

Funktionsweise:

Während des Ablaufs der IOC werden folgende Merkmale des Befehls in die IOC-Zone (ZES - ZRC) des Zentralspeichers eingegeben:

- Anzahl der Randeinheiten (N)
- Zonenadresse für jede Randeinheit ( $R_1 - R_n$ )
- Anschlußcode für jede Randeinheit ( $C_1 - C_n$ )

Am Ende dieser Eingabe stoppt das Programm und geht in Wartestellung.

Durch Absuchen der IOC-Zone werden die eingegebenen Merkmale ausgewertet und die entsprechenden Randeinheiten angeschlossen.

Durch gleichzeitiges Bearbeiten mehrerer Randeinheiten wird die Arbeit wesentlich beschleunigt.

Die genaue Funktion dieser "Simultaneität" wird in einem späteren Kapitel beschrieben.

Die Zeichen werden Byte für Byte in den Zentralspeicher geschrieben bzw. aus dem Speicher gelesen. Die Bearbeitung beginnt bei der Adresse +1, welche im entsprechenden Register steht. Diese Adresse wird nach jedem Zeichen + 1 gezählt.

Das Ende der jeweiligen Zone ist durch eine Trennmarke F4 gekennzeichnet. Diese Trennmarke steht in dem Byte, das dem letzten benutzten Byte der Speicherzone folgt. Die Trennmarke wird vor der Arbeit vom Programmierer durch eine INC gesetzt.

Diese Trennmarke zeigt auch das Ende der Arbeit mit der entsprechenden Randeinheit an.

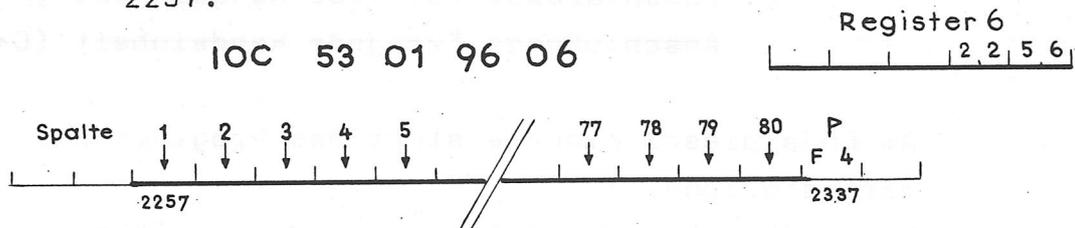
Wenn die Arbeit mit einer Randeinheit beendet ist, wird in der IOC-Zone (ZRC) die Anzahl der anzuschließenden Randeinheiten - 1 gezählt.

Wenn  $N = 0$ , wird das Programm wieder gestartet und der nächste, der IOC folgende Befehl ausgeführt.

Anmerkung: Die Wartestellung des Programmes während IOC, kann bei der Multiprogrammierung zur Bearbeitung eines anderen Programmes ausgenutzt werden.

Programmbeispiele für IOC (1. Randeinheit pro Befehl)

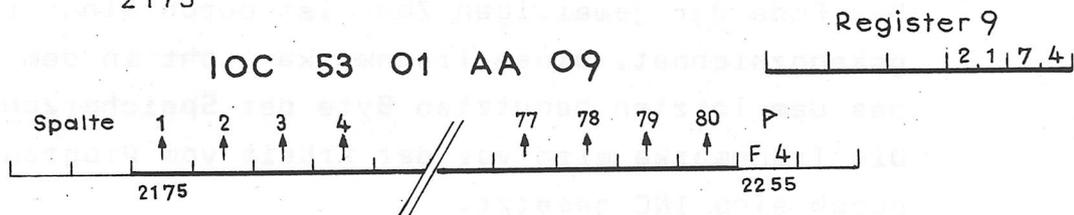
Leser: Speichern von 80 Spalten einer Karte, ab Adresse 2257:



Der Leser kann nicht zwischen 2 Zeichen halten. Es wird also jeweils eine Karte, ohne Unterbrechung gelesen.

Eine Speicherzone kann auch die Lesung mehrerer Karten hintereinander speichern. Hierzu muß die Trennmarke F4 hinter die Spalte 80 der letzten Karte gesetzt werden.

Stanzer: Stanzen von 80 Spalten einer Karte ab Adresse 2175



Der Stanzer kann in jeder Spalte halten. Es ist also möglich, eine Karte für mehrere IOC-Befehle zu verwenden.

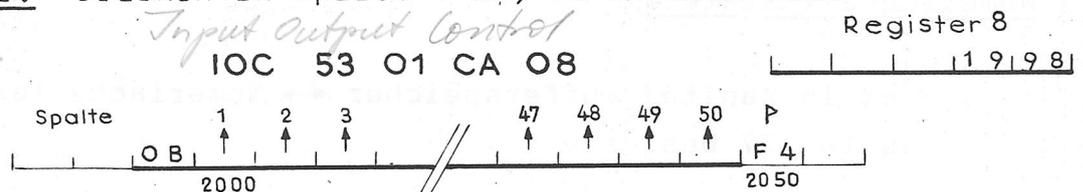
Das erste Zeichen der nächsten IOC wird in die Spalte gestanzt, in der bei der vorhergehenden IOC gestoppt wurde.

Man kann gleicherweise mehrere Karten in der selben IOC stanzen. Nach Stanzen der Spalte 80 wird automatisch eine neue Karte zugeführt. Das nächste Zeichen (81.) wird dann in die Spalte 1 dieser Karte gestanzt.

