,69	00,76,76,51,36
	(+ 76.932.169) + (- 167.033) = (+ 76.765.136)		

Es fand keine Kapazitätsüberschreitung statt.

b) vorher:	00	09,67,32,00,57	03,99,50,63,39
nachher:	13	09,67,32,00,57	13,66,82,63,96
	(+ 967.320.057+) + (+ 399.506.339) = (+ 1.366.826.396)		

Es fand eine positive Kapazitätsüberschreitung statt.

B) Substraktion, einfache Länge: SBD    | B1 | 50 | 97 |

	Reg. Kap.	Reg. 50	Reg. 97
a) vorher:	00	00,76,93,21,69	00,00,16,70,33
nachher:	00	00,76,93,21,69	99,23,23,48,64
	(+ 167.033) - (+ 76.932.169) = (- 76.765.136)		

Es fand keine Kapazitätsüberschreitung statt.

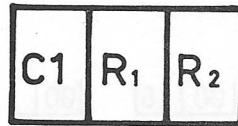
b) vorher:	00	07,36,29,92,02	95,21,07,63,22
nachher:	87	07,36,29,92,02	87,84,77,71,20
	(- 478.923.678) - (+ 736.299.202) = (- 1.215.222.880)		

Es fand eine negative Kapazitätsüberschreitung statt.

### 3.3.2.2. Multiplikation

MPD

Multi Ply Decimal



R1 = Einfachregister Multiplikator

R2 = Doppelregister Multiplikand

Der Inhalt des Doppelregisters R2 wird multipliziert mit dem Inhalt des Einfachregisters R1. Das Produkt befindet sich am Ende der Operation im Register 2. Ist das Register 1 ein anderes als Doppelregister 2, so wird sein Inhalt durch die Operation nicht verändert.

Die Operation erfolgt algebraisch.

- Größe der Faktoren:
- Multiplikator : 1 - 9 Ziffern + VZ
  - Multiplikand : 1 - 19 Ziffern + VZ
  - Produkt : 1 - 19 Ziffern + VZ

Die Maximalkapazität der Operanden errechnet sich nach der Formel:

$$M + m \leq 19$$

M = Anzahl Stellen Multiplikand

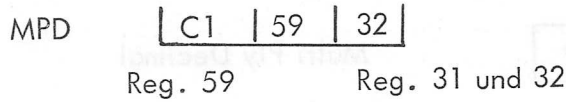
m = Anzahl Stellen Multiplikator

Ansonsten gelten die Regeln des Abschnittes 3.3.1.

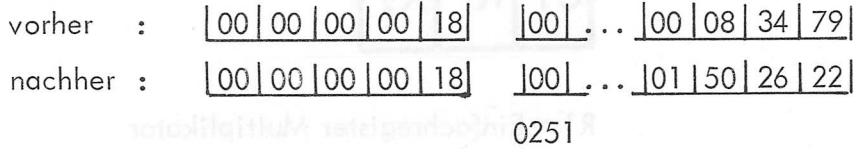
Überschreitet das Produkt 19 Ziffern, liegt Kapazitätsüberschreitung vor. Man kann es jedoch nur erkennen, wenn die 20. Ziffer ungleich 0 oder 9 ist. Das Resultat bleibt erhalten, das äußerste linke Byte des Registers 2 wird jedoch in das Register "Kap" (0091) übertragen. Überschreitet das Produkt 20 Stellen, gehen die überzähligen Stellen verloren.

Ist R1 = R2 wird die Zahl, die sich in R1 befindet, quadriert. Diese Zahl kann 9 Stellen nicht übersteigen. R1 kann auch gleich R2 - 1 sein.

Beispiele:

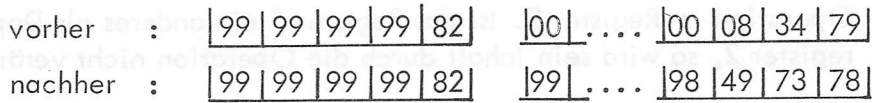


a) +x+



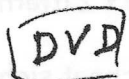
$(+ 83.479) \times (+ 18) = (+ 1.502.622)$

b) +x-



$(+ 83.479) \times (- 18) = (- 1.502.622)$

3.3.2.3. Division



Für die Division existiert ein Unterprogramm, das vom Benutzer frei in den Kernspeicher eingesetzt werden kann. Es muß in den Programmkartensatz zwischen KA17 und KA31/ 32 eingefügt werden.

Die UP-Karten sind aufsteigend von 0001 bis 0014 numeriert, die Reihenfolge darf nicht geändert werden.

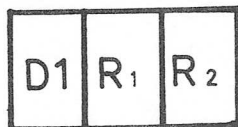
Das UP benötigt 249 Kernspeicherstellen. Die Adresse der UP-Division wird vom Ladeprogramm auf 2 Bytes gespeichert; und zwar bei einer Kernspeichergröße von

- 2500 Bytes auf die Stellen 2300 und 2301,
- 5000 " " " " 4800 und 4801,
- 10000 " " " " 9800 und 9801.

Der Dividend muß im Doppelregister (13) 14, der Divisor im Register 15 gespeichert sein. Das Register 00 muß stets Null sein.

Der Anruf dieses Unterprogramms erfolgt automatisch durch den Divisionsbefehl:

DVD DiVide Decimal



R1 = Einfachregister (Divisor) = R 15

R2 = Doppelregister (Dividend) = RR 14

Der Inhalt des Doppelregisters R2 (RR14) wird durch den Inhalt des Einfachregisters R1 (R15) dividiert.

Ergebnis:

- Der Quotient steht in den 5 linken Bytes  
(Einfachregister mit der Nummer  $R2 - 1 = R13$ )
- Der Rest steht in den 5 rechten Bytes  
(Einfachregister mit der Nummer  $R2 \approx R14$ )

Nach der Operation ist der Dividend (RR14) verändert; der Divisor (R15) bleibt erhalten.

Wenn das Register R1 ein anderes als das Doppelregister R2 ist, wird sein Inhalt durch die Operation nicht verändert.

Die Division erfolgt nicht algebraisch. Dividend und Divisor müssen positiv sein.

- Größe der Faktoren:
- Dividend : 1 - 17 Ziffern + VZ
  - Divisor : 1 - 8 Ziffern + VZ
  - Quotient : 1 - 9 Ziffern + VZ

Die Maximalgröße der Operanden errechnet sich nach folgender Formel:

$$\underline{D - d + 1 \leq 9}$$

- D = Anzahl Stellen Dividend  
- d = Anzahl Stellen Divisor

Ansonsten gelten die Regeln des Abschnittes 3.3.1.

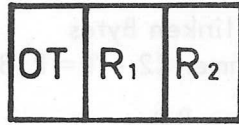
Überschreitet der Quotient 9 Stellen oder ist der Divisor Null, empfängt das Register "Kap" (0091) das Zeichen AA.

Beispiel:

	DVD	D1	15	14											
	Reg. 15		Reg. 13	Reg. 14											
vorher :	00	00	00	05	24	00	00	00	00	00	00	00	20	19	87
nachher:	00	00	00	05	24	00	00	00	03	85	00	00	00	02	47
						Quotient					Rest				
	(+ 201.987) : (+ 524) = (+ 385)														

3.3.2.4. Algebraischer Vergleich

CMD	CoMpare Decimal single length	91
CMDD	CoMpare Decimal Double length	92



OT = 91 oder 92

R1 + R2 = Einfach- oder Doppelregister, nach OT.

Der Inhalt des Registers 1 wird mit dem Inhalt des Registers 2 verglichen.  
Der Inhalt beider Register wird durch die Operation nicht verändert.

Der Vergleich erfolgt algebraisch. Das Ergebnis wird in dem Vergleichsregister (0086) in Form eines Zeichens gespeichert:

- 3 C bei  $R_1 < R_2$
- 3 D bei  $R_1 = R_2$
- 3 E bei  $R_1 > R_2$

Anmerkung:

Der Inhalt des Vergleichsregisters bleibt bis zum nächsten Vergleich, der den Inhalt verändert, erhalten.

Vor einer Veränderung muß der Inhalt ausgewertet worden sein.

Größe der Operanden:

- CMD : 9 Ziffern + VZ (Register mit 5 Bytes)
- CMDD : 19 Ziffern + VZ (Register mit 10 Bytes)

Beispiele:

CMD	91   67   59	R1 = Reg. 67	R2 = Reg. 59	VG (0086)
a) vorher :	...  00 46 38	...	...  00 46 38	xx
nachher :	...  00 46 38	...	...  00 46 38	3D
				R1 = R2
b) vorher :	...  00 03 78	...	...  99 86 54	3D
nachher :	...  00 03 78	...	...  99 86 54	3E
	(+ 378)		(- 1.346)	R1 > R2
c) vorher :	...  99 76 34	...	...  99 94 12	3E
nachher :	...  99 76 34	...	...  99 94 12	3C
	(- 2.366)		(- 588)	R1 < R2

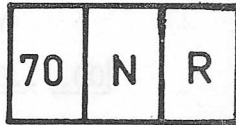


3.3.3. Versetzungs- und Rundungsbefehle

3.3.3.1. Versetzung nach links

SLD

Shift Left Decimal



N = kann variieren von 00 bis 19

R = Doppelregister

Der Inhalt des Doppelregisters R wird um N Dezimalstellen (Halbbytes) nach links verschoben. Die N ersten Stellen rechts im Register werden nach der Versetzung auf Null gesetzt. Das Vorzeichen des Operanden, der sich im Register R befindet, wird unter der Voraussetzung nicht verändert, daß der Operand beim Versetzen nicht die dafür vorgesehene Stelle erreicht.

Kapazitätsüberschreitung wird nicht angezeigt. Ist die Zahl N der Versetzungen größer als die Zahl N' der zur Verfügung stehenden Stellen links des Operanden, gehen N' - N Ziffern einschließlich des Vorzeichens verloren.

Bei N = 00 findet keine Versetzung statt, der Befehl wird nicht ausgeführt.

Bei N > 19 ist die Zahl der durchgeführten Versetzungen gleich dem Rest der Division von N durch 20.

Beispiel:

N = 77; Zahl der Versetzungen 17.

Das Register kann auch nicht numerische Zeichen enthalten.

Sie werden in der gleichen Weise, Halbbyte für Halbbyte, versetzt.

Anmerkung:

Dieser Befehl ermöglicht die Multiplikation einer Zahl mit  $10^n$ .

Beispiele:

SLD | 70 | 03 | 37 |

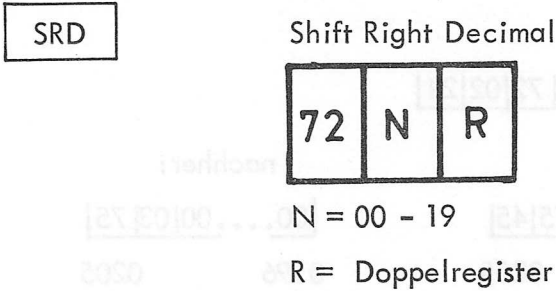
vorher:		nachher:	
00 ... 00 00 23 75		00 ... 02 37 50 00	
0276	0285	0276	0285

2.375 → 2.375.000

99 ... 99 99 96 25		99 ... 99 62 50 00	
0276	0285	0276	0285

- 375 → - 375.000

3.3.3.2. Versetzung nach rechts



Der Inhalt des Doppelregisters R wird um N Dezimalstellen (Halbbytes) nach rechts verschoben.

Die Ziffern, die sich an den N ersten Stellen rechts befinden, verschwinden und gehen verloren.

Die Ziffer, die sich an der ersten Stelle von links befindet, bleibt erhalten und wird auf den N folgenden Stellen eingesetzt. Dadurch bleibt die algebraische Form des Registerinhaltes erhalten.

Bei N = 00 findet keine Versetzung statt; der Befehl wird nicht ausgeführt.

Bei N > 19 ist die Zahl der durchgeführten Versetzungen gleich dem Rest der Division von N durch 20.

Beispiel: N = 35; Zahl der Versetzungen 15

Das Register kann auch nicht numerische Zeichen enthalten. Sie werden in der gleichen Weise, Halbbyte für Halbbyte, versetzt.

Anmerkung:

Dieser Befehl ermöglicht eine Division durch 10 unter Vernachlässigung des Restes. Ebenso kann eine Abrundung erfolgen.

Beispiele:

a) SRD

72|02|21

vorher:

nachher:

00...03|75|45

00...00|03|75

0196

0205

0196

0205

375.45 → 375

b) SRD

72|14|21

99|24|50.....00|00|00

99|99|99.....99|24|50

0196

0205

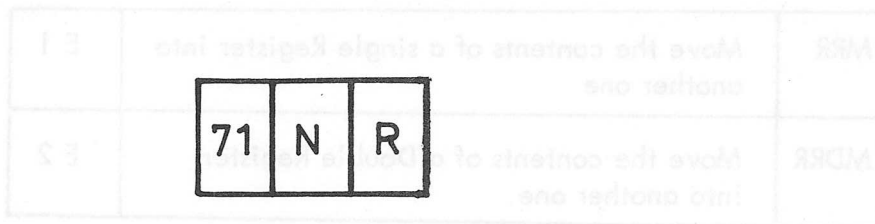
0196

0205

### 3.3.3.3. Rundungen

**ROUND**

Round



N = 00 - 99

R = Doppelregister

Die Zahl N wird ohne Berücksichtigung des Vorzeichens auf den Inhalt des Doppelregisters R addiert. Anschließend wird das Resultat, das sich in R befindet, um zwei Dezimalstellen (1 Byte) nach rechts versetzt, wobei die zwei rechten Stellen verloren gehen.

Die Operation erfolgt ohne Berücksichtigung des Vorzeichens.

#### Anmerkung:

Bei N = 00 erhält man eine Division durch 100 unter Vernachlässigung der Dezimalstellen.

Enthält das Register keine numerischen Werte, so wird das nicht angezeigt. Das Resultat ist jedoch falsch.

Das Resultat der Addition darf 19 Stellen nicht überschreiten, damit das Vorzeichen nicht zerstört wird. Kapazitätsüberschreitung wird jedoch nicht angezeigt.

#### Beispiele:

Round **71|50|85**

vorher:

nachher:

- |                |                       |
|----------------|-----------------------|
| a) ...00 07 25 | 00 00 07 = + 7.25 → 7 |
| b) ...00 07 85 | 00 00 08 = + 7.85 → 8 |
| c) ...99 92 75 | 99 99 93 = - 7.25 → 7 |
| d) ...99 92 15 | 99 99 92 = - 7.85 → 8 |

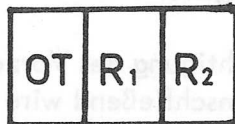
0525

0525

### 3.3.4. Übertragungsbefehle

#### 3.3.4.1. Überträge in einfacher und doppelter Länge

MRR	Move the contents of a single Register into another one	E 1
MDRR	Move the contents of a Double Register into another one	E 2



OT = siehe oben

R<sub>1</sub> = Abgebendes Register (einfach oder doppelt je nach OT)

R<sub>2</sub> = Empfangendes Register (einfach oder doppelt je nach OT)

Der Inhalt des Registers R<sub>1</sub> wird in das Register R<sub>2</sub> übertragen, nachdem dessen Inhalt gelöscht wurde. Der übertragene Wert erfährt keine Veränderung. Handelt es sich um verschiedene Register, wird der Inhalt von R<sub>1</sub> nicht verändert.

#### Länge:

- MRR : 5 Bytes oder 10 Dezimalziffern (R<sub>1</sub> + R<sub>2</sub> einfach)
- MDRR : 10 Bytes oder 20 Dezimalziffern (R<sub>1</sub> + R<sub>2</sub> doppelt)

Bei MDRR:

- Wenn  $R_1 = R_2 - 1$  erfolgt der Übertrag normal. Das kommt auf das gleiche hinaus, als wenn man den Inhalt des Registers 1 um 10 Dezimalstellen nach rechts verschoben hätte.
- Wenn  $R_1 = R_2 + 1$  erhält man dreimal den Inhalt der 10 Dezimalstellen von rechts des Registers 1 in den Einfachregistern: R<sub>1</sub>, R<sub>2</sub> und R<sub>2</sub> - 1.



Beispiele:

a) MRR  $\boxed{E1 \mid 19 \mid 28}$

	R 19	R 28
vorher :	$\boxed{0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9}$	$\boxed{0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0}$
nachher :	$\boxed{0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9}$	$\boxed{0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9}$

b) MDRR  $\boxed{E2 \mid 17 \mid 18}$   $(R1 = R2 - 1)$

Dopp. Reg. 1

Dopp. Reg. 2

vorher :	$\boxed{0 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 5 \mid 5 \mid 5 \mid 5 \mid 5 \mid 5 \mid 5 \mid 5 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0}$
nachher :	$\boxed{0 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 0 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 4 \mid 5 \mid 5 \mid 5 \mid 5 \mid 5 \mid 5 \mid 5 \mid 5}$
Einfachregister:	16                      17                      18

c) MDRR  $\boxed{E2 \mid 18 \mid 17}$   $(R1 = R2 + 1)$

Dopp. Reg. 1

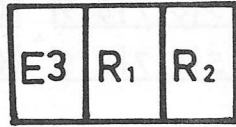
Dopp. Reg. 2

vorher :	$\boxed{0 \mid 3 \mid 3 \mid 3 \mid 3 \mid 3 \mid 3 \mid 3 \mid 3 \mid 0 \mid 1 \mid 1 \mid 1 \mid 1 \mid 1 \mid 1 \mid 1 \mid 1 \mid 0 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2}$
nachher :	$\boxed{0 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 0 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 0 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2 \mid 2}$
Einfachregister:	16                      17                      18

3.3.4.2. Überträge von einfachem noch doppeltem Register

**MSRDR**

Move the contents of a Single Register into a Double Register



R1 = Abgebendes Einfachregister

R2 = Empfangendes Doppelregister

Mit diesem Befehl kann man eine algebraische Zahl, die in einfacher Länge (9 Ziffern + VZ) dargestellt ist, in doppelter Länge (19 Ziffern + VZ) ausdrücken.

Der Inhalt des Einfachregisters R1 wird in das Doppelregister R2 übertragen, nachdem dessen Inhalt vorher gelöscht wurde. Die Ziffer (0 oder 9), die in R1 das Vorzeichen darstellt, wird 10 mal in das Register 2 nach links übertragen, um die algebraische Form der Zahl bei doppelter Länge zu erhalten.

Anmerkung:

Enthält das Register 1 eine Kapazitätsüberschreitung (Zeichen ungleich 0 oder 9), ist das Ergebnis der Operation fehlerhaft. Die ursprüngliche Zahl ist verändert worden.

Die Operation ist möglich mit  $R1 = R2$  oder mit  $R1 = R2 - 1$ . Diese zwei Methoden ermöglichen die "Erweiterung" einer Zahl bei Wiederverwertung des Registers, das sie enthält.

Der Inhalt des Registers R1 wird durch die Operation nicht verändert, außer in dem Fall, wo  $R1 = R2 - 1$  ist.

Beispiele:

a) MSRDR 

E3		17		83
----	--	----	--	----

		R 17		R 83																		
vorher :	0	1	2	3	4	5	6	7	8	9	9	9	9	9	9	9	9	9	9	9	9	9

nachher:	0	1	2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9
----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Zahl positiv

b) MSRDR | E3 | 17 | 18 |



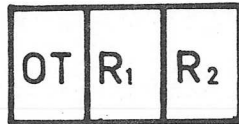
vorher : | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |

nachher : | 9 9 | 9 9 | 9 9 | 9 9 | 9 9 | 9 9 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |

Zahl negativ

3.3.4.3. Registeraustausch

EXR	EXchange single Registers	E 5
EXDR	EXchange Double Registers	E 6



OT = siehe oben  
 R<sub>1</sub>, R<sub>2</sub> = Einfach- oder Doppelregister je nach OT

Dieser Befehl vollzieht den Austausch des Inhaltes der Register R<sub>1</sub> und R<sub>2</sub>. Am Ende der Operation befindet sich im Register 1 der vorige Inhalt von R<sub>2</sub> und in R<sub>2</sub> der vorige Inhalt von R<sub>1</sub>.

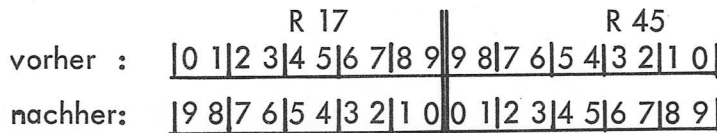
Größen:

- bei EXR : 5 Bytes oder 10 Dezimalstellen
- bei EXDR : 10 Bytes oder 20 Dezimalstellen

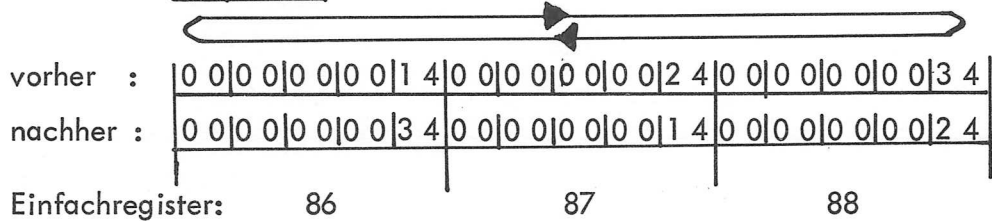
Bei EXDR ist R<sub>1</sub> = R<sub>2</sub> - 1 möglich. In diesem Fall erhält man einen rechtsläufigen Ringtausch des Inhaltes der drei Einfachregister.

Beispiele:

a) EXR | E5 | 17 | 45 |



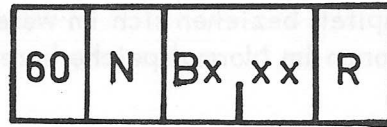
b) EXDR | E6 | 87 | 88 |



3.3.4.4. Laden eines Registers

LRN

Load Register N



N = 01 - 10

Bxxx = Adresse des zu übertragenden Wertes

R = empfangendes Einfach- oder Doppelregister

Mit diesem Befehl kann man in ein Register R eine Zahl von N Bytes (2N Dezimalziffern) übertragen, die in algebraischer Form gepackt im Normal-speicherbereich an der Adresse Bxxx steht.

Der Übertrag erfolgt in ein Einfachregister, wenn  $N \leq 5$  ist, und in ein Doppelregister, wenn  $N > 5$  ist. Der vorherige Inhalt des Registers wird gelöscht.

Die Zahl wird rechtsbündig in das Register eingesetzt. Wenn nicht alle Stellen des Registers belegt werden ( $N < 5$  oder 10), wird die äußerste linke Ziffer bis zum linken Ende des Registers wiederholt. Diese Ziffer bestimmt das Vorzeichen: Es muß unbedingt 0 oder 9 sein. Anderenfalls ist die Zahl im Register ungleich der ursprünglichen Zahl .

Anmerkung:

Es werden nur die Ziffern der Einer bei N berücksichtigt; die Ziffer 0 wird wie 10 behandelt.

Beispiel:

LRN	<u>60</u>   <u>03</u>   <u>71,52</u>   <u>14</u>	Basis 7 = 2000
	Normalspeicher	Reg. 14
vorher :	<u>2 0,9 5,4 2,1 3,2 0</u>	<u>0 0   0 0   0 0   0 0   0 0</u>
nachher :	<u>2 0,9 5,4 2,1 3,2 0</u>	<u>9 9   9 9   9 5   4 2   1 3</u>

### 3.4. Operationen auf Zeichenbasis

#### 3.4.1. Allgemeines

Die Befehle dieses Kapitels beziehen sich im wesentlichen auf die Bearbeitung von Informationen im Normalspeicherbereich des Zentralspeichers und zwar sind es:

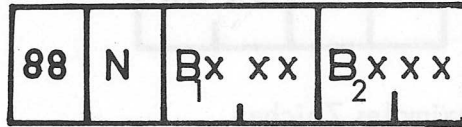
- logische Befehle
- Übertragungsbefehle.

Sie arbeiten in variabler Länge auf 8 Bit-Zeichen. Die Länge der bearbeiteten Worte wird durch die Zahl des Bytes spezifiziert, außer, ein Befehl bearbeitet nur ein Zeichen. Das trifft bei einigen logischen Befehlen zu. Die benutzte Adressierung ist die Basisadressierung.

3.4.2. Logische Befehle

3.4.2.1. Alphanumerischer Vergleich

**CMC** CoMpare Characters



N = Die Zahl der zu vergleichenden Bytes von 00 - 99  
 Bxxx = Adresse der rechten Stelle der zu vergleichenden Worte

Mit diesem Befehl können zwei Worte gleicher Länge N verglichen werden: Das Wort, das sich an B<sub>1</sub>xxx befindet, wird verglichen mit dem Wort an B<sub>2</sub>xxx. Die beiden Operanden werden durch die Operation nicht verändert. Der Vergleich verarbeitet Zeichen, die sich aus 8 Bits zusammensetzen. Die Worte können daher beliebige Zeichen des internen Codes enthalten.

Das Vergleichsergebnis beruht auf der Binärdarstellung des internen Codes und wird im Vergleichregister in Form eines hexadezimalen Zeichens gespeichert.

- 3 C bei  $B_1xxx < B_2xxx$
- 3 D bei  $B_1xxx = B_2xxx$
- 3 E bei  $B_1xxx > B_2xxx$

Bei N = 0 werden beide Worte wie "Nichts" behandelt. Das Register enthält das Ergebnis "Gleich" (3 D).

Beispiel:

CMC	88   06   6006   7006	Basis 6 = 1500	Basis 7 = 2000	
	Zone 6	Zone 7		VG
vorher :	3 0   4 5   3 1   3 8   5 4   4 F   3 4	3 0   4 C   3 0   3 3   4 2   5 6   3 9		00
nachher :	3 0   4 5   3 1   3 8   5 4   4 F   3 4	3 0   4 C   3 0   3 3   4 2   5 6   3 9		3C
	1500	1505	2000	2005
	$B_1xxx < B_2xxx$			

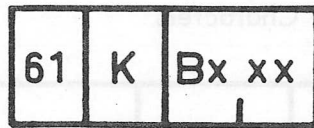


3.4.2.2. Logisches UND

NI

aNd

Immediat



K = hexadezimales Zeichen

Bxxx = Adresse des zu bearbeitenden Zeichens

Dieser Befehl setzt nach den Regeln des logischen UND das Zeichen K auf das Zeichen, das sich an Adresse Bxxx befindet.

Beispiel:

NI | 61 | 77 | 6004 | Basis = 1800

K (Binärdarstellung) | 0 1 1 1, 0 1 1 1 | (77)

Stelle 1804 vorher: | 1 1 1 1, 1 0 1 1 | (FB)

nachher: | 0 1 1 1, 0 0 1 1 | (73)

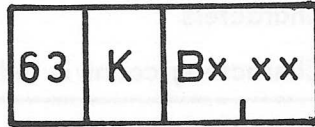
*Wenn = 1  
≠ 0*

*all 0*

3.4.2.3. Logisches ODER

$\emptyset I$

Or      Immediat



K = hexadezimaler Zeichen  
 Bxxx = Adresse des zu bearbeitenden Zeichens

Dieser Befehl setzt nach den Regeln des logischen ODER das Zeichen K auf das Zeichen, das sich an Adresse Bxxx befindet.

Das Ergebnis steht an dieser Adresse zur Verfügung. Das Zeichen, das ursprünglich dort stand, ist verändert worden.

Beispiel:

$\emptyset I$       | 63 | 34 | 6081 |      Basis 6 = 2300

K (Binärdarstellung)      | 0 0 1 1, 0 1 0 0 |      (34)

Stelle 2381 vorher:      | 1 1 1 0, 0 0 0 0 |      (D0)

nachher:      | 1 1 1 1, 0 1 0 0 |      (F4)

*alle 1*

*Wenn  $\neq \rightarrow 1$   
 $= = 0$  oder  $1$*

3.4.3. Übertragungsbefehle im Normalspeicherbereich

3.4.3.1. Einfachüberträge

MVC	MoVe Characters
MVIC	MoVe Characters; count is Indirect

80	N	Bx xx <sup>1</sup>	Bx xx <sup>2</sup>
8A	R	Bx xx	Bx xx

N = Länge des Übertrages in Anzahl Bytes

R = Nummer des Registers, das die Länge N enthält.

Die Länge N kann 00 - 99 sein.

B<sub>1</sub>xxx = rechte Adresse der abgebenden Zone

B<sub>2</sub>xxx = rechte Adresse der empfangenden Zone

Die Adresse B<sub>2</sub>xxx muß  $\geq$  als Adresse B<sub>1</sub>xxx oder  $\leq$  Adresse B<sub>2</sub>xxx - N sein.

Das Wort mit der Länge von N Bytes, das an Adresse B<sub>1</sub>xxx steht, wird nach Adresse B<sub>2</sub>xxx übertragen.

Der alte Inhalt der empfangenden Zone wird gelöscht. Der Inhalt der abgebenden Zone bleibt unverändert, wenn beide Felder unterschiedliche Adressen haben.

Die Länge des Wortes wird durch den Wert von N bestimmt:

- bei MVC innerhalb des Befehls
- bei MVIC durch den Inhalt des rechten Bytes des Registers R.

Der Übertrag erfolgt Stelle für Stelle von rechts nach links.

Anmerkung:

Befindet sich die Adresse B<sub>2</sub>xxx innerhalb der Adressen B<sub>1</sub>xxx - N und B<sub>1</sub>xxx, werden nur die Stellen zwischen den Adressen B<sub>2</sub>xxx und B<sub>1</sub>xxx richtig übertragen; sie wiederholen sich aber innerhalb des Empfangsfeldes (vgl. Beispiel -b)).

Beispiele:

a) Normaler Übertrag: MVC 

80	08	1008	1010
----	----	------	------

Basis 1 = 2000

vorher : 

3	P	A	R	I	S	2	4	5	3
---	---	---	---	---	---	---	---	---	---

nachher: 

3	P	P	A	R	I	S	5	3
---	---	---	---	---	---	---	---	---

2000                      2005                      2010

b) Unkorrekter Übertrag: MVIC 

8A	06	1008	1006
----	----	------	------

Basis 1 = 2000; Register 06 enthält: 6

vorher : 

8	4	3	P	A	R	I	S	2	4	5	3
---	---	---	---	---	---	---	---	---	---	---	---

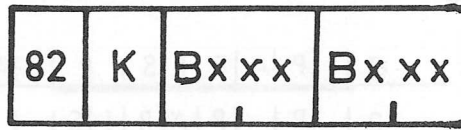
nachher: 

8	I	S	I	S	I	S	I	S	2	4	5	3
---	---	---	---	---	---	---	---	---	---	---	---	---

2000                      2005                      2010

3.4.3.2. Einsetzen eines Zeichens

INC      IN sert Character



- K = einzusetzendes Zeichen
- B<sub>1</sub>xxx = Adresse der 1. Stelle, die belegt werden soll (rechte Adresse)
- B<sub>2</sub>xxx = Adresse der letzten Stelle - 1, die belegt werden soll (linke Adresse)

Die Adresse B<sub>1</sub>xxx muß > B<sub>2</sub>xxx sein.

