Definition

Der Gamma 55 ist eine kleine Datenverarbeitungsmaschine in kompakter Bauweise (1,5 x 2,5 m).
Er ist außerdem gut geeignet, die normalen Büroarbeiten auszuführen. Z.B. Buchungsarbeiten, bei denen manuelle Eingaben während der Bearbeitung erforderlich sind.
Arbeiten also, die in den Bereich der Fakturiermaschinen fallen.

Zusammenstellung

Der Gamma 55 setzt sich zusammen aus der Zentraleinheit, welche die Informationen verarbeitet und den Randeinheiten, für die Ein- und Ausgabe der Informationen.

1. Die Zentraleinheit

Sie enthält einen Zentralspeicher (Kernspeicher), einen Festspeicher und die Register für Rechnen sowie Ein- und Ausgabe.

1.1 Im Zentralspeicher werden die Programme und die Informationen gespeichert. Der Speicher hat eine Kapazität von 2500, 5000 oder 10000 Bytes. Seine Zykluszeit (Lesen und Rückschreiben) beträgt 7,9 us.
Ein Byte besteht aus 8 Bits und Prüfbit.

1.2 Der Festspeicher enthält die Mikroprogramme, die arithmetischen Tafeln und die Tafeln für Code-Übersetzungen. Sein Inhalt kann gelesen aber nicht zerstört (verändert) werden.
Die Mikroprogramme bestehen aus einer Folge von Elementarfunktionen, die zur Durchführung eines Programmbefehls notwendig sind.
Alle Informationen dieses Speichers sind durch die Verdrahtung festgelegt. Der Festspeicher hat eine Kapazität von 1024 Worten à 36 Bits. Jedes Wort ist adressierbar und wird in 7,9 us (Kernspeicherzyklus) gelesen.

1024 x 36
3072
6144

36864 Speicherstelle

1.3 Die verschiedenen Register für die Verbindungen zwischen Zentralspeicher, Festspeicher und Randeinheiten bestehen aus Flip-Flops und Verstärker. Die Technologie dieser Register ist in TC1 beschrieben. Die Übertragungen werden durch die Rhythmen eines Taktgebers gesteuert.

2. Die Randeinheiten (standard)

Die an den Gamma 55 anschließbaren Kartenleser und Stanzer haben eine Platte mit gedruckter Schaltung, welche die Umschaltung von einem Kartencode auf einen anderen erlaubt. Z.B. von BULL auf IBM.
Für den Gamma 55 in Standardausführung sind nur die Randeinheiten der Klasse III vorgesehen. Diese Randeinheiten besitzen keinen Pufferspeicher. Die gelesenen Informationen werden direkt in den Kernspeicher geschrieben.

2.1 Kartenleser L617

Dieser Leser liest Spalte für Spalte(fotoelektrisch).
Das Zufuhrmagazin und das Ablagefach fassen je 500 Karten und können nur bei stehender Maschine bedient werden.
Die Maschine stopt nur vor der Zufuhr einer neuen Karte.
Die Übersetzung der Karte kann gelesen werden, bevor die Karte unter den Lesekopf gezogen wird. Die Karte bleibt zu diesem Zweck im Magazin liegen.
Die max. Geschwindigkeit des Lesers ist 150 K/min..
Diese Geschwindigkeit wird nur erreicht, wenn die Lesebefehle aufeinander folgen.
Der Leser besitzt 4 Tasten:

- die oberste Taste EJ. bewirkt die Ablage einer Karte.
- die Taste darunter MAN schaltet die automatische Zufuhr ab. Diese Taste muß auch gedrückt werden bei leerem Magazin, damit die Transportrollen stehen bleiben.
- links neben der Taste MAN liegt eine unbeschriftete Taste.
  Bei Druck auf diese Taste werden alle Karten im Magazin von den Transportrollen abgehoben. In den entstandenen Zwischenraum kann man 1 - 10 Karten einlegen.
- Ein Druck auf die untenliegende Taste SRD bewirkt, daß der Kartentransport nicht mehr von der Zentraleinheit gesteuert wird, sondern durch die Taste SRD auf der numerischen Tastatur.

2.2 Kartenstanzer P112A

Es handelt sich hierbei um die beschleunigte Bahn des Lochers P 112. Er stanzt 40 Z/sec.. Die Lochung wird kontrolliert durch Vergleich der versetzten Stanzstempel mit dem eingegebenen Code. Bei Fehler stopt der Stanzer sofort und zeigt Fehler an. Pu

Das Zufuhrmagazin und das Ablagefach fassen je 500 Karten und können nur bei stehender Maschine gefüllt bzw. geleert werden.

Die Bearbeitung der Karte ist Spalte für Spalte, so daß die Bearbeitung der Karte jederzeit unterbrochen werden kann. Die Ablage kann durch Programm gesteuert werden, es gibt aber keine Tabulation.

Der Stanzer hat 2 Tasten:

- die Taste EJ bewirkt die Ablage einer Karte und Zufuhr einer neuen Karte unter die Stanzstation.
- die Taste RES überbrückt die angerufene Kontrolle, so daß die Karte trotz Fehleranzeige weiter bearbeitet wird.

In den Stanzer P 112A kann die Abdruckeinrichtung des Lochers P 112 eingebaut werden. Diese Abdruckeinrichtung übersetzt 63 Zeichen im Code H1 4012 oder T121. Wenn diese Abdruckeinrichtung eingeschaltet ist, vermindert sich die Stanzgeschwindigkeit auf 20 Z/sec. Die Einrichtung wird mit einem zusätzlichen Schalter Ein- bzw. abgeschaltet.

2.3 Druckwerk MB1 (Olivetti)

Dieses Druckwerk druckt Zeichen für Zeichen mit einer Geschwindigkeit von 50 Z/sec.

Es kann zwischen 2 Zeichen gestopt werden.

Nachfolgend einige allgemeine Daten:

- Anzahl der Druckzeichen : 63
- Anzahl der Zeichen auf einer Linie : 132
- Abstand von Zeile zu Zeile : 4 mm
- Abstand von Zeichen zu Zeichen : 2,5 mm
- Zeit für Zeilensprung : 40 ms
- Zeit für Wagenrücklauf : max.450 ms
- Geschwindigkeit des Papiersprungs : 25 Zeilen/sec.
- max. Breite des Papiers : 38 cm
- min. Breite des Papiers : 7,62 cm

Die Zeiten von Zeilensprung und Wagenrücklauf können sich überlagern.
Das Papier wird transportiert durch eine Stachelkette (Paragon).
Der Transport beim Sprung ist gleich, gesteuert durch ein Steuerband (Bandpilot).

Das Druckwerk besitzt 3 Tasten:

- die grüne Taste links (RD) READ bewirkt ein Abheben des Farbbandes und erlaubt ein Betrachten des soeben geschriebenen, solange die Taste gedrückt ist.
- die weiße Taste in der Mitte (SCR) Saut et Retour startet einen Papiersprung mit Rücklauf des Wagens.
- die rote Taste rechts (RES) erlaubt nach Halt durch "Ende Papier" noch eine Zeile zu schreiben.

2.4 Numerische Tastatur

Die numerische Tastatur besitzt 12 Tasten für die Werte 0 - 9, 11 (:) und 12 (;).
Außerdem eine Taste Spaltensprung, Löschen, Übertrag und die Taste SRD für den Kartenleser.
Die Werte, die über die numerische Tastatur eingegeben werden, gelangen in einen Pufferspeicher und werden

auf der Leuchtanzeige sichtbar.
Die Tastatur ist nur zu benutzen, wenn die Lampe N aufleuchtet. Außerhalb dieser Zeit sind die Tasten blockiert.

mit Befehl Anschluß min. 1 Mot.

2.5 Alphanumerische Tastatur

Dies ist die BULL-Standard-Tastatur mit 47 Tasten und einer Leertaste für Spaltensprung. 45 Tasten tragen Buchstaben, Zahlen oder andere Symbole.
Die Alphatastatur wird, wie die numerische Tastatur, durch Programm von der Zentraleinheit freigegeben.
Die Freigabe wird angezeigt durch die Lampe A. Außerhalb dieser Zeit sind die Tasten blockiert.
Der Anschluß wird unterbrochen, wenn die im Kernspeicher vorgesehene Zone voll ist. Die Lampe A geht dann aus und die Tasten werden wieder blockiert.

2.6 Pufferspeicher

Die numerische Tastatur wird, unter Zwischenschaltung eines Speichers von 6 Zeichen à 4 Bits, mit dem Zentralspeicher verbunden.
Der Inhalt dieses Speichers wird auf einer Leuchtanzeige sichtbar gemacht.
Der Pufferspeicher kann Werte, die in die numerische Tastatur eingegeben werden, zum Kernspeicher übertragen und Stellen aus dem Kernspeicher empfangen und sichtbar machen.

a) Pufferspeicher nach Kernspeicher

Wenn die Zentraleinheit aufgrund des Programmes eine Information über die numerische Tastatur benötigt, wird die Tastatur mit dem Pufferspeicher verbunden und die Lampe N leuchtet.
Die nun eingegebenen Werte sind auf der Leuchtanzeige zu sehen.
Im Falle einer falschen Eingabe, kann der Pufferspeicher durch Druck auf die Taste "Löschen" gelöscht werden.

Ist die Eingabe korrekt, so kann der Inhalt des Pufferspeichers, durch Druck auf die Taste "Übertrag" in den Kernspeicher übertragen werden.
Die Verbindung Pufferspeicher-Tastatur wird unterbrochen und die Tasten sind wieder blockiert.

b) Kernspeicher nach Pufferspeicher

Um ein Resultat oder Zwischenergebnis sichtbar zu machen, wird der Kernspeicher mit dem Pufferspeicher verbunden und die Lampe V (Visualisation) leuchtet auf. Das Programm wird gestopt.
Ein Druck auf die Taste "Löschen" unterbricht die Verbindung Kernspeicher - Pufferspeicher und zeigt der Zentraleinheit, daß das Ergebnis von der Bedienung gelesen wurde.
Gleichzeitig wird der Puffer gelöscht und das Programm wieder gestartet.

3. Zusätzliche Randeinheiten

Diese Randeinheiten gehören nicht zur Standardausführung des Gamma 55, sind aber zusätzlich anschließbar:

- Lochstreifenleser
- Lochstreifenstanzer
- Schreibmaschine .

4. Hilfsspeicher

An den Gamma 55 können folgende, schnelle Randeinheiten der Klasse II angeschlossen werden:

4.1 Bandstationen 1,2 KH

Das Anschlußmodell trägt den Namen Uniselektor.
Dieser Uniselektor überwacht und steuert den Informationsaustausch zwischen Bandstation und Zentraleinheit des Gamma 55.
Ein Uniselektor kann 1, 3, 5 oder 7 Stationen steuern.
An 2 Kanäle der Klasse II können je 1 Uniselektor angeschlossen werden.
Die Plazierung der Daten erfolgt in Blöcken zu 4mal 12 Bytes. Zur Bearbeitung dreht sich der Lese-Schreibkopf und das Bandspringt von Block zu Block.
Die Speicherkapazität einer Spule beträgt 1,2 Millionen Bytes, d.h. 100000 Sprossen à 12 Bytes oder 25000 Blökke. Die Umspuldauer beträgt ca. 2 min. .

4.2 Platten

Der Typ des Plattenspeichers, der an den Gamma 55 angeschlossen werden soll, liegt noch nicht fest.

4.3 Trommel

Ein schneller Kanal ist für den Anschluß einer Magnettrommel mit 128 Bahnen à 700 Bytes (1 Byte = 8 Bits) vorgesehen.
Die Zugriffszeit schwankt zwischen 0 - 20 ms.
Die Trommel dreht mit einer Geschwindigkeit von 3000 Upm.

5. Grundausstattung

Der GE 55 besteht in seiner Grundausstattung aus folgenden Elementen:

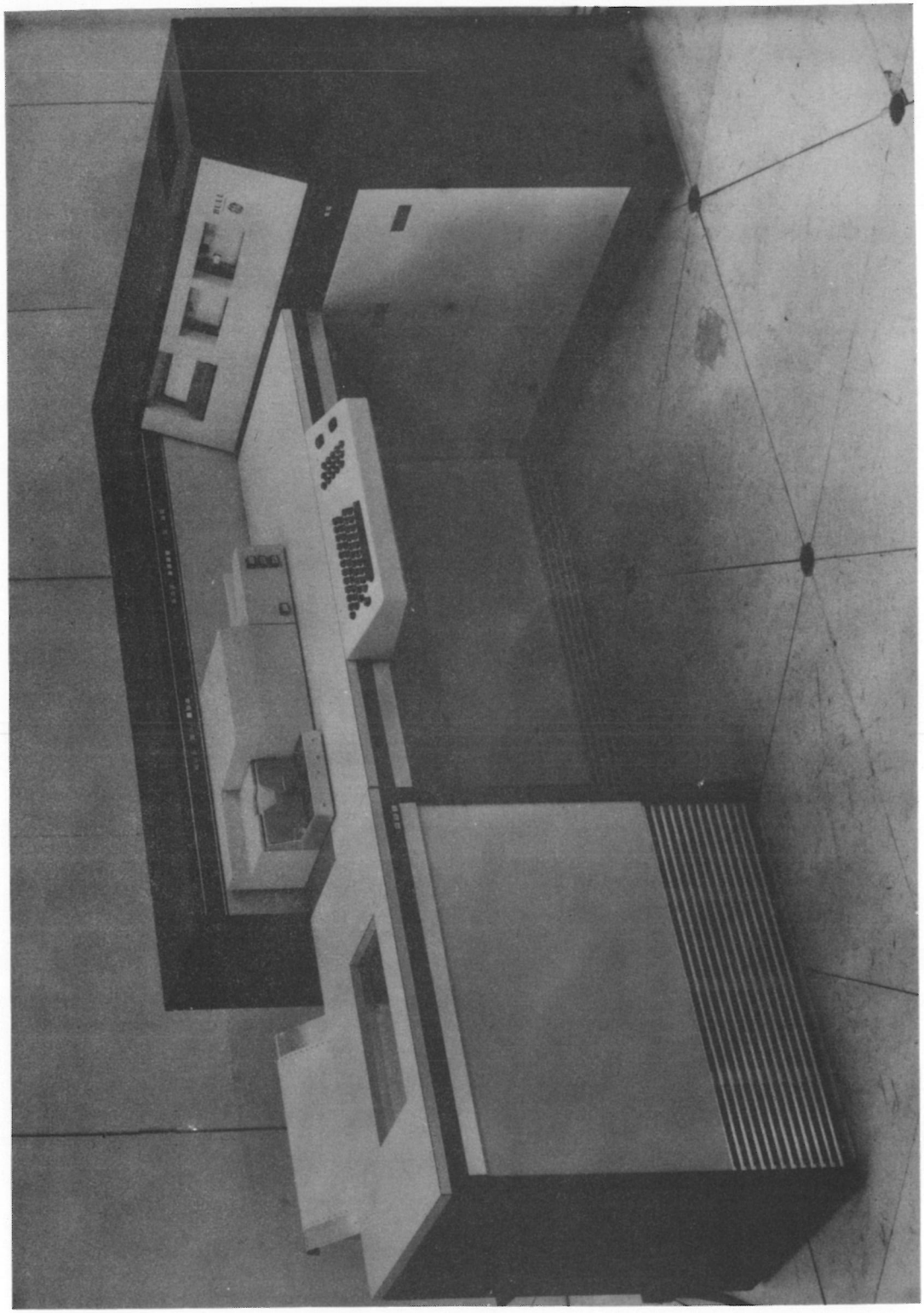
- Zentralspeicher (Kernspeicher) mit 2500 Bytes,
- Steuerpult,
- numerische Tastatur,
- Pufferspeicher,
- 1 Kanal der Klasse II,
- Kartenleser L617,
- Stanzer P112A,
- Drucker MB1,
- Festspeicher vom Typ MMS04.

Zusätzlich ist vorgesehen:

- alphanumerische Tastatur,
- Erweiterung des Zentralspeichers von 2500 auf 5000 Bytes,
- Trommel.

Wird an den GE 55 (Grundausstattung) eine zusätzliche Randeinheit angeschlossen, so ist die Erweiterung des Zentralspeichers, von 2500 auf 5000 Bytes, unbedingt erforderlich.

Implantation



1. Definitionen

1.1 Byte und Halbbyte

Eine Speicherstelle im Zentralspeicher besteht aus 9 Bits:

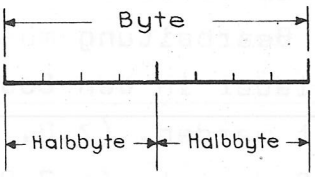
- 8 Bits zur Darstellung der Information (Byte),
- 1 Bit als Prüfbit, ergänzt auf Unpaarigkeit.

9 Bits

Ein Byte kann geteilt sein in zwei Halbbytes, zu je 4 Bits.

Anmerkung: 1. Die Bezeichnungen Byte und Halbbyte bezeichnen nur den Speicherplatz und nicht den Inhalt.

2. Das Byte im Gamma 55 entspricht dem Zeichen im Gamma 10 und Gamma 30, nur hat das Byte im Gamma 55 zwei Bits mehr.



1.2 Codierung der Information (Hexadezimal)

Um den Inhalt eines Bytes schneller zu erkennen, ohne den genauen Wert zu kennen, wird jedes Halbbyte durch ein Symbol dargestellt.
Da in jedem Halbbyte 16 verschiedene Kombinationen möglich sind, wird ein System mit 16 Symbolen benötigt.

Dieses System nennt man Hexadezimalsystem.

Hexadezimales Symbol	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Entsprechender Dezimalwert	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Beispiel:

	Byte								Byte							
Binär	0	L	L	L	L	L	L	L	L	0	0	L	L	0	L	0
Hexadezimal	7				F				9				A			

Anmerkung: Diess Form der Codierung ist auch sehr nützlich, bei einer Speicherentleerung. (Ausdrucken des Speicherinhaltes auf MB1). Dies wird notwendig bei der Aufstellung neuer Programme und der Störungssuche. In diesem Fall wird jedes Byte durch 2 hexadezimale Symbole dargestellt, die nebeneinander abgedruckt werden.

1.3 Interner Code (ISO)

Der interne Code des GE Gamma 55 ist der ISO-Code (internationaler Code).
Alle ankommenden Informationen für den Zentralspeicher sind in einem anderen Code, so daß diese Informationen vor der Weiterverarbeitung übersetzt werden müssen.
Nach der Bearbeitung müssen die auszugebenden Informationen wieder in den Code der entsprechenden Randeinheit übersetzt werden. (Z.B. BULL oder IBM-Kartencode).
Der ISO-Code ist ein 7-Bit-Code mit 128 möglichen Kombinationen.

1.4 Zeichen

Der Ausdruck "Zeichen" ist nicht mit der bei Gamma 10 und Gamma 30 üblichen Benennung einer Speicherstelle zu verwechseln (s.S. 13 Anmerkung).
Es handelt sich hier immer um eine Ziffer, einen Buchstaben, ein Sonderzeichen, einen Code oder eine Fahne.
Das Zeichen kann ein Byte oder ein Halbbyte belegen.
Ein Wort besteht aus mehreren Zeichen, die sich hintereinander im Speicher befinden.

2. Aufbau der Zentraleinheit

2.1 Der Zentralspeicher

Der Zentralspeicher ist ein klassischer Magnetkernspeicher zu 2500, 5000 oder 10000 Bytes. Er ist auf jeder Stelle dezimal adressierbar.

Die Zykluszeit (Lesen und Regenerieren) beträgt 8,5 us.

Der Zentralspeicher ist zur Aufnahme der Daten und der Programme bestimmt.
Im einzelnen enthält er auf Stelle:

01 - 30 eine logische Zone zur Abwicklung der Befehle.

31 - 84 eine zweite logische Zone, die für die Multiprogrammierung bestimmt ist. Darin findet sich:

DEFIO	(ZES)	Ein/Ausgabezone	31 - 54
CTPER	(ZRC)	Anschlußmerkmale	55 - 59
BGIN	(ZAR)	Originaladressen	60 - 69
STACK	(ZLP)	Programmablauf	70 - 74
STAT	(ZAT)	Wartezone	75 - 84

85 den Vorzeichenspeicher

86 das Vergleichsergebnis

87 - 88 die Adresse des Hauptprogramms, wohin nach Unterbrechung zurückgesprungen werden soll.

89 - 90 Die Adresse des nächsten Befehls, der ablaufen soll.

91 Am Ende einer Rechenoperation die höchste Wertstelle zur Feststellung der Kapazitätsüberschreitung, wenn diese nicht 0 oder 9 ist.

92 - 93 Die Adresse des Rücksprungs nach Ablauf des Fehlerprogrammes.

94 - 95 Die Adresse des Fehlerprogrammes.

96 -595 100 Register, numeriert von 00 - 99.