Wenn man nur einige Spalten stanzen will, kann man die Karte sofort nach Stanzen des letzten Zeichens ablegen. Dies geschieht durch den Code OD, der sofort vor der Trennmarke plaziert ist.

Es gibt keinen Code für Tabulation. Im Bedarfsfall kann hier mit wiederholtem Spaltensprung gearbeitet werden.

Drucker: Drucken in Spalte 1-50, ab Adresse 2000



Der Drucker kann in jeder Spalte halten.

Der Abdruck einer Speicherzone beginnt da, wo der Wagen steht.

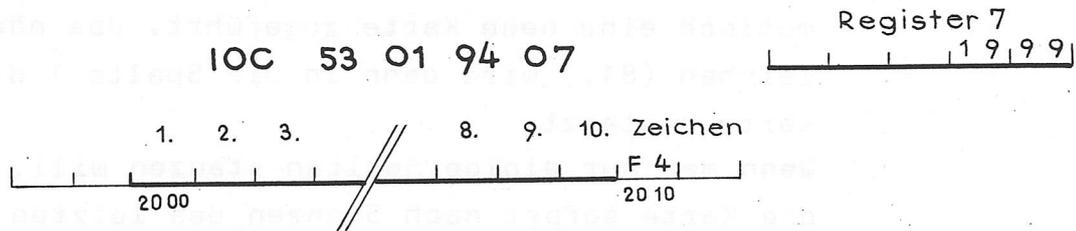
Für die genaue Plazierung des Abdrucks ist es daher notwendig, einen entsprechenden Programmcode in die Speicherzone zu plazieren.

Folgende Programmcodes können verwendet werden:

- OA: Zeilensprung
- OB: Zeilensprung mit Wagenrücklauf
- OC: Papiersprung mit Stop durch Steuerband (Band pilote)
- OD: Wagenrücklauf.

Nach Abdruck in Spalte 130 erfolgt automatisch ein Wagenrücklauf.

Alpha-Tastatur: Eingabe von 10 Zeichen, ab Adresse 2000.



Die Freigabe der Tastatur wird durch die Lampe A angezeigt.

Bei Lesen von F4 wird die Lampe gelöscht und die Tastatur verriegelt.

Numerische Tastatur:

Ist in Kapitel Pufferspeicher ↔ Numerische Tastatur, Seite 137 beschrieben.

		1. Randeinheit		2. Randeinheit		n. Randeinheit		
IOIC	TO	N	RC <sub>1</sub>	R <sub>1</sub>	RC <sub>2</sub>	R <sub>2</sub>	RC <sub>n</sub>	R <sub>n</sub>
	54							
	1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte

*Input Output Indirect Control.*

Dieser Befehl entspricht der IOC, mit dem Unterschied, daß der Anschlußcode nicht im Befehl festliegt, sondern im rechten Byte des Registers RC gespeichert ist.

## 5.9 Übersetzungen

5.9.1 Allgemeines: Nach Lesen einer Karte im BULL- oder IBM-Code, kann die entsprechende Speicherzone nicht sofort ausgewertet werden. Sie enthält einen Zwischencode, der erst in den internen Code übersetzt werden muß. (s. Tafeln Seite )

Das gleiche gilt auch für eine Eingabe über die Alpha-Tastatur (BULL-Kartencode).

Umgekehrt muß eine Speicherzone (im internen Code) vor dem Stanzen in den jeweiligen Zwischencode (BULL oder IBM) übersetzt werden.

Nach der Übersetzung besetzt das Zeichen den gleichen Platz (gleiches Byte) wie vorher. Es gibt keinen Zonenwechsel.

Die Übersetzung erfolgt Byte für Byte, von links nach rechts.

Sie beginnt bei der Basisadresse +1 und endet, wenn die Trennmarke (F4) gelesen wird.

Achtung: Eine zu übersetzende Zone darf keinen Programmcode (0A, 0B, 0C, 0D) enthalten.

Anmerkung: Es gibt im GAMMA 55 2 Platten (Standards), eine für den Leser und eine für den Stanzer, die den Wechsel der Codes erlaubt.

5.9.2 TRO

TO	
50	R
1.Byte	2.Byte

*Transcode*

Übersetzung des Code T 121 (8 Bits) in den ISO-Code (7 Bits)

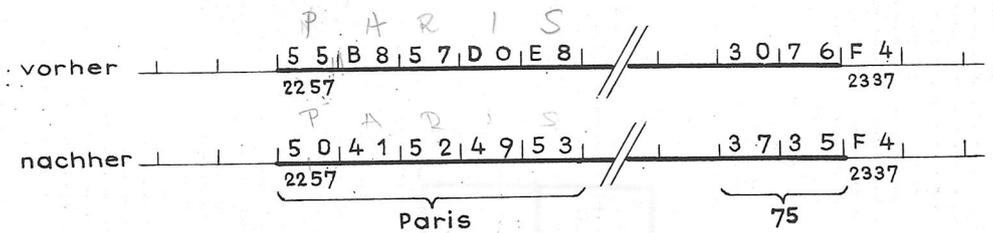
Beispiel:

Übersetzen von 80 Zeichen (T121 → ISO), ab Adresse 2257.

TRO 50 06

Register 6

| 2 | 2 | 5 | 6 |



5.9.3 TR1

TO	
51	R
1.Byte	2.Byte

Übersetzung des Code H14.012 (7Bits) in den ISO-Code (7 Bits).