Das Zeichen K wird auf jede Stelle einer Speicherzone eingesetzt, die durch die beiden Adressen begrenzt wird. Das Einsetzen erfolgt von rechts nach links von Stelle B<sub>1</sub>xxx einschließlich bis Stelle B<sub>2</sub>xxx ausschließlich.

Wichtig:

Ist die Adresse B<sub>1</sub>xxx ≤ Adresse B<sub>2</sub>xxx, erfolgt die Belegung bis Stelle 0000 des Speichers und vom Ende des Speichers bis Adresse B<sub>2</sub>xxx. Der Inhalt der speziellen Zonen, der Register und des Programmes werden zerstört.

Beispiel:

INC      |82|20|7117|7108|      Basis 7 = 1200

vorher : |39|39|37|36|35|34|33|32|31|30|20|5D|4D|

nachher : |39|38|37|20|20|20|20|20|20|20|20|5D|4D|

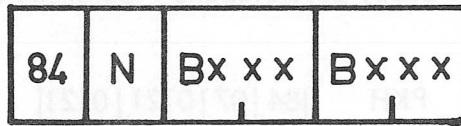
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17  
 1306                      1310                      1315

1308  
 1312

3.4.3.3. Packen hexadezimal

PKH

Pack Hexadezimal



- N = Zahl der zu packenden Bytes von 00-99
- B<sub>1</sub>xxx = rechte Adresse der abgebenden Zone
- B<sub>2</sub>xxx = rechte Adresse der empfangenden Zone

Die Adresse B<sub>2</sub>xxx muß  $\geq$  als B<sub>1</sub>xxx oder  $\leq$  B<sub>1</sub>xxx - N sein.

Dieser Befehl packt auf N Halbbytes ein Wort von N Bytes, das im ISO-Code ausgedrückt ist und nur die folgenden Werte enthält: Leerstelle, 0 - 9 und A - F. Die Umwandlung von Byte auf Halbbyte ist in der Tabelle dargestellt:

Zeichen	Leer- stelle	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ISO-Code (Byte)	20	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46
gepackt (Halb- byte)	0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Jede andere Kombination des ISO-Codes in den ursprünglichen Bytes gibt ein nicht auswertbares, sinnloses Ergebnis.

Das Wort von N Bytes, das gepackt werden soll, steht an Adresse B<sub>1</sub>xxx. Es wird in gepackter Form nach Adresse B<sub>2</sub>xxx übertragen und belegt dort N Bytes. Ist N ungerade, wird auf volle Bytes aufgerundet. In diesem 2 Fall enthält das äußerste linke Halbbyte Null.

Der alte Inhalt der empfangenden Zone wird gelöscht. Bei unterschiedlichen Feldern bleibt der Inhalt des abgebenden Feldes unverändert. Die Operation packt Stelle für Stelle von rechts nach links.

Anmerkung:

Befindet sich die Adresse B<sub>2</sub>xxx innerhalb der Adressen B<sub>1</sub>xxx - N und B<sub>1</sub>xxx, ist das Ergebnis nicht voraussehbar und hat keinerlei Wert,

Der Befehl PKH dient nicht dazu, Zahlen, mit denen gerechnet werden soll, von ungepackter in gepackte Form zu bringen. Diese Aufgabe wird von den Mehrfachübertragungsbefehlen übernommen. (s. 3.5.) Im wesentlichen dient PKH dazu, geschriebene Programme durch das Ladeprogramm zu verdichten.

Beispiel: PKH |84|07|0121|0121| Basis 0 = 2500

vorher : |45|42|30|46|41|32|31|20|20|

nachher: |45|42|30|0E|B0|FA|21|20|20|

2621

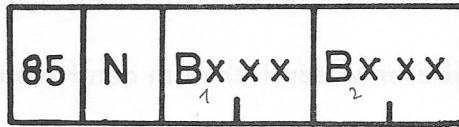
Zeichen	ISO-Codes (byte)	gepackt (Half-byte)
0	30	0
1	31	1
2	32	2
3	33	3
4	34	4
5	35	5
6	36	6
7	37	7
8	38	8
9	39	9
A	40	A
B	41	B
C	42	C
D	43	D
E	44	E
F	45	F



3.4.3.4. Entpacken hexadezimal

UPH

Un Pack Hexadezimal



N = Zahl der zu erzielenden Bytes von 00 - 99  
 B<sub>1</sub>xxx = rechte Adresse der abgebenden Zone  
 B<sub>2</sub>xxx = rechte Adresse der empfangenden Zone

Die Adresse B<sub>2</sub>xxx muß:

$$\geq B_{1xxx} + \frac{N}{2} \text{ oder } \leq B_{1xxx} - \frac{N}{2} \text{ sein.}$$

Ist N ungerade, wird  $\frac{N}{2}$  auf volle Bytes aufgerundet.

Dieser Befehl drückt im internen ISØ-Code ein Wort auf N Bytes aus, das auf N Halbbytes gespeichert ist. Die 16 hexadezimalen Werte, die jedes Halbbyte darstellen kann, ergeben folgende Zeichen ISØ-Codes:

Inhalt des Halbbytes	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
ISØ-Code (1 Byte)	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46		
Zeichen	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		

Das an Adresse B<sub>1</sub>xxx stehende, gepackte Wort wird in den ISØ-Code übersetzt und auf N Bytes an Adresse B<sub>2</sub>xxx übertragen.

Die abgebende Zone enthält  $\frac{N}{2}$  Stellen, aufgerundet auf volle Stellen, wenn N ungerade ist. In diesem Fall wird das äußerste linke Halbbyte dieser Zone nicht bearbeitet.

Der alte Inhalt der empfangenden Zone wird gelöscht. Bei verschiedenen Feldern bleibt der Inhalt der abgebenden Zone unverändert. Die Operation entpackt Stelle für Stelle von rechts nach links.

Anmerkung:

Befindet sich die Adresse B<sub>2</sub>xxx zwischen den Adressen  $B_{1xxx} + \frac{N}{2}$  und  $B_{1xxx} - \frac{N}{2}$  ist das Ergebnis nicht vorhersehbar und ohne Bedeutung.

Dieser Befehl dient nicht dazu, algebraische Zahlen aus gepackter in ungepackte Darstellung zu bringen.

Diese Aufgabe übernehmen die Mehrfachübertragungsbefehle (s. 3.5.).

Der Befehl UPH wird im wesentlichen in den Hilfsprogrammen benutzt.

Beispiel:

UPH            85 | 05 | 0150 | 0153            Basis 0 = 1000

vorher :      AB | 4C | 0B | 1E | 20 | 20 | 20 | A1

nachher:      AB | 4C | 43 | 30 | 42 | 31 | 45 | A1

1150

Zeichen	150-Code (1 Byte)	16-Bit-Code	16-Bit-Code
0	30	31	32
1	32	33	34
2	34	35	36
3	36	37	38
4	38	39	40
5	40	41	42
6	42	43	44
7	44	45	46
8	46	47	48
9	48	49	50
A	50	51	52
B	52	53	54
C	54	55	56
D	56	57	58
E	58	59	60
F	60	61	62

### 3.5. Mehrfachüberträge

#### 3.5.1. Darstellung

Die Mehrfachübertragungsbefehle gestatten es, ein Wort variabler Länge von einer Zone nach 1,2 oder 3 anderen Zonen gleichzeitig zu übertragen.

Die abgebende Stelle ist entweder ein numerisches Register oder eine von zwei Normalspeicherzonen; im allgemeinen Eingabezonen von Randeinheiten.

Die empfangenden Stellen sind: ein numerisches Register und/oder eine oder 2 Normalspeicherzonen; im allgemeinen Ausgabezonen für Randeinheiten.

Im Befehl kann nur ein Register entweder als abgebend oder als empfangend angegeben werden. Der Befehl erkennt die Normalspeicherzonen an den Adressen, die in den Basisregistern 6 - 9 stehen. Die Basisregister 6 und 7 entsprechen immer abgebenden Zonen; die Basisregister 8 und 9 immer empfangenden Zonen. Durch Vereinbarung und zur Unterscheidung bezeichnet man die Zonen mit der Nummer des ihnen zugeordneten Basisregisters.

Die verschiedenen Kombinationen der zu erzielenden Überträge, die durch Variation der abgebenden und empfangenden Zonen entstehen, ergeben eine Anzahl Befehle, die in Abschnitt 3.5.4 zusammengestellt sind.

Die Abschnitte 3.5.2 und 3.5.3 beschreiben die gemeinsamen Eigenschaften.

Drei Befehle geben die Möglichkeit, ein Zeichen in eine Zahl, die nach Zone 8 übertragen wurde, einzusetzen. Sie sind beschrieben in Abschnitt 3.5.5.

#### 3.5.2. Allgemeiner Aufbau und Darstellung

##### Symbolischer Operationstyp

Jedem Mehrfachübertragungsbefehl entspricht ein symbolischer Schlüssel der sich zusammensetzt aus dem Buchstaben M und 2, 3 oder 4 Zeichen, die von links nach rechts bedeuten:

die abgebende Stelle und die empfangende(n) Stelle(n). Die empfangenden Zonen stehen in folgender Reihenfolge:

numerisches Register, Zone 8, Zone 9.

Der Buchstabe M (Move) symbolisiert die Übertragungsoperation. Die abgebenden und empfangenden Zonen werden bezeichnet:

- durch den Buchstaben R für ein numerisches Register
- durch die Zahl (6, 7, 8 oder 9) des Basisregisters der entsprechenden Normalspeicherzone.

Beispiel:

<u>Abgebend</u>	<u>Empfangend</u>	<u>Symbolischer OT</u>
Zone 6	Reg., Zonen 8 + 9	M 6 R 8 9
Register	Zonen 8 + 9	M R 8 9
Zone 7	Zone 9	M 7 9

Das so gebildete Symbol kann noch ergänzt werden durch den Buchstaben G (Graphic), der anzeigt, daß es sich um einen Befehl handelt, der ein Zeichen in das zu übertragende Wort einsetzt.

Beispiel: M R 8 G

Format der Befehle

Die Länge eines Befehles hängt von der Anzahl der empfangenden Felder ab. Der Befehl enthält:

- den Operationstyp (OT)
- die Zahl N der zu übertragenden Stellen
- die Adresse des abgebenden Feldes
- die Adressen der empfangenden Felder in folgender Reihenfolge:  
Numerisches Register, Zone 8, Zone 9.

Theoretischer Aufbau:

OT	N	Adresse d. abgeb. Feldes	Adresse d. 1. empf. Feld. (Register)	Adresse d. 2. empf. Feld. (Zone 8)	Adresse d. 3. empf. Feldes (Zone 9)
----	---	--------------------------	--------------------------------------	------------------------------------	-------------------------------------

Betrifft der Befehl nur ein oder zwei empfangende Felder, werden die entsprechenden Adressen nach links verschoben.

Operationsschlüssel (OT)

Der Operationsschlüssel in Maschinensprache hängt von den abgebenden und den empfangenden Feldern ab. Er berücksichtigt auch das eventuelle Einsetzen eines Zeichens. Die verschiedenen Operationstypen, die möglich sind, stehen in den Abschnitten 3.5.4 und 3.5.5.

Länge N des Übertrages

Wenn der Befehl kein Register benutzt, kann die Länge N von 01 bis 99 zu übertragende Stellen betragen.

Wenn ein numerisches Register benutzt wird, wird die Länge N durch die Kapazität des Registers begrenzt.

Sie wird in Dezimalziffern dargestellt und beträgt:

- 01 - 09 bei einem Einfachregister
- 01 - 19 bei einem Doppelregister.

Wenn das Register als abgebende Stelle dient, wird die Länge N von der ersten Dezimalstelle rechts an gezählt.

Adressierung

Die Register werden normal mit ihrer Nummer adressiert. Die Adressierung der Normalspeicherzonen wurde vereinfacht:

Der Programmierer braucht nur die Verschiebungen der in Betracht kommenden Worte innerhalb der interessierenden Zonen in Betracht zu ziehen. Die zu jeder Zone gehörenden Basisregister werden automatisch durch die Entschlüsselung des Operationstyps bestimmt.

Dieses System, vereinfachte Basisadressierung genannt, funktioniert nur korrekt, wenn die Adressen (Registernummer und/oder Verschiebungen) im Befehl in der durch den Befehlsaufbau vorgesehenen Form von links nach rechts stehen. An diese Reihenfolge wird man durch den symbolischen Operationstyp erinnert.

Die Verschiebungen werden in folgender Form dargestellt:

- zwei Ziffern, wenn  $\leq 99$  : xx
- drei Ziffern, denen der Buchstabe A vorangestellt ist, wenn  $> 99$  : Axxx.

Sie belegen 1 oder 2 Bytes und werden im Befehl mit dem Buchstaben D bezeichnet, dem die Nummer der betreffenden Zone zugefügt wird.

Beispiele:

M6R89	27	09	38	16	A114	24
	OT	N	D6	R	D8	D9
MR89	16	12	42	A108	72	
	OT	N	R	D8	D9	

### 3.5.3.3. Wirkung der Befehle

Die Befehle der Mehrfachüberträge beachten die allgemeinen Regeln der Überträge im Zentralspeicher.

Jeder Übertrag vom abgebenden Feld nach dem entsprechenden empfangenden Feld erfolgt Zeichen für Zeichen von rechts nach links. Der alte Inhalt des empfangenden Feldes wird gelöscht. Der Inhalt des abgebenden Feldes bleibt erhalten.

Der Übertrag bringt das übertragene Wort in die richtige Form, wenn ein Register abgibt oder aufnimmt.

#### Übertrag einer Zone nach einem Register

Die Zahl, die ungepackt in dem abgebenden Feld steht, wird in das Register gepackt übertragen. Der Übertrag erfolgt in algebraischer Form, abhängig vom Inhalt des Vorzeichenspeichers, der unter Umständen vor dem Übertrag verändert werden muß. Durch die Operation wird der Vorzeichenspeicher gelöscht.

#### Übertrag Register nach Zone

Die algebraische Zahl, die gepackt im Register steht, wird komplementiert, falls sie negativ ist und anschließend in ungepackter Form in die empfangende Zone übertragen. Nullen links werden durch Leerzeichen (20) bis zur Länge N ersetzt. Der Vorzeichenspeicher wird abhängig vom Vorzeichen der Zahl geladen: 20 bei positivem oder keinem, 2D bei negativem Inhalt.

#### Überträge Zone nach Zone

Das übertragene Wort erfährt keinerlei Veränderung. Handelt es sich um eine ungepackte Zahl, die durch Nullen links ergänzt ist, bleiben diese erhalten.

#### Anmerkung:

Ist  $N = 00$ , findet keine Übertrag statt. Ist die abgebende Stelle ein Register, wird das Vorzeichen der Zahl, die in ihm enthalten ist, in den Vorzeichenspeicher übertragen.

3.5.4. Mehrfachüberträge (ohne Einsetzen eines Zeichens)

3.5.4.1. Die Befehle

von	nach			OT symb.	Befehle	Länge
	R	8	9			
R		X		MR8	12 N R D8	4/5
			X	MR9	14 N R D9	4/5
		X	X	MR89	16 N R D8 D9	5/7
Zone 6	X			M6R	21 N D6 R	4/5
	X	X		M6R8	23 N D6 R D8	5/7
	X		X	M6R9	25 N D6 R D9	5/7
	X	X	X	M6R89	27 N D6 R D8 D9	6/9
		X		M68	22 N D6 D8	4/6
			X	M69	24 N D6 D9	4/6
		X	X	M689	26 N D6 D8 D9	5/8
Zone 7	X			M7R	31 N D7 R	4/5
	X	X		M7R8	33 N D7 R D8	5/7
	X		X	M7R9	35 N D7 R D9	5/7
	X	X	X	M7R89	37 N D7 R D8 D9	6/9
		X		M78	32 N D7 D8	4/6
			X	M79	34 N D7 D9	4/6
		X	X	M789	36 N D7 D8 D9	5/8

L = Länge des Befehles in Bytes (min. und max. je nachdem, ob die Verschiebungen 1 oder 2 Bytes belegen).



3.5.4.2. Beispiele:

A Überträge von einem Register nach einer Zone (MR8, MR9, MR89)

a) MR 8 12|06|10|12| Basis 8 = 2100  
 Reg. 10 VZ Zone 8  
 vorher : 00|00|00|21|14| 00| 42|34|33|46|54|20|59|54|  
 nachher : 00|00|00|21|14| 20| 42|20|20|32|31|31|34|54|

b) MR 9 14|05|13|24| Basis 9 = 2175  
 Reg. 13 VZ Zone 9  
 vorher : 99|99|99|94|37| 00| 43|58|37|38|20|46|45|52|  
 nachher : 99|99|99|94|37| 2D| 43|58|20|20|35|36|33|52|

c) MR 89 16|07|12|A112| 58 Basis 8 = 2038  
 Basis 9 = 2175  
 vorher: nachher:  
 VZ 20| 20|  
 Reg. 12 00|00|05|48|32| 00|00|05|48|32|  
 Zone 8 20|20|20|20|20|20|20| 20|20|35|34|38|33|32|  
 Zone 9 41|42|43|44|45|46|47| 20|20|35|34|38|33|32|

B Überträge einer Zone nach Register (M6R, M7R)

a) M6R 21|05|32|15| Basis 6 = 2257  
 Zone 6 VZ Reg. 15  
 vorher : 20|20|32|34|39| 20| 99|99|99|83|14|  
 nachher : 20|20|32|34|39| 00| 00|00|00|02|49|  
 + 249

b) M7R 31 | 10 | 34 | 14 | Basis 7 = 1780

Vorzeichenspeicher vorher: nachher:

2D 00

Zone 7 20 | 20 | 20 | 34 | 33 | 38 | 35 | 32 | 31 | 39 |

Reg. 13/14 00 | 00 | 09 | 43 | 83 | 00 | 00 | 00 | 28 | 34 |

99 | 99 | 99 | 99 | 99 | 95 | 61 | 47 | 81 |

(- 4.385.219)

C Überträge von einer Zone nach einem Register und einer Zone (M6R8, M6R9, M6R89, M7R8, M7R9, M7R89)

M6R8 23 | 06 | 12 | 44 | A1 | 24 | Basis 6 = 2257

Basis 8 = 2038

vorher: nachher:

VZ 2D 00

Zone 6 42 | 20 | 30 | 38 | 33 | 30 | 30 | 46 | 42 | 20 | 30 | 38 | 33 | 30 | 30 | 46 |

Zone 8 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 30 | 38 | 33 | 30 | 30 | 20 |

Reg. 44 00 | 98 | 04 | 00 | 17 | 99 | 99 | 99 | 17 | 00 |

(- 8.300)

D Überträge von Zone nach Zone (M68, M69, M689, M78, M79, M789)

M789 36 | 05 | 32 | 12 | 80 | Basis 7 = 2257

Basis 8 = 2038

Basis 9 = 2175

vorher: nachher:

Zone 7 44 | 41 | 39 | 34 | 54 | 30 | 20 | 44 | 41 | 39 | 34 | 54 | 30 | 20 |

Zone 8 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 41 | 39 | 34 | 54 | 30 | 20 |

Zone 9 39 | 46 | 30 | 36 | 48 | 32 | 45 | 39 | 41 | 39 | 34 | 54 | 30 | 45 |



b) MR8G    1E|06|24|A1|12|C2    Basis 8 = 2038

	Reg. 24	VZ	Zone 8
vorher :	<u>00 00 00 00 05</u>	<u>00</u>	<u>20 44 52 43 56 58 41 49</u>
nachher:	<u>00 00 00 00 05</u>	<u>20</u>	<u>20 20 20 20 30 2C 30 35</u>

0000000005    →    0,05

3.6. Ein-/Ausgabeoperationen

3.6.0. Die Zone IØC

Sie zerfällt in 2 große Bereiche:

- Eingabe - Ausgabe
- Kennzeichnung der Verbindung

Der Bereich "Eingabe - Ausgabe" umfaßt 6 Gruppen zu je 4 Bytes. Jede der Gruppen kann einer Randeinheit zugeordnet sein. Ein IØC-Befehl belegt so viele Gruppen, wie er Randeinheiten verbindet.

Die Belegung dieses Bereiches wird im Augenblick der Aufnahme eines IØC durchgeführt, und zwar von links nach rechts.

Die Anzahl der gleichzeitig in diesem Bereich verbundenen Randeinheiten ist auf 6 begrenzt. Der Rechner berücksichtigt diesen Umstand.

Der Bereich wird von links nach rechts abgetastet. Während dieses Vorganges werden die einzelnen Randeinheiten entsprechend ihrer Verfügbarkeit in Arbeit gesetzt oder nicht. Wenn eine Gruppe entweder nichts oder die Elemente eines durchgeführten IØC enthält, wird sie nicht mehr abgetastet.

Der Bereich "Kennzeichnung der Verbindung" umfaßt 5 Bytes, die den Programmen 1 - 5 entsprechen.

Jeweils das rechte Halbbyte enthält die Anzahl der Randeinheiten, deren Arbeit noch nicht beendet ist. Bei gleitenden IØC-Befehlen können weitere IØC desselben Programms folgen. In diesem Fall wird eine Addition der Anzahl der durch einen jeden IOC angesprochenen Randeinheiten in diesem Halbbyte vorgenommen.

Das jeweils linke Halbbyte enthält:

- 8 bei einem blockierenden IØC
- 9 bei einem gleitenden IØC

### 3.6.1. Rolle und Aufgabe des Ein-/Ausgabebefehls

Jeder Informationsaustausch zwischen der Zentraleinheit und den verschiedenen Randeinheiten, die angeschlossen sind, wird durch einen einzigen Befehl vorgenommen: den Ein-/Ausgabebefehl.

Dieser Befehl, der in zwei Versionen existiert, steuert den Ablauf der Randeinheiten, indem er:

- entweder in den Zentralspeicher die gelesenen Daten zur Verarbeitung eingibt, die vom Kartenleser, dem Pufferspeicher oder der Trommel herkommen;
- oder Resultate nach durchgeführten Bearbeitungen durch die Zentraleinheit zum "Schreiben" auf externe Randeinheiten (z.B. Karten, Listen, Trommel etc.) bereitstellt;
- oder bestimmte Funktionen der Randeinheiten, wie z.B. Auswahl der Bahn auf der Trommel oder Papiervorschub auf dem Drucker etc. steuert.

Diese Funktionen sind in den Kapiteln beschrieben, die sich auf die Randeinheiten beziehen. Die allgemeinen Richtlinien, um diese Funktionen anzuwenden, befinden sich in Abschnitt 3.6.5.

Ein einziger Befehl kann mehrere Operationen (max. 6) unter den Bedingungen, die in Abschnitt 3.6.3 erwähnt sind, durchführen.

Der Ein-/Ausgabebefehl enthält:

- einen Operationstyp ( $\emptyset T$ ),
- die Zahl N der angerufenen Operationen,
- die entsprechenden Parameter für jede angerufene Operation.

Eine Operation, die sich auf eine Randeinheit bezieht, wird definiert durch zwei Parameter: den Anschlußcode und die Nummer des Basisregisters.

Der Anschlußcode

- identifiziert die angesprochene Randeinheit:
  - langsame oder schnelle Einheit;
  - Nummer des Kanals
  - Nummer der Einheit

- legt die durchzuführende Operation fest:

- Lesen,
- Schreiben,
- besondere Anweisungen (bei Trommel).

Er setzt sich aus 8 Bits zusammen und wird in der Form von zwei hexadezimalen Zeichen dargestellt.

Die Basisregister

geben die linke Adresse - 1 der Ein- oder Ausgabezone im Zentralspeicher an.

### 3.6.2. Organisation einer Ein-/Ausgabeoperation

Jeder angeschlossenen Randeinheit ist im Zentralspeicher eine Zone für die Ein- oder Ausgabe zugeordnet. Ihre Länge wird bestimmt durch die maximale Anzahl der im Laufe der Operation zu übertragenden Zeichen zuzüglich eventueller Hilfsschlüssel (s. 3.6.5.) und einer Trennmarke.

Der Übertrag einer Information erfolgt in reeller Länge.

Er beginnt:

- bei der Randeinheit an der ersten angesprochenen Stelle (normalerweise Stelle 1)
- im Zentralspeicher an der Stelle + 1, die als Basisadresse im angesprochenen Register steht.

Die Operation läuft im Speicher von links nach rechts bis zu einer Trennmarke ab, die anzeigt, daß das letzte Zeichen übertragen wurde.

Diese Trennmarke, dargestellt durch das Zeichen "F4" begrenzt eine Information nach folgenden Richtlinien:

#### a) Informationsausgabe aus dem Zentralspeicher

Die Trennmarke wird in der Ausgabezone des Zentralspeichers rechts vom letzten zu übertragenden Zeichen gesetzt. Bezieht sich die Ausgabe auf die Trommel, wird diese Trennmarke mitübertragen. In allen anderen Fällen zeigt sie lediglich das Ende des Übertrages an, ohne übertragen zu werden.



b) Informationsausgabe von der Trommel

Die Trennmarke, die vorher mit der Information auf die Trommel geschrieben wurde, zeigt das Ende des Übertrages an. Sie wird mit der Information in den Zentralspeicher übertragen.

c) Informationsausgabe von einer anderen Randeinheit

Da die Trennmarke von den Randeinheiten nicht übertragen wird, muß sie vorher im Zentralspeicher in der betreffenden Zone gesetzt werden. Sie belegt die erste Stelle rechts nach den für die Information vorgesehenen Stellen. Die Zentraleinheit analysiert den Inhalt der Stellen, die im Laufe des Übertrages einen Wert erhalten haben. Die Analyse der Trennmarke beendet die Operation.

Es ist Aufgabe des Programmierers, darauf zu achten, daß die Trennmarken an der richtigen Stelle stehen.

Die Trennmarke darf nicht an der Stelle + 1 stehen, die durch das Basisregister angegeben ist, weil sie dort nicht erkannt werden kann. Das bedeutet, daß die zu übertragende Information wenigstens eine Stelle umfassen muß.

Übertrag der Information

Die Randeinheiten stehen mit der Zentraleinheit über Kanäle in Verbindung. Jeder Kanal mündet in der Zentraleinheit in ein einstelliges Pufferregister, durch das die Information übertragen wird. Bevor die Operation gestartet wird, untersucht die Zentraleinheit, ob die angesprochene Randeinheit frei ist. Ist das der Fall, wird die Operation gestartet und läuft in folgender Weise ab:

a) Eingabe

Die Randeinheit überträgt die gelesenen Zeichen nacheinander in das Pufferregister. Anschließend erfolgt der Übertrag in die Eingabezone.

b) Ausgabe

Die Zentraleinheit entnimmt der Ausgabezone Zeichen und überträgt sie nacheinander in das Pufferregister unter der Voraussetzung, daß dieses frei ist.

Das erfolgt jedesmal, wenn die Randeinheit ein Zeichen übernommen hat, das vorher zur Ausgabe bereitgestellt wurde.

Bei jedem Übertrag eines Zeichens rechnet die Zentraleinheit die laufende Adresse der Ein- oder Ausgabebzone weiter.

Anmerkung:

Die Dauer der Operation wird durch die langsamste Randeinheit bestimmt.

Ist eine Randeinheit nicht frei, wartet die Operation solange, bis der Anschluß möglich ist.

3.6.3. Simultanbefehle mehrerer Operationen (Horizontale Simultaneität)

Ein einziger Befehl kann mehrere Randeinheiten ansprechen unter der Bedingung, daß diese sich an verschiedenen Kanälen befinden. So kann z. B. der Kartenstanzer (Kanal 2) und der Drucker (Kanal 3) gemeinsam arbeiten; dagegen müssen der Kartenleser und die numerische Tastatur durch verschiedene Befehle angesprochen werden, weil sie sich an einem gemeinsamen Kanal (1) befinden.

Gegenseitige Blockierung von Randeinheiten am gleichen Kanal

Zwischen Randeinheiten, die an denselben Kanal angeschlossen sind, ist keine Simultaneität möglich.

Beispiel: IØC Leser - num. Tastatur - Alphatastatur

Während des Ab tastens der IØC-Zone wird in diesem Fall nur die erste erkannte langsame Randeinheit bedient. Die Schaltkreise des GE-55 halten die Tatsache fest, daß diese eine Randeinheit verbunden ist und alle anderen langsamen Randeinheiten am gleichen Kanal noch zu bedienen sind.

Beispiel: 53 11 94 R  
simultanes Programm  
53 01 9A 12

Die Alphatastatur ist während der Kernspeicherarbeit simultan verbunden. Die numerische Tastatur wird erst entriegelt, wenn die Zone für die Alphatastatur gefüllt worden ist.

Ist beim Multiprogramming ein Programm in der Zone PRC in Wartestellung, führt der IØC für die numerische Tastatur einen Programmwechsel herbei. Dieses Programm läuft simultan mit der Verbindung der Alphatastatur ab. Ist die Arbeit an der Alphatastatur beendet, kommt es zu einer Simultaneität mit der numerischen Tastatur.

Ein Befehl ist abgeschlossen, wenn alle befohlenen Operationen durchgeführt worden sind.

Die Operationen werden in der Reihenfolge gestartet, in der die Einheiten zur Verfügung stehen. Die langsamen Randeinheiten können simultan arbeiten (s. 2.1.1.). Die Zentraleinheit löst, soweit es möglich ist, die Operationen parallel aus und kontrolliert den Ablauf Zeichen für Zeichen.

Dagegen erlaubt der Anruf der Trommel der Zentraleinheit nicht, mehrere Operationen simultan auszuführen, von denen eine diese Einheit betrifft. Diese wird vorrangig gestartet, sobald die Trommel zur Verfügung steht und läuft allein ab; die anderen Operationen werden erst ausgeführt, wenn diese vollständig durchgeführt wurde.

#### Kontrolle der Operationen

Die Zentraleinheit speichert die Ein-/Ausgabebefehle und führt die Operationen in der IØC-Zone aus, die im einzelnen enthält:

- die erforderlichen Angaben, spez. die Anschlußcodes, die durch den Befehl gegeben werden.
- die Informationen für den Ablauf der Operationen (lfd. Adressen der Zonen, Zahl der angeschalteten Randeinheiten etc.)

Nach jeder Beendigung einer Operation werden die entsprechenden Angaben in der IØC-Zone gelöscht und die Zahl der angeschalteten Randeinheiten wird um eine vermindert. Sobald diese Zahl Null erreicht, ist der Befehl beendet.

Wenn einer der Ein-/Ausgabebefehle, die gleichzeitig in der IØC-Zone gespeichert sind, den Kartenleser betrifft, dürfen es nur maximal 5 Befehle sein und zwar:

- höchstens 3 bei normalen Randeinheiten (einschl. Kartenleser),
- höchstens 2 bei schnellen Randeinheiten (z. B. Trommel).