2.2 Die Register sind Felder von 5 Bytes, bzw. 10 Halbbyte (= 10 Ziffern). In diesen Registern finden obligatorisch die Rechenoperationen, Versetzungen und Vergleiche statt, sowie einige Übertragungsoperationen.

Die Register 00 - 09 haben außerdem noch die Funktion einer Basis.
Die reelle Adresse eines Registers kann man auch der Formel: Nummer des Registers x 5 + 100 ableiten.

Zwei aufeinanderfolgende Register können zu einem Doppelregister zusammengezogen werden. Die Nummer dieses Registers entspricht der Nummer des rechten Registers.

Von Stelle 596 ist der Kernspeicher nicht spezialisiert.

Natürlich können die Register, wenn sie nicht in ihrer speziellen Funktion benutzt werden, als Normalspeicher verwendet werden.

3. Formate 18.1.68

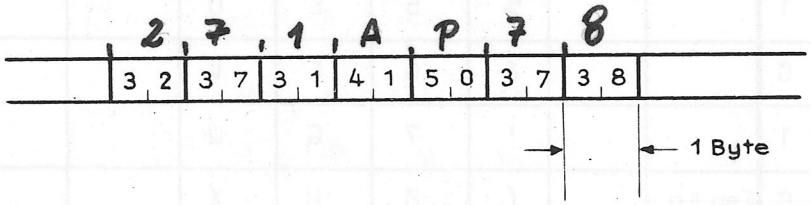
Eine Information kann im Kernspeicher in zwei verschiedenen Formaten gespeichert werden.

3.1 Format 1:

Ein in den Kernspeicher eingegebenes Zeichen, z.B. durch Lesen einer Karte, wird nach der Übersetzung in den ISO-Code, in einem Byte gespeichert. Man spricht in diesem Fall von "ungepackter" Darstellung.

Beispiel:

Die Zahl 271 AP 78 steht im Speicher im Format 1 wie folgt:



Im Format 1 sind die Zahlen immer in absoluten Werten ausgedrückt.

3.2 Format 2:

Das Programm und die Rechenfaktoren sind immer in "gepackter" Form dargestellt.
In diesem Format besetzt jedes Zeichen ein Halbbyte, d.h. je Byte zwei Zeichen. Zahlen werden in ihrem algebraischen Wert dargestellt. Eine negative Zahl wird durch ihr Komplement zu 10^{1o} oder 10^{2o} dargestellt.
Das äußerste linke Halbbyte bestimmt das Vorzeichen der Zahl (+ wenn 0 und - wenn 9).

3.3 Interner 7-Bit-Code (ISO)

Linkes Halbbyte →				0	1	2	3	4	5	6	7	Hexadezim. Darstellung
Binär-Code			K7	0	0	0	0	1	1	1	1	rechtes Halbbyte ↓
			K6	0	0	1	1	0	0	1	1	
			K5	0	1	0	1	0	1	0	1	
K4	K3	K2	K1									
0	0	0	0	Nul		Esp ƀ	0	@	P			0
0	0	0	1		Rub.R	!	1	A	Q			1
0	0	1	0			"	2	B	R			2
0	0	1	1			#	3	C	S			3
0	1	0	0		Rub.N	$	4	D	T			4
0	1	0	1			%	5	E	U			5
0	1	1	0			&	6	F	V			6
0	1	1	1			'	7	G	W			7
1	0	0	0	Esp An		(	8	H	X			8
1	0	0	1	Tab.		)	9	I	Y			9
1	0	1	0	Int.		*	:	J	Z			A
1	0	1	1	RC+Int.		+	;	K	[			B
1	1	0	0	Saut		,	<	L	\			C
1	1	0	1	RC ou EJ		-	=	M	]			D
1	1	1	0	Visu		.	>	N	↑			E
1	1	1	1	Val.Cl.		/	?	O	←			F

Die im Gamma 55 nicht verwendeten Symbole sind in dieser Tabelle nicht dargestellt.

3.4 BULL-Code (T 121) in internen Code (ISO)

Symbol	Kartencode	umges. Code	ISO
Esp.	. . .	70	20
!	0-12	F1	21
"	9- 7- 3	24	22
#	9- 7- 0	21	23
$	9- 7-12	A0	24
%	9- 7- 4	25	25
&	9- 7- 5	26	26
'	9- -12	E0	27
(	9- 8- 0	41	28
)	9- 8- 1	42	29
*	12	F0	2A
+	9- 8- 3	44	2B
,	9- 8-11	C8	2C
-	9- 8- 2	43	2D
.	11	FB	2E
/	9- 7-	20	2F
Ø	0	71	30
1	1	72	31
2	2	73	32
3	3	74	33
4	4	75	34
5	5	76	35
6	6	77	36
7	7-	30	37
8	8-	50	38
9	9-	60	39
:	9- 7- 1	22	3A
;	9- 7- 6	27	3B
<	12- 1	F2	3C
=	9- 8	40	3D
>	9- 8-12	C0	3E
?	9- 8- 5	46	3F
@	9- 8- 6	47	40

Symbol	Kartencode	umges. Code	ISO
A	7-11	B8	41
B	7- 0	31	42
C	7- 1	32	43
D	7- 2	33	44
E	7- 3	34	45
F	7- 4	35	46
G	7- 5	36	47
H	7- 6	37	48
I	8-12	D0	49
J	8-11	D8	4A
K	8- 0	51	4B
L	8- 1	52	4C
M	8- 2	53	4D
N	8- 3	54	4E
O	7-12	B0	4F
P	8- 4	55	50
Q	8- 5	56	51
R	8- 6	57	52
S	9- -11	E8	53
T	9- 0	61	54
U	9- 1	62	55
V	9- 2	63	56
W	9- 3	64	57
X	9- 4	65	58
Y	9- 5	66	59
Z	9- 6	67	5A
[	11- 1	FA	5B
\	9- 7- 11	A8	5C
]	9- 8- 4	45	5D
↑	0- 11	F9	5E
←	9- 7- 2	23	5F

Code von 0-6

11+12	$\overline{7}$	$\overline{8}$	$\overline{9}$	11			

$K1 \cdot \overline{K2} \cdot \overline{K3} = 0$

$K1 \cdot K2 \cdot K3 = 6$

3.5 IBM-Code (H14.012) in internen Code (ISO)

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

8 Code von 1-9

$11 \cdot \overline{12} + \overline{11} \cdot 12$ | $\overline{11} \cdot \overline{12} + 0(\overline{11}+\overline{12})$ | $0 \cdot \overline{12} + 11(\overline{0}+\overline{12})$

$K1 \cdot \overline{K2} \cdot \overline{K3} = 1$

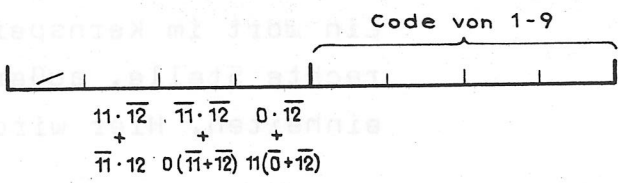
$K1 \cdot K2 \cdot K3 = 7$

K4

8 nach Zwischencode

3.6 ISO-Code in internen Code (ISO)

Symbol	Kartencode	ISO	Symbol	Kartencode	ISO
Esp.		20	@	12-	40
!	0 - 1	21	A	12- 1	41
"	0 - 2	22	B	12- 2	42
#	0 - 3	23	C	12- 3	43
$	0 - 4	24	D	12- 4	44
%	0 - 5	25	E	12- 5	45
&	0 - 6	26	F	12- 6	46
'	0 - 7	27	G	12- 7	47
(	0 - 8	28	H	12- 8	48
)	0 - 9	29	I	12- 9	49
*	0 - 8 - 2	2A	J	12- 8-2	4A
,	0 - 8 - 4	2C	L	12- 8-4	4C
-	0 - 8 - 5	2D	M	12- 8-5	4D
.	0 - 8 - 6	2E	N	12- 8-6	4E
/	0 - 8 - 7	2F	O	12- 8-7	4F
Ø	0	30	P	11-	50
1	1	31	Q	11- 1	51
2	2	32	R	11- 2	52
3	3	33	S	11- 3	53
4	4	34	T	11- 4	54
5	5	35	U	11- 5	55
6	6	36	V	11- 6	56
7	7	37	W	11- 7	57
8	8	38	X	11- 8	58
9	9	39	Y	11- 9	59
:	8 - 2	3A	Z	11- 8-2	5A
;	8 - 3	3B	[	11- 8-3	5B
<	8 - 4	3C	CS2	11- 8-4	5C
=	8 - 5	3D	]	11- 8-5	5D
>	8 - 6	3E	^	11- 8-6	5E
?	8 - 7	3F		11- 8-7	5F



4. Allgemeine Befehle

8.1.68

4.1 Definition

Makro-Basis

Dies sind die Elementarbefehle, die der Programmierer in die Maschine eingeben kann.

4.2 Zusammensetzung der Befehle

Operationstyp (TO)

(schnell zu verstehen)

Er ist auf 2 verschiedene Arten dargestellt:

- TO symbolisch oder mnemonisch. Er ist dargestellt in einem Code, der für die Maschine nicht lesbar und für den Befehl zu lang ist.
- TO intern. Er besetzt ein Byte im Zentralspeicher und besteht aus 2 Zeichen, hexadezimal codiert.

Konstante

Wenn in dem Befehl eine Konstante vorkommt, so besetzt diese das zweite Byte (außer C).
In der allgemeinen Darstellungsart eines Befehls steht das Zeichen:

K -als Symbol für ein Zeichen außer F1, F2, F3 und FF.
N -für 2 Ziffern im Format 2 (00 - 99), welche die Zeichenanzahl bei einer Übertragung, einem Vergleich und einer Versetzung angeben.
C -für den Anschlußcode einer Randeinheit.

Achtung: Dieses Zeichen besetzt das dritte Byte oder wenn mehrere Randeinheiten angeschlossen werden, eventuell das fünfte oder siebte Byte.

4.3 Adressierung

8.1.68

Ein Wort im Kernspeicher ist adressiert durch seine rechte Stelle, außer bei einem Befehl für die Randeinheiten. Hier wird die rechte Seite des Wortes

durch eine Trennmarke (F4) begrenzt. Die Adressierung ist je nach Befehl direkt oder durch eine Basis.

Direkte Adresse: Sie kann 2 Formen haben:

.XX.XX. die reelle Adresse ist dargestellt in 2 Bytes durch 4 Ziffern in Format 2.

.R. die Registernummer (00 - 99) steht in einem Byte, in Format 2.

Basisadresse mit Angabe des Basisregisters

Sie ist dargestellt durch 4 Ziffern in Format 2. Plaziert in 2 Bytes.

.BX.XX.

.B . . im 4. Halbbyte ist die Registernummer (0 - 9) angegeben, in der sich die Basis befindet. Diese Basis besteht aus 4 Ziffern, in den 4 rechten Halbbytes des Registers.

. X.XX. Diese 3 Ziffern geben die Versetzung der reellen zur Basisadresse an.
Man findet die Realadresse durch Addition dieser Ziffer (000 - 999) auf die Realadresse.

Beispiel: Das Register 8 enthält 1 5 2 5
In .BX.XX. steht 8015, dann ist die Realadresse 1525 + 015 = 1540.

Im Operationstyp enthaltene Adressen.

Dieser Adresstyp wird nur bei Übertragungen, in Verbindung mit den Ein- Ausgabezonen, verwendet. Entsprechend dem Operationstyp nimmt man den Inhalt eines Spezialregisters (4 rechte Halbbytes) als Basisadresse.

Diese Spezialregister sind:

Register 06: bestimmt die Eingangszone A, die für den Kartenleser reserviert ist.
Die Versetzung wird im Befehl unter D_6 angegeben.

Register 07: bestimmt die Eingangszone B.
Die Versetzung wird im Befehl unter D_7 angegeben.

Register 08: bestimmt die Ausgangszone C, die für den Drucker (MB1) reserviert ist.
Die Versetzung wird im Befehl unter D_8 angegeben.

Register 09: bestimmt die Ausgangszone D, die für den Stanzer reserviert ist.
Die Versetzung wird im Befehl unter D_9 angegeben.

Wenn die Versetzung $\leqslant 99$ ist, wird sie in einem Byte in Format 2, gespeichert. Z.B. | 8 | 7 |

Ist die Versetzung >99, (max. 999) wird sie durch 3 Ziffern, in Format 2, in 2 Bytes dargestellt.
Im ersten Halbbyte dieser beiden Bytes steht ein A.
Z.B. | A | 1 | 7 | 5 |

Dieses A im 1. Halbbyte zeigt der Maschine, daß die Versetzung in 2 Bytes gespeichert ist.

4.4 Länge der Befehle

Die Länge der Befehle ist variabel, besteht aber immer aus ganzen Bytes. (keine Halbbytes)
Der kürzeste Befehl besetzt 1 Byte und der längste Befehl besetzt 9 Bytes.

4.5 Anzahl der Befehle

Es gibt 67 Befehle, die entsprechend ihrem Zweck, in folgende Gruppen unterteilt sind:

- Übertragungsbefehle
- Rechenbefehle
- Befehle für Randeinheiten
- Sprungbefehle
- Logische Operationen
- Versetzungsbefehle
- Spezialbefehle für Programmerstellung
- Befehle für Multiprogrammierung.

5. Darstellung der Befehle

5.1 Übertragung

5.1.01 MRR

Move the content of a single Register into another one.

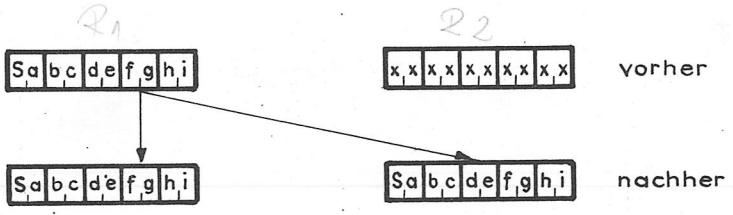
TO	Reg. Nr.	Reg. Nr.
E1	R1	R2
	00 - 99	00 - 99
1. Byte	2. Byte	3. Byte

Dauer: 1.6 ms

Übertragung des Inhalts eines Einfachregisters R1 in ein Einfachregister R2.

Die Übertragung erfolgt ohne Wechsel des Formats. Der Inhalt von R1 bleibt erhalten.

Beispiel



5.1.02 MDRR (double)

Move the contents of a Double Register into another one

E2	R1	R2
	01 - 99	01 - 99
1. Byte	2. Byte	3. Byte

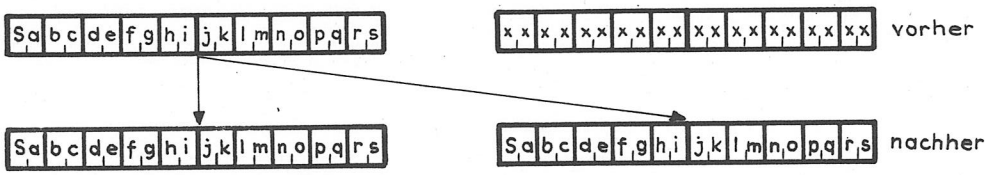
Dauer: 1.9 ms

Übertragung des Inhaltes eines Doppelregisters R1 in ein Doppelregister R2.

Wertmäßig gleich, R2 wird gelöscht

Die Übertragung erfolgt ohne Wechsel des Formats.

Beispiel



signe

5.1.03 MSRDR

TO	Reg. Nr.	Reg. Nr.
E3	R1	R2
	00 - 99	00 - 99
1. Byte	2. Byte	3. Byte

Dauer 1.8 ms

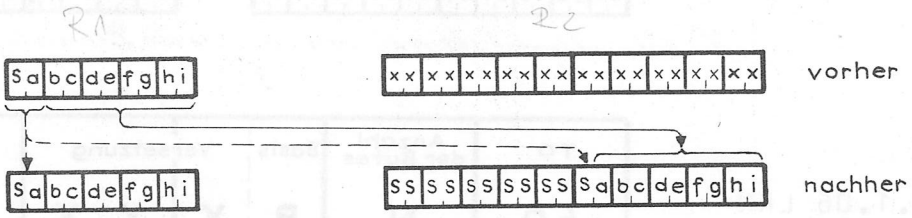
Move the content of a single Register into a double Register.

Übertragung des Inhaltes von Einfachregister R1 (in Format 2) in das Doppelregister R2. Das äußerste linke Halbbyte von R1 wird 10 x in R2 - 1 dupliziert.

Vorzeichen — Spicht Rechen

Anmerkung: Man kann R1 = R2 oder R1 = R2 - 1 haben.

Beispiel



5.1.04 EXR

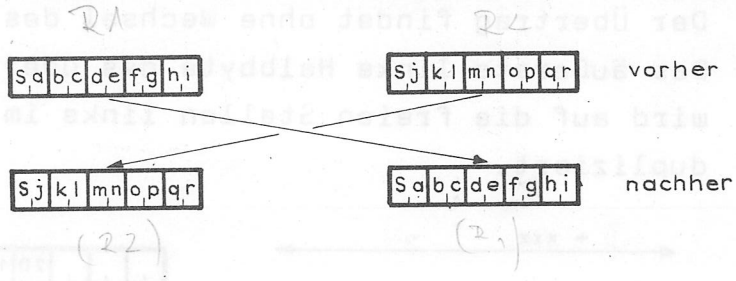
Exchange single Registe

TO	Reg. Nr.	Reg. Nr.
E5	R1	R2
	00 - 99	00 - 99
1. Byte	2. Byte	3. Byte

Dauer: 1.7 ms

Austausch des Inhaltes zweier Einfachregister.
Das Format wird nicht gewechselt.

Beispiel



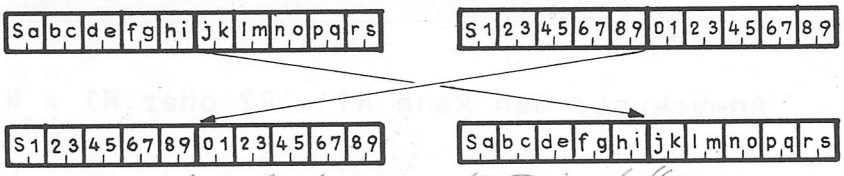
5.1.05 EXDR

Exchange Double Registers

TO	Reg. Nr.	Reg. Nr.
E6	R1	R2
	01 - 99	01 - 99
1. Byte	2. Byte	3. Byte

Dauer 2,2 ms

Austausch des Inhaltes zweier Doppelregister.
R1 - R2 - 1 ist möglich. Es findet dann ein Ringtausch dreier Einfachregister statt.



N = Nebenschlüssel dezimal, oder zweipgepackte Dezimalziffern.

5.1.06 LRN

Load Register N

TO	Anzahl der Bytes	Basis	Versetzung		Reg. Nr.
6.0	N	B	X	X X	R
	00 - 10	0-9	000	- 999	00 - 99
1. Byte	2. Byte	3. Byte	4. Byte		5. Byte

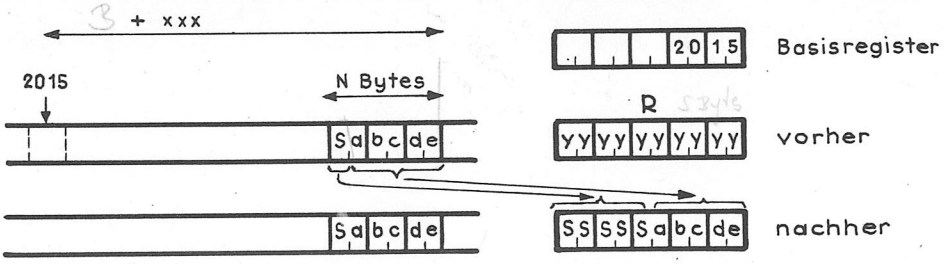
Dauer 1,4 + 0,08 Nms N = pro Byte

Übertrag eines Wortes von N (Anzahl) Bytes von der Basisadresse Bx.xx in das Register R.

R = Einfachregister, wenn N ≤ 5 Bytes
R = Doppelregister, wenn N > 5 Bytes
N = 0 wird wie 10 behandelt.

Wenn N > 10, werden nur die Einerstellen des angegebenen Wertes beachtet.

Der Übertrag findet ohne Wechsel des Formats statt. Das äußerste linke Halbbyte des übertragenden Wortes wird auf die freien Stellen links im Register dupliziert.