5.9.4 TRA

TO	
<b>5A</b>	<b>R</b>
1.Byte	2.Byte

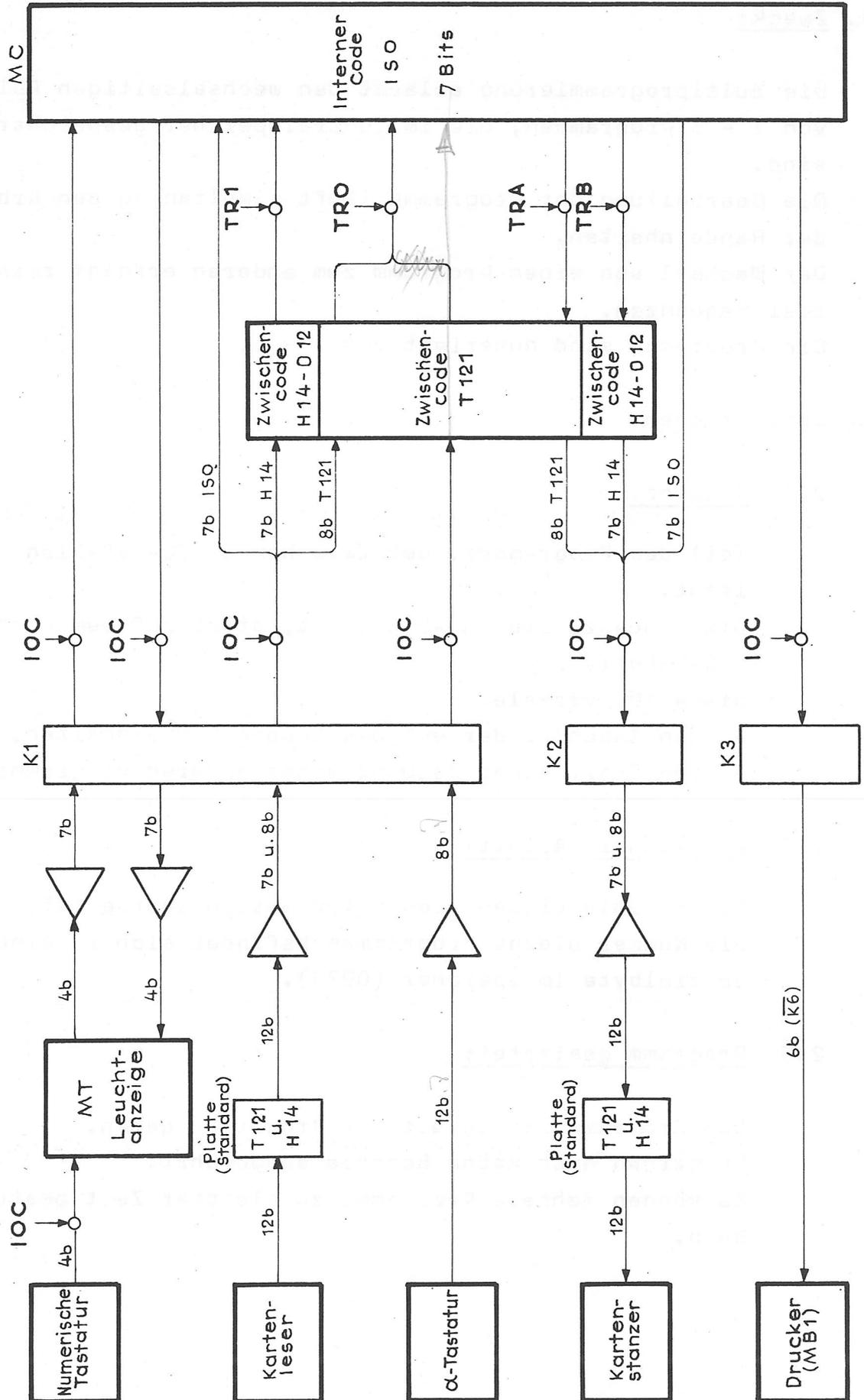
Übersetzung des ISO-Code (7Bits) in den  
Code T121 (8 Bits)

5.9.5 TRB

TO	
<b>5B</b>	<b>R</b>
1.Byte	2.Byte

Übersetzung des ISO-Code (7Bits)  
in den Code H14.012 (7 Bits)

Zusammenfassendes Schema für IOC und Übersetzung



## 1. Zweck:

Die Multiprogrammierung erlaubt den wechselseitigen Ablauf von 1 - 5 Programmen, die im Zentralspeicher gespeichert sind.

Die Bearbeitung der Programme läuft simultan zu den Arbeiten der Randeinheiten.

Der Wechsel von einem Programm zum anderen erfolgt zwischen zwei Sequenzen.

Die Programme sind numeriert von 1 - 5.

## 2. Definitionen:

### 2.1 Sequenz:

Teil des Programmes, der zwischen 2 IOC-Befehlen liegt.

Die Sequenz, die in Arbeit ist, stoppt auf dem nächsten IOC-Befehl.

Diese IOC veranlaßt:

- den Anschluß der entsprechenden Randeinheiten.
- den Start einer Sequenz eines anderen Programmes.

### 2.2 Programm in Arbeit:

Die Befehle dieses Programmes werden bearbeitet.

Die Nummer dieses Programmes befindet sich in einem Spezialbyte im Speicher (0070).