Daraus ergibt sich in diesen Fällen:

- ein einziger Ein-/Ausgabebefehl kann nicht mehr als 5 Operationen in der oben beschriebenen Weise bearbeiten.
- mehrere Ein-/Ausgabebefehle, die diese Konditionen nicht beachten können in Verbindung mit Multiprogramming nicht in der IØC-Zone eingespeichert werden. Es ist erforderlich, die notwendigen Schutzmaßnahmen vorzusehen (s. Kap. 4).

#### 3.6.4. Verhalten bei Störungen

Jede normale (leeres Fach bei Leser und Stanzer, Papierende am Drucker etc.) oder anormale (gestörte Schreib- oder Leseoperation etc.) Störung an einer Randeinheit bewirkt, daß diese Einheit nicht ansprechbar ist. Eine entsprechende Operation kann nicht gestartet werden. Ein Ein-/Ausgabe-befehl, der diese Einheit anspricht, wird in der IØC-Zone solange in Wartestellung gesetzt, bis die Störung beseitigt ist.

Aus diesem Grunde testet die Zentraleinheit vor jeder Ein-/Ausgabeoperation die angesprochene Randeinheit auf Bereitschaft. Ist das nicht der Fall, wird die Operation nicht gestartet und das Programm blockiert auf dem Befehl, weil es nicht beendet werden kann.

Der Bedienung wird das angezeigt durch Aufleuchten der entsprechenden Lampe am Bedienungspult. Sie hat daraufhin den Fehler zu beseitigen und die Randeinheit wieder betriebsbereit zu machen.

Wenn die Störungsanzeige aufgrund eines Lese- oder Schreibfehlers bei Trommelüberträgen erfolgte, oder wenn ein Stanzfehler durch die Kontrolleinrichtung des Stanzers angezeigt wurde, kann die Bedienung das Programm unterbrechen und ein "Fehlerprogramm" starten, das eingespeichert ist, um diese Störung zu beseitigen. Sie drückt hierzu auf die Taste "PI" (Program Interrupt) bevor sie die Randeinheit wieder einschaltet.

Nach Durchführung dieser Manipulationen und wiederhergestellter Arbeitsbereitschaft der Randeinheit startet die Zentraleinheit die in der IØC-Zone wartende Operation, führt sie aus und beendet den Ein-/Ausgabe-befehl, der blockiert worden war. Infolgedessen und je nachdem das Fehlerprogramm gestartet wurde oder nicht, springt das laufende Programm auf das Fehlerprogramm oder fährt in seinem normalen Ablauf fort.

#### Schlußfolgerungen

Der oben beschriebene Vorgang erfordert die Beachtung folgender Bemerkungen:

- a) Eine Störung, die durch eine Einheit im Laufe einer Operation X hervorgerufen wurde, wird durch die Zentraleinheit nur festgestellt und angezeigt, wenn die nächste Operation Y diese Randeinheit anspricht.
- b) Nach Beseitigung der Störung wird die in Wartestellung befindliche Operation Y durchgeführt.
- c) Eine Programmunterbrechung kann nur am Ende des Befehls erfolgen, der die Operation Y enthält.

Am Ende eines fehlerhaften Lesens oder Schreibens auf der Trommel (Operation X) hat man die Möglichkeit, die Operation zu wiederholen, bevor in der Bearbeitung fortgefahren wird. Dazu muß das Programm am Ende der Operation X angehalten werden, damit die fehlerhaften Daten nicht verarbeitet oder die Daten erneut zum Schreiben bereitgestellt werden können. Dazu muß man unmittelbar nach der Operation X eine Spezialoperation Y programmieren, deren einzige Aufgabe darin besteht, festzustellen, ob die Randeinheit zur Verfügung steht.

Diese besondere Operation ist in Abschnitt 3.6.5 beschrieben.

Das gleiche Verfahren ist auch beim Kartenstanzer anzuwenden, um sich die Möglichkeit zu lassen, bei fehlerhafter Stanzung den Stanzvorgang zu wiederholen.

### 3.6.5. Besonderheiten des Ein-/Ausgabebefehles

#### Anschlußfunktionen

Die Anschlußfunktionen der Randeinheiten werden entweder durch spezielle Anschlußcodes oder durch Hilfsschlüssel angerufen. Diese besonderen Fälle der Anwendung des Ein-/Ausgabebefehls sind in den in Abschnitt 3.6.6 angeführten Beispielen beschrieben.

#### 1. Trommel

Zwei spezielle Anschlußcodes ermöglichen es, die Bahn auf der Trommel auszuwählen, auf die geschrieben oder von der gelesen werden soll und den Zustand der Randeinheiten zu testen.

Für jede dieser Operationen drückt der Befehl, zusätzlich zum jeweiligen Anschlußcode, die Nummer des Basisregisters aus, dessen Aufgabe im folgenden für jeden Fall beschrieben wird.

#### Auswahl der Bahn

Die Operation besteht darin, die Nummer der gewünschten Bahn auszuwählen. Das Basisregister enthält die Adresse - 1 einer Zone von 4 Bytes, die die Nummer der Bahn auf drei Bytes enthält. Auf dem vierten Byte steht eine Trennmarke F4.

Die Auswahl einer Bahn kann nicht durch den gleichen Befehl kontrolliert werden, der auch die Lese- oder Schreiboperation kontrolliert.

Darum sollte die Auswahl der Bahn und das Lesen-Schreiben unabhängig voneinander programmiert werden, damit jede Operation gesondert getestet werden kann.

### Test der Einheit

Hier wird der Ein-/Ausgabebefehl dazu benutzt, zu testen, ob die letzte auf der Trommel durchgeführte Operation richtig durchgeführt wurde (s. 3.6.4.). Ist im Laufe dieser Operation ein Fehler aufgetreten, kann der Testbefehl nicht durchgeführt werden (Einheit ist nicht ansprechbar) und blockiert so das Programm.

Das spezifizierte Basisregister enthält die Adresse (nicht die Adresse -1) einer Stelle, die keinen Wert enthalten darf. Um nicht ein besonderes Testregister zu belegen, kann man die Stelle vor der Ein/Ausgabezone für die Trommel verwenden. Bei vorher richtig durchgeführter Operation wird das hexadezimale Zeichen "10" an der Basisadresse zur Verfügung gestellt. Durch Analyse des Testzeichens kann gegebenenfalls die falsch durchgeführte Operation wiederholt werden.

Der Test wird durch einen besonderen Ein-/Ausgabebefehl programmiert, der unmittelbar der Operation folgt, die man testen will: Auswahl, Lesen oder Schreiben.

### Anschluß von zwei Trommeln

Der Festspeicher MMS9 ermöglicht den Anschluß von zwei Magnettrommeln. (vgl. 2.8.3.2.)

### 2. Pufferspeicher, Kartenstanzer, Drucker

Diese Randeinheiten benutzen Hilfsschlüssel, deren Aufgabe in den Kapiteln beschrieben wurde, die sich auf sie beziehen.

Ein Hilfsschlüssel ist eine besondere Anweisung, die unter der Form eines 8 Bit-Zeichens dargestellt wird und aus der im internen Code zur Verfügung stehenden Kombinationen genommen worden ist (Spalten 0 und 1). Er wird im Zentralspeicher wie ein normaler Wert gespeichert und wird an die entsprechende Randeinheit durch einen Ein-/Ausgabebefehl entweder in Verbindung mit Daten oder allein übertragen. Seine Wirkung ist unmittelbar: Sobald eine Randeinheit einen Hilfsschlüssel empfangen hat, führt sie die entsprechende Funktion aus.



Hilfsschlüssel, die eine Information begleiten

Diese Schlüssel dienen dazu, am Ende einer Stanzoperation die Karte auszuwerfen oder am Ende einer Zeile den Papiervorschub zu ermöglichen.

Der Hilfsschlüssel wird in die Ausgabezone, die die auszugebende Information enthält, übertragen; und zwar an die Stelle, wo man die gewünschte Anschlußfunktion ausüben will unter Berücksichtigung, daß die Randeinheit Stelle für Stelle verarbeitet: Handelt es sich um ein gewöhnliches Zeichen, wird es geschrieben, ist es ein Hilfsschlüssel, wird die befohlene Funktion ausgeübt. Will man zum Beispiel einen Papiervorschub nach dem Drucken einer Zeile haben, setzt man den entsprechenden Hilfsschlüssel zwischen die zu druckende Information und die Trennmarke.

Mehrere Hilfsschlüssel können im Laufe einer Operation übertragen werden. Sie werden in der Reihenfolge, in der sie stehen, ausgeführt.

Hilfsschlüssel, die allein übertragen werden.

Diese Schlüssel werden speziell zum Einschalten der numerischen Tastatur und der Leuchtanzeige benutzt.

Der Übertrag dieser Schlüssel besteht aus einer besonderen Operation, die definiert wird durch:

- einen Anschlußcode, der die Randeinheit auswählt;
- die Nummer eines Registers, das die Adresse - 1 der Stelle enthält, an der der Hilfsschlüssel steht.

Diesem Schlüssel muß die Trennmarke F4 folgen.

Die Operation kann mit anderen Operationen der Ein-/Ausgabe im gleichen Befehl programmiert sein.

Anmerkung:

Der Pufferspeicher wird durch zwei verschiedene Anschlußcodes bezeichnet, und zwar je nachdem er empfangendes oder abgebendes Organ ist.

3. Kartenstanzer PS 40

Test der Einheit

Ein besonderer Anschlußcode ermöglicht die Feststellung, ob die zuletzt durchgeführte Operation zufriedenstellend abgelaufen ist.



Dieser Test benötigt einen speziellen Ein-/Ausgabebefehl, der unmittelbar dem Stanzbefehl folgt. Der spezielle Anschlußcode benutzt ein Basisregister, das die rechte Adresse - 1 einer Zone von 2 Bytes enthält, in der irgendein Zeichen und eine Trennmarke steht.

3.6.6. Die Befehle

$I\emptyset C$	Input Output Control
$I\emptyset IC$	Input Output Indirect Control

53	N	C <sub>1</sub>	R <sub>1</sub>	C <sub>2</sub>	R <sub>2</sub>	C <sub>3</sub>	R <sub>3</sub>	C <sub>4</sub>	R <sub>4</sub>	C <sub>5</sub>	R <sub>5</sub>	C <sub>6</sub>	R <sub>6</sub>
54	N	RC <sub>1</sub>	R <sub>1</sub>	RC <sub>2</sub>	R <sub>2</sub>	RC <sub>3</sub>	R <sub>3</sub>	RC <sub>4</sub>	R <sub>4</sub>	RC <sub>5</sub>	R <sub>5</sub>	RC <sub>6</sub>	R <sub>6</sub>

N = Zahl der Randeinheiten von 1 - 6

Für jede Einheit bedeutet:

C = Anschlußcode (s. Tafel)

RC = Nummer des Registers, das den Anschlußcode enthält

R = Nummer des Basisregisters

Die folgenden Angaben werden in die spezielle  $I\emptyset C$ -Zone übertragen:


- die Zahl N der Randeinheiten;
- je Einheit: - der entschlüsselte Anschlußcode,
- die Basisadresse des Registers R.

Der Anschlußcode ist enthalten:

- im  $I\emptyset C$  : im Befehl selbst;
- im  $I\emptyset IC$  : an der ersten, rechten Stelle des Registers RC.

Anmerkung:

Die Abschnitte 3.6.3 und 3.6.5 legten die Bedingungen zur Anwendung der Befehle  $I\emptyset C$  und  $I\emptyset IC$  fest und zeigten, welche Operationen simultan nicht vereinbar sind. Es ist darum erforderlich, daß man sich gut die Operationen merkt, die in einem Befehl angerufen werden können.

**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				
	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	1. Trommel - Lesen
		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

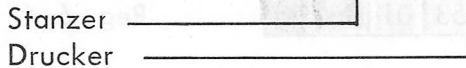
d) Freigabe Sichtanzeige Hilfsschlüssel Adresse - 1

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:

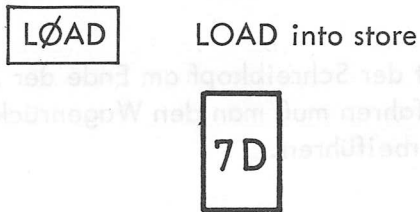
Diese Befehle können beim Test eines Programmes gebraucht werden, um den Inhalt bestimmter Kernspeicherzonen nach bestimmten Bearbeitungsfolgen auszudrucken. Sie werden an der Stelle in das Programm eingeschoben, die vom Programmierer gewählt wurde. Nach dem Test läuft die normale Bearbeitung weiter.

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.

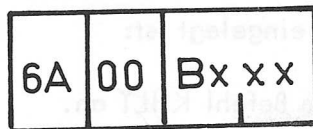
Wenn der Befehl "LØAD" ausgeführt ist, übernimmt das Programm den Befehl, der sich an der oben erwähnten Startadresse befindet, und der in das Programm-Adreß-Register übertragen worden ist.

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

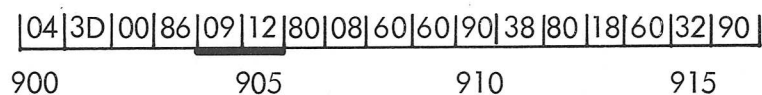
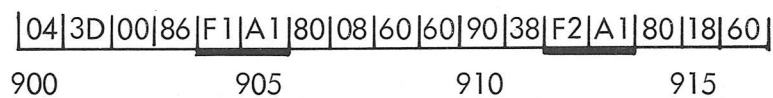
Dieser Befehl ersetzt in einem Programm die symbolischen Sprungadressen (siehe 3.2.) durch ihre wirkliche Adresse.

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:

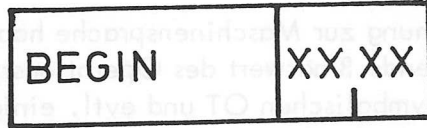
- organisatorische Pseudobefehle, die den Anfang und das Ende eines Programms festlegen sowie die vorläufigen Niveaus (Anfang der Sektionen).
- Pseudobefehle zur Erstellung von "Konstanten" (festen Werten), die feststehende und immer wieder beim Ablauf der Programme benutzten Daten eingeben.

3.9.1. Organisation eines Programmes

3.9.1.1. Programmanfang

BEGIN

BEGINning adress where the program is loading



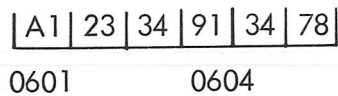
xxxx = absolute Adresse

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

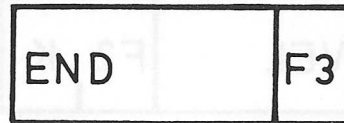




3.9.1.2. Programmende

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:

IOC | 53 | 01 | CA | 08 |

END | F3 |

Zentralspeicher



3.9.1.3. Anfang einer Sektion

LEVEL      LEVEL at which to stop jump.



F2 = festes Zeichen, das den Pseudobefehl darstellt.  
 K = 2 hexadezimale Zeichen, die das Etikett des Niveaus darstellen  
 (die Kombinationen F1 bis FF sind verboten).

Der Pseudobefehl LEVEL erfüllt in einem in die Maschinensprache umgewandelten Programm keine Aufgabe und erzeugt darum auch keinen Befehl.

Während des Ladens eines Programmes in den Kernspeicher wird das Bezugszeichen jedes Niveaus in allen Sprungbefehlen durch die reelle Adresse des Befehls ersetzt, der dem betreffenden Pseudobefehl unmittelbar folgt.

Die Pseudobefehle werden anschließend unterdrückt; das Programm entsprechend zusammengeschoben, wobei bei der Bestimmung der realen Adresse dieser Vorgang berücksichtigt wird.

Beispiel:

Anfangsprogramm		Programm in Maschinensprache	
Adr. A	Befehle	Adr. B	Befehle
0800	JRT 02 F1 1A	0800	02 09 54
0954	LEVEL F2 1A	0954	80 12 60 60 90 40
0956	MVC 80 12 60 60 90 40	0956	Fortsetzung des Programmes
	Fortsetzung des Programmes		

Adressen A: Adressen bei Laden vor dem Zusammenschieben

Adressen B: endgültige reelle Adressen.

3.9.1.4. Nullbefehl

NØP

No OPeration



Dieser Befehl erzeugt zwei gepackte Nullen an der Stelle, an der er im Programm steht.

Wenn das Programm auf diese Stelle stößt, wird keine Funktion ausgelöst. Das PAR erhöht sich aber um 1. Dadurch wird der Programmablauf nicht unterbrochen und der nächste Befehl kommt zur Durchführung.

Anmerkung:

Wenn dieser Pseudobefehl innerhalb einer Befehlsfolge steht, darf diese Stelle kein anderes Zeichen erhalten, da dieses von der Zentraleinheit als Operationstyp gewertet wird und ein nicht vorherzusehendes Resultat ergibt.

Beispiel:

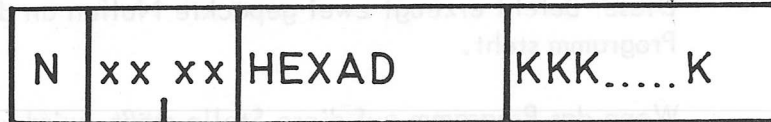
			PAR
800	NØP	} vorher nachher	0800
801	MVC		0801
	etc.		

### 3.9.2. Erstellung von Konstanten

Die Pseudobefehle zur Erstellung von Konstanten haben einen besonderen Aufbau, so daß sie nicht innerhalb eines Programmes auftreten dürfen. Ihr Platz ist entweder unmittelbar vor dem BEGIN oder nach dem END.

#### 3.9.2.1. Hexadezimale Konstanten

**HEXAD**      HEXADezimal data to load.



N            = Zahl der Bytes, die von den einzelnen Zeichen belegt werden (max. 32);

xxxx        = absolute Adresse, an der das 1. Zeichen von links eingespeichert werden soll;

KK...K     = hexadezimale Zeichen, die eingesetzt werden sollen (2-64 Zeichen, K = 2 Zeichen).

Dieser Pseudobefehl bewirkt das Erstellen von Zeichen K an den Stellen, die der absoluten Adresse xxxx folgen.

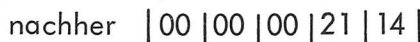
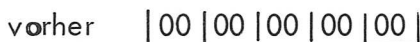
Die Zeichen K werden von links nach rechts eingespeichert, so daß je 2 ein Byte in aufsteigender Reihenfolge belegen.

#### Beispiel:

Erstellen einer Basisadresse im Register 8



Register 8



0140

3.9.2.2. Alphanumerische Konstanten

AN AlphaNumeric data to load.



- N = Zahl der Bytes, die von den einzelnen Zeichen belegt werden (max. 64);
- xxxx = absolute Adresse, an der das 1. Zeichen von links eingespeichert werden soll;
- AA...A = alphanumerische Zeichen, die eingesetzt werden sollen (1-64; A = 1 Zeichen).

Dieser Pseudobefehl bewirkt das Erstellen von Zeichen A an den Stellen, die der absoluten Adresse xxxx folgen.

Die Zeichen A werden von links nach rechts eingespeichert, so daß 1 Zeichen ein Byte in aufsteigender Reihenfolge belegt.

Beispiel:

Erstellen von konstanten Daten



vorher: 

00	00	00	00	00	00	00	00
----	----	----	----	----	----	----	----

nachher: 

00	20	31	30	38	25	20	00
----	----	----	----	----	----	----	----

0830

3.9.2.2. Alphanumerische Konstanten  
Alphanumeric data to load



xxxx = absolute Adresse, an der das 1. Zeichen von links einge-  
 speichert werden soll;  
 AA...A = alphanumerische Zeichen, die eingesetzt werden sollen  
 (1-64; A = 1 Zeichen)

Dieser Pseudobefehl bewirkt das Erstellen von Zeichen A an den Stellen,  
 die der absoluten Adresse xxxx folgen.

Die Zeichen A werden von links nach rechts eingezeichnet, so daß 1 Zeichen  
 ein Byte in aufsteigender Reihenfolge belegt.

Beispiel:

Erstellen von konstanten Daten



0830

#### 4. Simultaneität und Multiprogramming beim GE-55

##### 4.0. Vorbemerkungen

Der erste Abschnitt dieses Kapitels hat es sich zur Aufgabe gestellt, anhand eines ganz allgemeinen Plans eine Vorstellung der Simultaneität und des Multiprogramming, die wiederum nur eine erweiterte Form der Simultaneität ist, zu geben. Die verschiedenen Stufen des Simultanarbeitens werden dargelegt, mit denen eine Optimierung des Nutzen eines Rechners, der nur mit einem Processor ausgestattet ist, wie der GE-55 erreicht werden kann. Es wird gezeigt:

- wo jede Stufe gerechtfertigt ist und was sie für Folgen hat;
- wie man sie anwendet;
- wo die kritische Grenze ist, nach der eine Optimierung illusorisch ist.

Dieser erste Abschnitt nimmt darum auch keinen Bezug auf die Art und Weise, mit der diese Stufen der Simultaneität des GE-55 realisiert werden. Das ist Aufgabe der Abschnitte 4.2. und 4.3. Im besonderen soll man keine Schlüsse in bezug auf den GE-55 aus den Diagrammen ziehen, die ganz allgemein das in 4.1. gesagte illustrieren sollen.

##### 4.1. Allgemeine Einführung in die Simultaneität

Im Normalfall hat ein Programm folgenden Aufbau:

Lesen einer Reihe von Daten  
und Übertrag in die Lesezone

|  
Verarbeitung in der ZE

|  
Schreiben der Ergebnisse aus  
einer Ausgabezone

|  
Lesen einer neuen Reihe  
von Daten

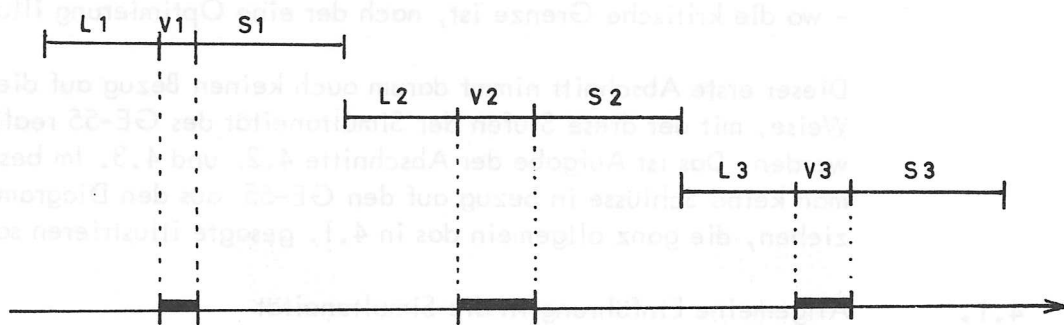
|  
etc.



Bezeichnet man mit:

- L 1 die Dauer des Lesens der Daten 1,
- V 1 die Dauer der Verarbeitung dieser Daten,
- S 1 die Dauer des Schreibens der Ergebnisse,
- L 2, V 2, S 2, die Dauer dieser Arbeiten für die Daten 2 etc.

So entsteht folgendes Diagramm der Operationen für eine Maschine, die keine Möglichkeit der Simultanarbeit hat:



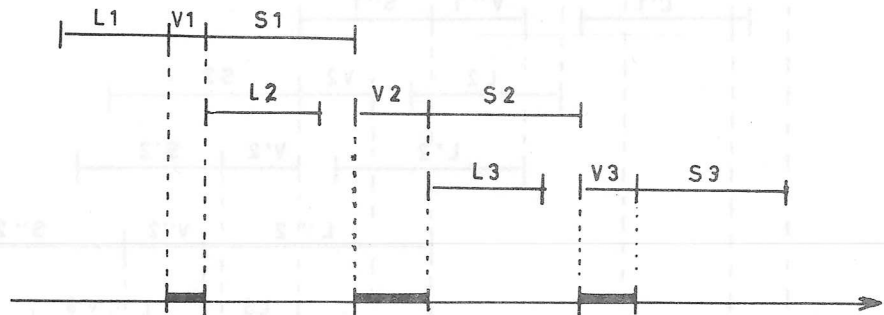
Belegung der Zentraleinheit (schraffiert dargestellt).

Aufgrund dieses Diagramms erkennt man, daß das Programm nur einen kleinen Bruchteil der Zeit beansprucht, die zur Bearbeitung durch die Zentraleinheit mit ihrer Elektronik zur Verfügung steht, während die Funktionen des Lesens und Schreibens durch relativ langsame elektromechanische Geräte erfolgen.

### 1. Abschnitt

Eine erste Verbesserung dieses Ablaufs erreicht man, wenn man das Lesen der zweiten Serie von Daten nach der Verarbeitung der Daten 1 vornimmt, d.h., gleichzeitig mit dem Schreiben der Ergebnisse 1. Da man nach der Verarbeitung V<sub>1</sub> die Daten 1 nicht mehr benötigt, kann man neue Daten in die Lesezone übertragen.

Das Diagramm sieht dann folgendermaßen aus:



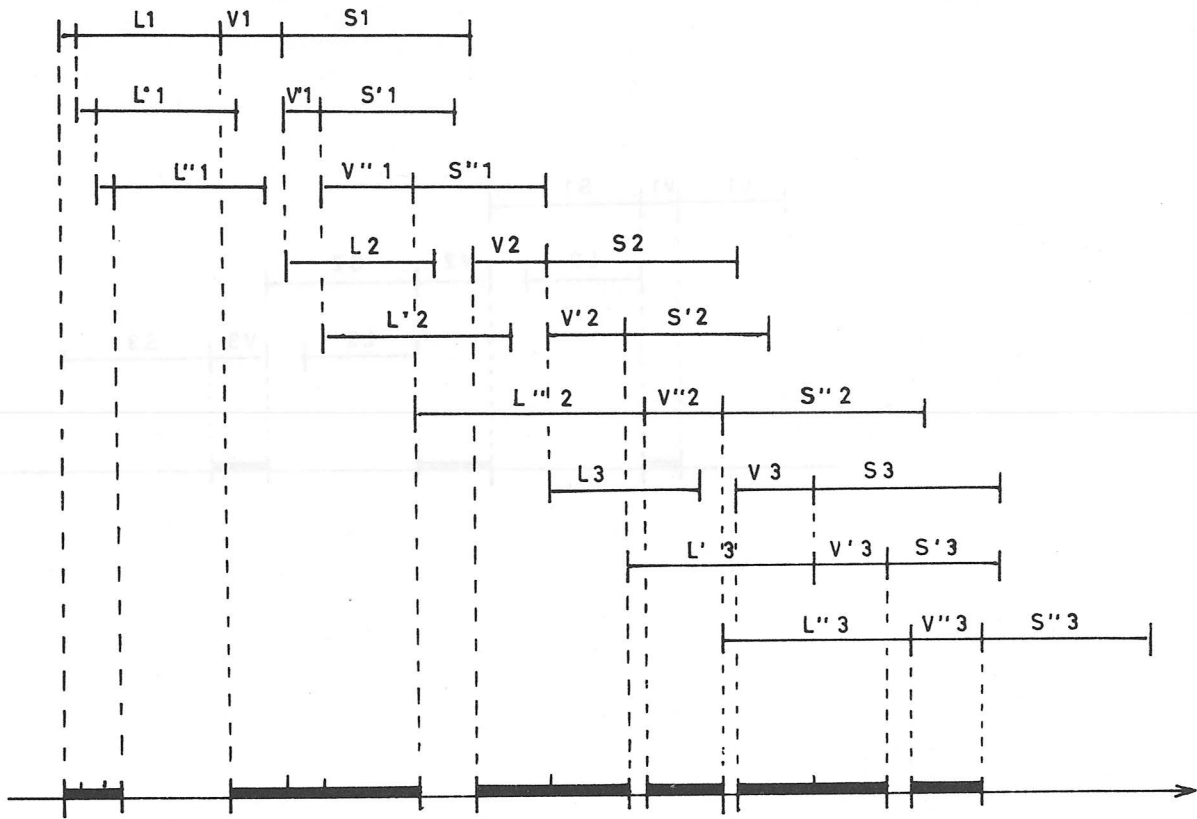
In diesem angenommenen Beispiel ist der Schreibvorgang länger als der Lesevorgang. Die Verarbeitung V 2 kann also nicht vor dem Ende des Schreibens S 1 beginnen, weil sonst die Ausgabezone zerstört würde.

Diese Simultaneität der Ein/Ausgaben entspricht der Hardware des GE-55. Die Gesamtdauer eines Programms ist also gleich der Summe der eigentlichen Verarbeitungszeit plus der Zeit für die längste Ein/Ausgabeoperation.

## 2. Abschnitt

Trotzdem ist man weit davon entfernt, alle Möglichkeiten der Verarbeitung innerhalb der Maschine voll auszuschöpfen. Um in dieser Richtung Fortschritte zu erzielen, öffnet das Multiprogramming einen Weg, der, gut angewendet, es erlaubt, fast das Optimum zu erzielen.

Das Multiprogramming besteht daraus, parallel mehrere Programme, die sich zur gleichen Zeit im Zentralspeicher befinden, auszuführen. Das folgende Diagramm zeigt den Ablauf von drei Programmen P, P' und P'' im Zusammenhang mit Multiprogramming. L 1 bezeichnet die Leseoperation der ersten Daten des Programms P; V3'' die Verarbeitung der dritten Daten des Programms P'' etc.



Anfang der Arbeit

Dieses Schema zeigt, daß bei einer solchen Organisation die drei Programme im ganzen genommen simultan, praktisch in der gleichen Zeit arbeiten, die sie auch allein brauchen würden.

Das Ergebnis ist wie folgt:

- 1) Die Möglichkeiten der Simultanarbeit der Ein/Ausgabeeinheiten werden auf breiter Basis angewendet.
- 2) Die elektronischen Organe werden zu einer Verarbeitung benutzt, die wechselweise den verschiedenen Programmen zur Verfügung steht.

Untersucht man das Diagramm, so ist folgendes festgehalten:

- a) in einem bestimmten Augenblick können die simultan ablaufenden Ein- und Ausgaben verschiedene Programme betreffen.
- b) Die Ein- und Ausgaben eines Programms werden simultan zur Verarbeitung der anderen Programme ausgeführt.
- c) Es erfolgt keinerlei Simultaneität in bezug auf die eigentliche Verarbeitung. Diese erfolgt durch die gleichen Schaltkreise der Zentraleinheit und zwar in der Reihenfolge, in der diese Schaltkreise zur Verfügung stehen. Das hat gewisse Wartezeiten im Ablauf der Programme zur Folge.

Zusammenfassung:

Es ist also letztlich nicht exact, zu sagen, daß die Programme simultan immer in der gleichen Zeit arbeiten, in der sie auch allein laufen würden. Dafür sind aber die Schaltkreise fast ständig in Arbeit. Je näher sie der Vollbelegung kommen, um so weniger treten Wartezeiten auf und die oben gezeigte Formel erhält ihren vollen Wert. Aber in dem Augenblick, in dem die Verarbeitungsmöglichkeiten der Maschine erschöpft sind, ist es evident, daß das Multiprogramming eine Täuschung sein muß. Für eine gegebene Maschine bestimmt das Verhältnis von internen Verarbeitungsgeschwindigkeiten zu der Geschwindigkeit der Ein/Ausgaben die maximale Zahl von Programmen, die man tatsächlich simultan und genau so schnell, als wenn sie allein wären, ablaufen lassen kann. Für den GE-55 ist diese Zahl 5. Das ist der Grund dafür, daß - wie man im Abschnitt über das Multiprogramming sehen wird - die logische Organisation der Maschine die mögliche Zahl der simultan arbeitenden Programme auf 5 begrenzt.