5.1.07 MVC

Moving

Move Caracter

TO	Anzahl der Bytes	1. Byte abgebender Basis	Versetzung		1. Byte empfangender Basis	Versetzung	
8.0	N	B_1	X	X X	B_2	X'	X' X'
	00 - 99	0 - 9	000	- 999	0 - 9	000	- 999
1. Byte	2. Byte	3. Byte		4. Byte	5. Byte		6. Byte

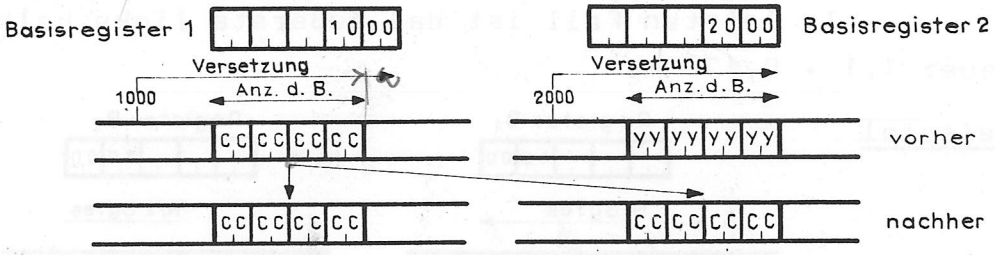
Zonen-Übertrag

Übertrag eines Wortes von N Bytes von Basisadresse B1x.xx nach B2x.xx

Der Übertrag erfolgt ohne Wechsel des Formats.

Bezeichnet A die abgebende und E die empfangende Adresse, so muß E ≤ A oder E > A-N sein.

Dauer: 1,9 + 0,05 Nms



5.1.08 MVIC

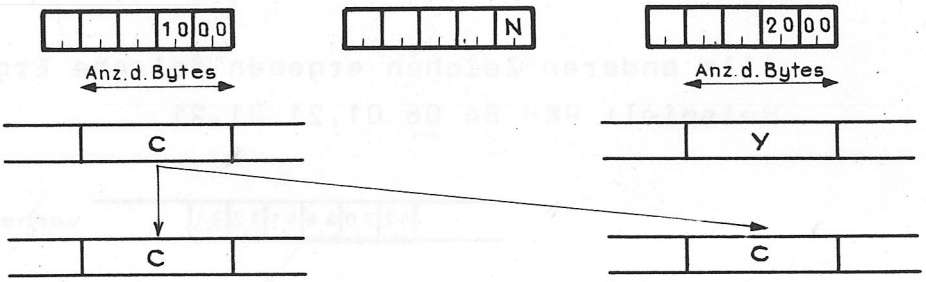
Move Caracter

Count is indirect

8A	R	B_1 XX	XX	B_2 X'	X' X'
	00 - 99				
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

Der Befehl entspricht MVC. Die Länge des zu übertragenden Wortes befindet sich in Register R und ist ≤ 99.

Dauer 2,2 + 0,05 ms



5.1.09 PKH

Pack Hexodecimal

	Nr. des abgebenden Byte	1. Byte abgebender		1. Byte empfangender	
		Basis	Versetzung	Basis	Versetzung
84	N	B_1	X | XX	B_2	X' | X'X'
	00 - 99	0 - 9	000 - 999	0 - 9	000 - 999
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

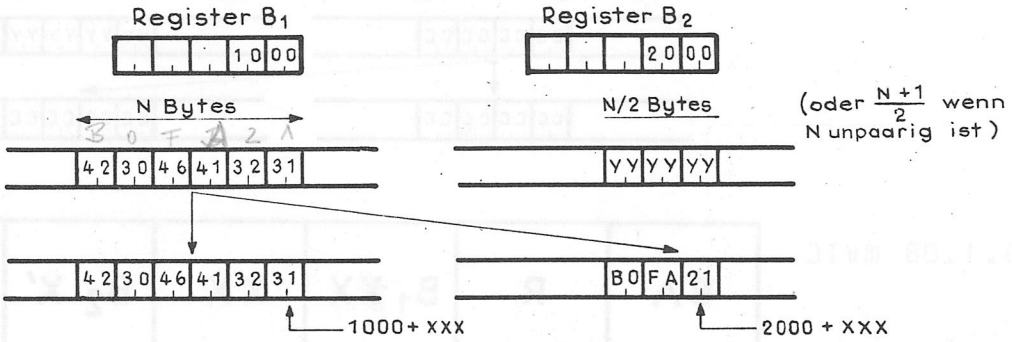
Übertrag mit Verdichtung eines Wortes mit der Länge von N Bytes in Format 1, welches durch die Basisadresse B1x xx bestimmt wird, nach B2x' x'x'. Das Resultat hat im Format 2 die Länge 1, die sich wie folgt errechnet:

$1 = \frac{N}{2}$ wenn N gerade ist. $1 = \frac{N+1}{2}$ wenn N ungerade ist.

Im letzten Fall ist das äußerste linke Halbbyte Null.

Dauer 1.1 + 0,17 Nms

Beispiel



Wenn B1x xx = B2x' x'x' ist, wird der linke Teil des Feldes durch den Übertrag nicht gelöscht.

Die Zeichen, die verdichtet, bzw. erweitert werden können, sind normalerweise:

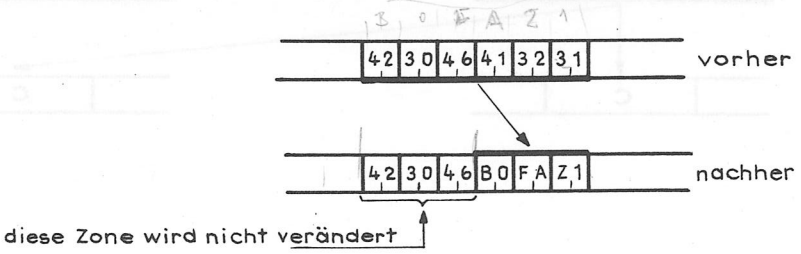
Die Zahlen 30 - 39 in hexadezimal (1 - 9)

Die Buchstaben 41 - 46 in hexadezimal (A - F)

Der Spaltensprung 20 in hexadezimal

Alle anderen Zeichen ergeben falsche Ergebnisse.

Beispiel: PKH 84 06 01.21 01.21



?

5.1.10 UPH

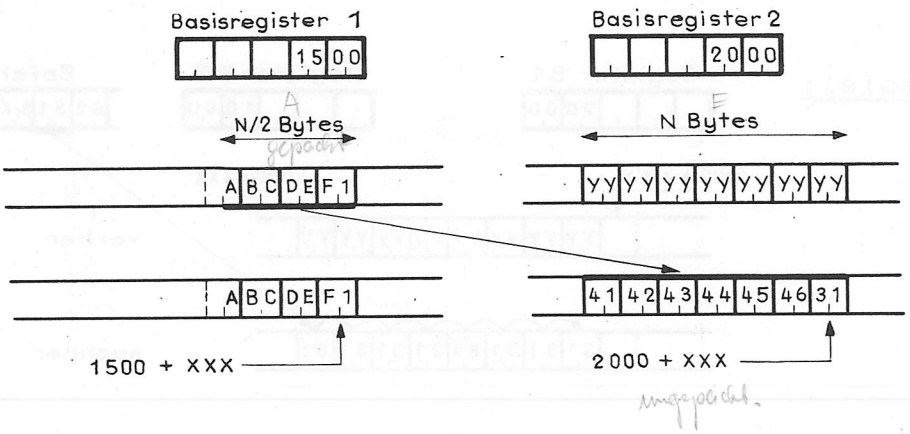
Un Pack Hexadecimal.

TO	Nr. des abgebenden Byte	1. Byte abgebender Basis	Versetzung		1. Byte empfangender Basis	Versetzung	
85	N (anzahl der abgeb. Halb-byte)	B_1	X	XX	B_2	X'	X'X'
	00 - 99	0 - 9	000	- 999	0 - 9	000	- 999
1. Byte	2. Byte	3. Byte		4. Byte	5. Byte		6. Byte

Übertrag mit Erweitern eines Wortes, das sich im Format 2 in der Basisadresse B1x xx befindet, auf ein Feld, das mit der Basisadresse B2x' x'x' beginnt. Das Resultat ist in Format 1 nach dem Erweitern von N:2 Zeichen des abgebenden Feldes.

Dauer: 06 + 0,2 N ms

Beispiel



A sei die abgebende und E die empfangende Adresse.

Dann ist:

$E > A + \frac{N}{2}$ $\left(A + \frac{N+1}{2}\right.$ wenn N ungerade ist)

oder

$E < A - \frac{N}{2}$ $\left(A - \frac{N+1}{2}\right.$ wenn N gerade ist)

K = Kennzeichen hexadezimal (Binärwert aus zwei gepackte hexadezimale Ziffern.

5.1.11 INC

Insert Character

Löschen

Werte einsetzen

Symbole

Werte

u. Beiden ersetzen

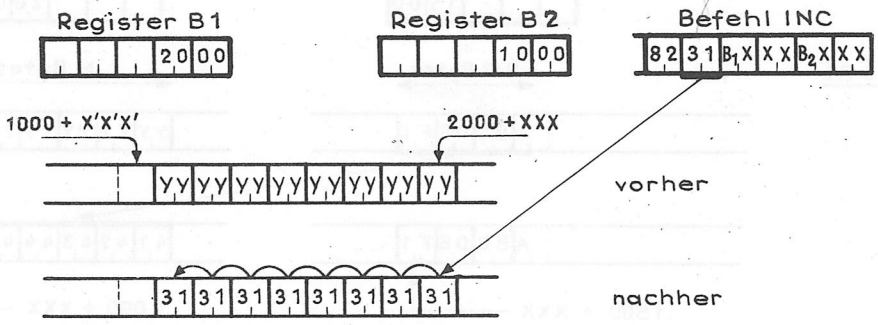
TO		1. Byte der Zone			1. Byte nach der Zone		
		Basis	Versetzung		Basis	Versetzung	
82	K	B_1	X	XX	B_2	X'	X'X'
	00 - FE*	0-9	000	- 999	0-9	000	- 999
1.Byte	2.Byte	3.Byte		4.Byte	5.Byte		6.Byte

Eingabe des 2. Bytes des Befehls (K) von der Basis B1x xx (einschließlich) bis zur Basis B2 x'x'x' (ausschließlich).

Dauer: 0,8 + 0,2 N ms

= Wert

Beispiel:



Anmerkung:

- B_1 xxx muß größer als B_2 x'x'x' sein, da sonst die logische Zone verändert wird.
- Wenn K = 00 entspricht dieser Befehl einem Löschbefehl.
- K darf nicht F1, F2, F3 oder FF sein, da dies die Programmeingabe stört.

5.2 Rechenbefehle

5.2.1 ADD

addition Decimal single length

TO	Reg.-Nr.	Reg.-Nr.
A1	R_1	R_2
	00 - 99	00 - 99
1. Byte	2. Byte	3. Byte

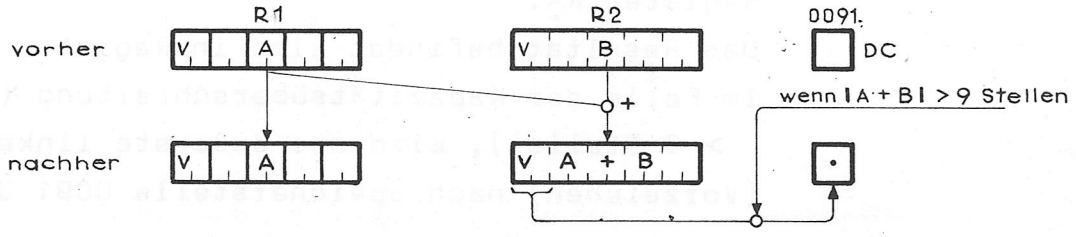
Dauer: 2,2 ms

Algebraische Addition der beiden Register R_1 und R_2.
Das Ergebnis steht im Register R_2.

Im Falle der Kapazitätsüberschreitung (Resultat > 9 Ziffern), wird das äußerste linke Halbbyte (Vorzeichen) nach Speicherstelle 0091 übertragen.

Die Auswertung der Stelle 0091 kann der Bedienung, durch Leuchtanzeige oder Abdruck, den Fehler anzeigen.

Beispiel:



Sonderfall: Wenn $R_1 = R_2$ wird der Inhalt der Register verdoppelt.

5.2.2 ADDD

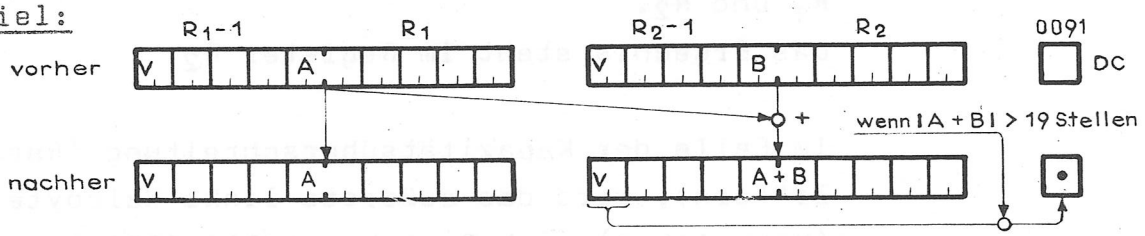
Addition Decimal Double length

TO	Reg.-Nr.	Reg.-Nr.
A2	R_1	R_2
	01 - 99	01 - 99
1.Byte	2.Byte	3.Byte

Dauer: 2,9 ms

Dieser Befehl ist gleich der ADD, jedoch werden hierbei 2 Doppelregister verwendet.

Beispiel:



5.2.3 SBD

Subtract Decimal single length

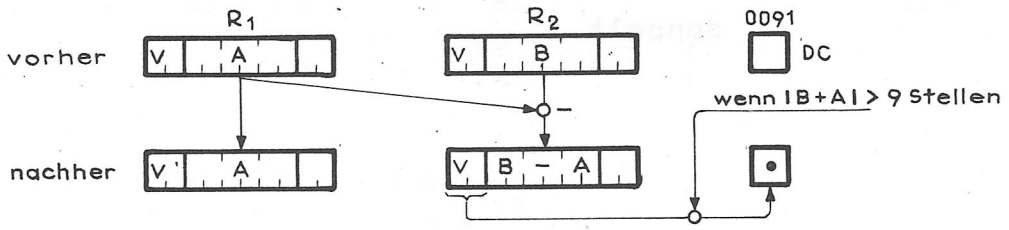
TO	Reg.-Nr.	Reg.-Nr.
B1	R_1	R_2
	00 - 99	00 - 99
1.Byte	2.Byte	3.Byte

Dauer: 2,2 ms

Algebraische Subtraktion des Register R_1 von Register R_2.
Das Resultat befindet sich in Register R_2.
Im Falle der Kapazitätsüberschreitung (Resultat > 9 Stellen), wird das äußerste linke Halbbyte (Vorzeichen) nach Speicherstelle 0091 übertragen.

Sonderfall: Wenn $R_1 = R_2$ wird R_2 gelöscht.

Beispiel:



5.2.4 SBDD

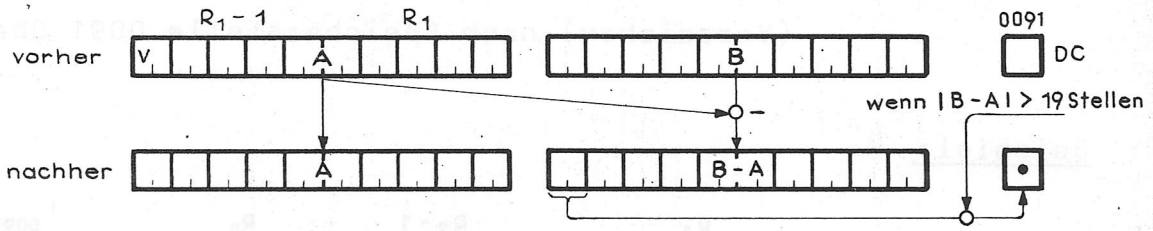
TO	Reg.-Nr.	Reg.-Nr.
B2	R_1	R_2
	01 - 99	01 - 99
1.Byte	2.Byte	3.Byte

Dauer: 2,9 ms

Subtract Decimal
Double length

Dieser Befehl ist gleich der SBD, jedoch werden hierbei 2 Doppelregister verwendet.

Beispiel:



5.2.5 RUUND

TO	Reg.-Nr.	Reg.-Nr.
7.1	K	R
	00 - 99	01 - 99
1.Byte	2.Byte	3.Byte

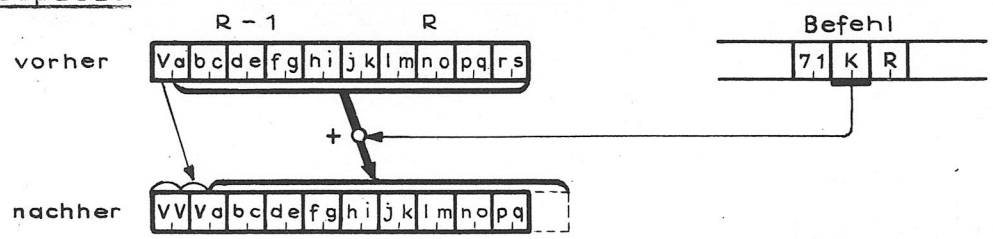
Dauer: 5,4 ms

Algebraische Addition des Zeichens K (2. Byte des Befehls) mit dem Doppelregister R.

2.R +50

Nach der Addition, Versetzung um 1 Byte nach rechts, mit Duplizierung des Vorzeichens und Wegfall der 1. zwei Ziffern rechts.

Beispiel:



5.2.6 MPD

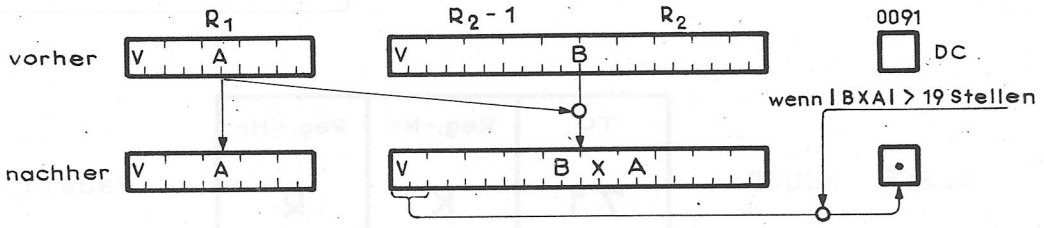
Multiply Decimal

TO	Register-Nr. Multiplikator	Register-Nr. Multiplikator
C1	R_1 00-99	R_2 01-99
1. Byte	2. Byte	3. Byte

Dauer: 50 (+.+) ms
52 (+.-) ms

Multiplikation des Inhaltes von Doppelregister R_2 (Multiplikant) mit dem Inhalt von Einfachregister R_1 (Multiplikator). Resultat in Doppelregister R_2.
Im Falle der Kapazitätsüberschreitung (Resultat > 10 Stellen) wird das äußerste linke Halbbyte (Vorzeichen) nach Speicherstelle 0091 übertragen.

Beispiel:



Sonderfall:
- Bei $R_1 = R_2$ wird der Inhalt des Registers R_2 zum Quadrat erhoben, unter der Bedingung, daß er kleiner als 10 Stellen ist.
- $R_1 = R_2 - 1$ ist möglich.

Leerseite

5.2.7 DVD

Divide Decimal!

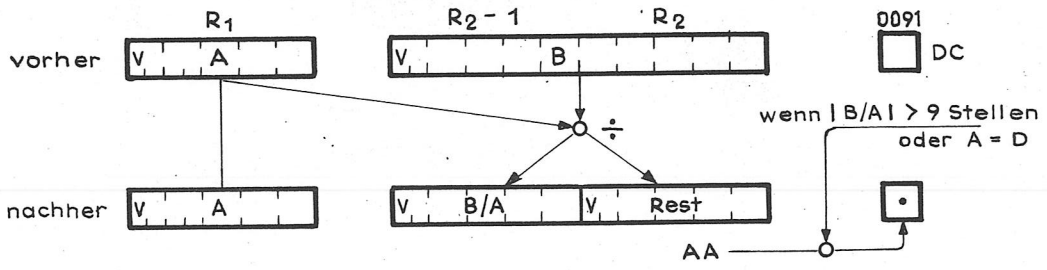
TO	Register-Nr. Multiplikator	Register-Nr. Multiplikator
D.1	R_1	R_2
	00 - 99	01 - 99
1.Byte	2.Byte	3.Byte

Dauer: 40 - 200 ms

Division des Inhaltes von Doppelregister R_2 (Dividend) durch den Inhalt des Einfachregisters R_1 (Divisor).
Am Ende der Operation enthält das Doppelregister:
- im linken Register den Quotient und
- im rechten Register den Rest.

Hat der Quotient mehr als 9 Stellen oder ist der Divisor gleich Null, so enthält die Stelle 0091, AA.

Beispiel:



Sonderfall:
- $R_1 = R_2$, Eingabe von 1 für Quotient, wenn $R_2 < 10$ Stellen ist.
- Ein Divisor Null ändert die Inhalte der Register 1 und 2 nicht.