### 2.3 Programm gestartet:

Das Programm ist bereit, in Arbeit zu gehen.

Es werden noch keine Befehle ausgeführt.

Es können mehrere Programme zu gleicher Zeit gestartet sein.

Die Nummern der gestarteten Programme stehen entsprechend ihrer Startfolge in der Zone ZLP (0070 - 0074).  
Ein Programm kan gestartet sein durch einen Spezialbefehl "Start".

#### 2.4 Programm in Wartestellung

Sobald bei der Bearbeitung eines Programmes eine IOC auftritt, geht dieses Programm in Wartestellung. Es bleibt in dieser Wartestellung, bis die angeschlossenen Rand-einheiten ihre Arbeit beendet haben.

Wenn ein Programm in Wartestellung geht, wird das, als nächstes gestartete Programm in Arbeit gesetzt.

Wenn die Randeinheiten ihre Arbeit beendet haben, wird das Programm wieder in die Zone ZLP eingesetzt.

Es steht dann hinter dem zuletzt gestarteten Programm und wird genau wie dieses behandelt.

#### 2.5 Programm gestopt:

In diesem Fall ist das Programm durch einen Befehl "Stop" angehalten worden.

Die Nummer dieses Programmes wird in die Zone ZAT (0076 - 0084) gespeichert.

Diese Zone wird bei den Befehlen "STOP", "START", "PRO" und "FREE" benutzt. Jedes Byte dieser Zone ist in die Halbbytes "n" (rechts) und "p" (links) unterteilt.

### 3. Spezielle Befehle für Multiprogrammierung

#### 3.1

STOP

TO
40
1.Byte

Dauer: 1,2 ms

Der Befehl "STOP" unterbricht das in Arbeit stehende Programm und speichert die Nummer dieses Programmes in die Zone ZAT. Das Programm wird gewechselt. Ein durch diesen Befehl gestopptes Programm kann nur durch den Befehl "START" wieder gestartet werden.

S T O P

↓

Programm in Arbeit stoppt

↓

Eingabe von n = 0 und p = Nummer des Programmes in die Zone ZAT.

↓

Wechsel des Programmes

3.2

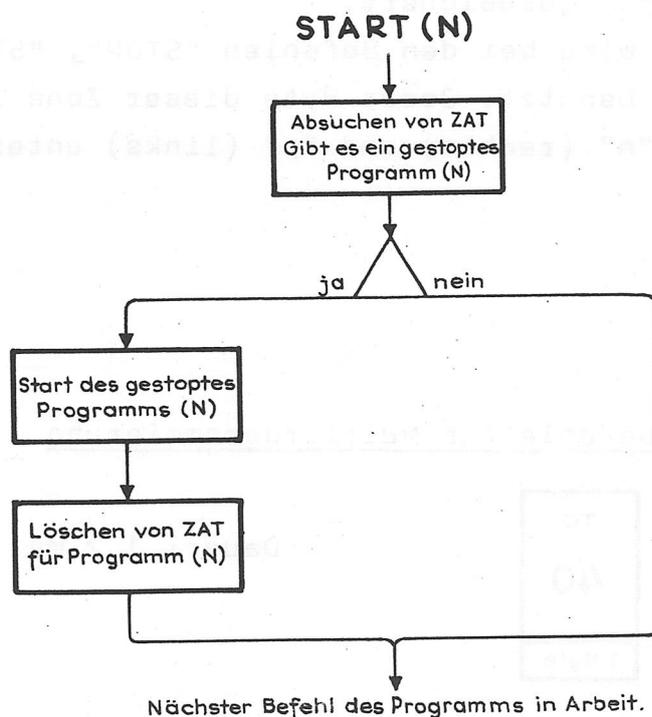
START

TO	Nummer des Progr.
41	N
1.Byte	2.Byte

Start eines Programmes, das durch einen Stop angehalten wurde.

Sonderfall:

Wenn das Programm, das gestartet werden soll, nicht durch den Befehl STOP angehalten wurde, bleibt der Befehl START ohne Wirkung.



3.3

PRO

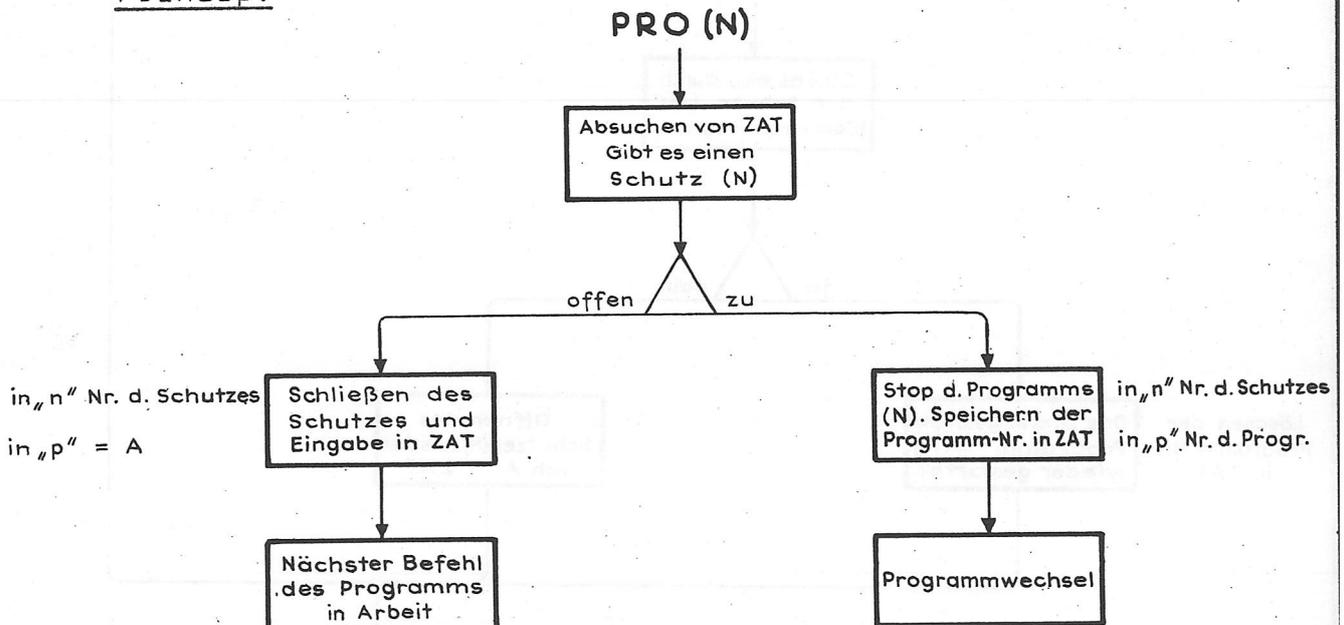
To	Nummer d. Schutzes
42	N
1. Byte	2. Byte