## 4.2. Die Simultaneität der Ein/Ausgaben des GE-55

### 4.2.1. Darstellung

Es ist bekannt, daß mehrere Randeinheiten zur gleichen Zeit im gleichen Ein/Ausgabebefehl (IØC oder IØIC - s.3.6.) unter der Voraussetzung, daß sie an verschiedenen Kanälen angeschlossen sind, angerufen werden können. Sie können aber auch durch mehrere Befehle in verschiedenen Programmen unter dem gleichen Vorbehalt angerufen werden. Diese Möglichkeit wird im Abschnitt über Multiprogramming beschrieben.

Die Aufgabe dieses Kapitels ist es, festzulegen, unter welchen Bedingungen die Randeinheiten arbeiten, wenn die Befehle wie folgt gegeben werden. Vorher müssen jedoch drei Typen von Ein/Ausgabeeinheiten festgelegt werden:

- 1) Normale Randeinheiten, die auf jedem Zeichen halten können, wie Drucker MB 50 und Stanzer PS 40.  
Der Übertrag des Inhalts einer Speicherzone nach einer solchen Einheit kann zu jedem Zeitpunkt unterbrochen und später fortgesetzt werden.
- 2) Normale Randeinheiten, die nicht auf jedem Zeichen halten können, wie der Kartenleser.  
Der Übertrag der Zeichen nach oder von einer solchen Einheit kann nicht unterbrochen werden.
- 3) Schnelle Randeinheiten wie die Trommel und der Drucker I 41.

Der Pufferspeicher, der im gleichen Rythmus mit der Zentraleinheit selbst arbeitet, wird nicht berücksichtigt.

### 4.2.2. Anwendung

Wie bekannt belegt ein Ein/Ausgabebefehl eine bestimmte Zone im Zentralspeicher, die IØC-Zone von Stelle 31 - 59. Infolgedessen steuert der verdrahtete Supervisor der Zentraleinheit aufgrund des Inhalts dieser Zone vollständig den Anschluß der Randeinheiten ohne weiteren Befehl durch das Programm.

#### 4.2.2.1. Die IØC - Zone

Sie besteht aus zwei Teilen:

- 24 Bytes (31 - 54) sind unterteilt in 6 Gruppen zu je 4 Bytes, die "IØC-Register" genannt werden. Jede dieser Gruppen enthält alle notwendigen Angaben zur Steuerung einer Einheit.
- 5 Bytes (55 - 59) werden entsprechend den 5 Programmen belegt, die simultan arbeiten können (Zusammenhang mit Multiprogramming).

Sie enthalten zu jedem Zeitpunkt die Anzahl der Randeinheiten, die für jedes Programm effektiv zur Verfügung stehen.

#### 4.2.2.2. Die IØC - Register

Sie enthalten für jede in Betrieb befindliche Einheit:

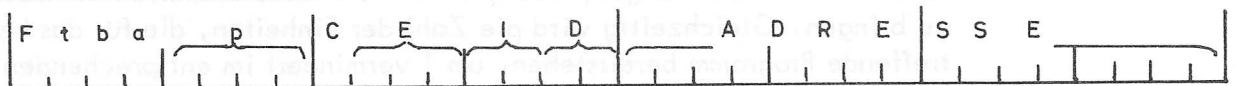
- in den ersten zwei Bytes:

- die Programmnummer für das sie arbeitet,
- die Nummer des Kanals, an den sie angeschlossen ist,
- die Nummer der Einheit an diesem Kanal,
- den Typ der Einheit (langsam, schnell, ohne oder mit Halt auf dem Zeichen),
- die durchzuführende Operation (Lesen, Schreiben etc.);

- in den anderen zwei Bytes:

- die Kernspeicherstelle des nächsten zu übertragenden Zeichens. Am Anfang ist diese Adresse gleich dem Inhalt des im IØC angegebenen Registers.

### IØC - Register



- F = 1 Freie Gruppe
- t = 0 Einheit ist angeschaltet; = 1 Einheit ist abgeschaltet
- b = 0 Einheit kann auf der Stelle halten oder (1) nicht
- a = 0 Einheit ist anzuschliessen; = 1 ist oder war angeschlossen
- p = Nummer des Programms
- C = 0 schnelle und (1) langsame Randeinheit
- E = Nummer der Einheit des Kanals
- I = Nummer des Kanals
- D = Aufgabe (Lesen, Schreiben etc.)

Zu einem gegebenen Zeitpunkt kann man also bis zu 6 Randeinheiten in dieser Zone haben, die simultan arbeiten und durch einen oder mehrere IØC (oder IØIC) Befehle angerufen wurden.

Vorstehendes ist die Beschreibung der Steuerung der Randeinheiten nach Belegung der IØC - Zone.

#### 4.2.3. Ablauf der Organisationen

Man hat drei Möglichkeiten zu unterscheiden:

- Alle Einheiten, die in der IØC - Zone gespeichert sind, sind langsame Einheiten mit Halt auf jedem Zeichen;
- Es sind langsame Einheiten ohne Halt auf jedem Zeichen;
- Es sind schnelle Einheiten.

4.2.3.1. Alle Einheiten sind langsam mit Halt auf jedem Zeichen.

Die Zentraleinheit tastet alle Steuergruppen ab, die effektiv belegt sind. Im Verlauf dieser Abtastung gibt sie einen Übertragungsbefehl für ein Zeichen an alle Einheiten, die, mechanisch gesehen, bereit sind. Am Ende der Abtastung wird ein Programm gestartet, das sich in der Bearbeitungsphase befindet (soweit eins vorhanden ist. Das ist nur in Verbindung mit Multiprogramming möglich) um einen Befehl durchzuführen, bevor eine neue Abtastung erfolgt und evtl. neue Befehle zur Übertragung eines Zeichens gegeben werden. Man sieht also, daß im Zusammenhang mit dem Multiprogramming eine effektive Simultaneität von mehreren Ein/Ausgaben und Verarbeitung erzielt werden kann. Ist eine Bearbeitung etwas länger, kann es sein, daß eine Einheit etwas auf den Befehl zum Übertrag eines Zeichens warten muß. Das hat jedoch keine Bedeutung, weil die Einheit zwischen zwei Zeichen halten kann. Die einzige Folge ist nur ein kurzer Aufenthalt.

Bei jedem von oder nach einer Randeinheit übertragenen Zeichen wird die in den Steuergruppen enthaltene Adresse um 1 weitergerechnet. Am Ende des Übertrags, der durch das Erkennen einer Trennmarke festgestellt wird, wird die Steuergruppe dieser Einheit gelöscht und die IØC - Zone zusammengeschoben, um evtl. andere Einheiten in Arbeit zu bringen. Gleichzeitig wird die Zahl der Einheiten, die für das betreffende Programm bereitstehen, um 1 vermindert im entsprechenden Byte des 2. Teiles der IØC - Zone.

4.2.3.2. Die IØC - Zone enthält Steuergruppen langsamer Randeinheiten ohne Halt auf jedem Zeichen

Die langsamen Randeinheiten, mit oder ohne Halt an jeder Stelle, können wegen der Abtastung der Steuergruppen der IØC - Zone simultan arbeiten. Aber im Gegensatz zu dem vorher Gesagten wird durch die Zentraleinheit kein evtl. in der Bearbeitungsphase befindliches Programm gestartet, weil Einheiten ohne Halt auf jeder Stelle sich in Arbeit befinden. Es erfolgt eine ständige Abtastung der IØC - Zone, um in der erforderlichen Zeit die Befehle zur Übertragung eines Zeichens an diese Randeinheit geben zu können. So könnte z.B. das Lesen einer Karte nicht unterbrochen werden, um einen Befehl abzuwarten, ohne daß Zeichen verlorengehen.

Die unterbrochene Abtastung der IØC - Zone bewirkt also, daß ein Simultanarbeiten in Verbindung mit einer Randeinheit ohne Halt auf dem Zeichen : nicht möglich ist.

Wie im vorigen Fall erfolgt auch hier nach Beendigung eines Übertrags ein Zusammenschieben der IØC - Zone. Es ist übrigens möglich, daß bei dieser Gelegenheit auf den vorhergehenden Fall genau zurückgekommen wird. (Durch die Möglichkeit der Simultaneität der Verarbeitung und der Ein/Ausgaben).



4.2.3.3. Die IØC - Zone enthält Steuergruppen für schnelle Randeinheiten (wie die Magnettrommel oder der Drucker I 41)

Die Natur dieser Randeinheiten erfordert es, daß Überträge nur ohne Unterbrechung zwischen den Zeichen erfolgen können. Weiterhin erlaubt es der Rhythmus des Übertrags der Zentraleinheit nicht, wie in den vorigen Fällen, mehrere Randeinheiten zu steuern. Wenn im Laufe der Abtastung der IØC - Zone die Zentraleinheit gleichzeitig auf folgende drei Bedingungen trifft:

- die IØC - Zone enthält Steuergruppen schneller Einheiten,
- diese schnellen Einheiten sind bereit, Überträge auszuführen,
- keine langsame Einheit ohne Halt auf dem Zeichen ist in Arbeit,

erfolgt der Übertrag in einem Zug. Die Abtastung der IØC - Zone wird unterbrochen. Das bedeutet, daß der Anschluß einer schnellen Randeinheit jede Simultanarbeit ausschließt, sowohl mit anderen Ein/Ausgaben als auch mit Bearbeitungen. Sobald der Übertrag beendet ist, beginnt die Abtastung erneut nach dem Zusammenschieben und man kommt auf einen der drei vorher genannten Fälle zurück.

4.2.3.4. Der Pufferspeicher

Wird im Laufe der Abtastung der IØC - Zone eine Steuergruppe ange-  
troffen, die sich auf den Pufferspeicher bezieht, erfolgt der Übertrag von 6 Zeichen in einem Zug, wie bei einer schnellen Randeinheit. Man kann aber nicht von einer Unterbrechung der Simultaneität sprechen, weil der Übertrag im Rhythmus des Zentralspeichers erfolgt und auf dieser Stufe keinerlei Simultaneität möglich ist.

4.2.4. Simultaneität des I 41

4.2.4.1. Simultaneität des I 41 mit einer langsamen Randeinheit.

Die Schaltkreise des GE-55 erlauben eine Simultaneität des I 41 mit einer einzelnen langsamen Randeinheit, gleich, ob diese Randeinheit auf dem Zeichen hält oder nicht.

4.2.4.2. I 41 und eine Randeinheit mit Halt auf dem Zeichen. Die Einheiten arbeiten voll simultan.

Beispiel: IØC mit I 41 und Stanzer

Die Placierung der Einheiten innerhalb des IØC-Befehls (I 41 - Stanzer oder umgekehrt) ist ohne Einfluß auf die Simultanarbeit.

#### 4.2.4.3. I 41 und eine langsame Randeinheit ohne Halt auf dem Zeichen

Entsprechend dem unter 2.7.1. aufgezeigten Verfahren ist eine Simultanarbeit nur dann möglich, wenn das Register AK im Augenblick der Verbindung des I 41 Null enthält.

Das bedeutet, daß der I 41 zuerst gestartet werden muß, wenn man eine Simultaneität erreichen will.

Beispiele: IØC mit Leser und I 41  
IØC mit I 41 und Leser

In dem ersten Beispiel wird der Leser zuerst gestartet. Das Register enthält dann "1", und die Verbindung des I 41 ist nicht möglich. Er wird erst dann gestartet, wenn das Lesen beendet ist.

In dem zweiten Beispiel ist eine volle Simultaneität gegeben. In dem Augenblick, in dem der I 41 zu verbinden ist, enthält das Register AK Null. Der I 41 wird gestartet. Die Schaltkreise des GE-55 erlauben eine Simultanarbeit des I 41 mit einer langsamen Randeinheit und somit wird auch der Leser gestartet.

#### 4.2.4.4. I 41 und mehrere langsame Randeinheiten

Das kann sich in den folgenden 3 Fällen ergeben:

- ein IØC spricht mehrere Randeinheiten an (maximal 6)
- in der vertikalen Simultaneität
- im Multiprogramming

Grundsätzlich tasten die Schaltkreise des GE-55, wenn der I 41 einmal gestartet worden ist, den ersten Bereich der IØC - Zone ab und ermitteln von links nach rechts die erste langsame Randeinheit, die schon verbunden ist oder noch zu verbinden ist. Das führt zu einer Simultaneität zwischen dem I 41 und dieser langsamen Einheit.

Beispiel: IØC - Stanzer - I 41 - Leser

Der Stanzer wird zuerst verbunden, dann der I 41 (AK = 0), und die erste langsame Randeinheit wird ermittelt. Die erste Einheit, die von links gefunden wird, ist der Stanzer. PS 40 und I 41 arbeiten also simultan. Wenn der I 41 seine Arbeit beendet, kommt es zu einer Simultaneität zwischen Stanzer und Leser. Wenn das Stanzen vor dem Drucken beendet ist, wird der Ausdruck allein weitergeführt, und erst danach wird der Leser verbunden.

Beispiel: (vertikale Simultaneität)  
53 11 AA 09

simultanes Programm  
53 12 1E 08 96 06

Durch den ersten IØC wird das Stanzen gestartet, und das anschließende Verarbeitungsprogramm läuft simultan dazu ab. In dem Fall, daß das Stanzen bei der Aufnahme des doppelten IØC I 41 - Leser noch nicht beendet ist, befinden sich zu diesem Zeitpunkt in der IØC - Zone:

PS 40 - I 41 - Leser

Der Stanzer ist verbunden.  
Das Register AK ist Null.

Der I 41 wird verbunden, und das Abtasten im ersten Bereich der IØC - Zone wird aufgenommen. PS 40 und I 41 arbeiten simultan, und wir kommen zurück auf den vorangegangenen Fall.

Bei Beendigung des Stanzens vor der Aufnahme des IØC I 41 - Leser kommt es zu einer Simultanarbeit I 41 - Leser.

4.2.4.5. Im Multiprogramming ist das Problem im Prinzip genau so wie bei der vertikalen Simultaneität; die einzelnen IØC in der IØC-Zone kommen hier lediglich aus verschiedenen Programmen.

#### 4.3. Multiprogramming beim GE-55

##### 4.3.1. Darstellung

Es ist beim GE-55 möglich, maximal 5 Programme, die sich gleichzeitig im Zentralspeicher befinden, nach den Regeln des Multiprogramming ablaufen zu lassen.

Wie am Anfang dieses Kapitels gezeigt wurde, bestehen diese Programme aus Verarbeitungsserien, die durch Ein/Ausgabebefehle ( $I\phi C$ ) getrennt werden. Trifft eines von ihnen im Laufe der Arbeit einen  $I\phi C$  - Befehl, setzt dieser Befehl eine oder mehrere Randeinheiten nach den im vorigen Abschnitt beschriebenen Regeln in Arbeit. Solange die Ein/Ausgabeoperationen dieses Programms nicht beendet sind, wird mit der Weiterverarbeitung dieses Programms ausgesetzt. Man sagt, daß ein solches Programm "erstarrt" ist. Das ist der Fall, wenn der Inhalt des PAR konstant bleibt.

Ein anderes Programm, das sich in der Verarbeitungsphase befindet, wird also zur Ausführung gebracht. Es läuft simultan mit den Ein/Ausgabeoperationen des "erstarrten" Programms ab, die Befehle schieben sich in die Abtastung der  $I\phi C$  - Zone unter den vorher gesagten Bedingungen ein. Sobald das zweite Programm in seinem Ablauf auf einen Ein/Ausgabebefehl trifft, "erstarrt" es ebenfalls und es wird zu einem weiteren Programm übergewechselt. Kann allerdings der Ein/Ausgabebefehl nicht in die  $I\phi C$  - Zone eingespeichert werden, weil die Zahl der zur Verfügung stehenden Steuergruppen nicht ausreicht, wird der Programmwechsel solange aufgeschoben, bis der Befehl effektiv zur Ausführung gelangt.

Nachdem die Ein/Ausgabeoperationen eines "erstarrten" Programms durchgeführt wurden, kann dieses seinen Ablauf wieder aufnehmen. Aber genau wie zu einem bestimmten Zeitpunkt mehrere Programme "erstarrt" sein können, gibt es auch mehrere Programme, die ablaufen können. Das freiwerdende Programm wird darum in eine Warteschleife eingereiht, um zum Laufen zu kommen, wenn es an der Reihe ist. Man sagt, ein solches Programm ist "gestartet".

Das Multiprogramming des GE-55 besteht also in einer Folge von Abläufen "gestarteter" Programme simultan mit den Ein/Ausgaben "erstarrter" Programme.

Außer den "gestarteten" und "erstarrten" Programmen gibt es noch ein oder mehrere "anhaltende" Programme. Das sind Programme, bei denen keine Operationen erfolgen, sondern die darauf warten, von einem anderen Programm gestartet zu werden. Der Halt eines solchen Programms kann durch das gleiche Programm oder durch die Einrichtung einer "Sicherung" (Schutz) erfolgt sein, der dazu bestimmt ist, zu vermeiden, daß Sequenzen, die unvereinbar mit den verschiedenen Programmen sind, gleichzeitig mit ihnen ablaufen. Das ist besonders der Fall, wenn mehrere

Programme gleiche Zonen im Zentralspeicher oder gleiche Randeinheiten, oder zwei am gleichen Kanal angeschlossene Einheiten benutzen. Es ist bekannt aus Abschnitt 4.2.1., daß zwei Einheiten, die an dem gleichen Kanal angeschlossen sind, nicht gleichzeitig von der Zentraleinheit gesteuert werden können. Das bedeutet also, daß nur I $\Phi$ C - Befehle übernommen werden können, die den gleichen Kanal benutzen, wenn sie einzeln in die I $\Phi$ C - Zone eingehen.

Das Zwischenspiel zwischen den Programmen wird durch Spezialbefehle gesteuert, die in Abschnitt 4.3.3. beschrieben werden.

Anmerkung:

Die Gesamtheit der simultan durchgeführten Programme ergibt eine "Verarbeitungseinheit". Die Programme werden numeriert von 1 - 5. Am Anfang der Arbeit sind alle Programme, außer einem, "angehalten". Nur das Programm 1 ist gestartet und kommt unmittelbar zur Durchführung. Dieses Programm muß die anderen im gewünschten Augenblick in der vom Programmierer gewünschten Reihenfolge starten.

4.3.2. Anwendungsweise

Es handelt sich hierbei im wesentlichen um verschiedene Zonen im Zentralspeicher, die von der Zentraleinheit ausgewertet werden, um den reibungslosen Ablauf der gemeinsam im Speicher befindlichen Programme zu steuern.

4.3.2.1. Die Registerzonen

Von den in der Standardregisterzone (Bytes 85-595) befindlichen Registern ist eine gewisse Zahl mit genau definierten Aufgaben für das gerade ablaufende Programm belegt.

Das sind:

- die spezialisierten Register (85-95), die notwendige Angaben zur Steuerung des ablaufenden Programms enthalten: PAR, Vergleichsergebnisse etc..
- die numerischen Register 0-9, in denen die Basisadressen, speziell die für die Mehrfachüberträge, gespeichert sind.

Bei Multiprogramming spielen diese Stellen die gleiche Rolle für das gerade "in Ausführung" befindliche Programm, wie bei Monoprogramming. Wenn aber in der Folge dieses Programm auf einen I $\Phi$ C - Befehl stößt und "erstarrt" und ein anderes Programm "gestartet" wird, nimmt es den Platz des vorherigen ein. Man muß also sowohl den Inhalt dieser Register fort speichern als auch sie mit den Angaben für dieses neue Programm belegen. Daher braucht man 5 Registerzonen, von denen eine bei Stelle 85 beginnt und die anderen nach Angabe des Programmierers. Jedesmal, wenn ein Programmwechsel erfolgt, wird durch die Zentraleinheit automatisch ein Austausch der zwei entsprechenden Zonen durchgeführt.

Was die numerischen Register betrifft, so können zwei Möglichkeiten vorliegen: Entweder übersteigt die Gesamtzahl aller von den verschiedenen Programmen benutzten Register die Zahl 100 nicht, dann werden die Register einfach auf die einzelnen Programme aufgeteilt, wie auch der Normalspeicherbereich; oder sie übersteigt die Zahl 100, dann muß eine Anzahl Register verschiedenen Programmen dienen und es muß bei jedem Programmwechsel der gleiche Austausch vorgenommen werden, wie auch bei den spezialisierten Registern.

Zusammenfassend dienen also die Registerzonen nicht nur zur Aufnahme der spezialisierten Register, sondern auch für eine gewisse Zahl von numerischen nach Angabe des Programmierers. Diese letzteren bestimmen also die Größe der Registerzonen und es ist klar, daß demnach die Länge der Registerzone die gleiche ist für alle im Speicher befindlichen Programme. Die Registerzone für das in Arbeit befindliche Programm beginnt also an Stelle 85 bis zu einer Stelle, an der sich eine Trennmarke FE befindet, die vom Programmierer gesetzt wird.

Diese Trennmarke begrenzt die Länge des Austausches im Augenblick eines Programmwechsels. Es versteht sich, daß diese Zone so klein wie möglich sein soll, damit die Dauer des Austausches verringert wird.

#### 4.3.2.2. Die Zonen zur Steuerung der Randeinheiten und der Programme

Das Multiprogramming benutzt folgende drei Zonen:

- 1) Die Zone der Ein/Ausgaben (IØC - Zone - Input - Output-Control von Stelle 31 - 59)

Diese Zone wurde bereits in Abschnitt 4.2.2. beschrieben. Sie steuert im wesentlichen die Simultaneität der Ein- und Ausgaben, und zwar sowohl bei Mono - als auch bei Multiprogramming.

Im folgenden soll daher nur noch eine Beschreibung der letzten 5 Stellen (55 - 59) dieser Zone gegeben werden, deren Aufgabe sich bei Multiprogramming zeigt.

Diese Stellen werden in bezug auf die Programme 1 - 5 belegt. Jedes von Ihnen enthält in seinem rechten Halbbyte die Zahl der im Laufe des Programms verwendeten Randeinheiten zum Abzählen durch das Programm, dessen Byte belegt ist.

In dem Augenblick, wo ein IØC - Befehl in einem Programm "P" zum Zuge kommt, wird die Zahl der durch den Befehl angeschalteten Randeinheiten an der Stelle (54 + P) gespeichert, und zwar im rechten Halbbyte. Jedesmal wenn der Rechner die auf eine dieser Einheiten bezogenen Überträge beendet, wird der Inhalt dieses Halbbytes um 1 vermindert. Wenn der Inhalt "0" erreicht, wird das entsprechende Programm "wieder gestartet". (s. Zone PRC unten).



- 2) Die Zone der "Registerzonenadressen" (RFA-Register Field Adresses  
Stelle 60 - 69)

Diese Zone enthält 5 Register zu je 2 Bytes, die entsprechend von rechts nach links für die Programme 1 - 5 belegt werden. Diese Register zeigen in jedem Moment die Adressen an, wohin die besonderen Register jedes Programms übertragen werden.

Nachdem vorher bereits Gesagten, ist eine dieser Adressen 85 und entspricht dem laufenden Programm. Jeder Wechsel der Registerzonen ruft daher die entsprechenden Adressen in der RFA - Zone an.

- 3) Die Zone der "Programmsteuerung" (PRC-Programs control, 70-84)

Diese Zone enthält zwei Teile, die durch eine Trennmarke "00" getrennt werden.

- a) Der linke Teile der PRC - Zone enthält die Warteschleife der "gestarteten" Programme (max.5). Jedesmal wenn ein Programm gestartet wird, empfängt das erste freie Byte von links:

- in seinem rechten Halbbyte die Nummer des Programms,
- in seinem linken Halbbyte den Wert 1, der anzeigt, daß es sich um das letzte "gestartete" Programm handelt. Das linke Halbbyte des vorhergehenden Bytes wird auf Null gesetzt.

Das Programm, dessen Nummer an Stelle 70 gespeichert ist, ist in Arbeit. Wenn dieses Programm auf einen IØC, STØP oder PRØ - Befehl (s.4.3.3.) trifft und ihm in der Warteschleife ein "gestartetes" Programm folgt, erfolgt Programmwechsel. Die Warteschleife wird um eine Stelle nach links versetzt und das Programm, dessen Nummer sich an Stelle 71 befand, kommt zur Ausführung (Stelle 70). Das "erstarrte" oder "angehaltene" Programm verschwindet aus diesem Teil der PRC - Zone. Die Programme werden so in der Reihenfolge, in der sie "gestartet" wurden, zur Ausführung gebracht.

Anmerkung:

Befindet sich ein arbeitendes Programm allein in der Warteschleife in dem Augenblick, in dem es auf einem IØC-Befehl "erstarrt" (Wert 1 im linken Halbbyte der Stelle 70), wird die Versetzung ausgesetzt bis dieses eine Programm erneut "gestartet" wird, das dann wiederum das Programm an Stelle 70 ist. Diese Besonderheit wird verwendet bei Monoprogramming.

- b) Der rechte Teil der PRC - Zone dient der Steuerung des Zusammenwirkens der Programme.

Er nimmt die Zeichen auf, die die Bedingungen dieses Zusammenwirkens darstellen. Diese Zeichen werden nacheinander von rechts nach links in der Reihenfolge, in der sie von den "Halt" oder "Schutz" - Befehlen erstellt werden, eingesetzt (s.4.3.3.)



Wenn man mit "n" das linke Halbbyte und mit "k" das rechte Halbbyte jeder Stelle dieses Teiles der PRC - Zone bezeichnet, werden die Bedingungen nach folgendem Schema dargestellt:

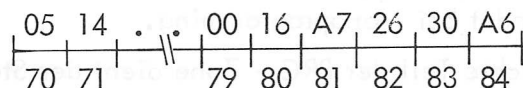
	n	k
Ein Programm "N" wurde durch einen STOP-Befehl angehalten.	Nummer des Programms	0
Ein Programm wurde durch den Schutz "k" geschlossen.	A	Nummer des Schutzes K
Ein Programm "N" wurde durch einen geschlossenen Schutz "k" angehalten	Nummer des Programms N	Nummer des Schutzes K

n = gepackte Dezimalziffer; k = hexadezimaler Zeichen

Die Zentraleinheit untersucht diese Zone durch Abtasten von rechts nach links. Die durch den gleichen Schutz "k" angehaltenen Programme werden in der Reihenfolge, in der sie blockiert wurden, wieder gestartet.

Diese Folge von Zeichen wird links durch eine Trennmarke 00 begrenzt. Die Grenze zwischen dem rechten und linken Teil dieser Zone ist jedoch fließend und zwar in Abhängigkeit von den plötzlich eintretenden Bedingungen des Zusammenspiels der Programme. Es kann sogar vorkommen, daß beide Teile dazu neigen, sich zu überschneiden, da die Gesamtlänge der Zone nicht ausreicht. In diesem Fall setzt der letzte Vorfall, der in dieser Zone ein neues Zeichen erzeugt, es an die Stelle der Trennmarke 00, und die Zentraleinheit zeigt Programmunterbrechung an. Das ist immer ein Zeichen für eine schlechte logische Konzeption in der Organisation der Simultaneität der Programme. Der Programmierer muss eingreifen, um die sich daraus ergebenden Änderungen vorzunehmen.

Beispiel:



- das Programm 5 ist in Arbeit,
- das Programm 4 ist gestartet,
- das Programm 3 wurde angehalten,
- der Schutz 6 und 7 ist geschlossen,
- die Programme 1 und 2 sind durch den Schutz 6 blockiert.

4.3.3. Die Befehle

4.3.3.1. Haltbefehle

STOP

STOP program execution

40

Der Befehl STOP, der in einem arbeitendem Programm mit der Nummer "N" steht, bewirkt den Halt des Programmablaufs und einen Programmwechsel.

Ein Zeichen "n0" wird an der ersten freien Stelle von rechts in der PRC-Zone eingesetzt (n - Nr. des Programms). Der linke Teil der PRC-Zone wird um eine Stelle nach links versetzt, wodurch die Nummer "n", die an Stelle 70 stand, verschwindet.

Anmerkungen:

1. Ein durch den Befehl STOP angehaltenes Programm kann nur durch einen Befehl START, der sich in einem anderen Programm befindet, das später in Aktion tritt, wiedergestartet werden (s.4.3.3.2).
2. Der Befehl STOP dient nicht dazu, ein Programm endgültig anzuhalten.
3. Er kann keine Kapazitätsüberschreitung der PRC-Zone hervorrufen.

4.3.3.2. Starten

START

reSTART a program halted by a stop.

41 N

N = Nummer des Programmes, das gestartet wird (01-05)

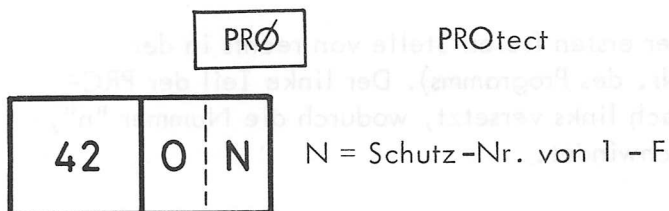
Der Befehl START, der in einem Programm, dessen Nummer ungleich "N" (im Befehl) ist, startet ein Programm "N", das vorher durch den Befehl STOP angehalten wurde.

Dazu wird der rechte Teil der PRC-Zone untersucht, ob das Zeichen "n0" eingespeichert wurde:

- a) wenn Ja : wird dieses Zeichen gelöscht, der rechte Teil der PRC-Zone wird zusammengeschoben und das Zeichen "1n" wird an der ersten freien Stelle von links in diese Zone eingesetzt. Das linke Halbbyte des vorhergehenden Bytes wird auf Null gesetzt.
- b) wenn Nein: ist der Befehl START ohne jede Wirkung.

In jedem Fall geht das laufende Programm zum nächsten Befehl über. Dieser Befehl kann keine Kapazitätsüberschreitung der PRC-Zone hervorrufen.

#### 4.3.3.3. Schutz



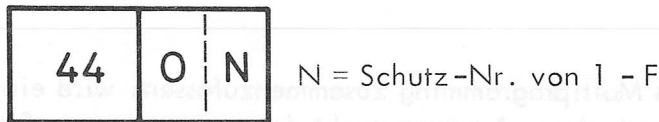
Trifft ein Programm "N" auf diesen Befehl, wird der rechte Teil der PRC-Zone untersucht, ob ein dem Befehl entsprechendes Zeichen "Ak" eingespeichert wurde.