5.3 Versetzungen

5.3.1 SLD

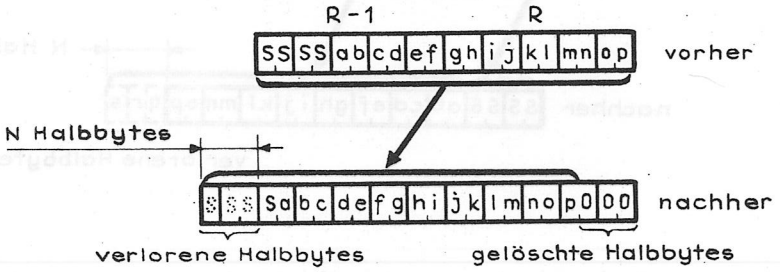
Shift left decimal

TO	Anzahl der Halbbytes	Reg. Nr.
7.0	N	R
	00 - 19	00 - 99
1. Byte	2. Byte	3. Byte

Dauer: 1 + N ms

Versetzung des Doppelregisters R um N Halbbytes.
Nach der Versetzung sind:
- die rechten N Halbbytes gelöscht
- die linken N Halbbytes verloren.

Beispiel:



Anmerkung: Wenn es sich bei der Versetzung um eine Zahl handelt, die nicht zerstört werden darf, so muß ihre Stellenzahl < (20 - N) sein.

Sonderfall: Die Zahl der durchgeführten Versetzungen ist gleich dem Rest der Division N : 20, wenn N ≥ 20.

5.3.2 SRD nrf

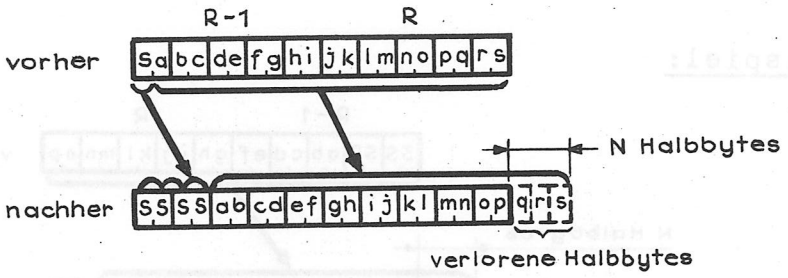
Ship right Decimal

TO	Anzahl der Halbbytes	Reg. Nr.
7.2	N 00-19	R 01-99
1. Byte	2. Byte	3. Byte

Dauer: 1 + N ms

Versetzung des Doppelregisters R um N Halbbytes.
Das äußerste linke Halbbyte wird dupliziert.
Die rechten Halbbytes sind verloren.

Beispiel:



Sonderfall: Wenn N ≥ 20 ist, ist die Zahl der durchgeführten Versetzungen gleich dem Rest der Division N : 20.

5.4 Vergleichsbefehle

5.4.1 CMD

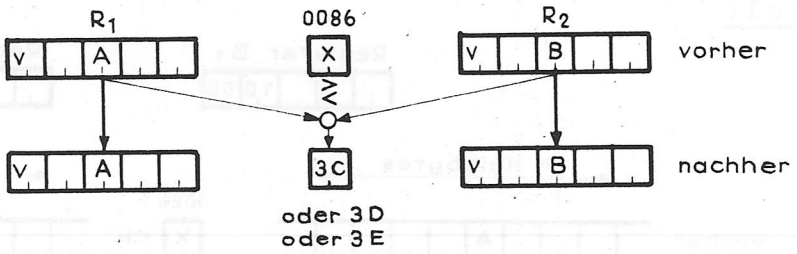
TO	Reg. Nr.	Reg. Nr.
9.1	R_1	R_2
	00 - 99	00 - 99
1. Byte	2. Byte	3. Byte

Dauer: 1,75 ms =
2,10 ms ≠

Algebraischer Vergleich des Inhaltes zweier Register (R_1 und R_2). Das Vergleichsergebnis steht in Stelle 0086 im internen Code.

3C wenn $(R_1) < (R_2)$
3D wenn $(R_1) = (R_2)$
3E wenn $(R_1) > (R_2)$

Beispiel:



5.4.2 CMDD

TO	Reg. Nr.	Reg. Nr.
9.2	R_1	R_2
	01 - 99	01 - 99
1. Byte	2. Byte	3. Byte

Dauer: 2,08 ms =
2,8 ms ≠

Vergleich entsprechend der CMD, jedoch mit zwei Doppelregistern.

Compare Caracter

5.4.3 CMC

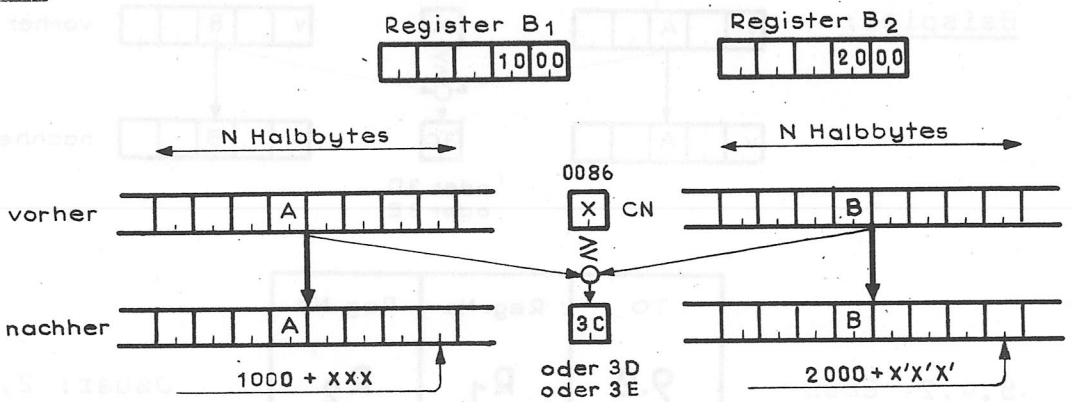
Dauer: 2+0,07ms = 2,3+0,1Nms ≠

TO	Anzahl der Halbbytes	Basis	Versetzung		Basis	Versetzung	
8.8	N	B_1	X	X X	B_2	X'	X' X'
	00 - 99	0-9	000	- 999	0-9	000	- 999
1. Byte	2. Byte	3. Byte		4. Byte	5. Byte		6. Byte

Vergleich von 2 Worten gleicher Länge in Basisadresse B_1 xxx und B_2x'x'x' .
Das Resultat steht in Stelle 0086 im internen Code.

3C = <
3D = =
3E = >

Beispiel:



Sonderfall: Wenn N gleich Null ist, enthält die Stelle 0086, 3D (=)

5.5 Logische Befehle

5.5.1 NI

K in JICode

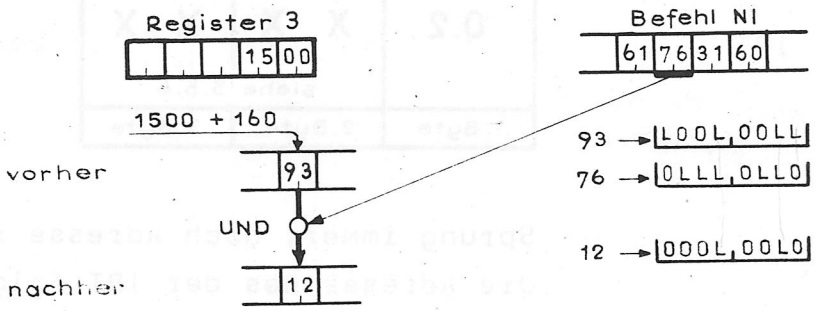
And immediat

TO		Basis	Versetzung	
6.1	K	B	X	X X
	00-FE*	0-9	000	- 999
1. Byte	2. Byte	3. Byte		4. Byte

Dauer: 1,6 ms

Zusammenfassung von 2 Zeichen, nach dem Prinzip des logischen "UND".
Das 1. Zeichen steht in Adresse Bxxx. Das 2. Zeichen ist das 2. Byte des Befehls.

Beispiel:



Anwendungsbeispiel:

wenn K = 01: Resultat = 00, (Bxxx) war PAARIG
wenn K = 01: Resultat = 01, (Bxxx) war UNPAARIG.

* K darf nicht F1, F2, F3 oder FF sein.

5.5.2 OI

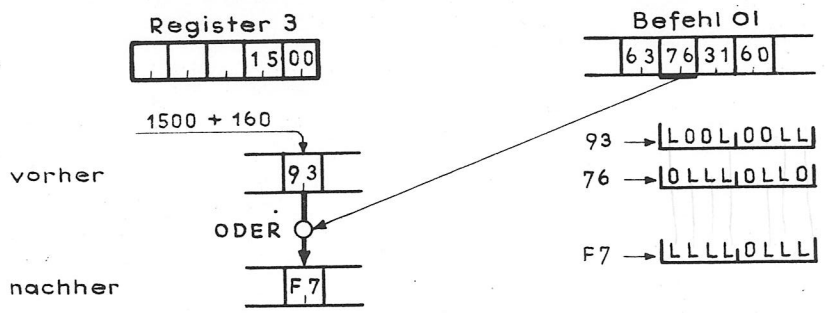
Or immediate

TO		Basis	Versetzung	
6.3	K	B	X	X X
	00-FE*	0-9	000	- 999
1. Byte	2. Byte	3. Byte		4. Byte

Dauer: 1,1 ms

Zusammenfassung von 2 Zeichen, nach dem Prinzip des logischen "ODER".

Beispiel:



5.6 Sprungbefehle

5.6.1 NOP

TO
0.0
1. Byte

Dauer: 0,35 ms

Sprung niemals. Es wird systematisch der nächste Befehl gelesen.

5.6.2 IRT

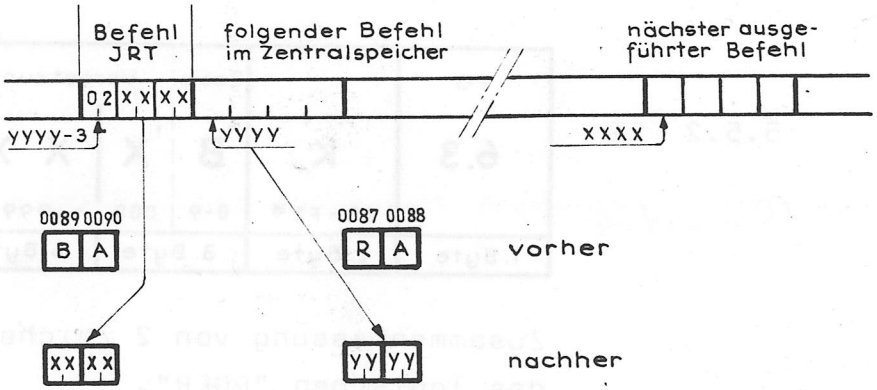
Jump, save Adress for Return

TO	Sprungadresse	
0.2	X X	X X
	siehe 5.5.6	
1. Byte	2. Byte	3. Byte

Dauer: 0,8 ms

Sprung immer, nach Adresse xxxx.
Die Adresse des der IRT folgenden Befehls, wird in Stelle 0087 + 0088 (Rücksprungadresse) gespeichert.

Beispiel:



Jump if immediate octet Equal to store octet.
Save adresse for return

5.6.3 JIERT

Dauer: bei Sprung 1,5 ms
ohne Sprung 1,3 ms

TO		Basis	Versetzung		Sprungadresse		
0.4	K	B	X	X X	X X	X X	
	00-FE*	0-9	000 - 999		siehe 5.5.6		
1. Byte	2. Byte	3. Byte		4. Byte	5. Byte	6. Byte	

Ein Sprung wird durchgeführt, wenn das Zeichen, das sich in Adresse Bxxx befindet, gleich dem Zeichen K ist.

K = XXXX

≠ In diesem Fall verhält sich JIERT wie JRT.

Die Sprungadresse befindet sich in xxxx.

Bei Ungleichheit wird nächster Befehl ausgeführt.

Jump if immediate octet is unequal to store octet.
Save adresse for return.

5.6.4 JIURT

Dauer: bei Sprung 1,5 ms
ohne Sprung 1,3 ms

TO		Basis	Versetzung		Sprungadresse		
0.5	K	B	X	X X	X X	X X	
	00-FE*	0-9	000 - 999		siehe 5.5.6		
1. Byte	2. Byte	3. Byte		4. Byte	5. Byte	6. Byte	

Ein Sprung wird ausgeführt, wenn das Zeichen, das sich in Adresse Bxxx befindet, ungleich dem Zeichen K ist.

K

≠ In diesem Fall, Sprung auf Adresse xxxx.

= Bei Gleichheit wird nächster Befehl ausgeführt.

* K darf nicht F1, F2, F3 oder FE sein.

5.6.5 Verwendung von JIERT und JIURT

In den meisten Fällen werden diese Befehle dazu benutzt, Vergleichsergebnisse auszuwerten und entsprechend dem Vergleichsergebnis den Programmablauf zu steuern. Man hat dann K = 3C (<), 3D (=) oder 3E(>)
und Bxxx = 0086 (CN)

Es sind dann folgende Fälle möglich:

Sprung bei	=	wenn eine	JIERT	und	K	=	3D
	≠	"	"	JIURT	"	K	=	3D
	<	"	"	JIERT	"	K	=	3C
	≥	"	"	JIURT	"	K	=	3C
	>	"	"	JIERT	"	K	=	3E
	≤	"	"	JIURT	"	K	=	3E

5.6.6

LEVEL

TO	/
F.2	X X 00 - FD*
1. Byte	2. Byte

* außer F1, F2 und F3

Dieser Befehl wird nur zur Programmaufstellung verwendet. Die Karte mit diesem Befehl liegt sofort vor der Karte des Befehls, auf den gesprungen werden soll. Der entsprechende Sprungbefehl (JRT, JIERT oder JIURT) hat dann nicht mehr die reelle Adresse xxxx, sondern die symbolische Adresse F1 xx.
xx entspricht dem xx der LEVEL-Karte.
Es ist also nicht mehr erforderlich, die direkte Adresse des Sprungs zu berechnen. Man legt an dieser Stelle eine LEVEL-Karte ein. Die Realadresse wird mit Programm errechnet.

Anmerkung: Der Befehl für das Errechnen der direkten Adresse ist RLA. Dieser wird auf Seite 73 beschrieben.

5.7 Übertragungen für Ein- und Ausgabezonen

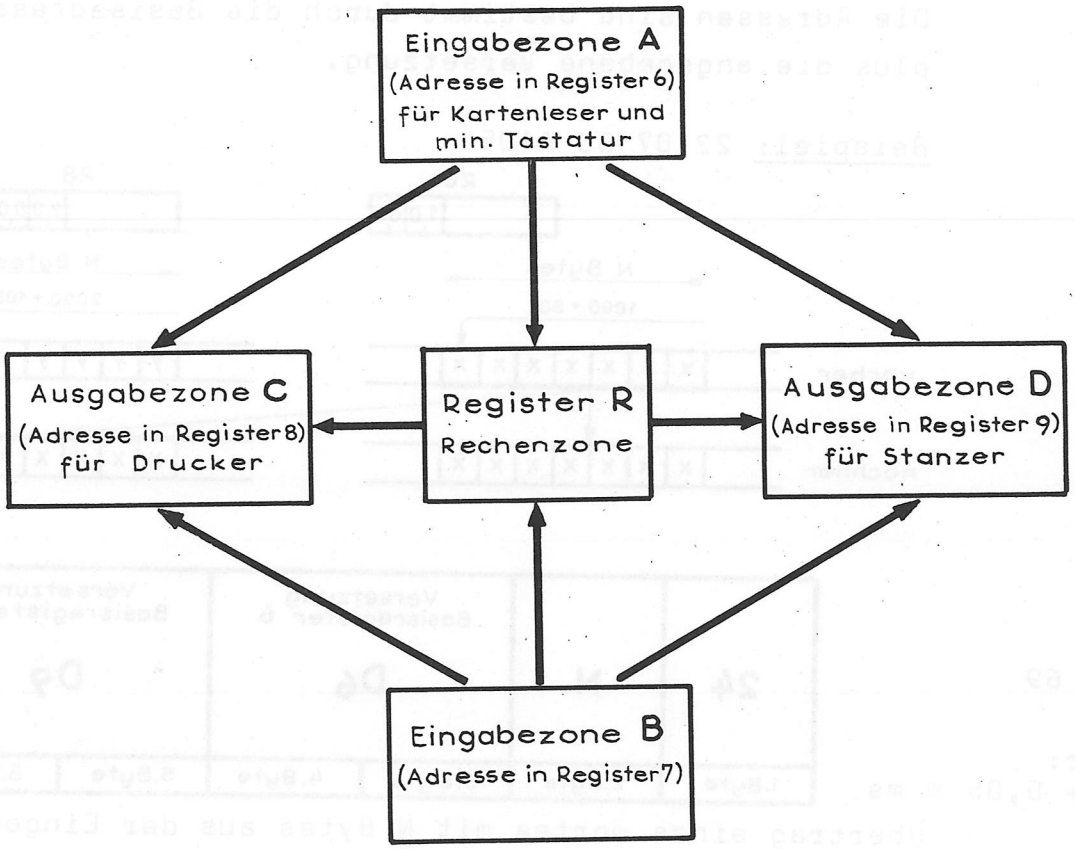
5.7.1 Allgemeines

Diese Befehle erlauben die simultane Übertragung eines Wortes von N Bytes, von

R - einem Register
A - Eingabezone (Basisadressregister 6)
B - Eingabezone (Basisadressregister 7)

nach

R - einem Register
C - Ausgabezone (Basisadressregister 8)
D - Ausgabezone (Basisadressregister 9)



5.7.2 Übertragungen von der Eingangszone zu den Ausgangszonen

Allgemeine Merkmale:
Diese Operation wird ausgeführt wie ein normaler Übertrag:
Es gibt keinen Wechsel des Formats.
Der Vorzeichenspeicher wird nicht verändert.

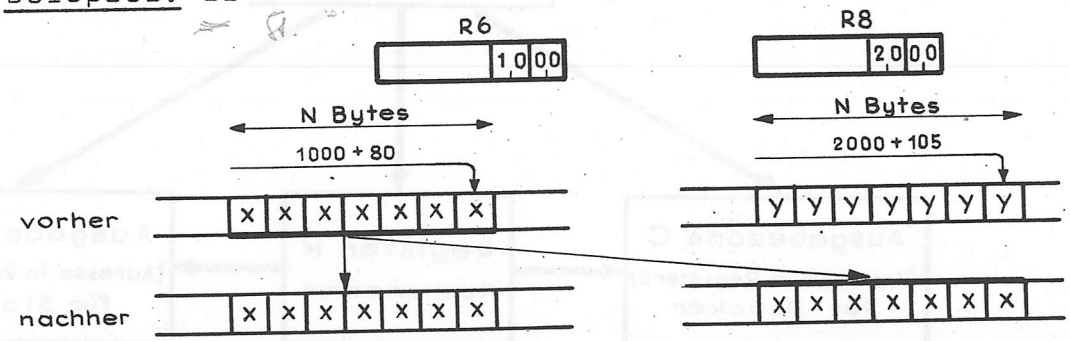
M 68

Dauer:
3,5 + 0,05 N ms

TO	Anz. d. Bytes	Versetzung Basisregister 6		Versetzung Basisregister 8	
22	N 00 - 99	D6		D8	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

Übertrag eines Wortes mit N Bytes aus der Eingangszone A in die Ausgangszone C.
Die Adressen sind bestimmt durch die Basisadressen plus die angegebene Versetzung.

Beispiel: 22 07 80 A105



M 69

Dauer:
3,5 + 0,05 N ms

24	N	Versetzung Basisregister 6		Versetzung Basisregister 9	
		D6		D9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

Übertrag eines Wortes mit N Bytes aus der Eingangszone A in die Ausgangszone D.
Die Operation ist die gleiche wie bei M 68; nur die Zone C wird durch die Zone D ersetzt.

M 689

Dauer:
4 + 0,1 N ms

TO		Versetzung Basisregister 6		Versetzung Basisregister 8		Versetzung Basisregister 9	
26	N 00 - 99	D6		D8		D9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte

Gleichzeitige Übertragung eines Wortes von N Bytes aus Register 6 in Register 8 und 9. Die Übertragung erfolgt Byte pro Byte.