Schutz eines Programmteils, welcher dasselbe Organ (Kanal, Speicherzone, Unterprogramm ...) wie ein anderes, simultan. ablaufendes Programm benutzt.

Ein Schutz "N" sagt "offen" oder "zu".

- offen: es steht kein Schutzmerkmal in ZAT. Das Programm läuft weiter ab.
- zu : ein Byte von ZAT enthält in n = Nummer des Schutzes p = A. Dies untersagt den weiteren Ablauf des Programmes.

Prinzip:

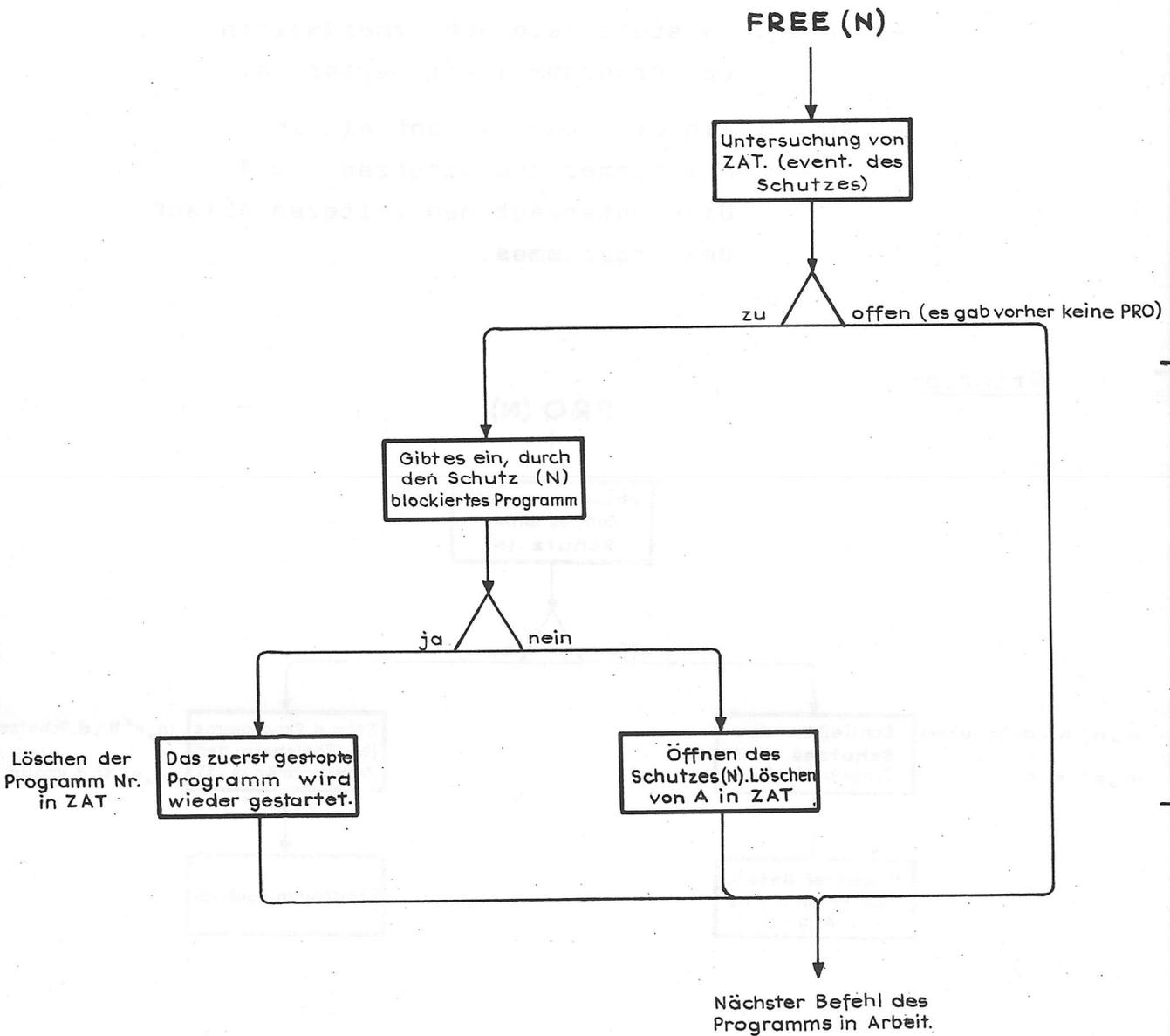


3.4 FREE

TO	Nummer d.Schutzes
44	N
1.Byte	2.Byte

Dieser Befehl hebt den Schutz (N) auf und erlaubt den weiteren Ablauf des Programmes.