- a) bei Nein: ist dieser Schutz offen .
  - das Zeichen "Ak" wird an die erste freie Stelle rechts in der PRC-Zone eingesetzt (k = Nummer des im Befehl genannten Schutzes).
  - das Programm "N" geht zum nächsten Befehl über.
- b) bei Ja : ist dieser Schutz geschlossen .
  - das Zeichen "nk" wird an die erste freie Stelle von rechts in die PRC-Zone eingesetzt (n = Nummer des Programms an Stelle 70; k = Nummer des im Befehl gegebenen Schutzes).
  - das Programm "N" wird angehalten : der linke Teil der PRC-Zone wird um eine Stelle nach links versetzt, wodurch die Nummer "n" an Stelle 70 verschwindet und ein Wechsel des Programms erfolgt.

Anmerkungen:

1. Ein durch den Befehl PRØ angehaltenes Programm kann nur durch den Befehl FREE gestartet werden, der sich in einem anderen Programm befindet, das später zur Ausführung gelangt (s.4.3.3.4).
2. Der Befehl PRØ kann im Fall (a) eine Kapazitätsüberschreitung der PRC-Zone und damit eine Programmunterbrechung nach sich ziehen.

4.3.3.4. Freimachung



Trifft ein Programm einen Befehl FREE, wird der rechte Teil der PRC-Zone untersucht, ob ein Zeichen "Ak" und ein oder mehrere Zeichen "nk", die dem angegebenen Schutz entsprechen, vorhanden sind. Drei Fälle können auftreten:

- a) Keines dieser Zeichen ist vorhanden: Der Schutz mit der Nummer "k" ist offen. Der Befehl FREE hat keine Auswirkung.
- b) Nur das Zeichen "Ak" befindet sich in der Zone : Der Schutz "K" ist geschlossen, aber es wird durch ihn kein Programm blockiert. Das Zeichen "Ak" wird gelöscht, wodurch der Schutz "K" geöffnet wird.
- c) Das Zeichen "Ak" und ein oder mehrere Zeichen "nk" befinden sich in der Zone: Der Schutz "K" ist geschlossen und blockiert ein oder mehrere Programme.
  - das erste Zeichen "nk" (k = im Befehl angegebene Nummer), das von rechts an angetroffen wird, wird gelöscht und das Zeichen "ln" (n = Nummer, die sich im gelöschten Zeichen "nk" befand) wird an die erste freie Stelle von links in die PRC-Zone eingesetzt (das linke Halbbyte der vorhergehenden Stelle wird auf Null gesetzt). Das drückt sich durch den Start des Programms "N" aus.
  - Das Zeichen "Ak" und die anderen evtl. Zeichen "nk" bleiben erhalten.

In allen Fällen geht das Programm zum nächsten Befehl über.

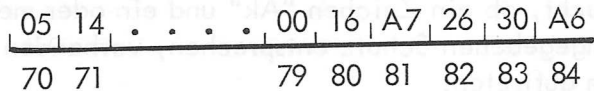
Anmerkung:

Die Fälle (b) und (c) ziehen ein Verschieben des rechten Teils der PRC-Zone nach sich. Der Befehl ruft keine Kapazitätsüberschreitung hervor.

Die beiden Befehle PRØ und FREE arbeiten analog dem Blocksystem der Eisenbahn. Der Befehl PRØ erzeugt Zeichen "Ak" oder "nk" (rotes Licht) bei jeder Ausführung und der Befehl FREE löscht jedesmal nur ein Zeichen (grünes Licht). Das bedeutet, daß jede Überschreitung eines PRØ, der einen Schutz "K" betrifft ein Übergang auf einen FREE entspricht, der den gleichen Schutz betrifft, wenn man nicht eine allgemeine Blockade des Programms riskieren will.

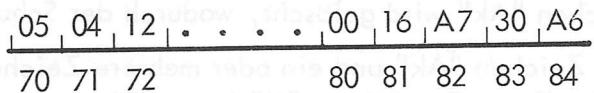
4.3.4. Beispiele

Um die Funktion des Multiprogramming zusammenzufassen, wird ein aussagefähiges Beispiel gegeben. Ausgangspunkt der angenommenen Situation ist das Ende des Abschnitts 4.3.2. Die PRC-Zone enthält:



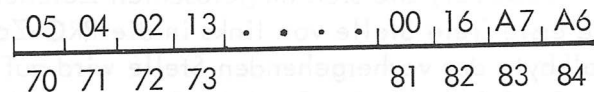
1. Das Programm 5 trifft den Befehl FREE 6

Das erste durch den Schutz 6 angehaltene Programm (Programm 2) wird gestartet.



2. Das Programm 5 trifft den Befehl START 3

Das Programm 3 wird gestartet.



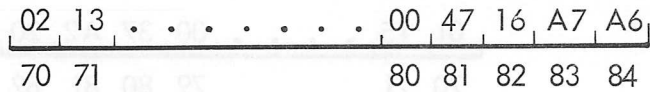
3. Das Programm 5 trifft einen IØC-Befehl

Es erstarrt während der Durchführung dieses Befehls. Das Programm 4 wird zur Ausführung gebracht.



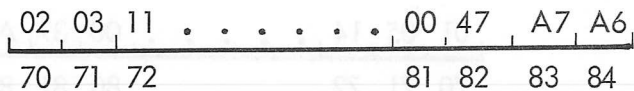
4. Das Programm 4 trifft einen Befehl PRØ 7

Der Schutz 7 ist geschlossen. Man geht zu Programm 2 über.



5. Das Programm 2 trifft einen Befehl FREE 6

Das durch den Schutz 6 blockierte Programm1 wird gestartet.



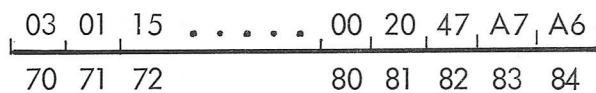
6. Der IØC-Befehl des Programms 5 ist beendet.

Das Programm 5 wird wiedergestartet.



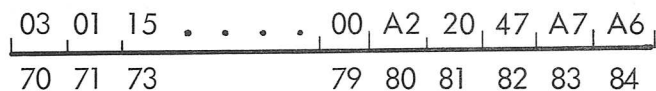
7. Das Programm 2 trifft einen Befehl STØP

Es wird angehalten und Programm 3 fährt fort.



8. Das Programm 3 trifft einen Befehl PRØ 2

Der Schutz 2 war "offen", es wird fortgefahren, nachdem er geschlossen wurde.

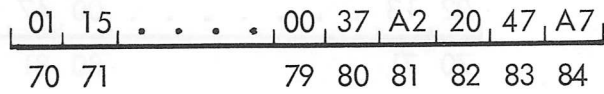


9. Das Programm 3 trifft einen Befehl FREE 6



10. Das Programm 3 trifft einen Befehl PRØ 7

Es wird blockiert. Es wird mit Programm 1 fortgefahren.



11. Das Programm 1 trifft einen Befehl FREE 7

Das Programm 4 wird entblockiert und gestartet.



Diese Folge von Vorfällen in einem aussagefähigen Beispiel erläutert alle Fälle, die bei Anwendung des Multiprogramming beim GAMMA 55 auftreten können.



5. Die Bedienung des GE-55

5.1. Einführung

Um den GE-55 in Betrieb zu nehmen, ist eine Reihe von Programmen erforderlich, die in den verschiedenen Phasen zur Anwendung kommen. Diese Programme bilden "Systeme", die in den folgenden Abschnitten beschrieben werden sollen. Außerdem werden die Richtlinien zur Erstellung und Anwendung der Arbeitsprogramme gegeben.

5.2. Die Software des GE-55

Die Methoden zur Benutzung eines GE-55 richten sich nach den angeschlossenen Randeinheiten. Man unterscheidet darum zwei Methoden, je nachdem, ob der GE-55 mit oder ohne Hilfsspeicher (Magnettrommel) ausgestattet ist.

Außerdem unterscheidet man noch zwei Elemente, ohne die der Rechner nicht arbeiten kann:

- die Programme, die seine Arbeit steuern;
- die Dateien, die verarbeitet werden sollen.

5.2.1. Lochkartenanlage GE-55 (ohne Trommel)

Bei dieser Konfiguration werden die durchzuführenden Programme und zu verarbeitenden Dateien obligatorisch durch Karten in den Kernspeicher eingegeben. Daher rührt auch der Name dieser Art der Konfiguration.

Um eine Arbeit zu beginnen, müssen die Parameterkarten und anschließend die Datenkarten in den Kartenleser eingelegt werden. Das Programm wird nun geladen, gestartet und die Daten in der Reihenfolge verarbeitet, wie sie gelesen werden. Um mehrere Arbeiten aneinanderzureihen, gibt man in der gleichen Art die entsprechenden Programme und Dateien ein.

Während die Dateien, die Karte für Karte gelesen werden, einen unbegrenzten Umfang haben können, ist der Umfang eines Programmes, das in Zentralspeicher gespeichert wird, begrenzt durch die Kapazität dieses Speichers. Das gleiche gilt für bestimmte Daten, Tabellen etc. die am Anfang der Arbeit eingespeichert werden, und auf die man ständig Zugriff hat.

### 5.2.2. GE-55 mit Trommel

Die Magnettrommel bietet eine zusätzliche Speicherkapazität, die bei Anschluß von zwei Trommeln verdoppelt werden kann. Es ist durchaus möglich, Programme auf die Trommel zu übertragen, anstatt sie im Kernspeicher zu speichern und jeweils nur den Teil in den Zentralspeicher abzurufen, der gerade benötigt wird.

Man kann also ein Programm in mehrere "Segmente" unterteilen, die nacheinander in derselben Zone des Zentralspeichers stehen, und zwar in der erforderlichen Reihenfolge. Diesen Vorgang nennt man "Segmentierung".

Das ist aber nicht der ganze Vorteil der Trommel. So kann man gewisse Hilfsoperationen übergehen und ihre Durchführung erleichtern, indem man sie am Anfang einer Arbeitsperiode, z.B. täglich, auf der Trommel speichert gemeinsam mit den durchzuführenden Programmen und den permanenten Dateien einschließlich der Dateien mit Direktzugriff, die sehr wichtig sein können. Man erreicht so eine Beschleunigung der Ladezeit für Programme und Daten, die man mehrmals innerhalb einer Arbeitsperiode benötigt.

Um diese Arbeiten durchzuführen, ist es nur noch erforderlich, die durchzuführenden Programme dem Rechner anzuzeigen und sie in den Zentralspeicher zu übertragen, wo sie dann zur Ausführung gelangen. Der Bedienungskraft obliegt es also nur noch, neue Werte einzugeben und die Resultate in Empfang zu nehmen.

#### Zusammenfassung:

Aufgrund dieser beiden schematisch dargestellten Anwendungsmethoden muß auch die Software den Besonderheiten einer jeden von ihnen Rechnung tragen, um die Forderungen des Benutzers zu erfüllen. Sie muß daher:

- sich nach den beiden Konfigurationen der Maschine richten;
- bei jeder Konfiguration eine Hilfe bieten, die es ermöglicht, jede Stufe erreichen zu können und dadurch den Programmierer zu entlasten.

#### 5.2.2.1. Die verschiedenen Systeme

Den zwei oben beschriebenen Konfigurationen entsprechen auch zwei Arten der Software, das "Kartensystem" und das "Trommelsystem". Von der Funktion her gesehen unterteilen sich diese wiederum in zwei Kategorien:

- Die "Programmiersysteme" dienen zur Vereinfachung der Erstellung von spezifischen Programmen, und zwar vom Schreiben des Programms ab bis zur endgültigen Fertigstellung in Maschinensprache.
- Die "Steuerungssysteme" steuern entweder die Durchführung der Programme in Maschinensprache oder - im Rahmen der zusätzlichen Speicher - die Organisation und die Auswertung der Dateien nach den vorher festgelegten Methoden.

Diese Systeme werden außerdem noch durch eine Bibliothek von Hilfs- und Standardprogrammen ergänzt, die dem Programmierer das Schreiben von Hilfsprogrammen zur Steuerung der Programme, der Dateien und deren Auswertung abnehmen.

Die Karten- und Trommelsysteme haben gemeinsame Berührungspunkte, die im folgenden zusammen beschrieben werden. Abweichungen werden jeweils angezeigt.

#### 5.2.2.2. Die Programmiersysteme

Die Programmiersysteme bieten dem Benutzer

1. die Programmiersprachen, und zwar die Grundsprachen und die Umwandlungssprache für die Trommel (siehe Kapitel 3). Diese beiden Sprachen sind leichter anzuwenden als die Maschinensprache, bieten eine größere Sicherheit und enthalten Makrobefehle.
2. ein Übersetz- und Kontrollsystem, daß
  - die in der Grundsprache abgefaßten Programme übersetzt,
  - Aufbau und Wahrscheinlichkeit der kodierten Befehle prüft,
  - die Programme in die Maschinensprache bringt und sie listet.

Beim GE-55 mit Trommel kommt zu diesem Übersetz- und Kontrollsystem noch ein Umwandlungssystem hinzu, das

- die zusätzliche Trommelsprache übersetzt,
- die verschiedenen Teile des Source-Programms (Sektionen, Unterprogramme usw.) umwandelt, indem es ihnen die effektiven Adressen im Speicher und auf der Trommel zuordnet und die notwendigen Verbindungen herstellt,
- die erhaltenen Programme in Segmente für die Verarbeitung aufteilt,
- die für die Programme erforderlichen Felder im Kernspeicher und auf der Trommel reserviert.

3. ein Listprogramm, das
  - beim Testen von Programmen Programmlisten erstellt, um Fehlersuche und -korrekturen zu erleichtern;
  - eine endgültige Programmliste nach dem Test, die zu den Programmunterlagen gehört, erstellt;
  - das Erstellen von Parameterkarten für das Programm "Verdichten" ermöglicht.
4. ein Programm "Verdichten", das die getesteten Programme "verdichtet", um die Anzahl von Programmkarten zu verringern und so die Ladezeiten jeder Arbeit zu optimieren.

#### 5.2.2.3. Die Betriebssysteme

Dem Benutzer stehen zur Verfügung:

Zum Fahren der Programme

Bei einer GE-55 Kartenanlage

- ein Ladeprogramm Kernspeicher, das
  - . die Verbindungszonen des verdrahteten Supervisors initialisiert (diese Zonen befinden sich am Anfang des Speichers) und den Kernspeicher evtl. ab Byte 161 löscht,
  - . das Programm "Unterbrechung" am Ende des Kernspeichers lädt,
  - . entweder das zu testende Source-Programm oder das auszuführende Teilprogramm in verdichteter Form in den Kernspeicher lädt,,
  - . bei einem Source-Programm die symbolischen Adressen in absolute Adressen umwandelt (siehe 3.2.),
  - . das Source-Programm oder Teilprogramm startet an der Anfangsadresse, die in einer Befehlskarte angegeben ist.

bei einem GE-55 mit Trommel

- ein Monitor-Programm, das von der Trommel abgerufen wird und
  - . das Programm "Supervisor TB" in den Kernspeicher lädt,
  - . die Verbindungszone (erweitert) des verdrahteten Supervisors und des Supervisors TB initialisiert,
  - . entweder das zu testende Programm oder das auszuführende Teilprogramm in den Kernspeicher lädt aufgrund des Kernspeicherladeprogramms, das zu einem Programm-Modul des Monitor-Programms wird.
- das Programm startet.

Anmerkung:

Das Kernspeicher-Ladeprogramm und der Monitor sorgen für die Betriebsfunktionen zwischen zwei Arbeiten, um sie aufeinander folgen zu lassen. Zu diesem Zweck werden sie in den Speicher geladen, sobald eine Arbeit beendet ist, und verschwinden, wenn die nächste Arbeit fertig zur Ausführung ist.

Bei einer GE-55 Kartenanlage

- ein "Unterbrechungsprogramm" P.G.I., das dem Bediener Standardverfahren zur Behebung von Störungen oder zum Informationsaustausch mit dem ablaufenden Programm liefert.

Dieses Programm wird vom Bediener wie unter 2.1.6. beschrieben in Betrieb gebracht. Es kann aufgrund angegebener Anweisungen folgende Funktionen ausführen:

- . das Lochen von Informationen wiederholen, die sich noch in der Stanzzone befinden,
- . eine Karte noch einmal lesen,
- . über die numerische Tastatur eine verschlüsselte Nachricht für das laufende Programm eingeben,
- . ein spezifisches Wiederanlaufprogramm zu starten, das evtl. unter mehreren solcher Programme ausgewählt wird,
- . vor allem das laufende Programm dort zum Wiederanlauf bringen, wo es unterbrochen worden ist.

Das P.G.I. kann ebenfalls auf der Sichtanzeige eine verschlüsselte Nachricht vom laufenden Programm anzeigen.

GE-55 mit Trommel

- ein Programm "Supervisor TB" (TB-Trommel)
  - . verarbeitet die Eingabe und Ausgabe der Trommel physikalisch, d.h. steuert (kontrolliert) die Lese/Schreiboperationen,
  - . verarbeitet Störungen auf den Randeinheiten und Programmunterbrechungen mittels eines P.G.I.-Moduls,
  - . verarbeitet die Anrufe von Segmenten eines gleichen Programms: Suche auf der Trommel, Laden in den Kernspeicher.
  - . Ruft schließlich das Monitor-Programm in den Kernspeicher, wenn eine Arbeit beendet ist, um die nächste anschließen zu können.

Anmerkung:

Das P.G.I. und der Subvisor TB befinden sich während des Arbeitsablaufes im Kernspeicher, um den Arbeitsablauf zu steuern.

GE-55 mit Trommel

- ein Ladeprogramm Trommel, das das oder die auszuführenden Programme in verdichteter Form auf die Trommel lädt und evtl. Trommelzonen für die Dateien reserviert. Diese Zonen werden anschließend vom Supervisor TB verarbeitet.

Zur Datenverarbeitung auf der Trommel

1. ein Programm "Dateienspeicherung", das die Dateien auf die Trommel speichert. Es gibt zwei Speicherungsverfahren:
  - sequentiell verkettet, d.h. jeder Satz auf der Trommel enthält die Adresse des nächstfolgenden Satzes,
  - indiziert sequentiell, d.h. die Sätze werden der Reihe nach nach ihren Kennzeichen gespeichert und können anhand einer Tabelle gesucht werden.
2. ein Programm "Dateifortschreibung", das es ermöglicht, in einer auf der Trommel gespeicherten Datei Sätze zu löschen, zu ersetzen oder hinzuzufügen. Die Datei kann wie oben angegeben gespeichert sein.
3. ein Programm "Eingabe und Ausgabe ausgewählter Informationen" das es ermöglicht, Teile einer Datei oder Sätze, die sich auf der Trommel befinden, in Karten zu lochen oder auszudrucken, um sie extern zu bearbeiten oder aus Sicherheitsgründen später wieder einzugeben.

5.2.3. Bibliothek der Service- und Standardverarbeitungsprogramme

Die Service-Programme, die die Programmier- und Betriebssysteme vervollständigen, sind in den vorangehenden Punkten beschrieben worden. Folgende Programme können noch dazugezählt werden:

- Erstellen Kartendateien
- Standardprogramm doppelte, Listen, Lochen
- Listen
- Bearbeitung von Kartendateien

Bei den Standard-Bearbeitungsprogrammen handelt es sich im wesentlichen um spezifische Unterprogramme:

- Division
- Kennzeichenkontrolle
- Verarbeitung von Überweisungen (franz. System)
- Verarbeitung von Daten (Datum)
- Verarbeitung von Währungsarten
- Lohnsteuer-Kirchensteuer
- Sozialversicherung
- Überlöcher ausfiltern
- Überlöcher einsetzen
- Daten auf Trommel
- Suchen Spur-Nummer
- Suchen Artikel auf Spur
- Rückübertragen Artikel auf Spur

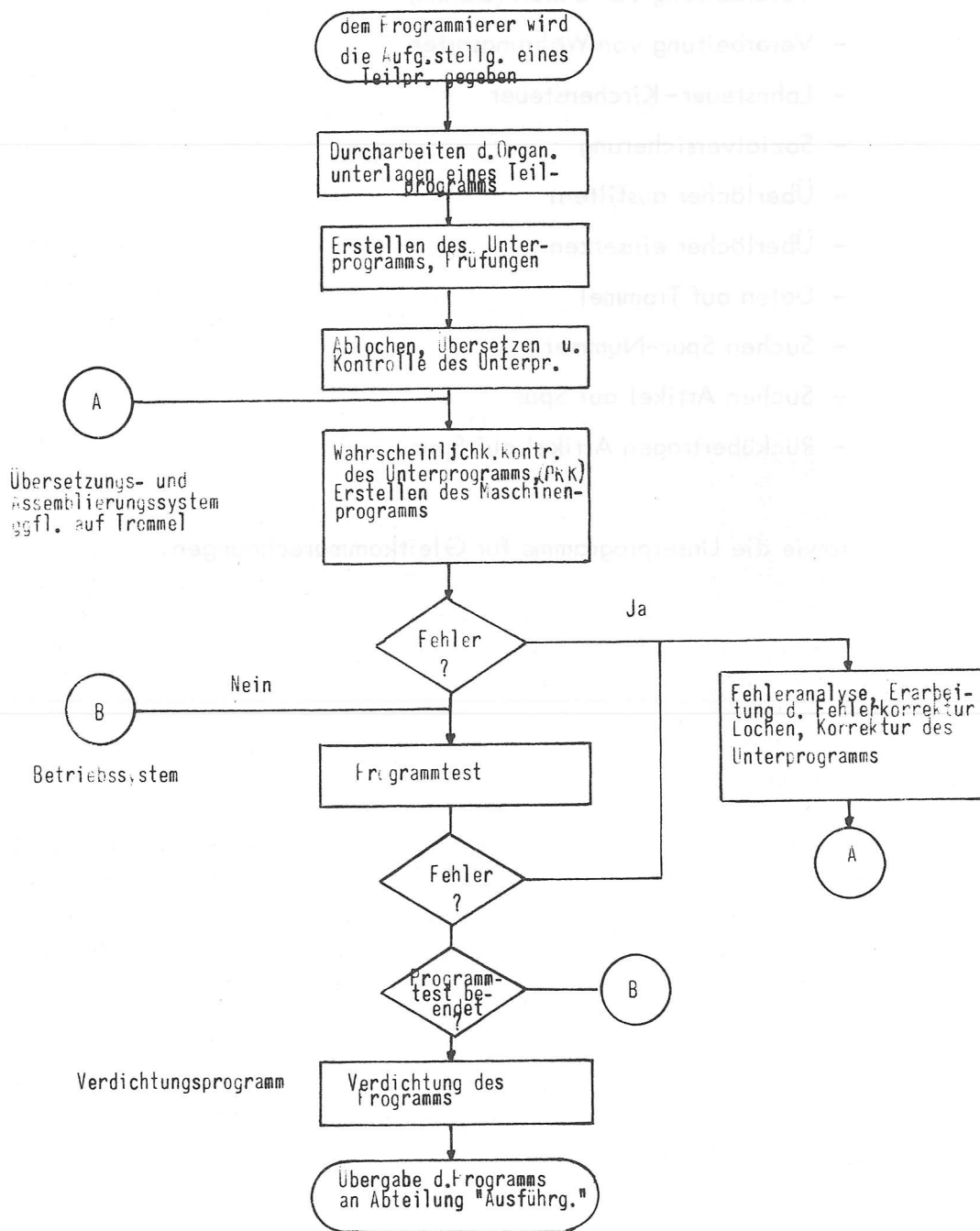
sowie die Unterprogramme für Gleitkommarechnungen.



5.3. Erstellen eines Programms

5.3.1. Verfahren

Die verschiedenen Operationen eines funktionsfähigen Programms gehen folgendermaßen ineinander über:



### 5.3.2. Erstellen der Unterprogramme

#### 5.3.2.1. Untersuchung und Organisation eines Teilprogramms

Vor dem Erstellen eines Programms untersucht der Programmierer die Spezifikationen, die in der Aufgabenstellung des jeweiligen Teilprogramms enthalten sind. Diese Spezifikationen beziehen sich auf die zu verarbeitenden Daten, die erwarteten Ergebnisse und die auszuführenden Verarbeitungen. Die Spezifikationen stellen den ersten Teil der Programmunterlagen dar.

Der Programmierer untersucht dann die Organisation dieser Verarbeitungen auf der Maschine und erstellt dabei folgende Programmunterlagen:

- Kernspeicherbelegung
- In diesem Schema sind die zu reservierenden Speicherzonen (Ein-/Ausgabe; Arbeitsfelder usw.), die Zuordnung der numerischen Register, d.h. also die Kernspeicherbelegung der Daten und Ergebnisse und die verbleibende Kapazität für das Programm angegeben.
- Ein detailliertes Flußdiagramm.

Dieses Flußdiagramm ermöglicht:

- Das Programm evtl. in mehrere Teilprogramme zu unterteilen, die in Multiprogramming laufen, und diese Programme in Sektionen aufzuteilen, die mit symbolischen Kennzeichen versehen sind. Diese Symbole werden vom Programmierer sofort festgelegt.
- Die Sektionen zu definieren, die als Unterprogramme aufgebaut werden können.
- Im Rahmen der Trommelprogrammierung diese Sektionen herauszulösen, die unabhängige Segmente darstellen können (z.B. die zu Beginn des Programms benutzten Sektionen können nach ihrer Ausführung durch Wiederholungssektionen im Kernspeicher überlagert werden), oder solche Sektionen, die über zwei Segmente gelegt werden können, die im Kernspeicher überlagert werden können.

Vor der Erstellung dieser Unterlagen hat der Programmierer die Normen und Einsatzbedingungen des betreffenden Betriebssystems zur Kenntnis genommen, um:

- die erforderliche Speicherkapazität für dieses System zu berücksichtigen (PGI. Supervisor, TB),
- die Standard-Zuordnung für Ein-/Ausgabefelder und Register zu berücksichtigen,
- zu berücksichtigen, welche Initialisationen beim Laden des Programms in den Kernspeicher vorgenommen werden, die nicht programmiert werden (Initialisation von Registern und Ein-/Ausgabefeldern), und diejenigen, die er vorsehen muß.
- Die Standard-Wiederanlaufverfahren nach Fehlern, um vor allem die spezifischen Wiederanlaufverfahren zu verkürzen, die aufgrund der Art der Verarbeitung und der Dauer programmiert werden müssen.
- Das Ladeverfahren von Segmenten, die von der Trommel kommen, um die bestmögliche Aufteilung festzulegen unter Berücksichtigung des Verhältnisses verfügbare Kernspeicherkapazität / voraussichtliche Ladezeit.
- Die Aufbaunormen dieser Segmente.

Der Programmierer muß außerdem die Standard-Unterprogramme kennen, um sie sinnvoll einzusetzen, und deren Einsatzbedingungen, die reservierten Symbole usw. kennen.

#### Symbolische Kennzeichnung der Sektionen

In der Basis-Programmiersprache stehen dem Programmierer 240 verschiedene Kennzeichnungen zur Verfügung, von denen er ggfl. die für die benutzten Software-Programme reservierten Bezeichnungen abziehen muß.

Was die Anwendung der Kennzeichnung betrifft, ist es gemäß den besonderen Kriterien in jedem Programm vorzuziehen, einheitliche Normen für alle Programme festzulegen, und die symbolischen Kennbezeichnungen anzusehen als ein praktisches Mittel zur Untersuchung und zur Verbindung zwischen dem Flußdiagramm und dem zugehörigen Programm. Zu diesem Zweck ordnet man die Kennzeichnung aufsteigend in der gleichen Ordnung wie im Flußdiagramm. Die Kennzeichnung wird aus zwei Zeichen gebildet: Das erste kann zum Beispiel die Seitennummer des Flußdiagramms sein und das zweite die laufende Nummer der Sektion auf der Seite. Auf diese Weise kann man 15 Seiten und 16 Sektionen pro Programm unterscheiden.

### 5.3.2.2. Ausarbeitung des Programms

Ist das Flußdiagramm detailliert aufgestellt, kann der Programmierer die Ausarbeitung des Programms (oder der Programme) in Angriff nehmen. Dazu benutzt er Programmformulare, die sowohl für die Basissprache als auch für die Assembler-Sprache in der Trommel gültig sind. Die ausgefüllten Programmformulare dienen als schriftliche Grundlage für die Ablochung der Programme. Jede Zeile entspricht einer Karte von 80 Spalten.

#### Das Programmformular

In 8.4. wird ein Beispiel für ein Programmformular gezeigt. Bei der Programmierung ist dem Aufbau dieses Formulars genau zu entsprechen. Die einzelnen Angaben sind nur in die zugehörigen Spalten einzusetzen.

#### Die Rubriken

Der Programmierer hat die Möglichkeit, in die einzelnen Zeilen neben den Befehlen zusätzliche Bemerkungen einzusetzen, um den Verlauf des Programms zu verdeutlichen.

#### Rubrik Kartenart (Spalte 2 und 3).

Diese Kartenart gibt an, wie der in der Zeile enthaltene Befehl verarbeitet werden muß.

Das erste Zeichen (Spalte 2) zeigt an, wo der Befehl gespeichert werden muß.

#### Beispiele:

Der Schlüssel 1 zeigt an, daß die Befehle der Reihe nach von der Anfangsadresse an gespeichert werden. Die Adresse wird durch den Pseudo-Befehl BEGIN definiert (Elementarbefehle der Basissprache etc.).

Der Schlüssel 3 zeigt an, daß der Befehl an einer absoluten Adresse zu speichern ist. Diese wird vom Programmierer angegeben.

Das zweite Zeichen (Spalte 3) gibt Art und Größe des Befehls an.

Beispiele:

Der Schlüssel 1 gibt an, daß der Befehl in entpackter Form anzuordnen ist, d.h. daß jedes Zeichen ein ganzes Byte belegt (alphanumerische Konstante).

Der Schlüssel 2 gibt an, daß der Befehl in gepackter Form gespeichert wird, d.h. daß jedes Zeichen ein Halbbyte belegt (Befehlsparameter, hexadezimale Konstanten).

Rubrik Zahl der Bytes (Spalte 4 und 5)

Der Programmierer gibt in dieser Rubrik die Anzahl der Bytes an, die durch die Parameter der Befehle ab Spalte 17 belegt werden (Befehle, Konstanten).

Eventuelle Kommentare sind in dieser Anzahl nicht enthalten.

Diese Zahl begründet einen der Parameter der Pseudo-Befehle der Konstantenerstellung (siehe 3.9.2.).

Rubrik Ordnungsnummer (Spalte 6 bis 9)

Diese Nummer setzt sich aus 4 Dezimalstellen zusammen und legt die Reihenfolge der Programmkarten fest.

Während der Anfangsausarbeitung eines Programms ist es empfehlenswert, die äußerste rechte Ziffer nicht zu benutzen und ihr systematisch den Wert 0 zu geben. Dieses erleichtert die späteren Modifikationen des Programms und erlaubt das Einsetzen von 1 bis 9 Zeilen zwischen zwei ursprünglich auf dem Formular vorhandenen Zeilen. Es ist nicht erforderlich, die bereits vorhandenen Nummern zu ändern; die neuen Linien werden dann beziffert von xxx1 bis xxx9.