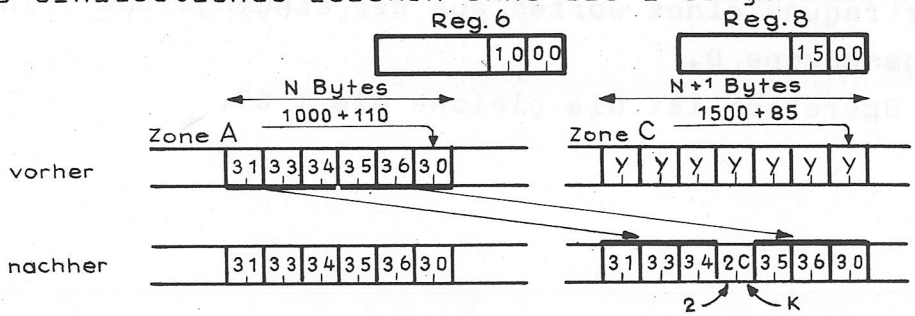
06 A 1100 85 C 3

M68 G

Dauer:
4 + 0,05 N ms

TO							
2E	N	D6		D8		K	N' (Anz. d. Dezimalst.)
						0-E	0-9
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	

Die Operation ist ähnlich wie bei M68. Der Unterschied besteht darin, daß in die Ausgangszone C ein Zeichen (Komma, Punkt usw. unter Berücksichtigung der Dezimalstellen eingesetzt werden kann. Das letzte 7. Byte enthält in seiner linken Hälfte das einzusetzende Zeichen und in der rechten Hälfte die Anzahl der Dezimalstellen. Die Zeichen, welche eingesetzt werden können, sind hexadezimal verschlüsselt. Es sind folgende:
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E.
Die 2 wird immer von der Maschine forciert, so daß in K nur das einzusetzende Zeichen ohne die 2 eingesetzt wird.



Wenn N = 00, findet kein Übertrag statt, aber das Komma ist auf seine Stelle in der Zone C gesetzt worden. Wenn der Wert des Wortes kleiner als 1 ist, wird links vor das eingesetzte Zeichen eine Null gesetzt. Alle im Befehl vorgesehenen Stellen hinter dem eingesetzten Zeichen werden geschrieben.

Auch wenn es eine Leerspalte ist wird eine „0" im ISO-Code = 30 eingesetzt.

M 78

Dauer: 3,5 + 0,05 N ms

TO	Anz. d Bytes	Versetzung Basisregister 7		Versetzung Basisregister 8	
32	N 00 - 99	D7		D8	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

Übertragung eines Wortes aus der Eingangszone B in die Ausgangszone C.
Die Operation ist die gleiche wie M68.

M 79

Dauer: 3,5 + 0,05 N ms

TO	Anz. d. Bytes	Versetzung Basisregister 7		Versetzung Basisregister 9	
34	N 00 - 99	D7		D8	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

Übertragung eines Wortes aus der Eingangszone B in die Ausgangszone D.
Die Operation ist die gleiche wie M 69.

TO	Anz. d. Bytes	Versetzung Basisregister 7		Versetzung Basisregister 8		Versetzung Basisregister 9	
36	N 00 - 99	D_7		D_8		D_9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte

M 789

Dauer: 4 + 0,1 N ms

Übertragung eines Wortes aus der Eingangszone B in die Ausgangszonen C und D.
Die Operation ist die gleiche wie M 689

TO	Anz. d. Bytes	Versetzung Basisregister 7		Versetzung Basisregister 8			Anz. d. Dezimalst.
3E	N 00 - 99	D_7		D_8		K	N'
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	

M78G

Dauer: 4 + 0,05 N ms

Übertragung eines Wortes aus der Zone B in die Zone C mit Einsetzen eines Zeichens (20 bis 2E) entsprechend des 7. Halbbytes dieser Instruktion.
Diese Operation ist die gleiche wie M68G.

5.7.3 Übertragung vom Register zur Ausgangszone

Diese Übertragungen geben entsprechend ihrem Wert in den Vorzeichenspeicher (MS 0085) ein:

2D wenn der Wert < 0 ist, d.h. eine 9 im linken Halbbyte des Registers

20 wenn der Wert ≥ 0 ist, d.h. eine 0 im linken Halbbyte des Registers.

Beim Übertrag vom Register in die Ausgabezonen, wird der Inhalt des Registers, wenn dieser negativ ist, ins Komplement gesetzt. Zu 10^N 10^20.

Zahlen, die sich im Register in Format 2 befinden, erscheinen in der Ausgabezone im Format 1.

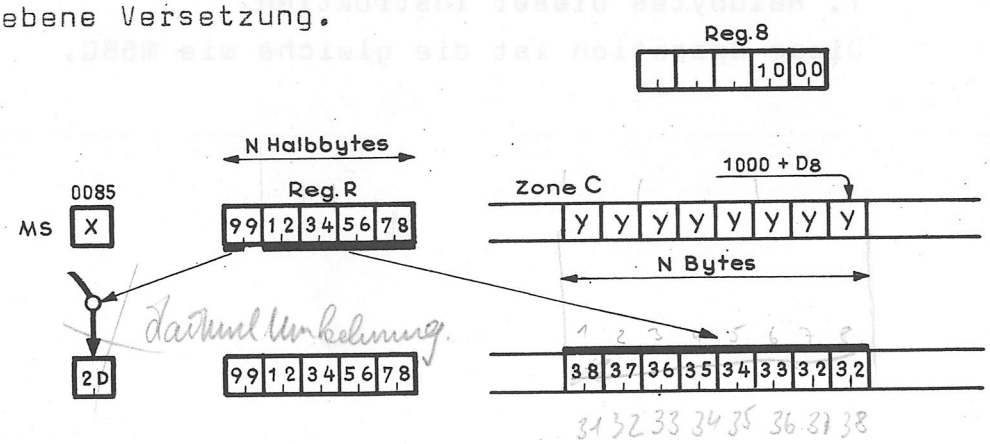
Die Nullen links werden unterdrückt und durch Spaltensprünge ersetzt. (20)

MR8

TO	Anzahl der Halbbytes	Register-Nr.	Versetzung Basisregister 8	
12	N	R	D8	
	00 - 19	00 - 99		
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte

Dauer:
5 + 0,2 N
(+ 0,4 wenn -)ms

Übertragung des Inhaltes von N ~~rechten~~ Halbbytes von Register R in die Zone C.
Die Zone C ist definiert durch Reg. 8 und die angegebene Versetzung.



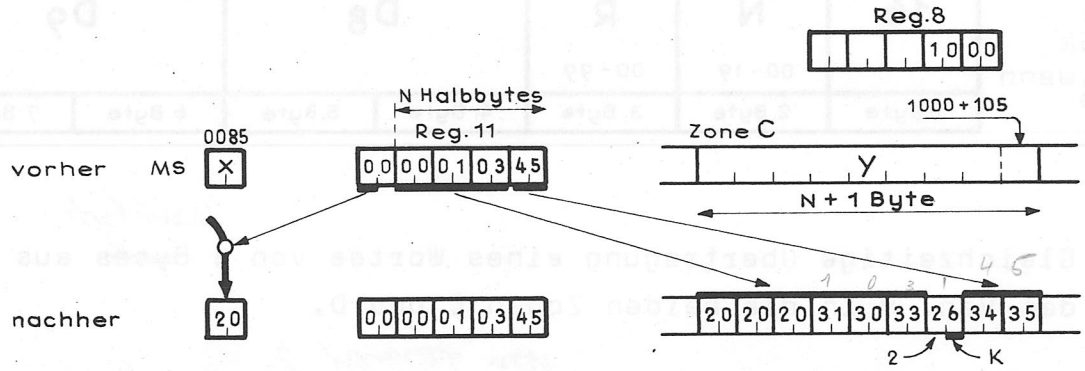
MR8G

Dauer:
5,2 + 0,2 N ms
(+ 0,4 wenn -)

TO	Anzahl der Halbbytes	Register-Nr.	Versetzung Basisregister 8			Anz.d. Dezimalst.
1E	N	R	D8		K	N'
	00 - 99	00 - 99			0-E	0-9
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	

Die Operation ist ähnlich wie bei MR8. Der Unterschied besteht darin, daß in die Zone C ein Zeichen (Komma, Punkt usw.) unter Berücksichtigung der Dezimalstellen eingesetzt werden kann. Das 7. Byte enthält in seiner linken Hälfte das einzusetzende Zeichen und in der rechten Hälfte die Anzahl der Dezimalstellen. Die Zeichen, die eingesetzt werden können, sind hexadezimal verschlüsselt. Es sind folgende:
20, 21, 22, 23, 24, 25, 26, 27, 28, 29 2A, 2B, 2C, 2D, 2E.
Die 2 wird immer von der Maschine forciert, so daß in K nur das einzusetzende Zeichen steht.

Beispiel: 1E.08.11.A105.C2



Wenn N = 00, findet kein Übertrag statt, aber das Komma ist auf seine Stelle in der Zone C gesetzt worden.

Wenn der Wert des Wortes kleiner als 1 ist, wird links vor das eingesetzte Zeichen eine Null gesetzt.
Alle im Befehl vorgesehenen Stellen hinter dem eingesetzten Zeichen werden geschrieben.

MR9

Dauer: 5 + 0,2N
(+ 0,4 wenn -)ms

TO	Anzahl der Halbbytes	Register-Nr.	Versetzung Basisregister 9	
14	N 00 - 19	R 00 - 99	D9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte

Übertragung des Inhaltes von N rechten Halbbytes aus dem Reg. R in die Zone D.
Die Instruktion ist der von MR8 ähnlich. Anstelle der Zone C tritt hier die Zone D.

von Format 2 (gepackt)
in Format 1 ungepackt.

MR89

Dauer:
6+0,25N
(+0,4,wenn -) ms

TO	Anzahl der Halbbytes	Register-Nr.	Versetzung Basisregister 8		Versetzung Basisregister 9	
16	N 00 - 19	R 00 - 99	D8		D9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

Gleichzeitige Übertragung eines Wortes von N ~~Bytes~~ Halbbytes aus dem Reg. R in die beiden Zonen C und D.

von Format 2
in Format 1.

5.7.4 Übertragung der Eingangszone in ein Register

Bei dieser Übertragung ist der algebraische Wert entsprechend dem Inhalt des MS (0085 im MC).

20 oder 00 - der Wert ist ≥ 0 Memoire signe -

2D - der Wert ist < 0

Es ist also eventuell eine Änderung des Inhaltes von MS vor der Übertragung erforderlich.

Am Ende der Operation ist der MS auf 00.

Zwischencode Umspeichl ISO Code

Umsetzen von Format 1 in Format 2.

Vor der Übertragung ist der Wert normal codiert, also in Format 1.

Nach der Übertragung steht der Wert in seiner algebraischen Form also in Format 2.

	TO	Anzahl der Bytes	Versetzung Basisregister 6		Register-Nr.
	21	N 00 - 19	D6		R 00 - 99
M6R	1. Byte	2. Byte	3. Byte	4. Byte	5. Byte

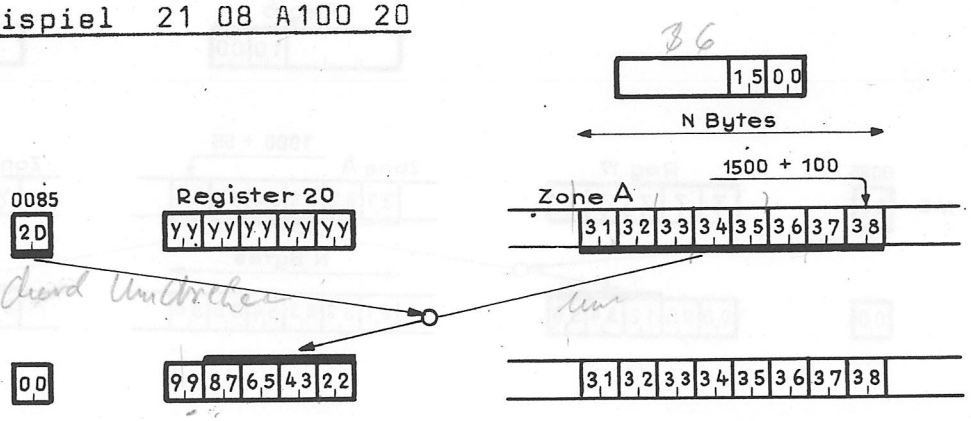
Dauer: 4,7 + 0,2 N ms

Übertragung eines Wortes mit N Bytes aus der Eingangszone in ein Register R.

Einfachregister wenn N < 10

Doppelregister wenn N ≥ 10

Beispiel 21 08 A100 20



M7R

Dauer: 4,7+0,2 N ms

TO	Anzahl der Bytes	Versetzung Basisregister 7		Reg.-Nr.
31	N 00 - 19	D7		R 00 - 99
1.Byte	2.Byte	3.Byte	4.Byte	5.Byte

Übertragung eines Wortes mit N Bytes aus der Eingangszone in ein Register R.
Die Operation ist ähnlich wie M6R. Anstelle Zone A tritt hier die Zone B.

5.7.5 Übertragung der Eingangszone in ein Register und eine Ausgangszone

Dieser Befehl erlaubt zwei verschiedene Übertragungen simultan, wenn die abgebende Zone gemeinsam ist:

- Eine Übertragung Eingangszone zum Register (siehe 5.7.4)
- Eine Übertragung Eingangszone zur Ausgangszone (s. 5.7.3)

TO N Da R Dc

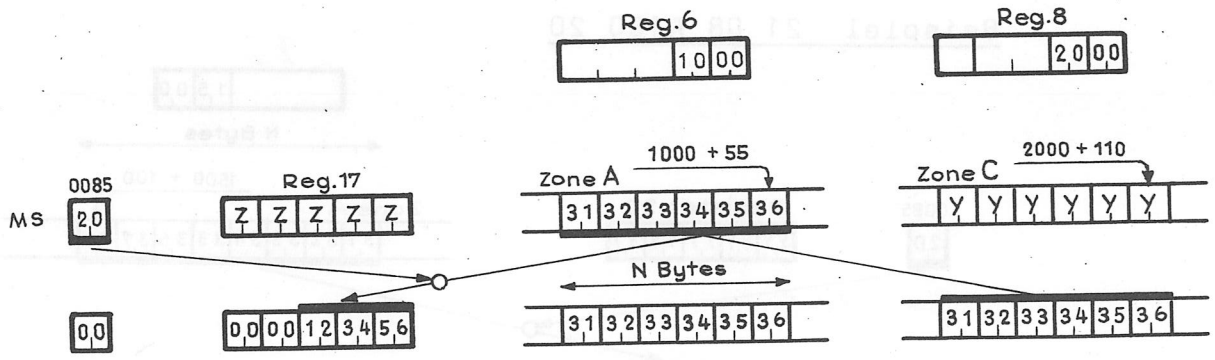
TO	Anz. d. Bytes	Versetzung Basisregister 6		Reg.-Nr.	Versetzung Basisregister 8	
23	N 00 - 19	D6		R 00 - 99	D8 Dc	
1.Byte	2.Byte	3.Byte	4.Byte	5.Byte	6.Byte	7.Byte

M6R8 ✓

Dauer: 5,8+0,25N ms

Übertragung eines Wortes mit N Bytes aus der Eingangszone A in das Register R und die Ausgangszone C.

Beispiel 23 06 55 17 A110



M6R9

TO	Anz. d. Bytes	Versetzung Basisregister 6		Reg-Nr.	Versetzung Basisregister 9	
25	N 00 - 19	D6		R 00 - 99	D9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

Dauer: 5,8+0,25N ms

Übertragung eines Wortes mit N Bytes aus der Eingangszone A in das Register R und in die Ausgangszone D. Die Instruktion ist ähnlich wie M6R8, die Zone C wird durch die Zone D ersetzt.

M6R89

TO	Anz. d. Bytes	Versetzung Basisregister 6		Reg-Nr.	Versetzung Basisregister 8		Versetzung Basisregister 9	
27	N 00 - 19	D6		R 00 - 99	D8		D9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte	9. Byte

Dauer: 7+0,3N ms

Übertragung eines Wortes mit N Bytes aus der Eingangszone A in ein Register R und die Ausgangszonen C und D.

M7R8
Dauer: 5,8+0,25N ms

TO	Anz. d. Bytes	Versetzung Basisregister 7		Reg.-Nr.	Versetzung Basisregister 8	
33	N	D7		R	D8	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

Übertragung eines Wortes mit N Bytes aus der Zone B in das Register R und die Ausgangszone C.

Diese Instruktion ist ähnlich wie M6R8.
Anstelle der Eingangszone A tritt die Eingangszone B.

M7R9
Dauer: 5,8+0,25N ms

TO	Anz. d. Bytes	Versetzung Basisregister 7		Reg.-Nr.	Versetzung Basisregister 9	
35	N	D7		R	D9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

Übertragung eines Wortes mit N Bytes aus der Eingangszone B in ein Register R und die Ausgangszone D.

Diese Instruktion ist ähnlich wie M6R9.
Anstelle der Eingangszone A tritt die Eingangszone B.

M7R89
Dauer: 7+0,3N ms

TO	Anz. d. Bytes	Versetzung Basisregister 7		Reg.-Nr.	Versetzung Basisregister 8		Versetzung Basisregister 9	
37	N	D7		R	D8		D9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte	9. Byte

Übertragung eines Wortes mit N Bytes aus der Eingangszone B in ein Register R und die Ausgangszonen C und D.

Diese Instruktion ist ähnlich wie bei M6R89.
Anstelle der Eingangszone A tritt die Eingangszone B.

5.8 Anschlußbefehle für Randeinheiten

			1. Randeinheit		2. Randeinheit		n. Randeinheit (<7)	
IOC	53	Anzahl d. Randeinheiten N	C_1	R_1	C_2	R_2	C_n	R_n
	1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte

N = Anzahl der anzuschließenden Randeinheiten (N < 7)

R = Nummer des Registers, welches die Adresse der Zone enthält, mit der die Randeinheit korrespondiert.

C = Anschlußcode entsprechend folgender Tafel:

Kanal	abgebende Randeinheit MHZ	OHZ*		annehmende Randeinheit MHZ	OHZ*	
1	92	95	Pufferspeicher	9A	9D	Pufferspeicher
1		96	Kartenleser			
1	94	97	α-Tastatur			
2				AA	AD	Stanzer
3				CA	CD	Drucker

In einer IOC dürfen nicht 2 Randeinheiten an den selben Kanal angeschlossen sein.

* MHZ = mit Halt pro Zeichen, OHZ = ohne Halt pro Zeichen.
Diese Unterscheidung ist interessant bei Multiprogrammierung. Bei Monoprogramm kann das eine oder andere verwendet werden.

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 stopt 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.

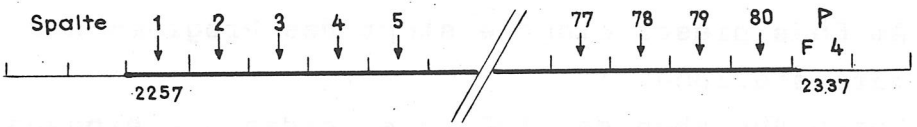
Anmerkung: Die Wartestellung des Programmes während IOC, kann bei der Multiprogrammierung zur Bearbeitung eines anderen Programmes ausgenutzt werden.

Programmbeispiele für IOC (1. Randeinheit pro Befehl)

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

IOC 53 01 96 06

Register 6 | 2 2 | 5 6



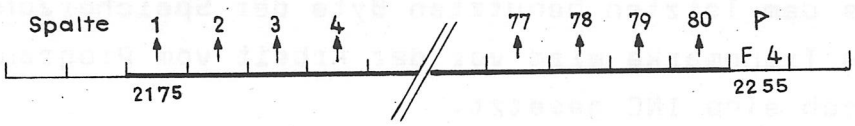
Der Leser kann nicht zwischen 2 Zeichen halten. Es wird also jeweils eine Karte, ohne Unterbrechung gelesen.

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

IOC 53 01 AA 09

Register 9 | 2 1 | 7 4



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 gestopt 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

Input output control

IOC 53 01 CA 08

Register 8

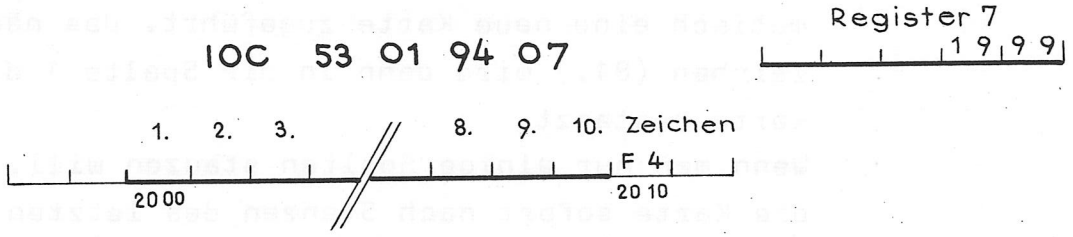
Der Drucker kann in jeder Spalte halten.
Der Abdruck einer Speicherzone beginnt da, wo der Wagen steht.
Für die genaue Plazierung des Abdrucks ist es daher notwendig, einen entsprechenden Programmcode in die Speicherzone zu plazieren.