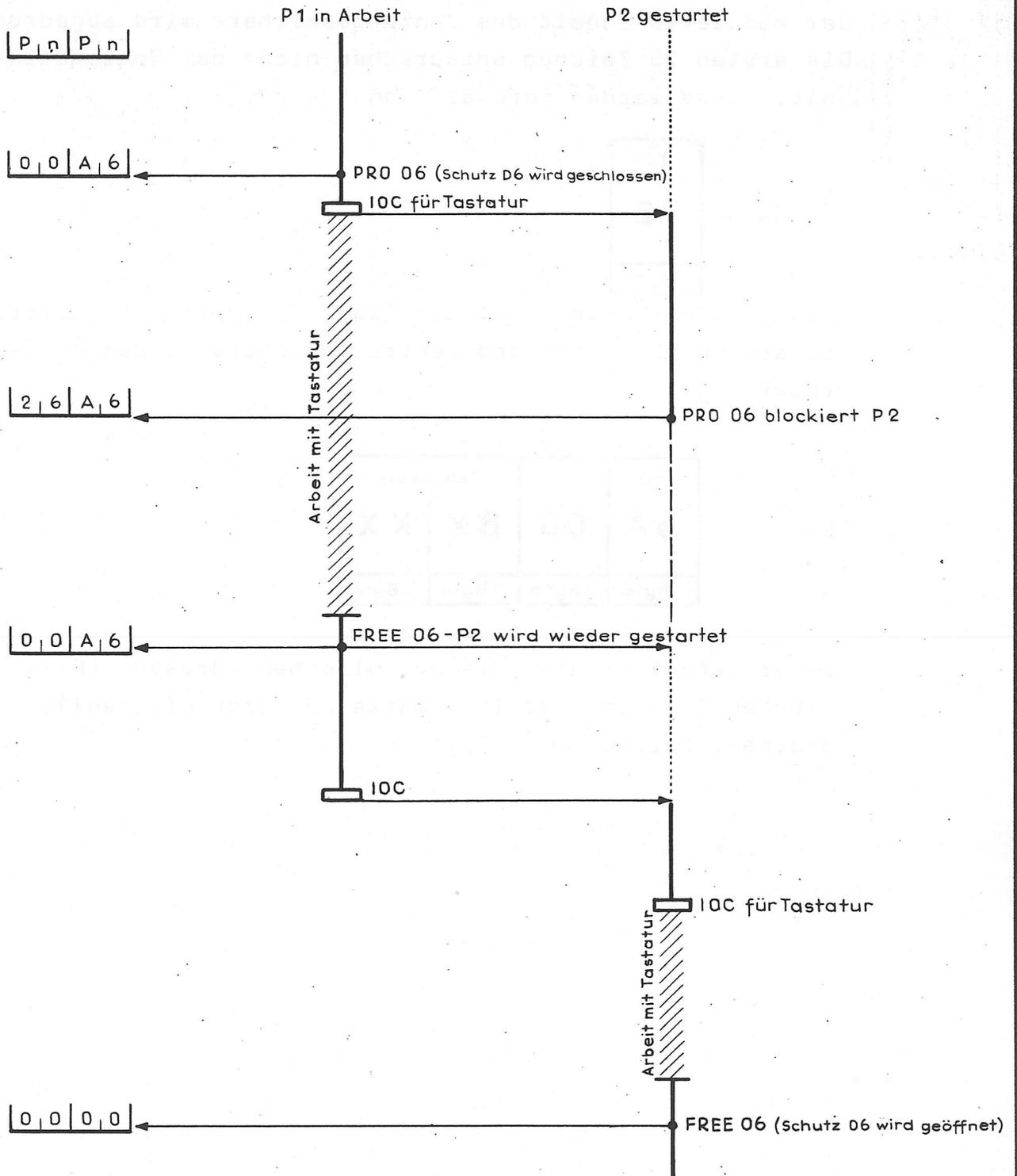
Prinzip:



4. Beispiel für die Verwendung von PRD und FREE

Zwei Programme, P1 + P2 sollen mit der numerischen Tastatur arbeiten. P1 ist in Arbeit, P2 ist gestartet.

Der verwendete Schutz trägt die Nummer 6.



Die Blockierung von P2 verhindert die gleichzeitige Benutzung der Tastatur durch 2 Programme.

Befehle für die Programmerstellung

PRSTO

TO
<b>7.8</b>
1.Byte

Dieser Befehl entspricht der Taste PRS (print store).  
 Der komplette Inhalt des Zentralspeichers wird abgedruckt.  
 Die ersten 25 Zeichen entsprechen nicht dem Speicherinhalt.  
 Diese werden forciert von 1 - 25.