Bemerkung

Der Programmierer beziffert nicht die Pseudo-Befehle der Konstantenerstellung, die sich in einer beliebigen Reihenfolge befinden können. Er vermerkt vielmehr in den Spalten 6 bis 9 einen der Parameter dieser Pseudo-Befehle, nämlich die Adresse der durch die spezifischen Konstanten belegten Zone in der betreffenden Linie.

Der Befehl

Rubrik Symbolischer Operationstyp. . (Spalte 10 bis 16)

Die Buchstaben und Ziffern, die zum symbolischen Operationstyp zusammengefügt sind, werden von Spalte 10 an ohne Zwischenraum geschrieben. Die rechts nicht gebrauchten Spalten bleiben frei.

#### Rubrik Befehl (Spalte 17 bis 44)

Die Parameter werden von Spalte 17 an nacheinander ohne Zwischenraum geschrieben auf der Grundlage eines alphanumerischen oder hexadezimalen Zeichens pro Spalte. Die rechts unbenutzten Spalten bleiben frei. Beim Schreiben der Parameter sind die Formvorschriften und Kodierungsregeln, die in den Kapiteln 3 und 4 angegeben sind, genau einzuhalten.

Was indessen die Grundbefehle betrifft, sind die Spalten 17 und 18 reserviert. Der Programmierer kann hier dem hexadezimalen Operationstyp angeben.

#### Die Kommentare (Spalten 34 oder 46 bis 80)

Die Kommentare sind Informationen in Klartext, um die Sektionen, Sequenzen oder Befehle zu erläutern.

Das Programmiersystem führt keine Bearbeitung des Kommentartextes durch. Die Kommentare werden nur auf den Programmlisten wiedergegeben.

Der Programmierer kann sowohl Befehlslinien durch Kommentare vervollständigen als auch zwischen den Befehlslinien besondere Kommentarzeilen einfügen. Im ersten Falle können die Kommentare in Spalte 46, im zweiten Falle in Spalte 34 begonnen werden.

Der Programmierer kann zu diesem Zweck alle druckbaren Zeichen gebrauchen und so viele Abstände hinzufügen, wie er benötigt. Die Kommentare können eine beliebige Stelle innerhalb der zugeordneten Zonen belegen.

#### Bemerkungen:

Wird ein Kommentar in eine Zeile gesetzt, in der auch ein Befehl enthalten ist, muß die Spalte 45 frei bleiben. In einer Zeile, die nur einen Kommentar enthält, müssen die Spalten 10 bis 33 frei bleiben.

Kommentare können nach den absoluten spezifischen Werten eines Pseudo-Befehls der Konstantenerstellung geschrieben werden. Es ist indessen dringend erforderlich, sie durch Abstände zu trennen.

#### Kennzeichen (Spalte x bis 80)

Es ist empfehlenswert, die letzten Spalten jeder Linie zu reservieren, um hier das Kennzeichen des Programms zu vermerken (Nummer, Code, etc.). Dieses Kennzeichen wird in alle Karten eines bestimmten Programms wiederholt.

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.

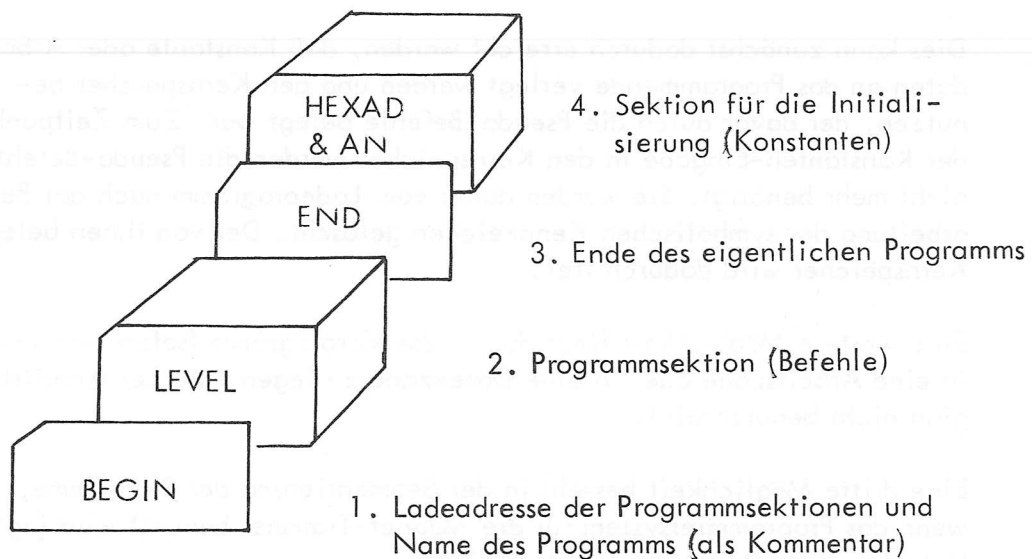
### 5.3.3. Erstellen und Test von Maschinenprogrammen

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

Zeichen	Zwischen code	Lochcode			ISO - code = HEX	Zeichen	Zwischen code	Lochcode			ISO - code = HEX
⌘	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

A

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	
SRD	1.0 + 0.9 V	
R <del>U</del> ND	5.0	
	V = Zahl der Versetzungen	

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)	
$\emptyset$ C $\emptyset$ IC	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.
IOC	Ein/Ausgabebefehl	3.6.6.
IOIC	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> 3.9.13
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> 3.9.14
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.

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		
Zeileuschaltung 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
00	1	NØP	3.2.4.	60	5	LRN	3.3.4.4.
02	3	JRT	3.2.2.1.	61	4	NI	3.4.2.2.
04	6	JIERT	3.2.2.2.	63	4	ØI	3.4.2.3.
05	6	JIURT	3.2.2.2.	6A	4	RLA	3.8.3.
12	4/5	MR8	3.5.4.1.	70	3	SLD	3.3.3.1.
14	4/5	MR9	3.5.4.1.	71	3	RØUND	3.3.3.3.
16	5/7	MR89	3.5.4.1.	72	3	SRD	3.3.3.2.
1E	5/6	MR8G	3.5.5.	78	1	PRSTØ	3.8.1.
				7D	1	LØAD	3.8.2.
21	4/5	M6R	3.5.4.1.				
22	4/6	M68	3.5.4.1.				
23	5/7	M6R8	3.5.4.1.				
24	4/6	M69	3.5.4.1.	80	6	MVC	3.4.3.1.
25	5/7	M6R9	3.5.4.1.	82	6	INC	3.4.3.2.
26	5/8	M689	3.5.4.1.	84	6	PKH	3.4.3.3.
27	6/9	M6R89	3.5.4.1.	85	6	UPH	3.4.3.4.
2E	5/7	M68G	3.5.5.	88	6	CMC	3.4.2.1.
				8A	6	MVIC	3.4.3.1.
31	4/5	M7R	3.5.4.1.				
32	4/6	M78	3.5.4.1.				
33	5/7	M7R8	3.5.4.1.	91	3	CMD	3.3.2.4.
34	4/6	M79	3.5.4.1.	92	3	CMDD	3.3.2.4.
35	5/7	M7R9	3.5.4.1.				
36	5/8	M789	3.5.4.1.				
37	6/9	M7R89	3.5.4.1.	A1	3	ADD	3.3.2.1.
3E	5/7	M78G	3.5.5.	A2	3	ADDD	3.3.2.1.
				B1	3	SBD	3.3.2.1.
40	1	STØP	4.3.3.1.	B2	3	SBDD	3.3.2.1.
41	2	START	4.3.3.2.				
42	2	PRØ	4.3.3.3.	C1	3	MPD	3.3.2.2.
44	2	FREE	4.3.3.4.				
				D1	3	DVD	3.3.2.3.
50	2	TRØ	3.7.2.				
51	2	TRI	3.7.2.	E1	3	MRR	3.3.4.1.
53	4/14	IØC	3.6.6.	E2	3	MDRR	3.3.4.1.
54	4/14	IØIC	3.6.6.	E3	3	MSRDR	3.3.4.2.
57	1	KHLT	3.8.4.	E5	3	EXR	3.3.4.3.
5A	2	TRA	3.7.2.	E6	3	EXDR	3.3.4.3.
5B	2	TRB	3.7.2.				
				F2	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.



### 7.1.1. Aufbau der Basis-Softwareprogramme

Jedes Programm ist in Karten gelocht.

Es gibt mehrere Karten für jedes Programm. Die ersten 68 Spalten jeder Karte sind für das eigentliche Programm reserviert und die 12 letzten sind den Angaben, die zum Erkennen des Programms dienen, reserviert. Das sind:

- in der Spalte 69 und 70 der Schlüssel 23, der besagt, daß es sich um den GE-55 handelt (s. Ref.-Nr. der Dokumentation),
- in den Spalten 71 bis 74 die Programmnummer,
- in Spalte 75 den Lochcode, und zwar:
  1. wenn es sich um den Code H14.012 handelt,
  2. wenn es sich um den Code T121 handelt,
- in Spalte 76 steht die letzte Stelle des Jahresdatums der Erstellung dieses Programms,
- in Spalten 77 bis 79 die Nummer des lfd. Tages der Erstellung des Programmes,
- in der Spalte 80 eine 1, wenn es die erste Karte, eine 9, wenn es die letzte Karte und nichts, wenn es andere Karten sind. Man muß unbedingt die Karten, die eine 1 in Spalte 80 hat, an den Anfang legen und die Karte, die eine 9 in Spalte 80 hat, an den Schluß. Die Reihenfolge der anderen Karten ist ohne Bedeutung.

Die Anzahl der Karten der Software-Programme wird später angegeben.

7.2. Die Basis-Softwareprogramme

7.2.1. Die Programmkartenkontrolle (PKK)

Mit diesem Programm werden:

- die Lochungen in den Programmkarten geprüft
- die Konstantenkarten in Format 1 und 2 kontrolliert
- die Maschinen-OT aufgrund des symbolischen Operationstyps gestanz
- die Befehle aufgelistet.

Die Programmnummer lautet für:

2500 Bytes : 23.30.12 ( 91 Karten) und 23.30 14 (77 Karten)

5000 Bytes : 23.30 15 (168 Karten)

10000 Bytes : 23.30 18 ( 168 Karten)

Die Kontrolle der Programm- und Konstantenkarten erfolgt Karte für Karte. Eine Liste der angezeigten Fehler befindet sich in Abschnitt 7.2.1.3.

7.2.1.1. Handhabung

- Eingabe der Programm-Kartenkontrolle.
- Eingabe einer Parameterkarte, die die gewünschte Funktion anzeigt (siehe 7.2.1.2.).
- Eingabe des zu kontrollierenden Kartenpaketes: (unverdichtetes Programm).

Das Programm muß von normalem Aufbau sein, d.h. eine Programm-Leitkarte, die Karte F3 und die Startkarte enthalten.

Ergebnisse:

Jede gelesene Karte wird gelistet mit der reellen Adresse und evtl. aufgetretenen Fehlern.

Wenn die Startkarte gelesen ist, werden sämtliche fehlenden LEVEL gelistet. Sobald die Programmkartenkontrolle beendet ist, wird auf der Leuchtanzeige 6 x die 7 angezeigt. Zu diesem Zeitpunkt wird die numerische Tastatur freigegeben und man kann mit Druck auf die Taste CLB den Puffer löschen.

Wiederanlauf nach Fehler

Siehe Abschnitt 7.5.

Begrenzung

Für das zu kontrollierende Kartenpaket ist keinerlei Einschränkung vorgesehen.



7.2.1.2. Parameter

<b>Listen und PKK</b>						
1	2	3	4	5	6	

1	<b>Listen, PKK und Stanzen des Maschinen-OT aufgrund des symb. OT</b>					
1	2	3	4	5	6	

### 7.2.1.3. Fehlerarten

- Fehler 1: Der Schlüssel in Spalte 2 + 3 stimmt nicht.
- Fehler 2: Der symbolische Operationstyp existiert nicht.
- Fehler 3: Der symbolische Operationstyp und der Maschinencode stimmen nicht überein (wenn der Maschinencode bereits gelocht ist).
- Fehler 4: Übertragungsbefehle stimmen nicht überein (die Verschiebungen der Basen sind nicht numerisch oder unterschiedlich von Axxx).
- Fehler 5: Anschlußcode des IØC oder des IØIC existieren nicht oder die angegebenen Register sind nicht numerisch.
- Fehler 6: Fehler in den Parametern (der Parametertyp entspricht nicht 0, 1, 2, 3, 4, 5, 6).
  - Die Nummer der Karte ist entweder nicht numerisch oder größer 999.
  - Die Feldnummer ist unterschiedlich von 1 - 9.
- Fehler 7: Die Karte enthält Buchstaben oder Sonderzeichen, die verschieden sind von 1 - 9, A - F oder Leerzeichen (eine Ausnahme besteht für die Kartenart 31).
- Fehler 8: Der Operationstyp existiert nicht.
- Fehler 9: Die in den Spalten 4 + 5 angegebene Länge entspricht nicht der Länge, die dem Operationstyp entspricht.
- Fehler 10: Die Anzahl der in der Karte gelochten Zeichen ist verschieden von der Zahl, die in den Spalten 4 - 5 gelocht ist.
- Fehler 11: Die Länge ist nicht numerisch und entspricht auch nicht einer freien Stelle.
- Fehler 12: Die Register sind nicht numerisch.
- Fehler 13: Bxxx ist nicht numerisch.
- Fehler 14: Ein Sprung ohne F1 mit Buchstaben in der Rücksprungadresse.
- Fehler 15: Die Rücksprungadresse ist größer 2499 (4999, 9999).
- Fehler 16: Sprung ohne F2.
- Fehler 17: 2 Sprungstufen sind einander sehr ähnlich.
- Fehler 18: Fehler in der Sortierfolge der Ordnungsnummer (leere Felder werden nicht berücksichtigt).
- Fehler 19: Fehler in der Anzahl der Karten, d.h. die Anzahl der gelesenen Karten ist nicht identisch mit der gelochten Anzahl). Die Anzahl der gelesenen Karten wird rechts der Ziffer 19 angeschrieben.

- Fehler 20: Die Startadresse ist nicht korrekt.
- Fehler 21: In der Sprungangabe befindet sich eine Trennmarke.
- Fehler # : Die Adresse in der Programm-Leitkarte ist nicht numerisch (In diesem Fall blockiert der GAMMA 55).
- Fehler !: Die Anzahl der gelochten Zeichen, die in Spalten 4 + 5 gelocht sind, ist nicht numerisch (In diesem Fall blockiert der GAMMA 55).
- Fehler \* : Eine LEVEL-Karte hat in den Spalten 4 + 5 eine Zahl, die unterschiedlich von 02 ist.
- Fehler ?: Die Konstantenkarte befindet sich vor der Programmendkarte F3. Anschließend liegen noch Befehlskarten.
- Fehler < : Die Verschiebung 1 ist kleiner als Verschiebung 2 in einem MVC-Befehl, falls die gleiche Basis benutzt wurde.

Außerdem werden bei der PKK, sobald die Startkarte gelesen ist, die Sprungbefehle angeschrieben, zu denen kein entsprechender LEVEL gefunden wurde.

Die Programmkarten-Kontrolle prüft die ersten 206 LEVEL-Nummern. Werden in einem Programm mehr als 206 Nummern belegt, oder ein LEVEL mit einer größeren Nummer wie 206 belegt, und die Sprungbefehle stimmen nicht mit den dazu gehörigen LEVEL überein, so schreibt die PKK statt der Fehlerliste: Fehler VV.

KA	K	Ord.Nr.	Speicher- stelle	OT symb.	Befehle	Fehlerart							
1	2	4	5	7	10	12	15	17	23	25	}}	89	90

7.2.2. Das Listprogramm (PK-Listen)

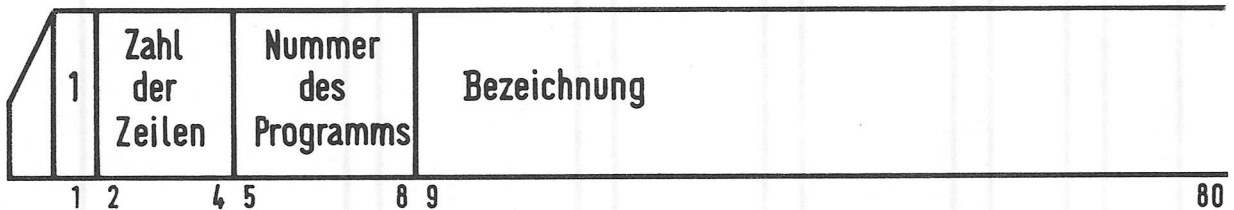
Mit diesem Programm werden die Programmkarten Karte für Karte aufgelistet. Gleichzeitig werden die Parameter für das Verdichtungsprogramm gestanzt.

Das Listprogramm besteht aus 45 Karten. Die Programmnummer lautet für:

- 2500 Bytes : 23.32 41
- 5000 Bytes : 23.32 45
- 10000 Bytes : 23.32 48

7.2.2.1. Handhabung

- Eingabe des Listprogramms
- Eingabe einer Parameterkarte



In den Spalten 2 - 4 der Parameterkarte werden die gewünschten Zeilen je Blatt gelocht. Ist diese Lochung vorhanden, erfolgt ein entsprechender Papiersprung bei der gewünschten Zeile.

Die angegebene Anzahl der Zeilen umfaßt die Kopfzeile, die Leerzeile zwischen der Kopfzeile und der 1. zu druckenden Zeile und den zu druckenden Zeilen. Die Programmbezeichnung wird auf der Höhe der Kopfzeile angeschrieben.

Das Vorhandensein einer Lochung in den Spalten 5 - 8 der Parameterkarte (Programmnummer) ermöglicht das Stanzen der für das Verdichtungsprogramm erforderlichen Parameterkarten (siehe Abschn. 7.2.3.). Diese Karten enthalten eine oder mehrere Parameter, die Anfangsadresse und die Endadresse der oder des zu verdichteten Programmes und eine Startkarte. Diese Startkarte dient nur dazu, um im Verdichtungsprogramm dupliziert und ergänzt zu werden mit der Anzahl der Karten des verdichteten Programms, die in die Spalten 17 - 20 gelocht wird.

Die Parameterkarte ist obligatorisch. Fehlt diese Parameterkarte, so wird die erste zu listende Karte wie eine Parameterkarte behandelt. Da im Prinzip die erste zu listende Karte die Programm-Leitkarte ist, die die Anfangsadresse enthält, wird diese nicht beachtet. Wird keine der gegebenen Möglichkeiten des Listprogramms gewünscht, so ist es möglich, eine Leerkarte als Parameterkarte zu nehmen.

- Eingabe der oder des zu listenden Programmes, das sich wie folgt zusammensetzen kann:
  - . Eine Programmleitkarte
  - . Programmkarten
  - . Programmendkarte.
- Eingabe der Konstantenkarten in Formates 1
- Eingabe der Konstantenkarten in Formates 2
- Eingabe der Programmstartkarte.

Wiederanlauf nach Fehlerhalt

Wiederanlauf nach Halt des Druckers (Papierende).

Leuchtet die Lampe PR und bleibt die Maschine stehen, muß das Papier gewechselt werden, wobei darauf zu achten ist, daß das erste Blatt auf der ersten zu druckenden Zeile steht. Anschließend ist der Sprungstreifen einzustellen und das letzte Blatt des Papierstapels in den Kontakt für Papierende einzuschieben. Ein Druck auf die Taste RES des Druckers läßt die Arbeit normal weiterlaufen.

Wiederanlauf nach Fehlerhalt des Stanzers (siehe Abschn.7.5)

KA	K	Ord.Nr.	Speicherstellen	OT symb.	Befehle						
11	2	4	5	7	10	12	15	17	23	25	88



### 7.2.3. Das Verdichtungsprogramm (MEFE)

Mit diesem Programm werden getestete Arbeitsprogramme verdichtet. In einem nicht fertig getesteten Programm enthält jede Programmkarte nur einen Befehl. In dem verdichteten Programm sind mehrere Befehle je Karte gelocht.

Das Verdichtungsprogramm besteht aus **19 Karten**. Die Programmnummer lautet für:

2500 Bytes : 23.32 43  
5000 Bytes : 23.32 46  
10000 Bytes : 23.32 49

Ein verdichtetes Programm bietet zwei wesentliche Vorteile:

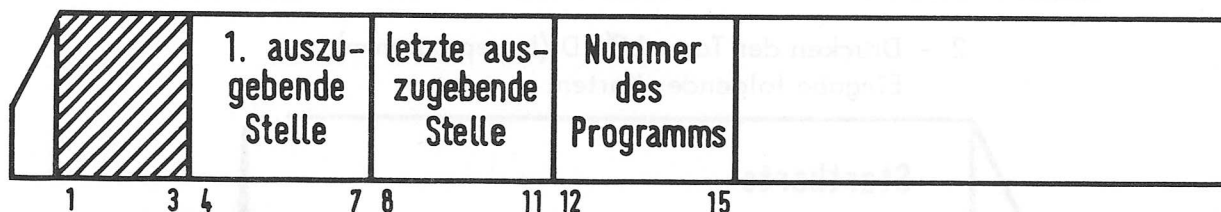
- Es gibt eine wesentliche Verringerung der Kartenanzahl.
- Die gelochte Adresse in den Spalten 6 - 9 gestattet es, die Karten in beliebiger Reihenfolge einzugeben.

#### 7.2.3.1. Handhabung

Das Programm "Listen der Programmkarten" liefert die für das Verdichtungsprogramm erforderlichen Parameter, die folgendes enthalten:

- Eine oder mehrere Parameterkarten des Types, der unten beschrieben ist und eine Startkarte.

Es ist ratsam, sich von jedem Programm eine Liste anzufertigen, bevor es verdichtet wird, da man dann automatisch die gewünschten Parameterkarten für das Verdichtungsprogramm erhält.



- Eventuell ist das Löschmodul einzugeben.
- Anschließend wird das getestete Programm mit dem Lademodul (ICARE) eingegeben. Es ist unbedingt erforderlich, daß die Startkarte vor dem Eingeben herausgenommen wird.
- Danach wird das Verdichtungsprogramm (MEFE) eingegeben.

- Eingeben der Parameterkarten und der Startkarte, die beim Listen des Programmes gestanzt wurden. Vorher ist das Zufuhrmagazin des Stanzers mit Leerkarten zu füllen.

Anlauf nach Fehlerhalt

Der einzige, vorgesehene Fehlerhalt liegt bei Stanzfehler vor.

Begrenzungen

Es ist nur möglich, Programme von Stelle 0161 bis zur Stelle

1732 bei 2.500 Bytes

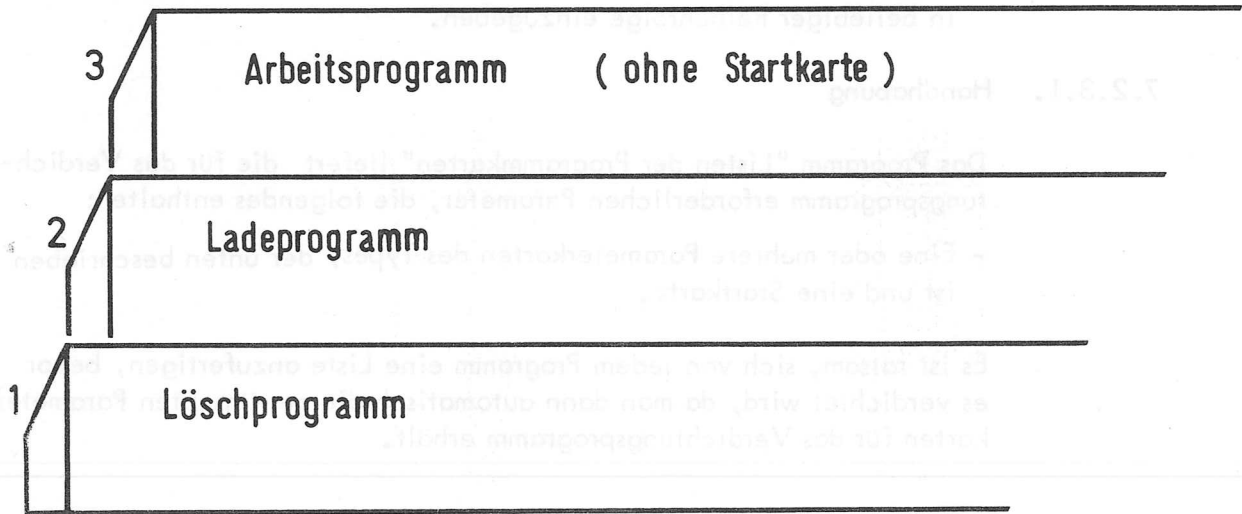
4040 bei 5.000 Bytes

9040 bei 10.000 Bytes

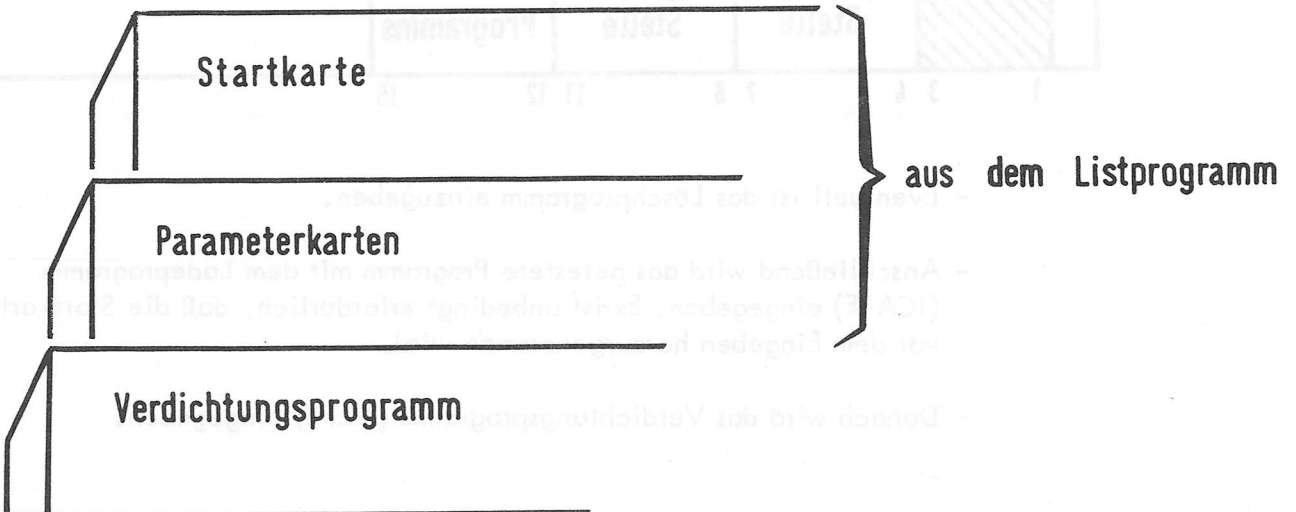
einschließlich einzugeben.

### 7.2.3.2. Arbeitsablauf

- 1 - Drücken der Taste LØAD  
Eingabe folgender Karten:



- 2 - Drücken der Taste LØAD (Ladeprogramm)  
Eingabe folgender Karten:



GE-55

#### 7.2.4. Das Ladeprogramm (ICARE)

Das Ladeprogramm gibt ein Arbeitsprogramm in den Kernspeicher ein, ohne es zu starten. Zusätzlich wird das Unterbrechungsprogramm eingespeichert (s. 7.2.6.).

Die Programmnummer lautet für:

2500 Bytes	:	23.33 32	(32 Karten)
5000 Bytes	:	23.33 33	(37 Karten)
10000 Bytes	:	23.33 34	(37 Karten)

Durch das Ladeprogramm erfolgt außerdem:

- das Einsetzen der reellen Adresse anstelle der Symbole in den Sprungbefehlen.
- eine Kontrolle der Zahl von Programm- und Konstantenkarten.
- ein Freimachen der von den LEVEL belegten Stellen (Zahl der LEVEL x 2) und ein Zusammenschieben des Programms danach.
- ein Belegen des Kernspeichers mit bestimmten Werten (s. Formular "Kernspeicherbelegung").
- der Start des eingegebenen Programms.

Das gesamte System besteht aus:

- dem eigentlichen Ladeprogramm für unverdichtete oder verdichtete Programme.
- dem Unterbrechungsprogramm (s. 7.2.6.).

#### 7.2.4.1. Die Handhabung

Laden eines unverdichteten Programms

- Laden des Löschmoduls.
- Laden des Ladeprogramms und des Unterbrechungsprogramms.
- Eingabe des zu ladenden Programms, das sich wie folgt zusammensetzen muß:
  - Eine Programmbeginnkarte
  - Die Programmkarten
  - Eine Endkarte
- Eingabe der Konstantenkarten, Format 1,
- Eingabe der Konstantenkarten, Format 2,
- Eingabe der Startkarte.

Laden eines verdichteten Programms:

- Laden des Löschmoduls
- Laden des Ladeprogramms und des Fehlerprogramms
- Eingabe des verdichteten Programms, das sich wie folgt zusammensetzt:
  - . Konstantenkarten oder Programmkarten, Format 2,
  - . eine Programm-Startkarte.

#### Wiederanlauf nach Unterbrechung

Es gibt eine Unterbrechung, wenn die Anzahl der eingegebenen Karten nicht mit der in der Programmstartkarte angegebenen Anzahl der Karten übereinstimmt. In diesem Fall wird an der Leuchtanzeige dreimal die 1 sichtbar. Man kann diese Unterbrechung übergehen, indem man den Wiederanlauf mit einer fixen Adresse, die im Fehlerprogramm vorgesehen ist, bewerkstelligt:

- Löschen des Pufferspeichers (Taste CLB)
- Eintasten (auf der numerischen Tastatur): 88
- Übertrag Puffer an Kernspeicher (Taste MVB).

#### Die Begrenzungen

Es ist möglich, ein Programm einzugeben, das von Stelle 0161 bis zur Stelle 1732 bei 2.500 Bytes  
4040 bei 5.000 Bytes  
9040 bei 10.000 Bytes

einschließlich eingespeichert wird.

Die maximale Anzahl der benutzbaren LEVEL ist 256.

GE-55

Ref.-Nr.: 23.20.001 D

Juli 1968

GE-55

© 100.05.28.1-1-1968

8891.1101

Ref.-Nr.: 23.20.001 D

Juli 1968

### 7.2.5. Das Löschmodul

Das Löschmodul löscht den Kernspeicher von Stelle 161 an bis zum Ende. Das ist je nach Kernspeicherkapazität die Stelle 2499, 4999 oder 9999.

Es besteht aus 3 Karten. Die Programmnummer (Sp. 71 - 74) lautet für:

2500 Bytes	:	23.32 42
5000 Bytes	:	23.32 44
10000 Bytes	:	23.32 47



### 7.2.6. Das Unterbrechungsprogramm (PGI)

Ziel des Unterbrechungsprogramms ist es, die Arbeit des Programmierers zu erleichtern, indem sie ihm eine Standardlösung für die Wiederanläufe nach Unterbrechungen gibt, die er in seinem Programm vorsehen muß. Außerdem soll es in gleichem Maße die Arbeit des Bedieners erleichtern durch die Normalisierung der zu bewerkstelligenden Handgriffe im Falle einer Unterbrechung.