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 54	N	RC_1	R_1	RC_2	R_2	RC_n	R_n
	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

Translate

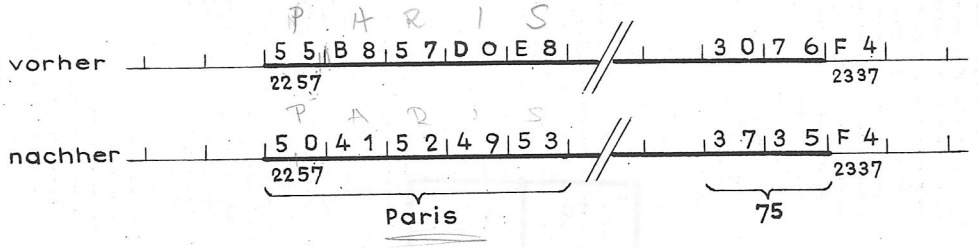
Übersetzung des Code T 121 (8 Bits) in den ISO-Code (7 Bits)

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	
5A	R
1.Byte	2.Byte

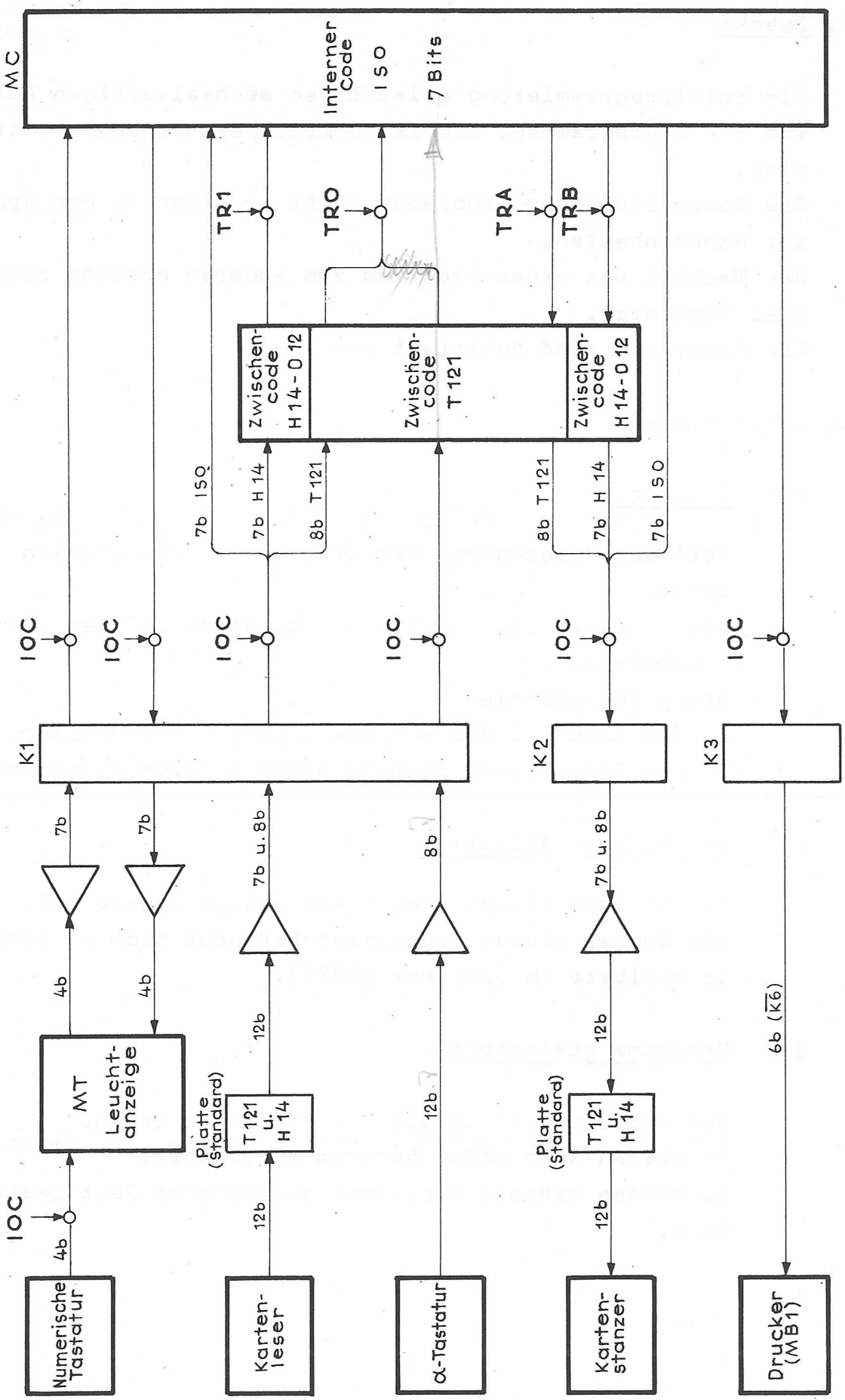
Übersetzung des ISO-Code (7Bits) in den Code T121 (8 Bits)

5.9.5 TRB

TO	
5B	R
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, stopt 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 Randeinheiten ihre Arbeit beendigt haben.
Wenn ein Programm in Wartestellung geht, wird das, als nächstes gestartete Programm in Arbeit gesetzt.
Wenn die Randeinheiten ihre Arbeit beendigt 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 gestoptes Programm kann nur durch den Befehl "START" wieder gestartet werden.

S T O P
↓
Programm in Arbeit stopt
↓
Eingabe von n = 0 und p = Nummer des Programmes in die Zone ZAT.
↓
Wechsel des Programmes

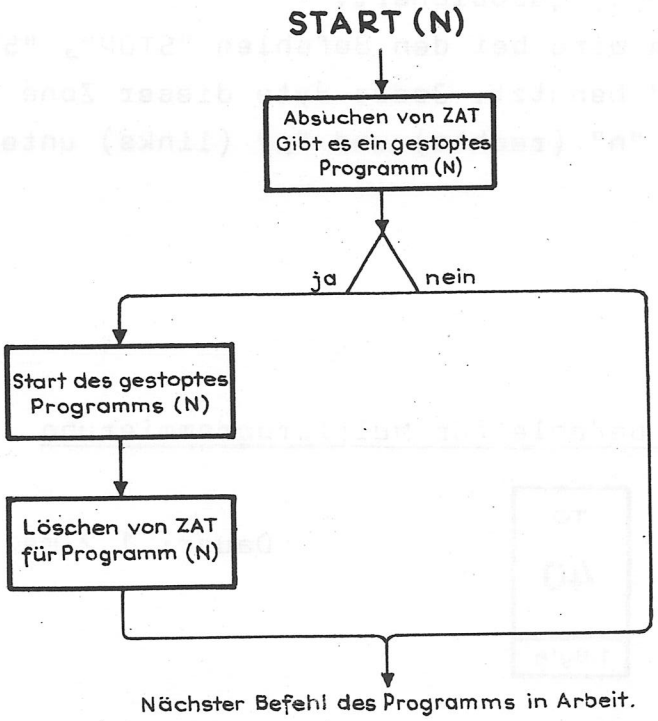
3.2

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.



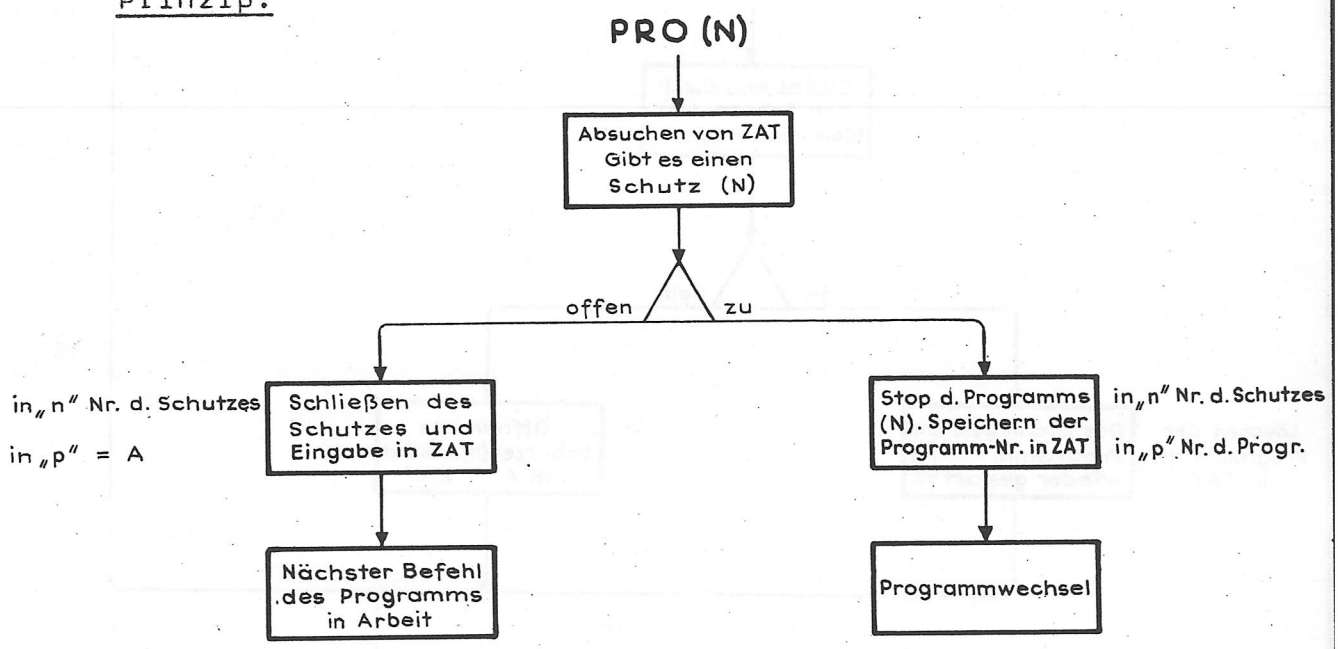
3.3 PRO

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:

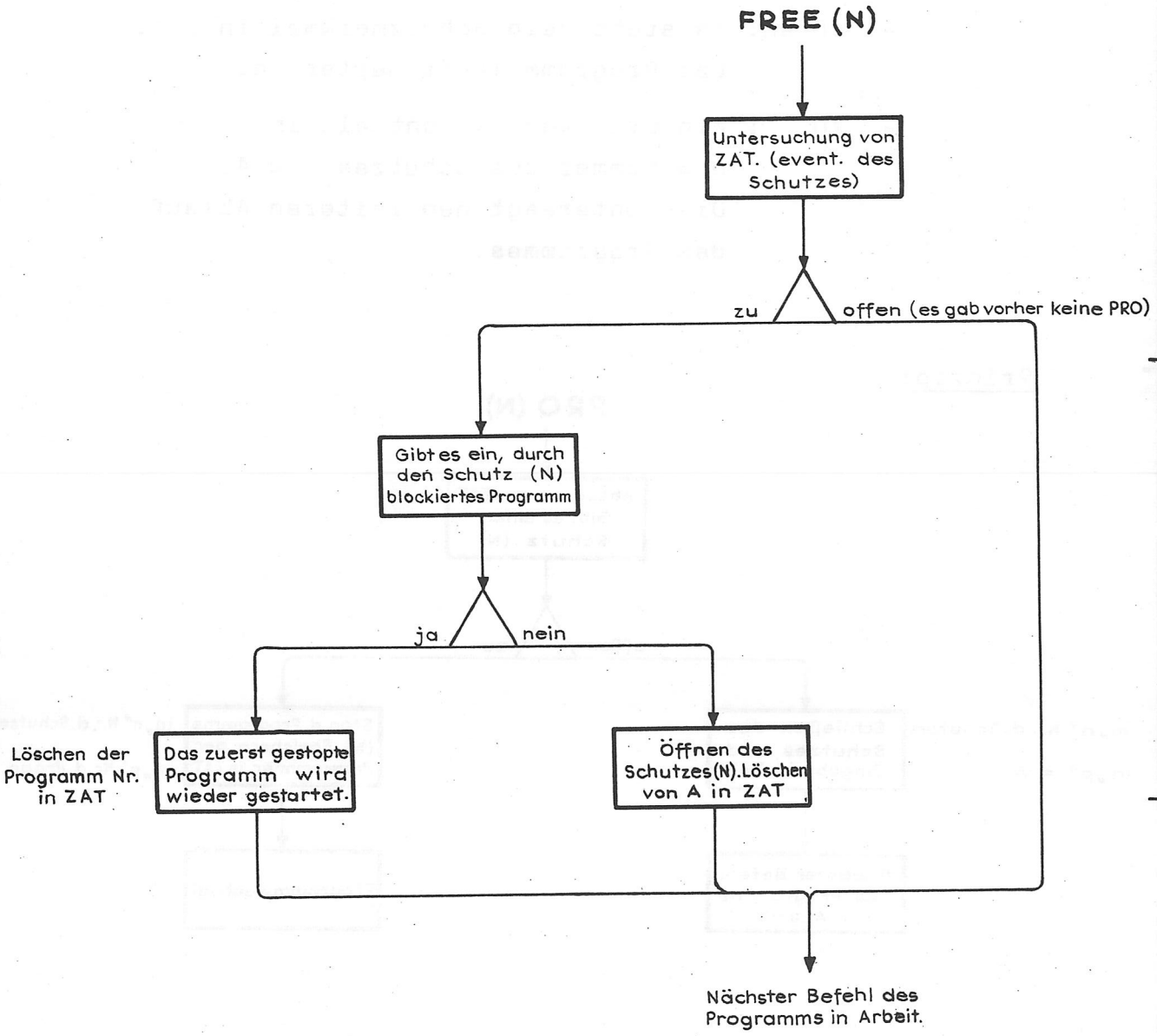


3.4 FREE

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

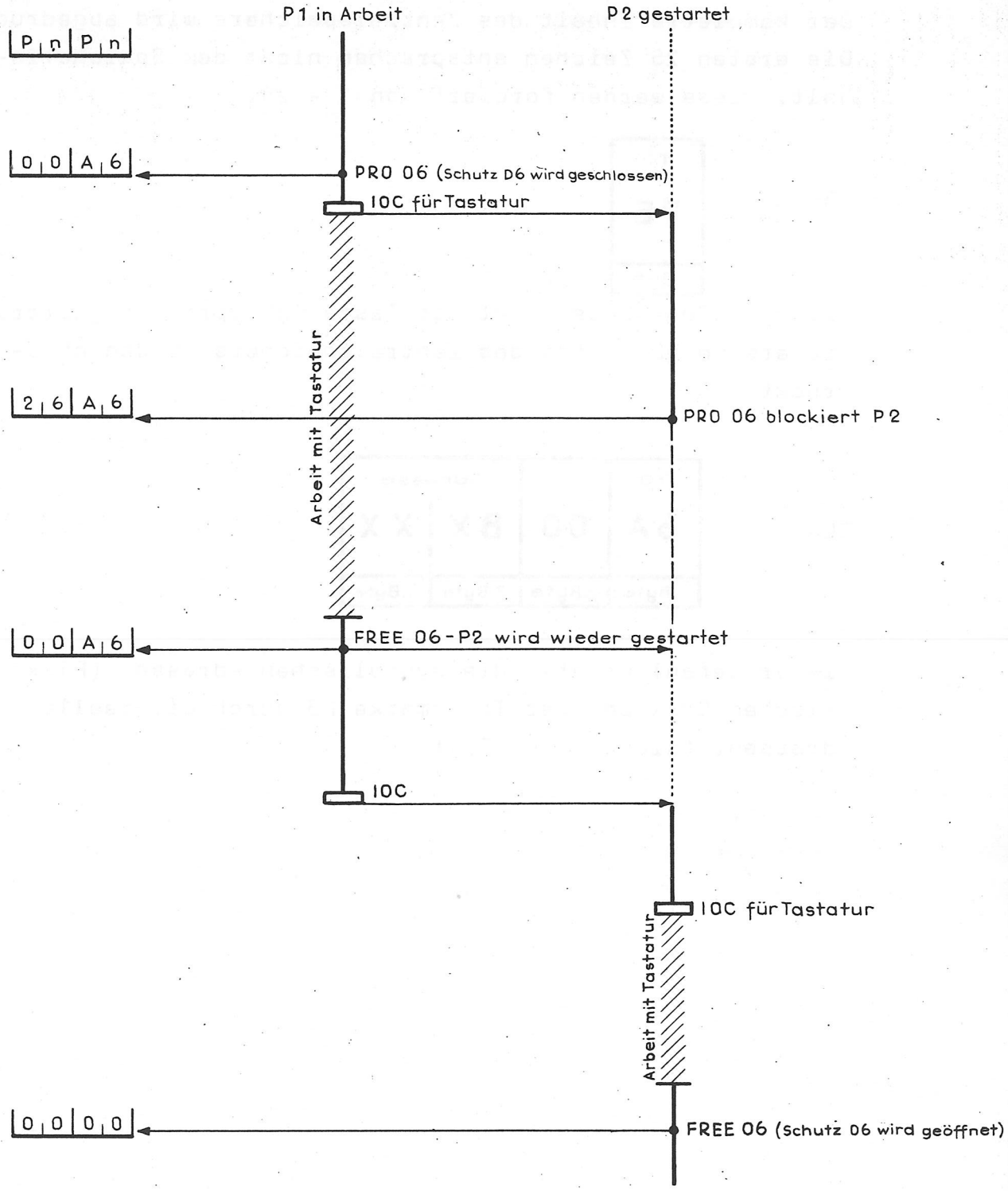
Dieser Befehl hebt den Schutz (N) auf und erlaubt den weiteren Ablauf des Programmes.

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
7.8
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
7E
1.Byte

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

RLA

TO		Adresse	
6A	00	BX	XX
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:

JIURT	05	3C	0086	F100	0396
CMC	88	04	0175	0205	0402
JIURT	05	3C	0086	F101	0408
CMC	88	04	0175	0210	0414
JIURT	05	3C	0086	F102	0420
M6R	21	02	A245	23	0426
JRT	02	F1	03		0431
LEVEL	F2	02			0434
M6R	21	02	A247	23	0436
JRT	02	F1	03		0441
LEVEL	F2	01			0444
M6R	21	02	A249	23	0446
JRT	02	F1	03		0451
LEVEL	F2	00			0454
M6R	21	02	A251	23	0456
LEVEL	F2	03			0461
MPD	C1	23	19		0463
	F3				

0396	JIURT	05	3C	0086	0450
0402	CMC	88	04	0175	0205
0408	JIURT	05	3C	0086	0442
0414	CMC	88	04	0175	0210
0420	JIURT	05	3C	0086	0434
0426	M6R	21	02	A245	23
0431	JRT	02	04	55	
0434	M6R	21	02	A247	23
0439	JRT	02	04	55	
0442	M6R	21	02	A249	23
0447	JRT	02	04	55	
0450	M6R	21	02	A251	23
0455	MPD	C1	23	19	
		00	21		

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
57
1.Byte

Dieser Befehl stopt das Programm, wenn die Taste KHLT betätigt ist.
Durch Einlegen dieser Befehlskarte,zwischen die übrigen Karten des Programmes, kann man das Programm an jeder beliebigen Stelle anhalten.

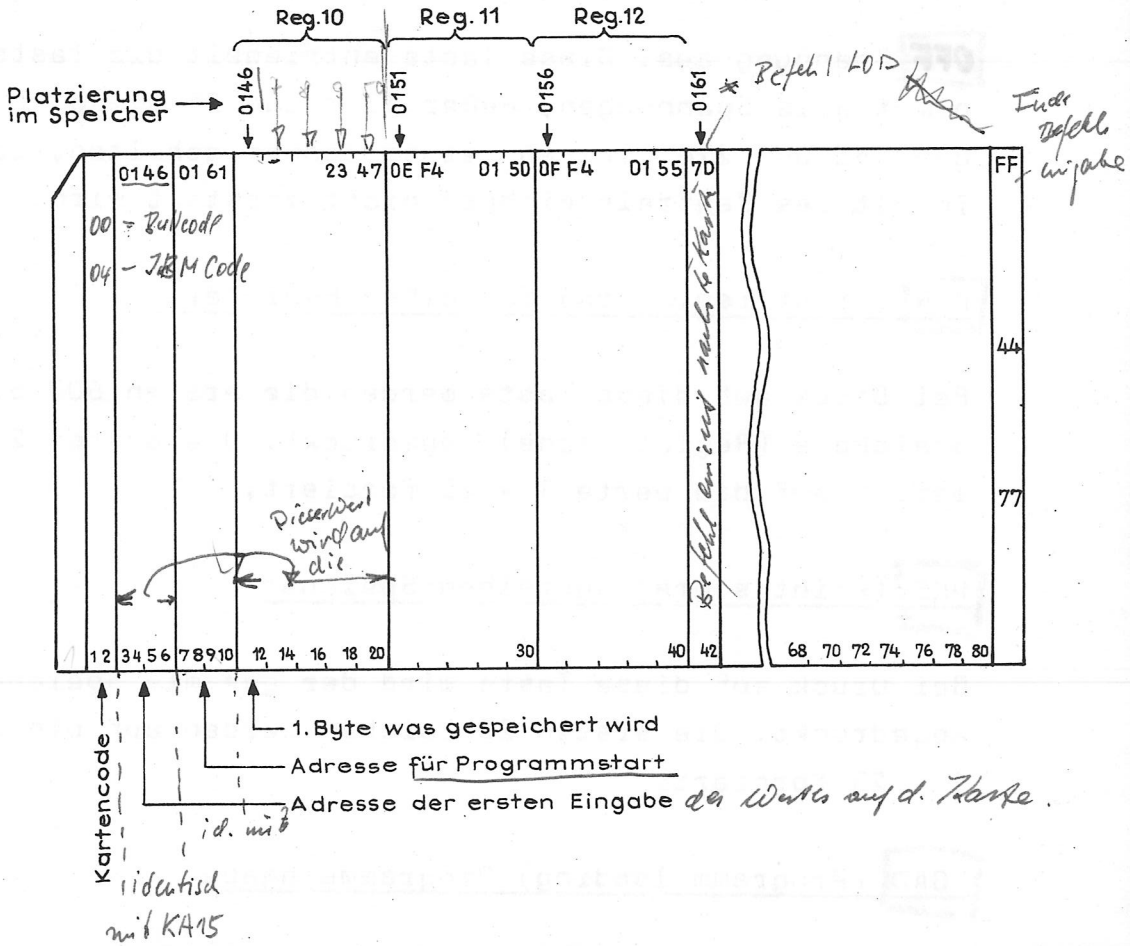
LOAD

TO
7D
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 gespeichert 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 Angabe

Beispiele auf schriftlich

8.168

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.