PRREG

TO
<b>7E</b>
1.Byte

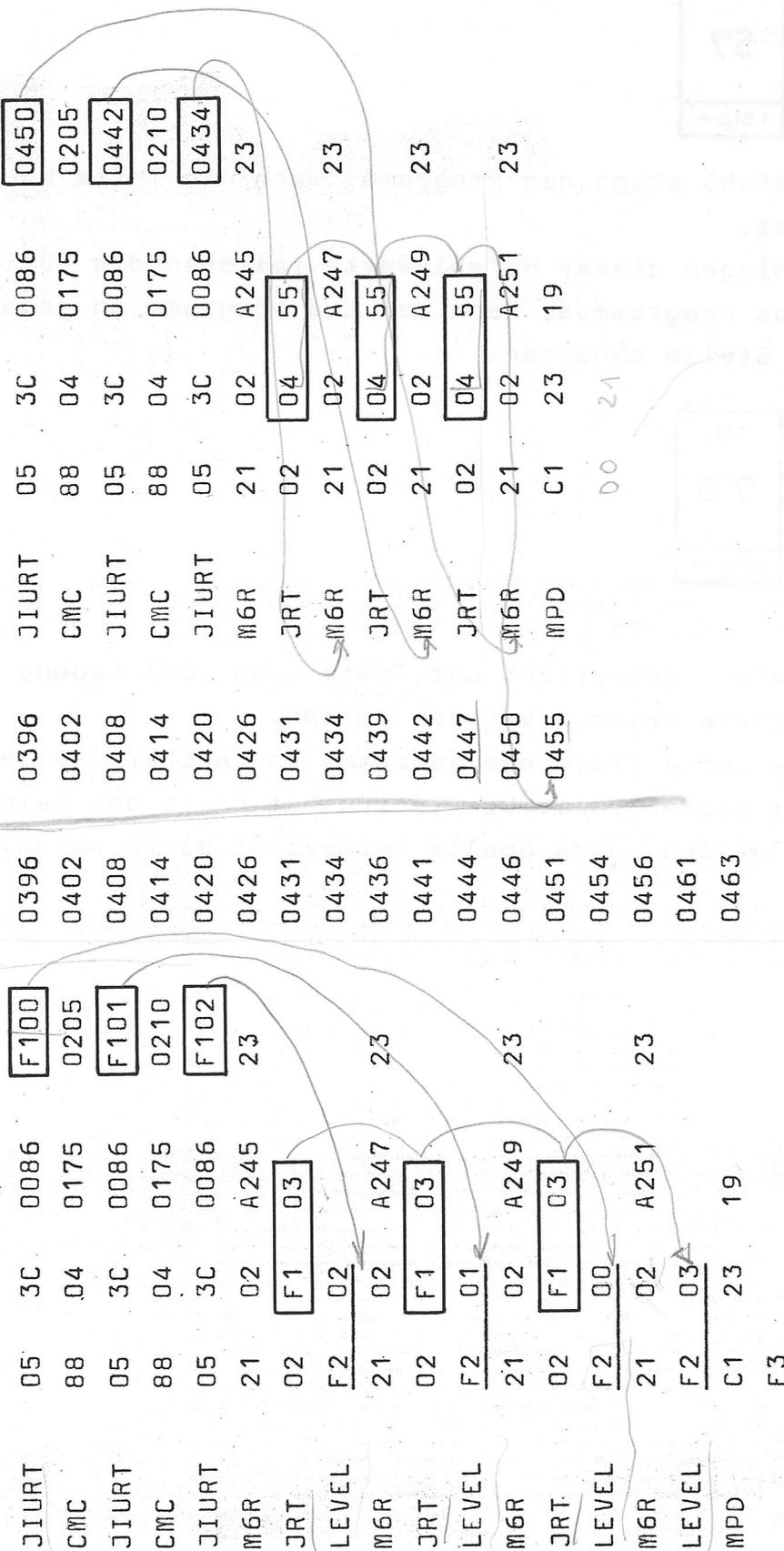
Dieser Befehl entspricht der Taste PRR (print registers).  
 Die ersten 600 Bytes des Zentralspeichers werden abgedruckt.

RLA

TO		Adresse	
<b>6A</b>	<b>00</b>	<b>BX</b>	<b>XX</b>
1.Byte	2.Byte	3.Byte	4.Byte

Dieser Befehl ersetzt die symbolischen Adressen (F1xx) zwischen Bxxx und der Trennmarke F3 durch die reelle Adressen. (siehe LEVEL Seite *46*)

Beispiel:



Durch diesen Befehl wird das Programm außerdem komprimiert, da alle Levels entfallen.  
 Der gewonnene Speicherplatz ist (Anzahl F2 · 2) Bytes lang. Es ist zu beachten, daß diese Stellen nicht gelöscht werden.

KHLT

TO
<b>57</b>
1.Byte

Dieser Befehl stoppt das Programm, wenn die Taste KHLT be-  
tätigt ist.

Durch Einlegen dieser Befehlskarte, zwischen die übrigen  
Karten des Programmes, kann man das Programm an jeder be-  
liebigen Stelle anhalten.