Es besteht darin:

- die Art der Unterbrechung sichtbar zu machen
- einen automatischen Wiederanlauf für die nichtprogrammierten Unterbrechungen zu gewährleisten
- ein spezielles Wiederanlaufprogramm für die Unterbrechungen zu starten.

#### 7.2.6.1. Beschreibung des Programms

##### Anruf des Programms

Der Anruf des Unterbrechungsprogramms kann auf zwei Arten erfolgen:

- durch Druck auf die Taste PI (Programmunterbrechung). Der Druck auf die Taste PI bewirkt:
  - daß die Programmadresse von PAR nach ANU übertragen wird,
  - daß die Adresse von AFP nach PAR übertragen wird.
- durch Belegung der Pufferzone mit den Fehlercodes und durch einen Sprung auf die Adresse 4676 (bei 5000 Bytes). Der Programmierer hat die Möglichkeit, die drei linken Stellen der Pufferzone mit Signal-codes für Unterbrechungen zu belegen. Durch das PGI werden an der Leuchtanzeige die drei rechten Stellen mit 111 belegt.

##### Ablauf des Programms

Nacheinander erfolgt:

- die Speicherung des Inhalts der Pufferzone
- die Speicherung der Rücksprungadresse
- die Speicherung der drei ersten Bytes der PRC-Zone
- Setzen eines Testes für Monoprogramming
- Sichtbarmachung von 3 x einer '1' auf der Leuchtanzeige und evtl. auch der spezielle Code für Unterbrechungen.

Jetzt kann durch Druck auf die Taste CLB der Inhalt des Pufferspeichers gelöscht und über die numerische Tastatur die Art des Wiederanlaufs angegeben werden:

- Lesen einer Karte
- Stanzen einer Karte von der Stanzzone her
- Setzen eines Tests im Zentralspeicher auf der Adresse 4661 (bei 5000 Bytes)

Anschließend erfolgt:

Beim Wiederanlauf nach einer Unterbrechung des Lesers, Stanzers oder nach dem Test wird der ursprüngliche Zustand der logischen Zone wieder hergestellt. Dieses Herstellen des ursprünglichen Zustandes umfaßt:

- den Start des Programmes, das gestartet war im Augenblick der Unterbrechung,
- die Annullierung des Tests für Monoprogrammierung, falls ein Test gesetzt wurde,
- die Wiederherstellung des ursprünglichen Zustandes in der Pufferzone,
- den Rücksprung ins Hauptprogramm,
- den Start eines Programmes mit einem spezifischen Wiederanlauf, dessen Adresse sich an den Stellen 4659 und 4660 befinden muß (bzw. 2355 und 2356)
- den Start eines spezifischen Programms, dessen Anfangsadresse über die Tastatur eingegeben wird.

#### Möglichkeiten des Wiederanlaufs

Es ist möglich, alle diese Starts durch das Unterbrechungsprogramm durchzuführen, wenn das Programm durch die Taste PI oder durch Programm angerufen wird.

Es ist aber zu bemerken, daß der Anruf des Unterbrechungsprogramms (PGI) durch Druck der Taste PI automatisch die Rücksprungadresse nach Unterbrechung in die Programmadresse setzt.

Dadurch wird dem PGI ermöglicht, das unterbrochene Programm auf der Adresse neu zu starten, die der unterbrochenen Adresse folgt.

Daraus ergibt sich, daß, wenn das PGI durch ein Programm angerufen wurde, um ein Lesen oder Stanzen oder das Setzen eines Testes zu bewirken, ein Einsetzen der Adresse nach Unterbrechung nicht erfolgt. Daher muß der Programmierer wie folgt verfahren:

Anstatt sofort in das Fehlerprogramm zu springen, erfolgt ein Sprung in ein Unterprogramm:

```
LEVEL F 2 ..  
MVC 80 02 0088 0093  
MVC 80 02 0095 0090
```

Vorsichtsmaßnahmen

#### Monoprogramming

Der Wiederanlauf nach einer Unterbrechung, die das PGI anruft, schließt den Gebrauch der numerischen Tastatur und des Pufferspeichers ein.

Der Programmierer muß daher beachten, daß er die numerische Tastatur nicht benutzt, wenn das PGI angesprochen werden kann.

#### Beispiel:

Dieses Programmierbeispiel ist zu vermeiden:

- ∅C Freigabe numerischer Tastatur
- ∅C Stanzen
- ∅C Übertrag Puffer an Zentralspeicher.

Wird ein Fehler im Stanzer festgestellt, so wird die Freigabe für die numerische Tastatur durch den Sprung ins Fehlerprogramm (PGI) annulliert.

#### Multiprogramming

Es existieren zwei Arten des Multiprogramming:

Die eine Methode besteht darin, ein Hauptprogramm und mehrere Unterprogramme (1 - 4) dazu zu schreiben. In diesem Falle genügt es, als Wiederanlauf die Arbeit von Beginn einer Gruppe an neu aufzunehmen. Dafür kann ein spezifisches Programm angewendet werden, das unter 7.2.6.2. erläutert ist. Mit ihm können 2 - 5 Unterprogramme bearbeitet werden.

Ein Wiederanlauf nach einem Fehler an den Randeinheiten kann durch das PGI nach der alten klassischen Methode erfolgen, ohne daß man mit Beginn einer neuen Gruppe neu anfangen muß.

Die zweite Art von Multiprogramming besteht darin, daß man mehrere Programme hat, die in keiner Verbindung zueinander stehen und die verschiedene Probleme behandeln. In diesem Falle wird jedes Programm als ein lineares Programm, wie bei Monoprogramming, betrachtet.

Jedenfalls muß der Programmierer beachten, daß der gleiche Kanal nicht im selben Moment von mehreren Programmen beansprucht wird. Vor jedem IØC muß ein Schutz (PRØ) vorgesehen werden. Es wird empfohlen, für einen Schutz die Nummer anzugeben, die mit der Kanalnummer übereinstimmt, z.B. für einen IØC, der den Kanal 1 benutzt, wird PRØ 01 gesetzt.

Beispiel:

PRØ	02	
IØC	Stanzen	
FREE	02	Programm 1
PRØ	02	
IØC	Stanzen	
FREE	02	Programm 2

Der Wiederanlauf nach einer Unterbrechung durch die Taste PI läuft normal weiter. Der Wiederanlauf nach einer programmierten Unterbrechung wird mit einem Sprung auf die Adresse 4726 (2422) gestartet, so daß die anderen Programme nicht blockiert werden. Der Programmierer kann daher in dem Pufferspeicher einen Code eingeben, der die Art der Unterbrechung kennzeichnet.

7.2.6.2. Anwendung

Anruf eines Lesebefehls

- Drücken der Taste P.I.
- 111 wird an der Leuchtanzeige sichtbar.
- Löschen des Pufferspeichers (Druck auf die Taste CLB).
- Eintasten der Zahl 55 (51).
- Übertrag Puffer an Zentralspeicher (Druck auf die Taste MVB).

Wiederanlauf nach einem Halt des Stanzers

- Drücken der Taste P.I.
- Drücken der Taste Auswurf (EJ), um die letzte richtig gestanzte Karte auszusteuern.
- Drücken der Taste RES. Zu diesem Zeitpunkt wird das Ende der fehlerhaften Karte in eine Leerkarte gestanzt und dreimal die '1' auf der Leuchtanzeige sichtbar.

- Drücken zweimal auf die Taste EJ.
- Herausnehmen der fehlerhaften Karten, die sich im Ablagefach befinden.
- Löschen des Pufferspeichers mit Taste CLB.
- Eintasten auf der numerischen Tastatur den Code 48 (44).
- Drücken der Taste MVB (Übertrag Pufferspeicher an Zentralspeicher).

Die Arbeit läuft normal weiter. In jedem Falle liegt eine Leerkarte im Ablagefach.

Setzen eines Tests

- Drücken der Taste PI.
- Es muß dreimal die '1' auf der Leuchtanzeige sichtbar gemacht werden, sofern deren Ablauf nicht gestört und kein Fehlerfall angezeigt wurde.
- Löschen des Pufferspeichers durch Drücken der Taste CLB.
- Eintasten auf der numerischen Tastatur des einzugebenden Testwert und anschließend den Code 62 (58)
- Übertrag Puffer an Zentralspeicher durch Drücken der Taste MVB.

Dadurch wird erreicht, daß der Test im Zentralspeicher die Stelle 4661 (2357) besetzt und daß man auf das Hauptprogramm zurückkommen kann.

- Starten eines Programms an einer fixen Adresse.

Bei Multiprogramming muß der Programmierer das spezifische Wiederanlaufprogramm vorsehen.

Um dies zu starten, muß man nach Auftreten des Fehlercodes und der angezeigten '111' wie folgt verfahren:

- Löschen des Pufferspeichers.
- Eintasten der Zahl 88
- Übertragen dieses Wertes in den Kernspeicher mit der Taste MVB.

Dadurch wird das Programm mit der Adresse gestartet, die im Kernspeicher an Stelle 4659 und 4660 (2355/56) steht. Es ist zu beachten, daß das Ladeprogramm in der Programmadresse die Startadresse, die in der Startkarte gelocht ist, speichert. Der Programmierer hat die Möglichkeit, an einer anderen Adresse als der Startadresse zu starten. Die Stellen 4659 und 4660 (2355/56) sind dann fixe Adressen.

- Starten eines Programms an eine Adresse, die mit der Tastatur eingegeben wird.

Bei Multiprogramming muß der Programmierer das spezifische Wiederanlaufprogramm vorsehen.

Um das Programm starten, wird nach Erscheinen des Fehlercodes und der '111' auf der Leuchtanzeige wie folgt verfahren:

- Löschen des Pufferspeichers (Taste CLB).
- Eingeben der reellen Adresse des Programms, auf welches man springen will, anschließend Eingabe des Codes 98 (94).
- Übertrag des Pufferspeichers an den Zentralspeicher (Taste MVB).

Dadurch wird das Programm an der angezeigten Adresse gestartet.

#### Bemerkungen:

Der Programmierer muß sich immer die Tatsache vor Augen halten, daß das Unterbrechungsprogramm sich der numerischen Tastatur bedient und daß jede Freigabe der numerischen Tastatur nicht in dem Moment erfolgen darf, wo ein Anruf des PGI erfolgt.

- Platzbelegung

Das Unterbrechungsprogramm (PGI) belegt 142 (145) Bytes, wovon 128 für das Programm und 14 (17) zum Speichern benutzt werden.

#### Einschränkungen

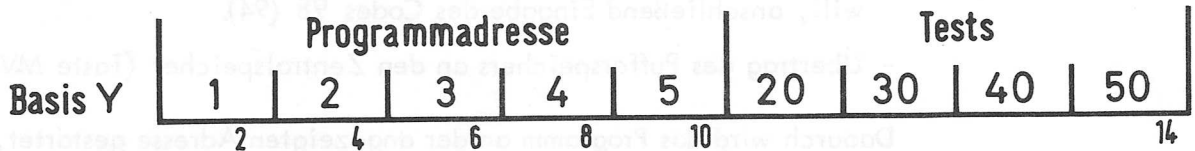
Die Programme, die sich des Unterbrechungsprogrammes bedienen, müssen zwangsläufig in jeder ihrer Registerzonen, die dem Programmwechsel unterliegen, folgendes berücksichtigen:

- Das Register 00 muß immer leer bleiben.
- Die Register 6, 8 und 9 müssen immer die Standardbasen für das Lesen, den Druck und für das Stanzen beinhalten.
- Das Register 10 muß immer die Standardbasis für den Pufferspeicher enthalten.
- Das Register 11 muß immer den Code für die Leuchtanzeige, die Trennmarke F4 und die Adresse 0150 enthalten.
- Das Register 12 muß immer den Code für den Übertrag des Puffers an den Zentralspeicher, die Trennmarke F4 und die Adresse 0155 enthalten.

### 7.2.6.2. Das spezifische Wiederanlaufprogramm

Dieses Programm benötigt im Kernspeicher eine Arbeitszone von 14 Bytes, die folgendes enthält:

Die Adresse des Unterprogramms des spezifischen Wiederanlaufs, die Anfangsadressen der Untergruppenprogramme und die Tests "Halt-Programm" der Programme 2 - 5. Beispiel:



- es benutzt einen LEVEL (AA)
- es belegt ein Basisregister, das durch den Programmierer zu bestimmen ist, der dann die Programm-Linien, die dieses Register benutzen, ergänzen muß.

In der Basis Y finden wir folgendes:

- 1 : Adresse des Unterprogramms des spezifischen Wiederanlaufs
- 2 - 5 : Anfangsadressen der Untergruppen-Programme
- 20 - 50: Die wirklichen Werte, die in der PRC-Zone zu untersuchen sind.

#### Plazierung

Das Programm belegt 94 Bytes, wenn 4 Unterprogramme benutzt sind.

Es belegt 2 Basisregister X + Y und eine Arbeitszone von 14 Bytes.

#### Einschränkungen

Dieses Programm wird den normalen Bedingungen des Anwendungsbereichs der Maschine angepaßt, d.h. es benutzt die normale Basis-Software. Das Register 0 muß immer vor einem Sprung ins PGI auf 0 stehen.

#### Beispiel:

siehe Seite 233



Erläuterung	Sprünge		KA	Zahl der Bytes	Ord.-Nr.		Symbolischer Operationstyp	Befehl																				
	F 1	F 2			2	3		4	5	6	9	10	16	17	20/21	24/25	28/29	32										
			1	2	0	6	0	0	0	J	I	U	R	T	0	5	1	1	0	0	7	0	F	1	A	A		
Übertrag für P2			1	2	0	6	0	1	0	M	V	C			8	0	0	2	0	0	6	7	Adresse eines Basisregisters (x)					
			1	2	0	6	0	2	0	M	V	C			8	0	0	2	Y	0	0	4	X	0	0	5		
			1	2	0		0	3	0																			
Übertrag für P3			1	2	0	6	0	4	0	M	V	C			8	0	0	2	0	0	6	5	Adresse eines Basisregisters (x)					
			1	2	0	6	0	5	0	M	V	C			8	0	0	2	Y	0	0	6	X	0	0	5		
			1	2	0		0	6	0																			
Übertrag für P4			1	2	0	6	0	7	0	M	V	C			8	0	0	2	0	0	6	3	Adresse eines Basisregisters (x)					
			1	2	0	6	0	8	0	M	V	C			8	0	0	2	Y	0	0	8	X	0	0	5		
			1	2	0		0	9	0																			
Übertrag für P5			1	2	0	6	1	0	0	M	V	C			8	0	0	2	0	0	6	1	Adresse eines Basisregisters (x)					
			1	2	0	6	1	1	0	M	V	C			8	0	0	2	Y	0	1	0	X	0	0	5		
			1	2	0		1	2	0																			
Löschen PRC			1	2	0	6	1	3	0	I	N	C			8	2	0	0	0	0	8	4	0	0	7	0		
Übersetzen "HALT" (PRC)			1	2	0	6	1	4	0	M	V	C			8	0	0	4	Y	0	1	4	0	0	8	4		
			1	2	0		1	5	0																			
			1	2	0	6	1	6	0	J	R	T			0	2	Anfangadresse des Programms											
			1	2	0		1	7	0																			
			1	4	0	2	1	8	0	L	E	V	E	L	F	2	A	A										
			1	2	0		1	9	0																			
Übertrag für P1			1	2	0	6	2	0	0	M	V	C			8	0	0	2	0	0	6	9	Adresse eines Basisregisters (x)					
			1	2	0	6	2	1	0	M	V	C			8	0	0	2	Y	0	0	2	X	0	0	5		
			1	2	0		2	2	0																			
			1	2	0	4	2	3	0	N	I				6	1	0	F	0	0	7	0						
PRC 11			1	2	0	6	2	4	0	I	N	C			8	2	1	1	0	0	7	1	0	0	7	0		
			1	2	0		2	5	0																			
Programmwechsel			1	2	0	1	2	6	0	S	T	O	P		4	0												
			1	2	0		2	7	0																			
			1	2	0		2	8	0																			
			1	2	0		2	9	0																			
			1	2	0		3	0	0																			
			1	2	0		3	1	0																			

12.66 / 1009

Buchstabe O = Ø, Ziffer Null = 0

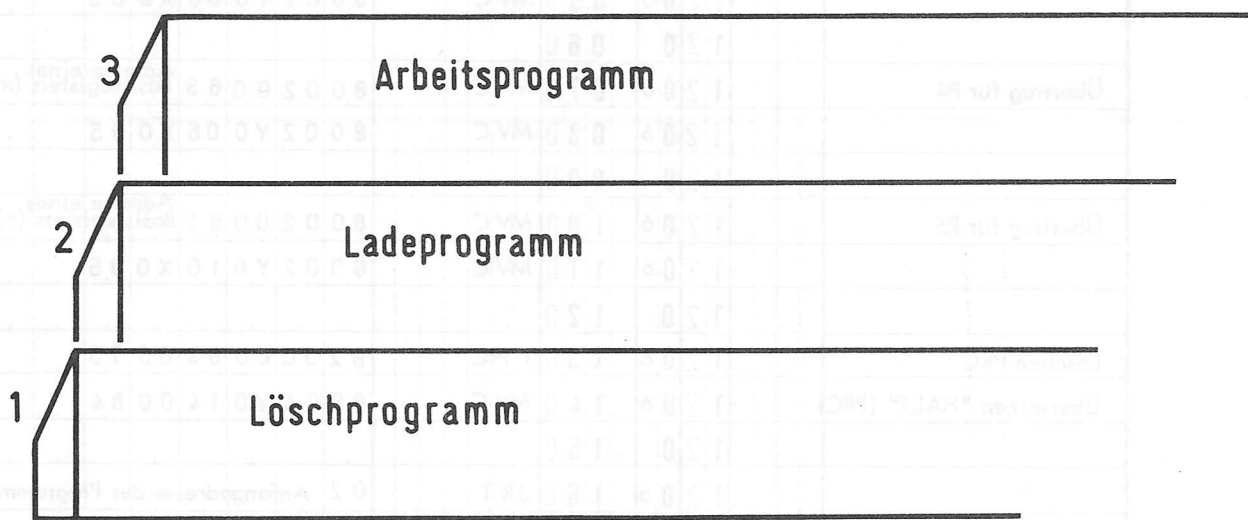
### 7.3. Anwendung der Basis-Software-Programme

#### 7.3.1. Das Laden eines Arbeitsprogrammes

Um ein Arbeitsprogramm zu laden, muß man folgende Software-Programme benutzen:

- das Löschmodprogramm. In gewissen Fällen wird dieses Programm nicht benutzt. Die genaue Handhabung muß auf der Bedienungsanweisung eingetragen sein.
- das Ladeprogramm.

Die verschiedenen Programme müssen in folgender Reihenfolge eingegeben werden:



##### 7.3.1.1. Handhabung

- Taste LØAD drücken.
- Die Karten in den Kartenleser in oben angezeigter Reihenfolge einlegen.
- Taste RES drücken.

Jetzt werden die Karten gelesen. Nach dem Lesen der letzten Karte, die immer eine Startkarte mit einem Schlüssel 16 oder 36 in Spalte 2 und 3 sein muß, können sich zwei Möglichkeiten ergeben.

- a) Auf der Sichtanzeige wird 3 x eine 1 (111) angezeigt. In diesem Fall stimmt die Anzahl der Programmkarten nicht mit der abgelochten Zahl überein. Es ist folglich notwendig, das Programm in die richtige Ordnung zu bringen und von vorne anzufangen.
- b) Es erscheint auf der Leuchtanzeige keine 1.
  - Die Datenkarten in das Zufuhrmagazin legen.
  - Auf RES drücken.

7.3.1.2. Wiederaufnahme bei Fall a)

Vorher wurde gesagt, daß ein Fehler vorliegt, wenn die Anzahl der eingelesenen Karten unterschiedlich ist von der Zahl, die auf der Startkarte gestanzt ist. Man kann diesen Fehler übergehen, wenn man:

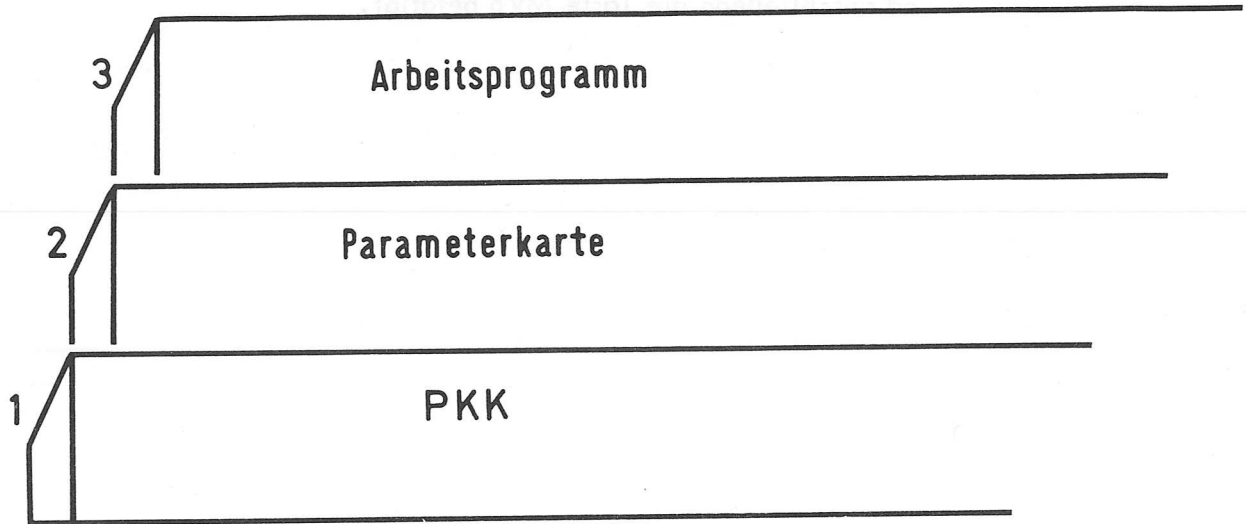
- Den Puffer löscht, indem man auf die Taste CLB drückt,
- auf der numerischen Tastatur 88 eingibt,
- und anschließend die Taste MVB betätigt.



### 7.3.2. Die Programmkartenkontrolle

Um ein Arbeitsprogramm zu testen, ist es notwendig, das Programm Programmkartenkontrolle anzuwenden (s. 7.2.1.).

Am Anfang liegt die PKK, gefolgt von einer Parameterkarte und dem zu testenden Programm.



#### 7.3.2.1. Handhabung

- Taste LØAD drücken.
- Die Karten, wie oben, in den Kartenleser einlegen.
- Taste RES drücken.

Jetzt werden die Karten gelesen.

#### 7.3.2.2. Die Wiederaufnahme bei Fehlerfällen

s. 7.5.

### 7.3.3. Das Listen eines Arbeitsprogrammes

Um ein Arbeitsprogramm zu listen, wird das Programm 'Listen' benötigt. Hinter dem Programm 'Listen' muß eine Parameterkarte liegen und dahinter das zu listende Programm.

#### Die Parameterkarte

Eine Karte mit 1 in Spalte 1 bewirkt einen Papiersprung und die Erstellung einer oder mehrerer Parameterkarten, die die Anfangsadresse und die Endadresse des eingespeicherten Programmes angeben. Diese Karten werden für das Verdichten des Arbeitsprogrammes benötigt.

1	Zahl der Zeilen	Nummer des Programms	Bezeichnung
1	2	4 5	8 9

80

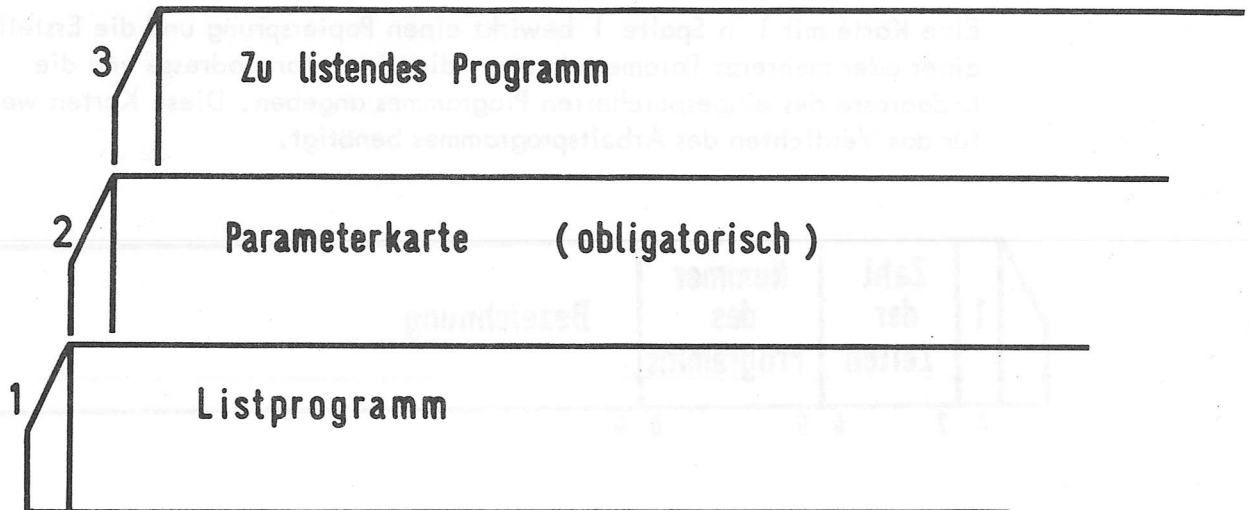
Sind die Spalten 2 bis 4 gelocht, erfolgt ein Papiersprung, der in dem Moment ausgeführt wird, wo die Anzahl der bedruckten Zeilen mit dieser gelochten Zahl übereinstimmt. In dieser Zahl ist eine Zeile für die Überschrift, eine Leerzeile zwischen dieser Titelzeile und der ersten Listzeile des zu listenden Programmes und die erforderlichen Zeilen für das Programm enthalten.

Eine Lochung in den Spalten 5 bis 8 bewirkt das Stanzen der Parameterkarten für das Verdichtungsprogramm (MEFE). Am Ende der Liste erhält man:

- eine oder mehrere Parameterkarten MEFE,
- eine Startkarte.

Die Bezeichnung des Programmes wird auf der Höhe der Titelzeile angeschrieben. Wenn keine der angebotenen Möglichkeiten eingehalten werden, ist es möglich, eine Leerkarte als Parameterkarte zu verwenden, da diese Parameterkarte obligatorisch ist.

Am Anfang liegt das Programm 'Listen', anschließend eine Parameterkarte und danach das zu listende Programm.



#### 7.3.3.1. Die Handhabung

- Taste **LØAD** drücken.
- Die Karten sind in der obengenannten Reihenfolge in den Kartenleser zu legen.
- Drücken der Taste **RES.**

Anschließend werden die Karten gelesen und das Programm gelistet.

Nach dem Lesen der letzten zu listenden Karte werden die Parameterkarten für das Verdichten (MEFE) gestanzt, wenn eine Parameterkarte mit 1 in Spalte 1 und der Programmnummer vorhanden ist.

Die Wiederanläufe nach Unterbrechungen 'Papier Ende' und 'Ende des Stanzens' sind vorgesehen.

##### 1) Wiederanlauf nach Papier Ende

Der Drucker blockiert auf der Höhe der ersten zu druckenden Zeile des Blattes nach Abfühlen des ersten Sprungloches, wenn der erste Sprung ausgeführt ist.

Die Lampe 'PR' leuchtet.

Folgende Handgriffe sind erforderlich:

- Wechseln des Papiers und richtige Einstellung des ersten Blattes auf die Höhe der ersten zu druckenden Zeile des Blattes.
- Einklemmen des letzten Blattes zwischen die Kontakte für Papierende.
- Druck auf die Taste RES des Druckers.

Bei einer Liste ohne Sprung bei vollem Blatt ist es erforderlich, das Papier zu wechseln, nachdem Papierende festgestellt wurde.

2) Wiederanlauf nach Unterbrechung des Stanzers.

s. 7.5.



#### 7.3.4. Das Verdichten eines Programmes

Um ein Programm zu verdichten, ist es erforderlich, das Programm MEFE (Verdichtungsprogramm), hinter dem die Parameterkarten liegen, zu verwenden, wenn das Programm einmal in den Kernspeicher mit Hilfe des Ladeprogramms eingespeichert wurde.

Die Parameterkarten sind die Karten, die bei der Liste des Programms erstellt wurde.

Die Karten müssen in der unten angegebenen Reihenfolge liegen: ( s.S. 220)

##### 1) Einspeichern des Programms

- Löschmodul
- Lademodul
- Arbeitsprogramm (ohne Startkarte).

##### 2) Das Verdichten des Programms

- Verdichtungsprogramm
- Parameterkarten
- Startkarte.

Diese Arbeit besteht aus zwei Arbeitsgängen:

- dem Einspeichern des Programms mit Hilfe des Lademoduls und anschließend dem Verdichten dieses Programms.

#### 7.3.4.1. Die Handhabung

##### 1) - Drücken der Taste LOAD.

- Einlegen der Karten in den Leser in der richtigen Reihenfolge.
- Drücken der Taste RES.

Nachdem alle Karten gelesen sind:

2) - Druck auf die Taste LOAD

- Eingabe der Karten in den Leser in der richtigen Reihenfolge.
- Drücken der Taste RES.  
(Prüfen, ob Leerkarten im Stanzer liegen)

Wiederanlauf nach Unterbrechung

s. 7.5.

7.4. Wiederanlauf nach Unterbrechungen bei Kartenstau auf dem Leser

Stau am Schlitz

- Herausnehmen der Karten.
- Evtl. Erneuern der Karte, die den Stau verursacht hat und diese Karte neu an den Kopf des Kartenpaketes legen.
- Neues Füllen des Zufuhrmagazins.
- Drücken der Taste RES.

Stau auf der Kartenbahn oder am Ablagefach

Wenn es erforderlich ist, das Kartenpaket aus dem Zufuhrmagazin nehmen.

- Öffnen der Verkleidung der Kartenbahn.
- Evtl. das Plastikführungsband herausnehmen. Um dies zu bewerkstelligen, ist der bewegliche Block der Bahn zu öffnen, indem man die Verriegelung öffnet, die sich über dem Block befindet und den beweglichen Block mittels eines Knopfes in der Mitte des Blocks herauszieht.
- Entfernen der zerstörten Karten.

Erneutes Einlegen des Plastikbandes. Um das Band einzulegen, nimmt man das Band mit beiden Händen und führt es um den beweglichen Block, so daß die Lochungen für die Synchronisation nach unten liegen. Nun läßt man das Band herabgleiten, bis es auf den Führungsrollen liegt.

Den Block neu verriegeln

Verschieben des Führungsbandes mittels Daumen und Zeigefinger nach rechts, bis die eine der drei Richtschlitze mit dem Führungsloch am beweglichen Teil des Blocks genau übereinstimmt.