Die 4 Tasten PRR, LOAD, PRS und RUN sind beleuchtet, sobald die Temperatur des Kernspeichers auf 40° ist.
Während der Aufheizung des Speichers kann das Programm nicht ablaufen und die Tasten sind dunkel.
Außerdem haben diese 4 Tasten folgende Wirkung gemeinsam:

- Stop der Rhythmen
- Löschen der Anschlußregister (V)
- Löschen des Flip-Flop KP (Kapazitätsüberschreitung)
- Forcierung auf eine Adresse des Festspeichers
- Start der Rhythmen.

STOP

Bei Druck auf diese Taste werden die Rhythmen am Ende des laufenden Zyklus, gestopt. Das heißt, am Ende von T2.

START (Start timing)

Diese Taste startet den Startkreis für die Rhythmen.
Erst nach Freigeben der Taste START, werden die Rhythmen gestartet.

PI (Programm interrupt) Programmunterbrechung

Der Druck auf diese Taste forciert das Flip-Flop IP (B023) auf 1. Der nächste Rhythmus T0 bringt das Flip-Flop B024 und hiermit die, in der Taste IP eingebaute Lampe. Sobald die Mikrofunktion VIP gestartet ist, wird das Flip-Flop B024 wieder auf 0 gesetzt und die Lampe erlischt.
Durch VIP wird, vor Ablauf des nächsten Befehls, das Hauptprogramm (Unterbrechungsprogramm) gestartet.
D.h. die Adresse in (0089 - 0090) wird in (0092 - 0093) gespeichert und die Adresse des Fehlerprogrammes wird von (0094 - 0095) in (0089 - 0090) eingegeben.
(Siehe auch PGI im Band Software).

RPI (Reset program interrupt) Aufheben Programmunterbrechung

Mit dieser Taste wird die Programmunterbrechung unter der Bedingung, daß das Fehlerprogramm noch nicht gestartet ist, aufgehoben.

SEQ (step by step sequence) Schritt für Schritt, Sequenz

Bei Druck auf diese Taste werden die Rhythmen bei Auffinden des nächsten Befehls KHLT, gestopt. Die Taste bleibt verriegelt. Der erneute Start ist nur über die Taste START möglich. In diesem Fall startet das Programm auf der gespeicherten Adresse und läuft bis zur nächsten KHLT.

INS (Step by step instruction) Schritt für Schritt, Befehl

Bei Druck auf diese Taste werden die Rhythmen vor Ablauf der Befehlsanalyse für den folgenden Befehl (A.P. - 0.0.00.01) gestopt.
Jeder Druck auf die Taste START bringt den Ablauf eines Befehls.

ACHTUNG !

- Um zu verhindern, daß beim Abdruck des Register bzw. Speicherinhaltes, (gestartet durch PRR oder PRS) das Programm bzw. die logische Zone zerstört wird, ist es unbedingt erforderlich, vorher das Programm durch Druck auf die Taste INS zu stoppen.
  Bei Beendigung des Abdruckes (oder Stop durch Taste STOP) wird diese Taste wieder auf Normal gestellt und das Programm durch die Taste START wieder gestartet.

- Das gleiche ist zu beachten, wenn man die Maschine ausschaltet.

Man drückt erst INS und anschließend STOP.
Wenn jetzt durch die Taste OFF die Spannung abgeschaltet wird, hat man die Gewähr, daß das Programm nicht zerstört wird, da die Maschine zwischen 2 Befehlen steht.

INH Diese Taste blockiert die Rhythmen und setzt den Betriebsstundenzähler für den Kunden außer Betrieb.
Der Techniker kann nun, durch Einschalten eines Schlüsselschalters am Netzteil, die Rhythmen wieder starten, setzt aber gleichzeitig hiermit den Betriebsstundenzähler für Wartungsarbeiten in Betrieb.

Lampen:

A (Alpha)

Diese Lampe leuchtet, wenn eine Eingabe über die Alphatastatur erforderlich und diese angeschlossen ist.

N (Numerisch)

Diese Lampe leuchtet, wenn eine Eingabe über die numerische Tastatur erforderlich und diese angeschlossen ist.

PR = Störung durch Drucker

PU = Störung durch Kartenstanzer

R = Störung durch Kartenleser

EXT = Störung durch eine entfernt stehende Randeinheit.

PAR = Schlüsselfehler (Paarigkeitskontrolle) oder Kapazitätsüberschreitung (Adresse über 2499 bei 2500 Speicherstellen)

RDO = (Leuchtanzeige) Eine Information ist vom Zentralspeicher im Pufferspeicher angekommen und auf der Leuchtanzeige sichtbar gemacht worden.

Wartungspult

Das Wartungspult befindet sich auf der Rückseite der Zentraleinheit unter der Verkleidung und kann nur durch den Techniker bedient werden.

Die Taste "Départ Horloge" startet die Rhythmen und entspricht der Taste START.

Die Taste "Arrêt Horloge" stopt die Rhythmen und entspricht der Taste STOP.

Ein Schalter Pas à pas Cycle, erlaubt den Ablauf eines einzelnen Zyklus, durch Druck auf die Taste "Depart Horloge".

Ein Potentiometer erlaubt die Regulierung des Heizstroms für den Kernspeicher. Diese Regulierung geschieht nur durch den Techniker, mit Hilfe eines Amperemeter. (100 uA).

"Marges mémoire", ist ein Schalter mit 12 Positionen. Mit diesem Schalter wird die Vorspannung der Lese-Vorverstärker (Préamplis) eingestellt.
Seine normale Stellung ist in Position 7.

Eine Serie von Meßbuchsen erlaubt die einzelnen Spannungen der Zentraleinheit zu kontrollieren.

Durch einen Schalter mit den Positionen A - J können die Register der Zentraleinheit, mit Hilfe von DM 160, beobachtet werden.

Eine nichtmarkierte Position hinter J erlaubt die richtige Funktion der DM 160, zu prüfen.

Durch eine Serie von Schaltern kann LART forciert werden.

Durch eine zweite Serie von Schaltern kann auf einer gewünschten Adresse des Festspeichers gestopt werden.

Auf dem Stanzer P112 A sitzt ein Schalter zum Abschalten des Motors.
In seiner oberen Stellung ist der Motor freigegeben.

Links von diesem Schalter sitzt ein zweiter Schalter, der die Warnerkupplung permanent erregt.
In seiner oberen Stellung ist die Warnerkupplung immer erregt.

Allgemeiner Aufbau

Für die Benutzung des Zentralspeichers, der die Programme und die benötigten Daten enthält, gibt es eine Anzahl Register und Flip-Flops (Seite 87).

Die verschiedenen Verbindungen zwischen den Registern und Flip-Flops werden durch die Mikrofunktionen hergestellt. Im Laufe eines Mikrofunktion (7,9 us), führt die Zentraleinheit eine Elementarfunktion aus.

Z.B.: Die Übertragung von einem Register zum anderen, der Wechsel zwischen zwei Register, die Addition oder Subtraktion einer Dezimalstelle von einem Register usw..

Jeder Programmbefehl besteht aus einer Folge von Mikrofunktionen.
Diese Mikrofunktionen sind im Festspeicher fest verdrahtet.

Register M (M = mémoire). Dies ist das Auswahlregister des Zentralspeichers.
Es besitzt 16 Binärpositionen (16 Flip-Flops).
Da es sich hierbei um eine Dezimalauswahl handelt, sind die 16 Binärpositionen in 4 Gruppen aufgeteilt. Jede Gruppe stellt bei der Auswahl die Werte 0 - 9 dar.
Man kann also die Adressen 0000 - 9999 darstellen.

Register C (C = Compteur decompteur = Adressrechner)

Dieses Register entspricht inseinem Aufbau dem Register M.
Durch Mikrofunktionen kann man 1 (dezimal) zuzählen oder abzählen.
C enthält die für M bestimmte Adresse.

Sonderfälle: Wenn der Zentralspeicher 2500 Bytes hat:
2499 + 1 = 0000, 0000 - 1 = 2499

Wenn der Zentralspeicher 5000 Bytes hat:
4999 + 1 = 0000, 0000 - 1 = 4999

Wenn der Zentralspeicher 10000 Bytes hat:
9999 + 1 = 0000, 0000 - 1 = 9999

Register D (D = Décalage = Versetzregister)

Dieses Register besitzt ebenfalls 16 Binärpositionen und dient, in Verbindung mit Register L, zur Versetzung der Adressen.
Es besteht folgende Besonderheit: Die rechte Hälfte des Registers kann in die linke Hälfte des Registers übertragen werden (u - d nach c - m).

1 mal 10, Hundert mal tausend

Register G und H

Diese beiden Register haben je 4 Binärpositionen, d.h. sie entsprechen 1 Dezimalstelle.
Beide Register zusammen, erlauben die Darstellung der Dezimalwerte 00 - 99.
Nur das Register G kann plus 1 oder minus 1 zählen.
Da man die beiden Register wechseln kann, ist es möglich, von 00 - 99 und umgekehrt zu zählen.

Register L (L = Liaison = Ein-Ausgabe-Register)

Dieses Register stellt die Verbindung zwischen Zentralspeicher und Festspeicher sowie Zentralspeicher und den Kanälen her.
Es besteht aus 9 Binärpositionen, welche das gelesene Zeichen empfangen oder in den Zentralspeicher eingeben.
Die neunte Position empfängt den Paarigkeitsschlüssel.

Register A (A = auxiliaire = Hilfsregister)

Dies ist ein Hilfsregister für das Register L und besteht aus 8 Binärpositionen.
Die 4 rechten Binärpositionen haben eine besondere Aufgabe. Dieser Teil entspricht einem Zähler. Der Inhalt kann um 1 (dezimal) addiert oder subtrahiert werden.
Der Inhalt kann auf 10 forciert werden.

Flip-Flops R und T

Diese beiden Flip-Flops formen mit Lu und Au die Auswahladresse für den Festspeicher.
Außerdem wird in R, bei Benutzung der Rechentafeln, der eventuelle Übertrag gespeichert.
T erlaubt die Verbindung der Festspeicheradressen zu je 2.

L.A.R.T.

Dies ist kein Register, sondern die Verbindung von:

der rechten Hälfte Register L
der rechten Hälfte Register A
dem Flip-Flop R
und dem Flip-Flop T

L.A.R.T. kann eine Adresse (Auswahladresse für Festspeicher) nach Register S übertragen oder eine Adresse von Register N empfangen.
Register N ist selbst mit dem Festspeicher verbunden.

Flip-Flop IP: (IP = interruption programme)

Dieses Flip-Flop wird bei einer Programmunterbrechung gesetzt. Durch Auswertung dieses Flip-Flops, nach jedem Programmbefehl, kann das Fehlerprogramm gestartet werden. (s. auch Teil Programmierung)

Flip-Flop KP

Dieses Flip-Flop wird gesetzt, wenn ein Fehler bei der Paarigkeitskontrolle im Zentralspeicher auftritt.
Es bewirkt einen sofortigen Stop der Rhythmen.
Die Lampe KP leuchtet.

Festspeicher und Mikrofunktionen

Allgemeines: Der Festspeicher enthält, bei der Fabrikation, festgelegte Informationen.

- Die Informationseinheit des Festspeichers ist das Wort. Ein Wort enthält 36 Bits.
- Der Festspeicher enthält 1024 Wörter, aufgeteilt in 512 Doppelwörter. Jedes Wort kann adressiert werden.
- Man kann in jedem Zyklus nur 1 Wort lesen.

Ein Wort kann enthalten:

- einen Teil der Tafeln (Rechnen, Übersetzen usw.)
- eine Elementarfunktion, welche innerhalb eines Zyklus, die Verbindung zwischen zwei Registern oder zwischen einem Register und dem Zentralspeicher steuert.

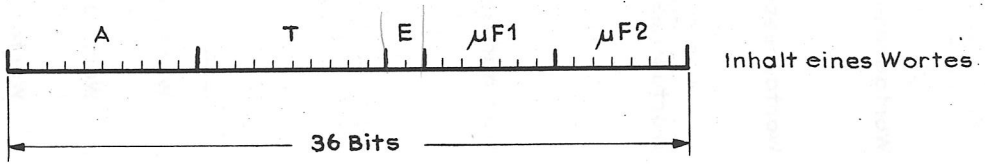
Diesen Elementarbefehl nennt man Mikrofunktion.

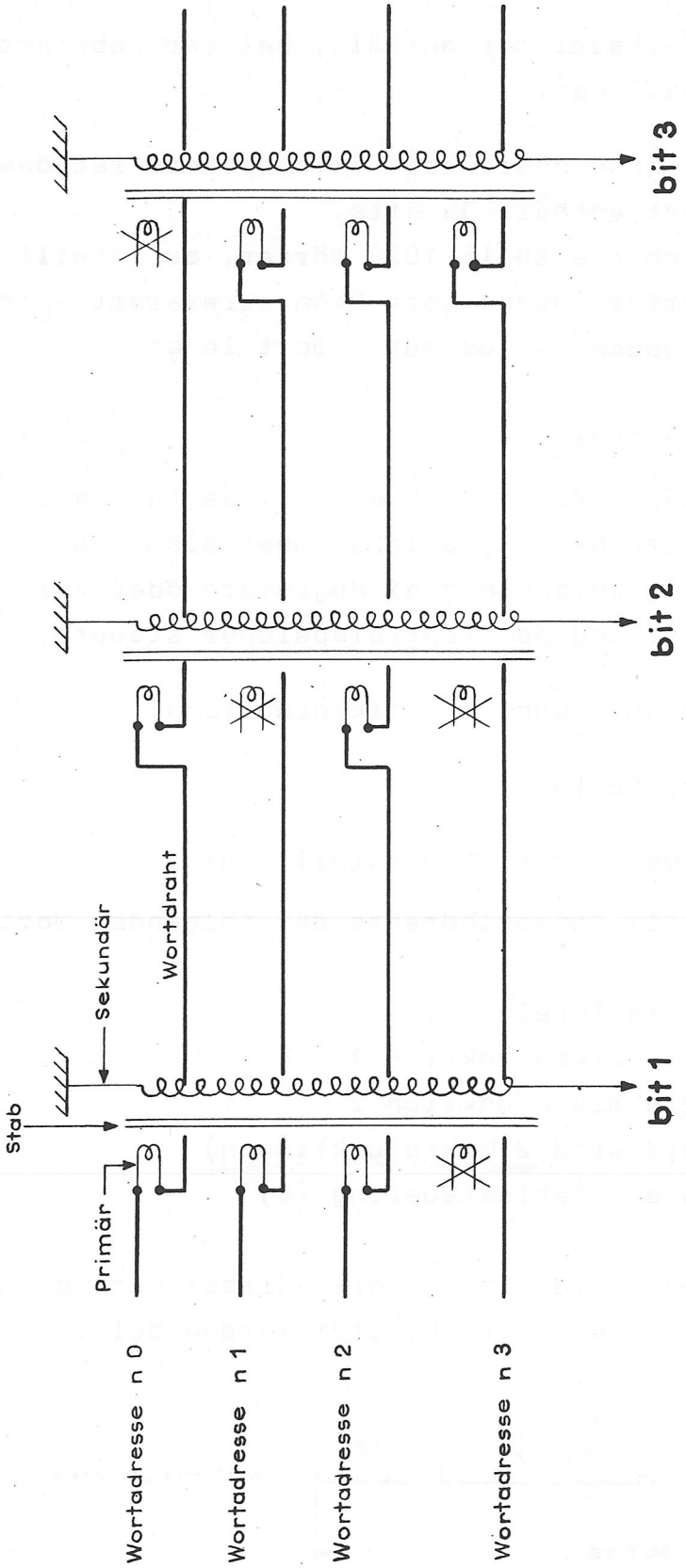
Das Wort: (s.hierzu Seite 87)

Ein Wort besteht aus 36 Bits, aufgeteilt in:

- 10 Bits für die Auswahladresse des folgenden Wortes. (A)
- 10 Bits für die Tafeln (T)
- 7 Bits für die Mikrofunktion 1
- 7 Bits für die Mikrofunktion 2
  (In jedem Wort sind 2 Mikrofunktionen)
- 2 Bits für die Befehlssteuerung (E)

Die Befehlssteuerung zeigt an, ob die Adresse der folgenden Mikrofunktion oder eine andere benutzt werden soll.

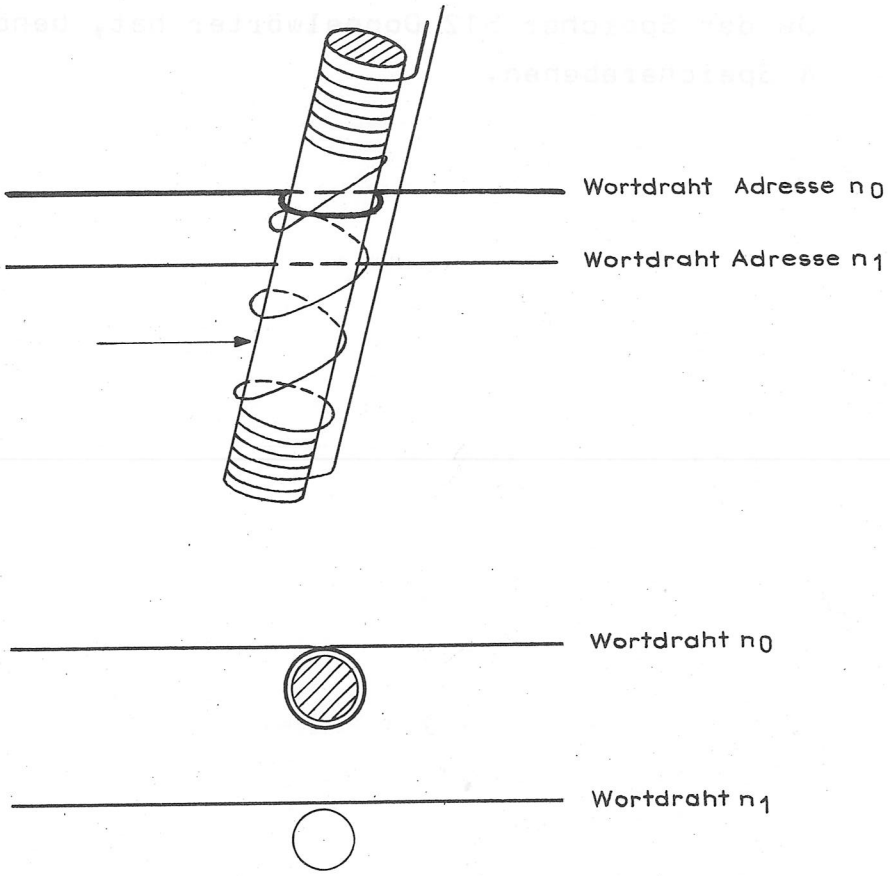




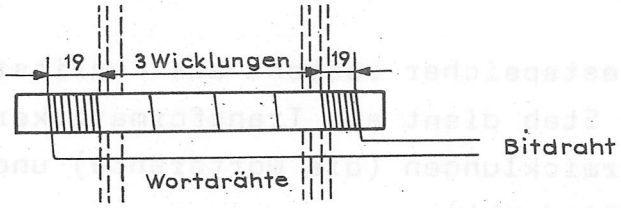
Wortadresse n 0	1	1	0
Wortadresse n 1	1	0	1
Wortadresse n 2	1	1	1
Wortadresse n 3	0	0	1

Aufbau des Festspeichers

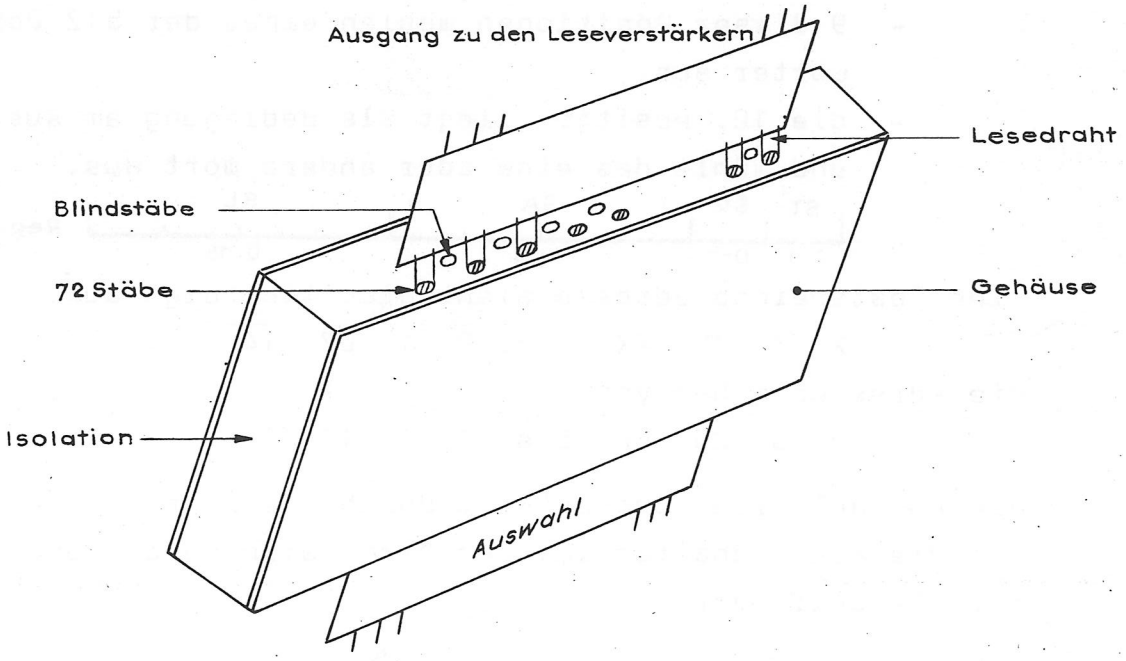
Der Festspeicher besteht aus Ferritstäben.
Jeder Stab dient als Transformatorkern, mit mehreren Primärwicklungen (die Wortdrähte) und einer Sekundärwicklung (dem Bitdraht).
Der Wortdraht liegt mit einer Windung um die Stäbe, deren Bit bei dieser Adresse auf 1 stehen soll.
Soll das Bit, also die Sekundärseite eines Stabes, auf 0 stehen, so geht der Wortdraht an diesem Stab vorbei. (s. Abb.)