LOAD

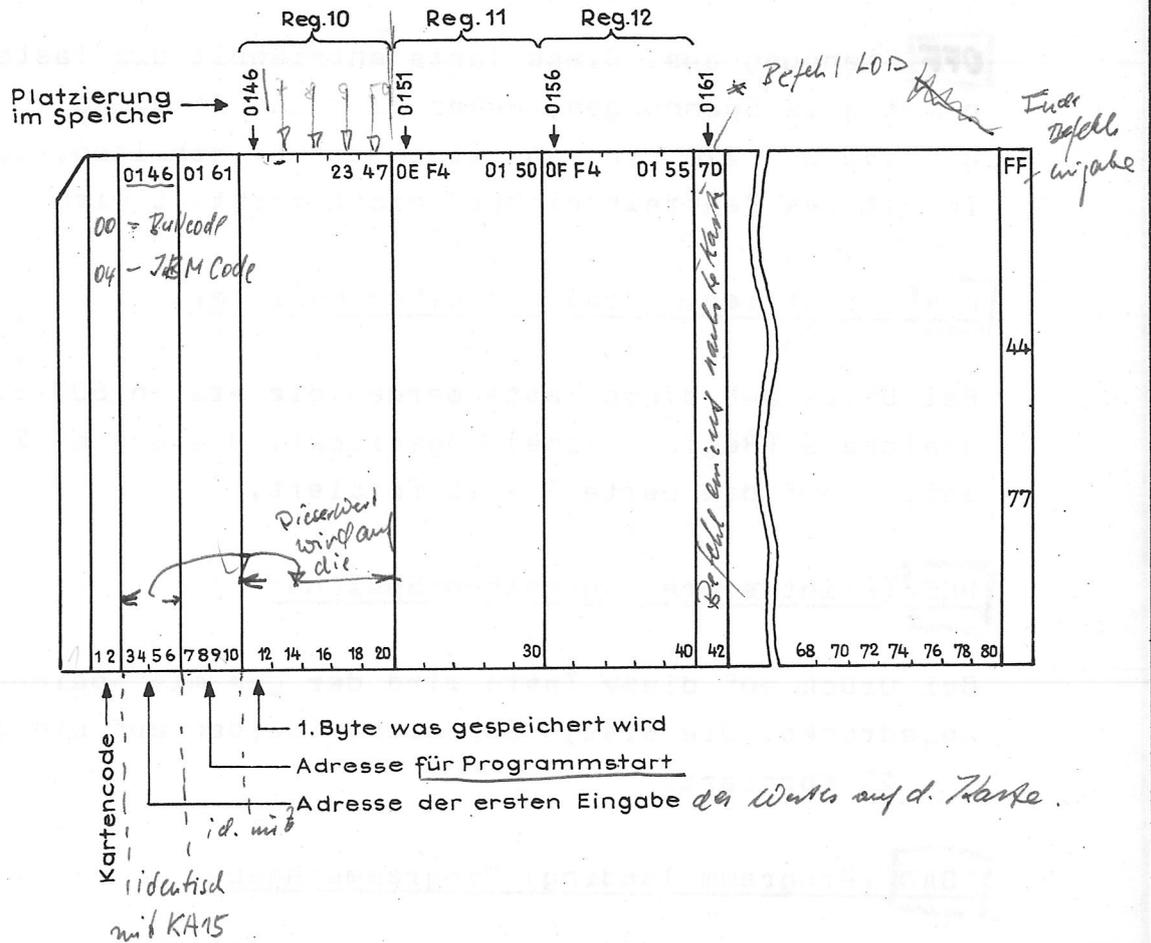
TO
<b>7D</b>
1.Byte

Dieser Befehl entspricht der Taste LOAD (CP) Ladung des  
Kernspeichers durch Lesen von Karten.

Die erste Karte trägt die Adresse, ab welcher Stelle ge-  
speichert werden soll. Die letzte hat FF in den beiden  
letzten Spalten. Jede Spalte entspricht einem Halbbyte.

Anwendungsbeispiel:

Um in die 3 Register 10, 11 und 12 bestimmte Werte einzugeben, erstellt man eine Ladekarte wie folgt und legt diese Karte vor das eigentliche Programm



Durch Druck auf die Taste LOAD wird diese Karte gelesen und in den Speicher eingegeben. Wenn FF gelesen wird, startet das Programm auf Adresse 0161. Dies ist der Befehl LOAD. Dieser Befehl startet jetzt die Eingabe des übrigen Programmes

Vorteil der Ladekarte: mehr auf eine Karte.  
 u. genaue Eingabe

Beispiele auf schriftl.

2.168 Bedienungspult:

Tasten:

**ON** Spannung ein. Diese Taste rastet ein und legt 220 V an das Netzteil. Nach Aufbau der Spannungen 15 M und 27 M leuchtet die Lampe dieser Taste auf.

**OFF** Spannung aus: Diese Taste entriegelt die Taste ST und nimmt alle Spannungen, außer 15 M und 27 M, weg. Die Spannungen 15M und 27M bleiben, etwas länger gehalten, damit der Inhalt des Zentralspeichers nicht zerstört wird.

**PRR** (Print registers) Schreiben Register

Bei Druck auf diese Taste werden die ersten 600 Stellen des Speichers (Registerzone) abgedruckt. Die ersten 25 Zeichen werden auf die Werte 1 - 25 forciert.

**PRS** (Print store) Schreiben Speicher

Bei Druck auf diese Taste wird der gesamte Speicherinhalt abgedruckt. Die ersten 25 Zeichen werden auf die Werte 1 - 25 forciert.

**LOAD** (Programm loading) Programmeingabe

Der Druck auf diese Taste startet ein Ladeprogramm, welches die Eingabe der Programme über den Kartenleser erlaubt. Dieses Programm ist im Festspeicher verdrahtet

**RUN** (Start run) Programmstart

Der Druck auf diese Taste startet das Programm auf der Adresse, die in Stelle 0089 - 0090 steht.