Es ist darauf zu achten, daß das Führungsband nicht in umgekehrter Richtung verschoben wird, da man dabei die Führungsschlitze beschädigt.

Anschließend läßt man eine Leerkarte durch den Leser laufen, indem man auf die Taste EJ des Lesers drückt, um zu sehen, ob noch ein Rest aus einer zerstörten Karte in der Lesebahn steckt und ob das Führungsband genau abläuft.

Danach den Leser verschließen.

Drücken der Taste RES.

Bei vollem Fach:

Herausnehmen des Kartenpaketes und Drücken auf die Taste RES.

#### 7.5. Wiederanlauf nach Unterbrechung beim Stanzen in den Software-Programmen

Die Lampe PU leuchtet.

- Drücken auf die Taste PI.
- Drücken der Taste EJ, um die letzte, korrekt gelochte Karte auszusteuern.
- Herausnehmen der Karten aus dem Ablagefach.
- Drücken auf die Taste RES. Nun wird das Ende der fehlerhaften Karte in eine Leerkarte gestanzt und am Ende dieses Lochvorgangs wird auf der Sichttafel 3 x 1 sichtbar.
- Zweimal Taste EJ drücken.
- Herausnehmen der fehlerhaften Karten aus dem Ablagefach.
- Drücken der Taste CLB.
- Die Arbeit wird normal wieder aufgenommen. In jedem Falle ist die erste Karte im Ablagefach eine Leerkarte.

8. Programmunterlagen

8.1. Allgemeines

Die Programmunterlagen werden vom Programmierer erstellt.

Die in den folgenden Abschnitten beschriebenen Formulare und Listen sind zwar für die Programmierung erforderlich, gehören aber nicht alle zu den obligatorischen Unterlagen. Diese sind in Abschnitt 8.4. besonders erwähnt.

8.2. Die verwendeten Formulare

8.2.1. Block- und Flußdiagramm

Auf diesem Formular wird für jede Arbeit der detaillierte Ablauf aufgezeichnet. Für eine Arbeit können mehrere Blätter verwendet werden.

Bestell-Nr.: 218 (Abb. siehe Seite 247)

8.2.2. Karteneinteilung

Alle Karten, die in einem Programm Verwendung finden, werden hier mit ihrer Einteilung aufgezeichnet. Es müssen sowohl die Eingabekarten wie auch die Summenkarten berücksichtigt werden.

Bestell-Nr.: 202 (Abb. siehe Seite 248)

8.2.3. Druckbild

Auf diesem Formular wird das gewünschte Druckbild schreibstellengerecht eingetragen. Die senkrechte Zeilennumerierung erleichtert das spätere Lochen des Vorschubstreifens.

Bestell-Nr.: 375 (Abb. siehe Seite 249)

8.2.4. Programmliste

Während des Testens entsteht durch das Softwareprogramm "Programmkartenlisten" auf Blankopapier eine Liste der Programmbefehle. Eine Beschreibung dieser Liste ist in dem entsprechenden Absatz in Kapitel 7 gegeben.

Diese Liste ersetzt nach Fertigstellung des Programmes das "Programmformular" (s. 8.2.8.).

(Abb. siehe Seite 255 )

### 8.2.5. Kernspeicherausdruck

Während des Testens wird man häufig einen Kernspeicherausdruck vornehmen.

Nach dem Test, wenn das Programm fertiggestellt ist, gehört ein Kernspeicherausdruck zu den Programmunterlagen. Dieser Ausdruck erfolgt unmittelbar nach dem Laden des Programmes.

Der techn. Kundendienst hat mit dieser Unterlage die Möglichkeit, bei evtl. Störungen die korrekte Eingabe zu kontrollieren.  
(Abb. siehe Seite 256)

### 8.2.6. Bedienungsanweisung

Auf diesem Formular müssen alle notwendigen Angaben enthalten sein, die der Maschinenbediener am Anfang und während der Arbeit benötigt. Ebenfalls müssen die Anweisungen für sein Verhalten bei Störungen gegeben werden.

Bestell-Nr.: 1003 (Abb. siehe Seite 250)

### 8.2.7. Kernspeicherbelegung

Dieses Blatt gibt einen Überblick über die Kernspeicherbelegung mit der Belegung durch das Ladeprogramm.

Dieses Blatt, evtl. in Verbindung mit dem neutralen Speicherbelegungsformular (Best.-Nr. 233) dient dazu, eine Übersicht über die Belegung der Speicherstellen durch Programm, Tabellen und Konstanten zu geben.

Dieses Blatt soll die Belegung nach dem Laden eines Programmes am Anfang der Arbeit wiederspiegeln.

Bestell-Nr.:	1010 (Abb. siehe Seite 251)	1016 für	2500 Bytes
		1014 für	5000 Bytes
		1015 für	10000 Bytes

### 8.2.8. Programmformular

Dieses Formular ist das Grundformular der Kodierung eines Programmes. Es ist gleichzeitig Lochbeleg. Nach Fertigstellung des Programmes werden diese Formulare durch die Programmliste ersetzt.

Bestell-Nr. 1009 (Abb. siehe Seite 252)

### 8.2.9. Allgemeine Beschreibung

Es ist empfehlenswert eine genaue Beschreibung der Arbeit bei den Programmunterlagen zu haben. Diese Beschreibung kann durch die Aufgabenstellung ersetzt werden.

8.2.10. Symbole und Register

Dieses Formular kann als Ergänzung zu dem Formular "Kernspeicherbelegung" dienen; indem es eine Übersicht über die belegten numerischen Register gibt.

Gleichzeitig können auf ihm die verwendeten LEVEL notiert werden, um bei umfangreichen Programmen die Gefahr einer Doppelbelegung zu vermeiden.

Bestell-Nr.: 1001 ( Abb. siehe Seite 253)

8.3. Testunterlagen

Unabhängig von den Programmunterlagen gehört zu jeder Arbeit ein Satz Probe- oder Testkarten und ein mit diesen Karten geschriebenes Druckformular. Mit diesen Unterlagen kann der Operateur vor jeder Programmeingabe die korrekte Arbeit der Maschine prüfen.

8.3.1. Probekarten und 80/80-Liste

Nach dem Laden eines Programmes werden zuerst diese Karten in die Maschine eingelesen. Der Kartensatz wurde vom Prommierer erstellt und berücksichtigt alle in der wirklichen Arbeit vorkommenden Möglichkeiten. Jede Programmserie (Verzweigung) muß von ihm angesprochen werden.

Die 80/80-Liste dieser Karten, wird als Unterlage für den Fall aufbewahrt, daß die Probekarten neu gelocht werden müssen.

8.3.2. Druckformular und Summenkarten

Nach dem Durchlauf dieser Karten durch die Maschine ist das erhaltene Druckbild und die Summenkarten mit der Originalvorlage auf Übereinstimmung zu vergleichen.

#### 8.4. Formularmuster

Auf den folgenden Seiten sind die in Abschnitt 8.2. beschriebenen Formulare in verkleinerter Form abgebildet. Von diesen Formularen gehören die folgenden obligatorisch zu den Programmunterlagen:

- Block- und Flußdiagramm
- Karteneinteilung
- Kernspeicherbelegung
- Druckbild
- Programmliste
- Kernspeicherausdruck
- Bedienungsanweisung
- Allgem. Beschreibung
- Probekarten und 80/80-Auflistung
- Druckformulare und Summenkarten als Muster einer Arbeit mit den Probekarten.



9.65 / 218

Darstellung		T e x t		Verwei- sung
0	1	2		3

Geschäftsstelle :  
Firma :

Name des Programms :  
Nr. :

Erstellt am :  
durch :  
berichtigt am :  
durch :

BLOCKDIAGRAMM  
FLUSSDIAGRAMM


BULL  
GENERAL ELECTRIC

Symbole: Funktion - 1 Ausgang, Entscheidung - 2 Ausgänge (linker Ausgang - Bedingung erfüllt / rechter Ausgang - Bedingung nicht erfüllt)

GE-55

Ref.-Nr.: 23.20.001 D

Juli 1968

Geschäftsstelle: Firma:	Name des Programms: Nr.	Erstellt am: durch: berichtigt am: durch:	<b>KARTENEINTEILUNG</b> 																																																																																																																							
<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1</td><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td><td>50</td><td>55</td><td>60</td><td>65</td><td>70</td><td>75</td><td>80</td></tr> </table> </div> <div style="width: 20%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1</td><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td><td>50</td><td>55</td><td>60</td><td>65</td><td>70</td><td>75</td><td>80</td></tr> </table> </div> <div style="width: 20%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1</td><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td><td>50</td><td>55</td><td>60</td><td>65</td><td>70</td><td>75</td><td>80</td></tr> </table> </div> <div style="width: 20%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1</td><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td><td>50</td><td>55</td><td>60</td><td>65</td><td>70</td><td>75</td><td>80</td></tr> </table> </div> <div style="width: 20%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1</td><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td><td>50</td><td>55</td><td>60</td><td>65</td><td>70</td><td>75</td><td>80</td></tr> </table> </div> <div style="width: 20%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1</td><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td><td>50</td><td>55</td><td>60</td><td>65</td><td>70</td><td>75</td><td>80</td></tr> </table> </div> <div style="width: 20%;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1</td><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td><td>30</td><td>35</td><td>40</td><td>45</td><td>50</td><td>55</td><td>60</td><td>65</td><td>70</td><td>75</td><td>80</td></tr> </table> </div> </div>				1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80																																																																																																										
1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80																																																																																																										
1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80																																																																																																										
1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80																																																																																																										
1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80																																																																																																										
1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80																																																																																																										
1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80																																																																																																										

8 66 / 202

GE-55

Ref.-Nr.: 23.20.001 D

Juli 1968

Geschäftsstelle:										Name des Programms:										DRUCKBILD 10 Stellen je Zoll										BULL GENERAL ELECTRIC									
Firma:										Nr.:										Vorschub 1/2										Druckbild Nr.:									
Erstellt am:										berichtigt am:																													
durch:										durch:																													
1										20										90										140									
2										25										95										145									
3										30										100										150									
4										35										105										155									
5										40										110										160									
6 1"										45										115										165									
7										50										120										170									
8										55										125										175									
9										60										130										180									
10										65										135										185									
11										70										140										190									
12 2"										75										145										195									
13										80										150										200									
14										85										155										205									
15										90										160										210									
16										95										165										215									
17										100										170										220									
18 3"										105										175										225									
19										110										180										230									
20										115										185										235									
21										120										190										240									
22										125										195										245									
23										130										200										250									
24 4"										135										205										255									
25										140										210										260									
26										145										215										265									
27										150										220										270									
28										155										225										275									
29										160										230										280									
30 5"										165										235										285									
31										170										240										290									
32										175										245										295									
33										180										250										300									
34										185										255										305									
35										190										260										310									
36 6"										195										265										315									
37										200										270										320									
38										205										275										325									
39										210										280										330									
40										215										285										335									
41										220										290										340									
42 7"										225										295										345									
43										230										300										350									
44										235										305										355									
45										240										310										360									
46										245										315										365									
47										250										320										370									
48 8"										255										325										375									
49										260										330										380									
50										265										335										385									
51										270										340										390									
52										275										345										395									
53										280										350										400									
54 9"										285										355										405									
Druckzeit, jäh.										1 2 3 4 5 6 7 8 9 10 11 12										1 2 3 4 5 6 7 8 9 10 11 12										1 2 3 4 5 6 7 8 9 10 11 12									
Blätter je Abz.										0 1 2 3 4 5 6 7 8 9 10 11 12										0 1 2 3 4 5 6 7 8 9 10 11 12										0 1 2 3 4 5 6 7 8 9 10 11 12									
Zahl der Kopien										0 1 2 3 4 5 6 7 8 9 10 11 12										0 1 2 3 4 5 6 7 8 9 10 11 12										0 1 2 3 4 5 6 7 8 9 10 11 12									

Bei Druckvorlage gelten nicht die Abmessungen dieses Formulars, sondern folgende Maße für  
 1 Druckstelle: Höhe 1/6 Zoll  
 Breite 1/10 Zoll

GE-55

Ref.-Nr.: 23.20.001 D

Juli 1968

Geschäftsstelle: Firma:		Name des Programms: Nr.:		Erstellt am: durch:		Berichtigt am: durch:		BEDIENUNGSANWEISUNG GE 55				BULL GENERAL ELECTRIC					
				B E A R B E I T U N G		S T Ö R U N G E N											
Bezeichnung der benutzten For- mul. u. Karteien		Kartenart		Eingabe über numer. Tastatur		Eingabe über α-numer. Tastatur		Bemerkungen		Leuchtanzeige		Randeinheit		Art d. Wiedernlaufs		Tastatureingabe	
Kartenleser																	
Kartenschanzer																	
Drucker																	
benutzte Randeinheiten		Karten- leser		Drucker		Karten- stanzer		numerische/numerisch Tastatur									
Allgemeine Angaben zum Pro- gramm																	

GE-55

Ref.-Nr.: 23.20.001 D  
Juli 1968

Geschäftsstelle:		Name des Programms:		Erstellt am:		KERNspeicherBELEGUNG		BULL	
Firma:		Nr.:		durch:		GE 55		GENERAL ELECTRIC	
				berichtigt am:		5000 Bytes			
				durch:					
00	00	00	00	00	00	00	00	00	00
01	01	01	01	01	01	01	01	01	01
02	02	02	02	02	02	02	02	02	02
03	03	03	03	03	03	03	03	03	03
04	04	04	04	04	04	04	04	04	04
05	05	05	05	05	05	05	05	05	05
40									
41									
42									
43									
44									
45									
46									
47									
48									
49									

GE-55



Erläuterung		Sprünge		KA		Zahl der Bytes	Ord.-Nr.		Symbolischer Operationstyp		Befehl					
		F 1	F 2	2	3		4	5	6	9	10	16	17	20 21	24 25	28 29
				1	2	0		0	0							
				1	2	0		0	1							
				1	2	0		0	2							
				1	2	0		0	3							
				1	2	0		0	4							
				1	2	0		0	5							
				1	2	0		0	6							
				1	2	0		0	7							
				1	2	0		0	8							
				1	2	0		0	9							
				1	2	0		1	0							
				1	2	0		1	1							
				1	2	0		1	2							
				1	2	0		1	3							
				1	2	0		1	4							
				1	2	0		1	5							
				1	2	0		1	6							
				1	2	0		1	7							
				1	2	0		1	8							
				1	2	0		1	9							
				1	2	0		2	0							
				1	2	0		2	1							
				1	2	0		2	2							
				1	2	0		2	3							
				1	2	0		2	4							
				1	2	0		2	5							
				1	2	0		2	6							
				1	2	0		2	7							
				1	2	0		2	8							
				1	2	0		2	9							
				1	2	0		3	0							
				1	2	0		3	1							

12.66 / 1009

Buchstabe O = Ø, Ziffer Null = 0

GE-55

Ref.-Nr.: 23.20.001 D

Juli 1968







C.NK.	NUM.	IMP.	TO.	CODES
15	5	1101		1101
14	02	10	1101 LEVEL	F201
12	03	15	1101 MRR	E12406
12	06	20	1104 INC	8200104400220
12	04	25	1110 IOC	53019A12
12	04	30	1114 IOC	53019210
12	04	35	1118 IOC	53019A12
12	04	40	1122 M6R	21029725
12	04	45	1126 IOC	53019210
12	04	50	1130 M6R	21029726
12	03	55	1134 MRR	E11902
14	02	60	1137 LEVEL	F202
12	03	65	1137 MSRDR	E30014
12	03	70	1140 MRR	E11501
12	03	75	1143 ADD	A12501
12	06	80	1146 MVC	800110000235
12	03	85	1152 MRR	E11501
12	03	90	1155 ADD	A12601
12	06	95	1158 MVC	800110000240
12	03	100	1164 MSRDR	E30030
12	03	105	1167 MRR	E11501
14	02	110	1170 LEVEL	F203
12	06	115	1170 MVC	800110010165
12	03	120	1176 ADD	A11329
12	03	125	1179 ADD	A10514
12	03	130	1182 ADD	A10501
12	03	135	1185 CML	911425
12	06	140	1188 JIERT	043C0086F103
12	03	145	1194 MRR	E11501
12	03	150	1197 MSRDR	E30014
14	02	155	1200 LEVEL	F204
12	06	160	1200 MVC	800110010165
12	03	165	1206 ADD	A11330
12	03	170	1209 ADD	A10514
12	03	175	1212 ADD	A10501
12	03	180	1215 CML	911426
12	06	185	1218 JIERT	043C0086F104
12	03	190	1224 MRR	E11501
12	03	195	1227 MRR	E10906
12	06	200	1230 INC	820014121088
12	04	205	1236 IOC	53019606
12	02	210	1240 TR1	5106
14	02	215	1242 LEVEL	F205
12	06	220	1242 MVC	800260826162
12	06	225	1248 INC	822060826080
12	03	230	1254 ADD	A11706
12	04	235	1257 IOC	53019606
12	02	240	1261 TR1	5106
12	03	245	1263 MSRDR	E30014
12	03	250	1266 MRR	E10906
12	03	255	1269 ADD	A12906

Programmliste des GE-55

GE-55

Ref.-Nr.: 23.20.001 D

Juli 1968

0026	2020321100	2191934680	0000008000	0000800000	0080000000
0051	8000000081	0000000000	0000000000	0000008531	0000000000
0076	0000000010	5040302000	3C92459738	0000009672	0000000000
0101	0000000801	0000000270	0000000005	0000000334	0000000001
0126	0000009255	0000000000	0000009040	0000009173	0000009346
0151	800EF40151	110FF40156	0000000000	0000000000	0000000000
0176	0000000000	0000000000	0000000000	0000000000	0000000000
0201	0000000000	0000000000	0000000000	0000000000	0000000000
0226	0000000000	0000000000	0000000000	0000000000	0000000000
0251	0000000060	0000000060	0000000750	0000000006	0000000000
0276	0000000000	0000000000	0000000000	0000000000	0000000000
0301	0000000000	0000000000	0000000000	0000000000	0000000000
0326	0000000000	0000000000	0000000000	0000000000	0000000000
0351	0000000000	0000000000	0000000000	0000000000	0000000000
0376	0000000000	0000000000	0000000000	0000000000	0000000000
0401	0000000000	0000000000	0000000000	0000000000	0000000000
0426	0000000000	0000000000	0000000000	0000000000	0000000000
0451	0000000000	0000000000	0000000000	0000000000	0000000000
0476	0000000000	0000000000	0000000000	0000000000	0000000000
0501	0000000000	0000000000	0000000000	0000000000	0000000000
0526	0000000000	0000000000	0000000000	0000000000	0000000000
0551	0000000000	0000000000	0000000000	0000000000	0000000000
0576	0000000000	0000000000	0000000000	0000000000	0000000000
0601	5349454754	414C202D20	4252415545	5245495341	43484B4F4E
0626	54454E2020	202A202020	4B52454449	544F52454E	313936384B
0651	544F2E4E52	2E44415435	4D20472E4B	542E42454C	2E234A2D53
0676	5445585420	2F2053594D	424F4C534B	4F4E544F53	4F4C4C2020
0701	4D57535448	4142454E4D	5753545355	4D4D452020	534F4C4C20

GE-55

UMSCHLÜSSELUNGSTABELLE

Zeichen	Zwischen code	Lochcode	Lochcode	Zeichen	Zwischen code	Lochcode	Lochcode	ISO code	
0	20	0	0	P	57	11	7	50	
1	30	1	1	Q	58	11	8	51	
2	31	2	2	R	59	11	9	52	
3	32	3	3	S	22	0	2	53	
4	33	4	4	T	23	0	3	54	
5	34	5	5	U	24	0	4	55	
6	35	6	6	V	25	0	5	56	
7	36	7	7	W	26	0	6	57	
8	37	8	8	X	27	0	7	58	
9	38	9	9	Y	28	0	8	59	
•	4B	12	8	Z	29	0	9	5A	
/	21	0	1	⌋	4C	12	8	4	5D
#	3B	8	3	@	3C		8	4	40
%	2C	0	8	&	40	12		26	
-	50	11	2D	⌋	2B	0	8	3	2C
*	5C	11	8	⌋	5F	11	8	7	27
A	41	12	1	(	4D	12	8	5	28
B	42	12	2	<	4E	12	8	6	3C
C	43	12	3	\	4F	12	8	7	5C
D	44	12	4	↑	70	11	0	5E	
E	45	12	5	:	5E	11	8	6	3B
F	46	12	6	=	2D	0	8	5	3D
G	47	12	7	"	2E	0	8	6	22
H	48	12	8	!	2F	0	8	7	21
I	49	12	9	>	3E		8	6	3E
J	51	11	1	?	3F		8	7	3F
K	52	11	2	←	2A	0	8	2	5F
L	53	11	3	:	3D		8	5	3A
M	54	11	4	+	60	12	0	2B	
N	55	11	5	⌋	3A		8	2	5B
O	56	11	6	)	5D	11	8	5	29

I 41 halbe Druckgeschwindigkeit

SPALTE	0	1	2	3	4	5	6	7	ISO CODE
Binär darstell.	0 0	0 1	0 1	0 1	0 1	0 1	0 1	0 1	ZEILE
0000	0	1	2	3	4	5	6	7	0
0001	1	2	3	4	5	6	7	8	1
0010	2	3	4	5	6	7	8	9	2
0011	3	4	5	6	7	8	9	A	3
0100	4	5	6	7	8	9	A	B	4
0101	5	6	7	8	9	A	B	C	5
0110	6	7	8	9	A	B	C	D	6
0111	7	8	9	A	B	C	D	E	7
1000	8	9	A	B	C	D	E	F	8
1001	9	A	B	C	D	E	F		9
1010	A	B	C	D	E	F			A
1011	B	C	D	E	F				B
1100	C	D	E	F					C
1101	D	E	F						D
1110	E	F							E
1111	F								F

ANSCHLUSSCODES FÜR RANDEINHEITEN

CC	R	Funktion
96	06	Kartenleser
AA	09	Kartenstanzer
CA	08	Drucker MB 50
9A	12	Freigabe num. Tastatur N
9A	11	Freigabe Sichtanzeige V
94	R	Alphatastatur A
9A	10	Übertrag Zentralspeicher Puffer
92	10	Übertrag Puffer Zentralspeicher
10	R	Trommel, Lesen
12	R	Trommel Schreiben
13	R	Trommel Auswahl der Bahn
11	R	Trommel Test
1E	08	Drucker I 41 Druck
1D	08	Drucker I 41 Test

FUNKTIONSSCHLÜSSEL I 41

48Z	64Z	Funktion
3C	4C	keine Zwischenzeile
3D	4D	Einfacher Zeilenvorschub
3E	4E	Doppelter Zeilenvorschub
3F	4F	Dreifacher Zeilenvorschub
21	51	Papieranschub und Stop auf Kanal A
22	52	Papieranschub und Stop auf Kanal B

Testzeichen an Basisadresse: 10 = nicht volles Blatt  
30 = volles Blatt

### GE 55 Befehle

Sprünge		Logische Operationen		
NØP	00	CMD	91 R1 R2	
JRT	02 F1 xx	CMDD	92 RR1 RR2	
JIRT	04 K Bx xx F1xx	CMC	88N Bx xx Bx xx	
JUIRT	05 K Bx xx F1xx	NI	61K Bx xx	
LEVEL	F2 xx	ØI	63K Bx xx	
Überträge		Arithmetische Operationen		
MRR	E1 R1 R2	ADD	A1 R1 R2	
MRRR	E2 RR1 RR2	ADDD	A2 RR1 RR2	
MSRR	E3 R1 RR2	SBD	B1 R1 R2	
EXRR	E5 R1 R2	SBDD	B2 RR1 RR2	
EXDR	E6 RR1 RR2	RØUND	71 K RR	
LRN	60 N Bx xx R(R)	MPD	C1 R1 RR2	
MVC	80 N Bx xx Bx xx	DVD	DI R1 RR2	
MVIC	8A R Bx xx Bx xx	Verschiebungen		
PKH	84 N Bx xx Bx xx	SLD	70 N RR	
UPH	85 N Bx xx Bx xx	SRD	72 N RR	
INC	82 K Bx xx Bx xx	EIN-/AUSGABE - Überträge		
MR8		12 N R Dc	M7R	31 N Db...
MR8G		1E N R Dc	M78	32 N Db...
MP9		14 N R Dc	M78G	3E N Db...
MR89		16 N R Dc	M7R8	33 N Db...
M6R		21 N Da R	M79	34 N Db...
M68		22 N Da Dc	M789	35 N Db...
M68G		2E N Da Dc	M7R89	36 N Db...
M6R8		23 N Da R Dc		37 N Db...
M6R9		24 N Da Dd		
M689		25 N Da R Dd		
M689		26 N Da Dc Dd		
M6R89		27 N Da R Dc Dd		
EIN-/AUSGABE		EIN-/AUSGABE		
IOc	53 N CC1 R1--CCn	Rn		
IOiC	54 N RCC1 R1--CCn			
TR1	51 R H14.012 → ISO			
TRB	5B R ISO → H14.012			
TRD	5D R (141 → 64 Zeichen)			
Hilfsbefehle		Multiprogramming		
PRSTØ	78	PRØ	42 N	
LØAD	7D	FREE	44 N	
RLA	6A 00 Bxxx	START	41 N	
KHLT	57	STØP	40	
Reservierte Speicherstellen		Reservierte Speicherstellen		
0085	1st. Vorzeichen	0091	1st. Kapazitätsüber-	
0086	1. Vergleichs-Ergeb.	0093	schreibung	
0088	2. Rücksprung-		RSA nach FP	
	Adresse			
0090 (2)	P A R	0095	2 Fehleranzeigepro.	
Division: RI=15, RR2=14; UP Div. ≈ 250 Bytes				

2 Byte, daher Adressen 89, 90

\* Halt: Nur wirksam, wenn Taste gedrückt ist.





**KARTENEINTEILUNG**

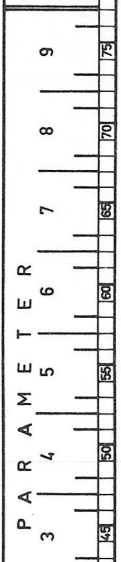


Erstellt am:  
durch:  
berichtigt am:  
durch:

Name des Programms:  
Nr.

Geschäftsstelle:  
Firma:

KA 15 PROGRAMMLEITKARTE																				
Sp. 6 - 9 Ordnungsnummer																				
Sp. 17 - 20 Speicherstelle des ersten einzugehenden Befehls																				
15	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80					
Ord. Nr.			1. Stelle																	
KA 12 BEFEHLSKARTE																				
Sp. 4 - 5 Anzahl der Stellen																				
Sp. 6 - 9 Ordnungsnummer																				
Sp. 10 - 16 symbolischer OT																				
Sp. 17 - 32 Befehlscode																				
KA 14 NIVEAUKARTE																				
Sp. 6 - 9 Ordnungsnummer																				
Sp. 17 - 20 F2 und 14d. Nr. des Niveaus ( von 00 - 99)																				
KA 17 PROGRAMMENDKARTE																				
Sp. 6 - 9 Ordnungsnummer																				
KA 31 KONSTANTENKARTE																				
Sp. 4 - 5 Anzahl der Stellen																				
Sp. 6 - 9 1. Stelle im Speicher																				
Sp. 17 - 80 Konstanten im Format 1 (1 Spalte je Byte)																				
KA 32 KONSTANTENKARTE																				
wie KA 31, jedoch Konstanten im Format 2 ( 2 Spalten je Byte)																				
KA 16/36 PROGRAMMSTARTKARTE																				
Sp. 6 - 9 Ordnungsnummer																				
Sp. 10 - 13 Stelle der ersten Adresse des Befehls, mit dem angefangen werden soll.																				
KA 18/19 PROGRAMMLISTKARTE																				
Ka 18 Startkarte																				
Ka 19 Stopkarte																				







# **BULL**

## **GENERAL ELECTRIC**

### **BULL GENERAL ELECTRIC GmbH Deutschland**

5000 Köln-Mülheim  
Wiener Platz 2, Bullhochhaus  
Ruf 61 63 11, Telex 08 873372

Technischer Kundendienst  
5000 Köln-Niehl  
Nesselrodestraße 26-30  
Ruf 74 53 88/89  
Telex 08 881 141

5000 Köln-Dellbrück  
Lochkartendruckerei  
Bergisch-Gladbacher Straße 1025  
Ruf 68 18 46

#### Geschäftsstellen in:

1000 Berlin 30, Keithstraße 2-4  
Ruf 24 90 66

4800 Bielefeld, Bahnhofstraße 47  
Ruf 6 93 96/97

2800 Bremen, Breitenweg 25  
Ruf 31 32 90

4000 Düsseldorf, Immermannstr. 15  
Ruf 36 07 21/25

4300 Essen, Kopstadtplatz 13  
Ruf 22 73 45/46

6000 Frankfurt a. M., Telemannstraße 1-3  
Ruf 72 79 41/43  
Umstellungszentrum GE-55

2000 Hamburg 1, Steindamm 62  
Ruf 24 01 45

3000 Hannover, Podbielskistraße 32  
Ruf 66 19 31/66 24 86  
Umstellungszentrum GE-55

3500 Kassel, Friedrich-Ebert-Straße 48  
Ruf 7 48 11

5000 Köln, Barbarossaplatz 3-5  
Ruf 21 03 43/44

Umstellungszentrum GE-55  
Bonner Straße 180, Tel. 38 01 85

6800 Mannheim, Augustaanlage 13  
Ruf 4 69 44/45

8000 München 2, Rosental 19  
Ruf 26 40 05  
Umstellungszentrum GE-55

8500 Nürnberg, Dennerstraße 4  
Ruf 26 81 11

6600 Saarbrücken  
Saargemünder Straße 18  
Ruf 6 70 61

7000 Stuttgart N, Hauptmannsreute 13  
Ruf 62 69 64/65

### **Österreich BULL GENERAL ELECTRIC Aktiengesellschaft**

1030 Wien III  
Gigergasse 1 / 3. Stock  
Ruf 72 26 78, Telex 01 27 43  
Telegramm bullge wien

#### Niederlassungen in:

4020 Linz, Humboldtstraße 39  
Ruf 2 76 25

8010 Graz, Schmiedgasse 40  
Ruf 8 86 70

#### Lohnarbeitsbetriebe in:

1030 Wien III, Gigergasse 1  
Ruf 72 26 87

4020 Linz, Fadingerstraße 26  
Ruf 2 41 03

### **Schweiz BULL GENERAL ELECTRIC**

8021 Zürich, Lagerstraße 47  
Telefon 051 23 67 60

#### Geschäftsstellen in:

8021 Zürich, Lagerstraße 47  
Telefon 051 23 67 60

4000 Basel, Dufourstraße 32  
Telefon 061 24 88 40

3000 Bern, Mittelstraße 32  
Telefon 031 23 17 22

1200 Genf, 47, rue du 31-Décembre  
Telefon 022 35 74 10