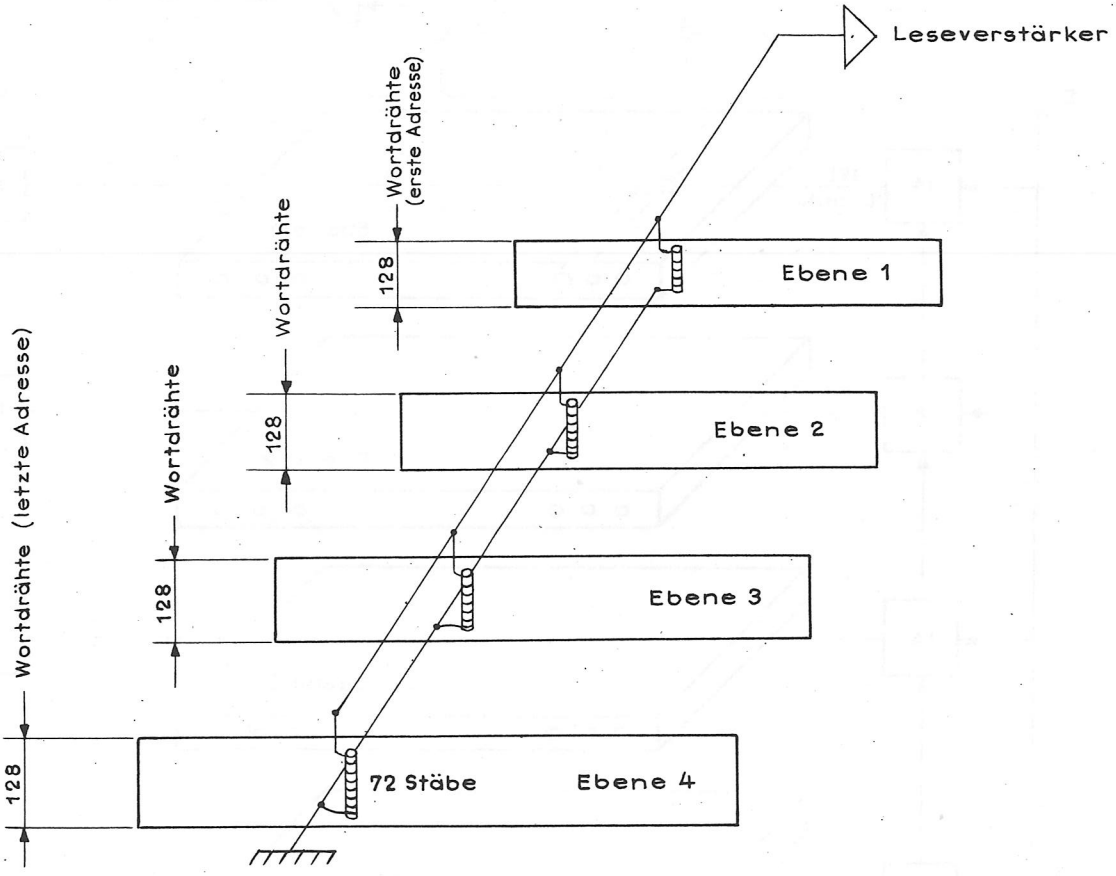


Die Wicklung des Bitdrahtes ist so ausgeführt, daß durch den auftretenden Sekundärimpuls kein anderer Wortdraht beeinflußt wird.



Ein Wortdraht durchläuft 2 Worte à 36 Bits, also 72 Stäbe. Diese 72 Stäbe stellen eine Speicherebene dar. Zwischen diesen 72 aktiven Stäben liegen 71 Blindstäbe, damit sich diese nicht gegenseitig stören.
Auf einer Speicherebene liegen 128 Wortdrähte.
Da der Speicher 512 Doppelwörter hat, benötigen wir 4 Speicherebenen.

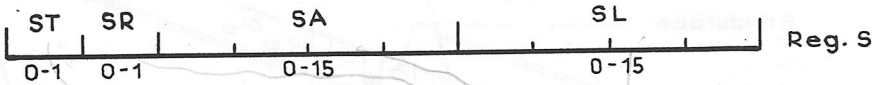

Ausgang zu den Leseverstärkern
Lesedraht
Blindstäbe
72 Stäbe
Gehäuse
Isolation
Auswahl


Leseverstärker
Wortdrähte (erste Adresse)
Wortdrähte
Wortdrähte
Wortdrähte (letzte Adresse)
128
128
128
128
Ebene 1
Ebene 2
Ebene 3
Ebene 4
72 Stäbe

Auswahl

Die Speicherauswahl geht über das Register S, das 10 Binärpositionen hat.

- 9 dieser Positionen wählen eines der 512 Doppelwörter aus.
- die 10. Position liegt als Bedingung am Ausgang und wählt das eine oder andere Wort aus.



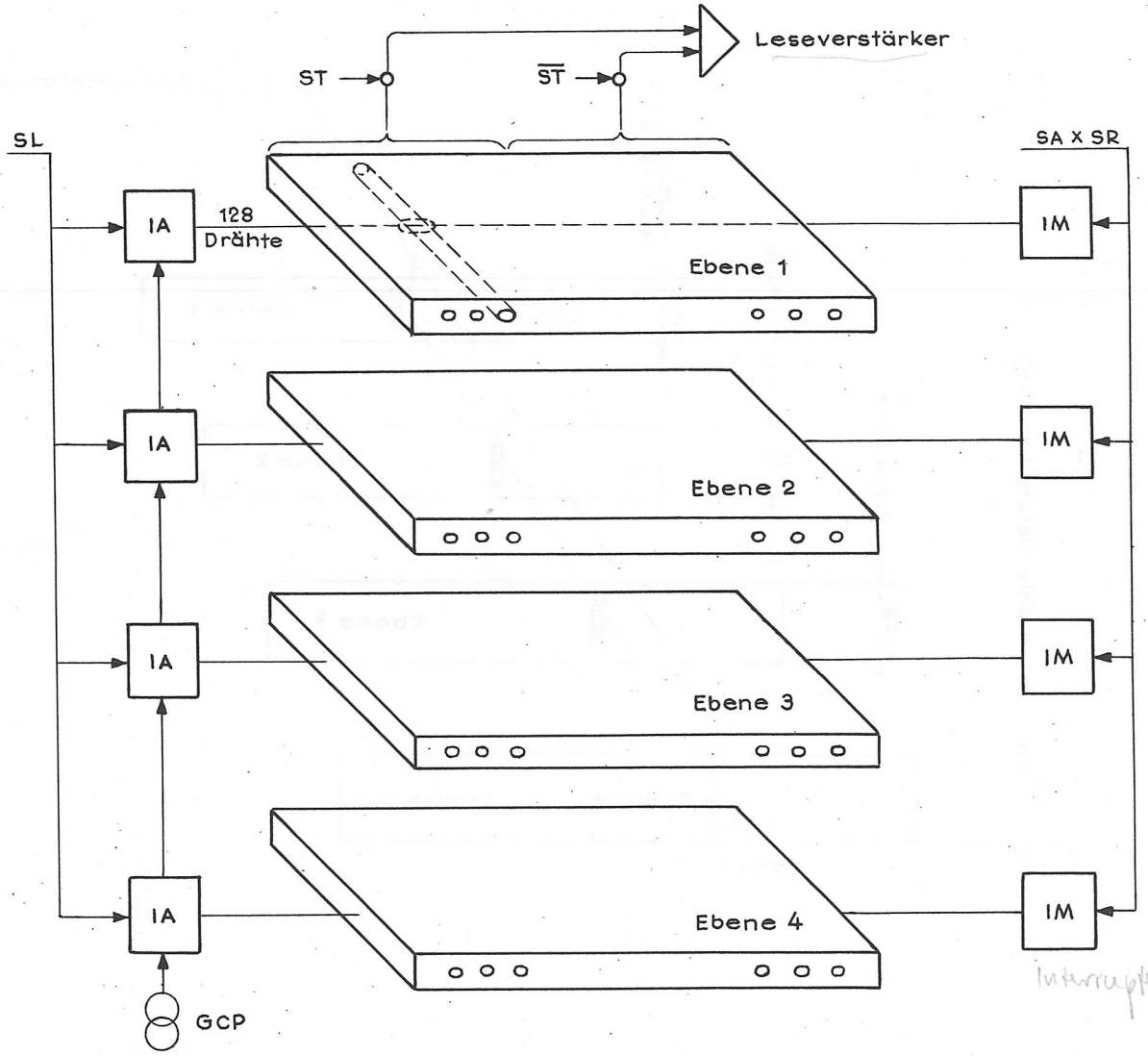
Eine Festspeicheradresse sieht also wie folgt aus:

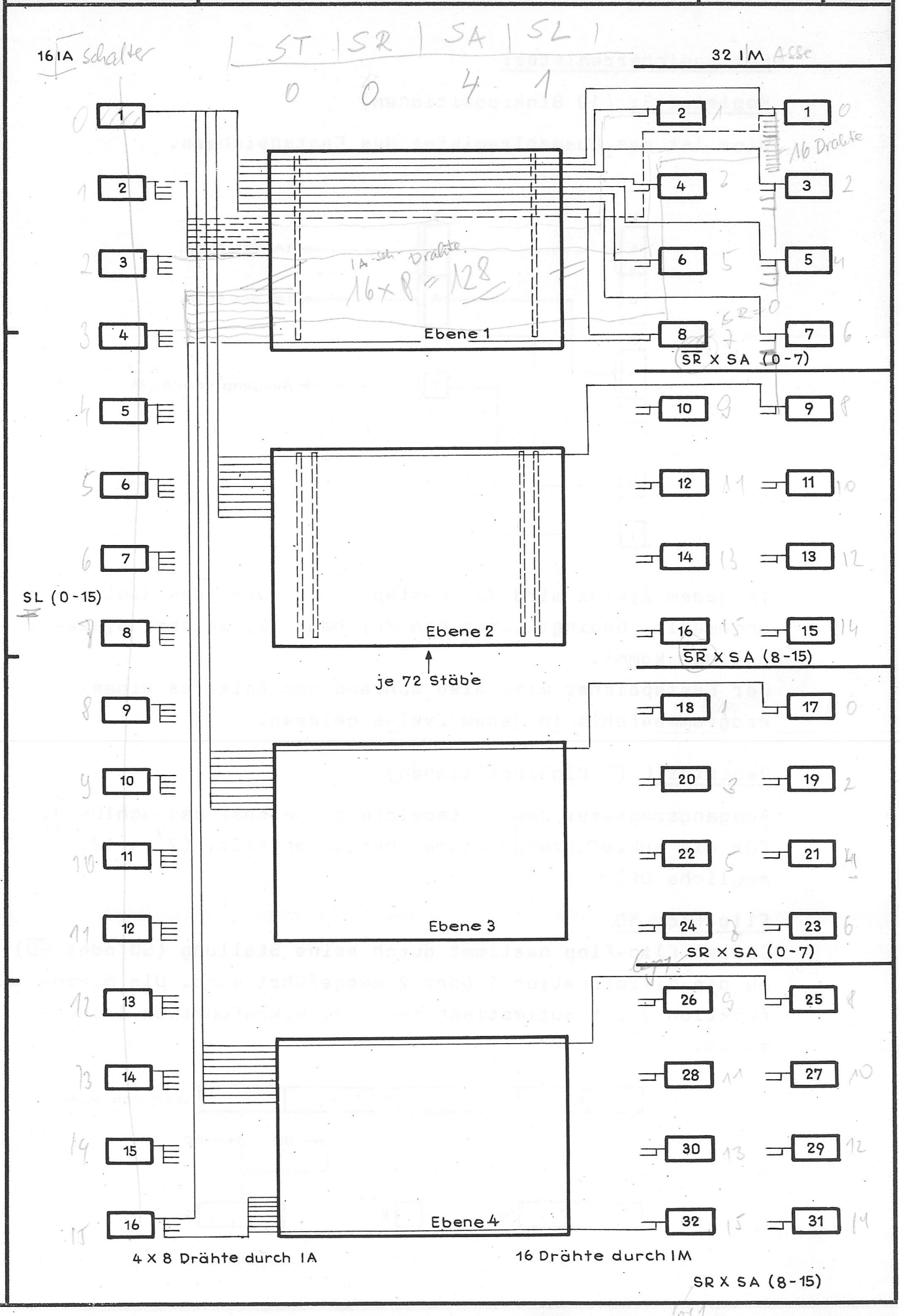
X X XX XX z.B. 0 1 07 14

Die Adressen gehen von:

0 0 00 00 bis 1 1 15 15

Die Auswahl wird durchgeführt durch die Luftschalter IA und die Masseschalter IM. Der Strom wird durch den Generator GCP geliefert.

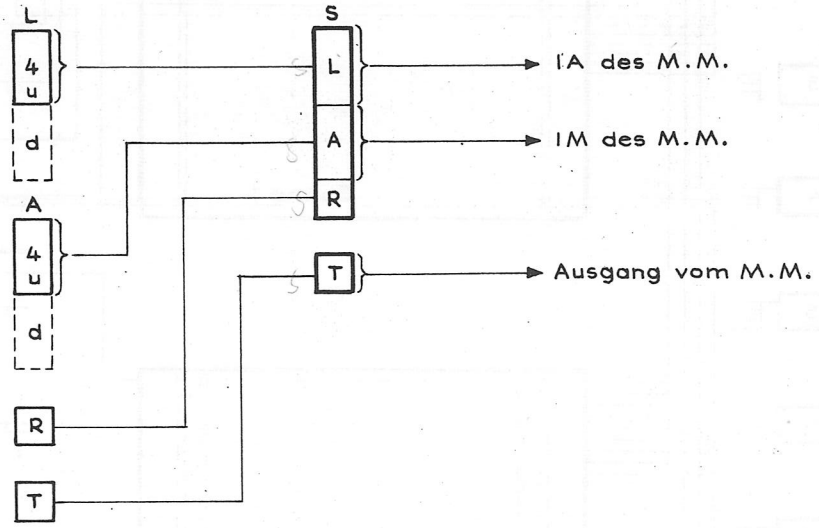



16 IA Schalter
ST SR SA SL
0 0 4 1
32 IM Asse
Ebene 1
Ebene 2
Ebene 3
Ebene 4
16 Drähte
SR X SA (0-7)
SR X SA (8-15)
SR X SA (0-7)
SR X SA (8-15)
SL (0-15)
je 72 Stäbe
4 X 8 Drähte durch IA
16 Drähte durch IM

Festspeicherregister:

Register S: (10 Binärpositionen)

Dies ist das Auswahlregister des Festspeichers.



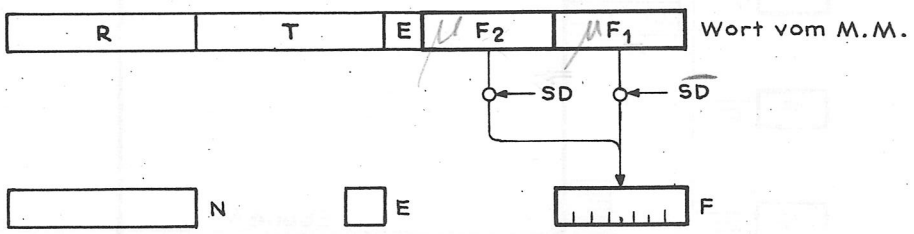
In jedem Zyklus wird der Festspeicher durch das Register S angewählt, bedingt durch den Rhythmus T2, welcher systematisch kommt.
Der Festspeicher wird also während des Ablaufes eines Programmbefehls in jedem Zyklus gelesen.

Register F (7 Binärpositionen)

Ausgangsregister des Festspeichers, welches den Schlüssel für die auszuführende Mikrofunktion enthält. (2^7 = 128 mögliche uF)

Flip-Flop SD

Dieses Flip-Flop bestimmt durch seine Stellung (SD oder $\overline{SD}$) ob die Mikrofunktion 1 oder 2 ausgeführt wird. Die Mikrofunktion 2 ist automatisch nach der Mikrofunktion 1 angerufen.

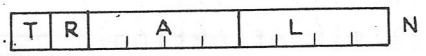


Flip-Flop ND

Dieses Flip-Flop hat den gleichen Zweck wie SD.

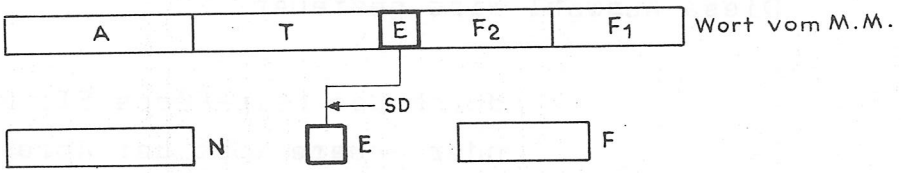
Register N (10 Binärpositionen)

Ausgangsregister des Festspeichers, welches die folgende Adresse oder ein Ergebnis der Tafeln enthält.



Register E (2 Binärpositionen)

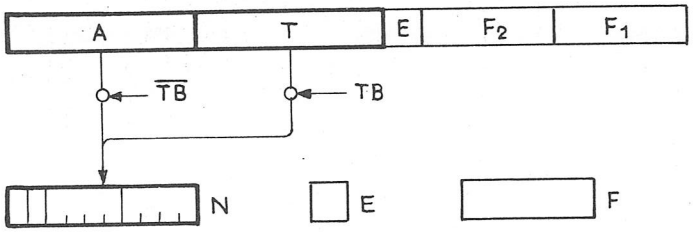
Dieses Register empfängt die 2 Stellen Programmsteuerung des Festspeicherwortes.
Es gibt 4 verschiedene Codes, von 0 - 3 (s.Kap.Programmst.)



Der Programmsteuercode wird erst in Mikrofunktion 2 übertragen.

Flip-Flop TB

Dieses Flip-Flop steuert die Eingabe in das Register N. (Adresse oder Ergebnis von Tafeln)



Flip-Flop B

Dieses Flip-Flop wird gesetzt, wenn mit einer Randeinheit der Klasse II gearbeitet wird. (Trommel, Band usw.)

Lesung

Leseverstärker:

Diese Leseverstärker geben in jedem Festspeicherzyklus nur ein Wort ab und in diesem Wort:

- ein Ergebnis der Tafeln oder die nächste Adresse
- die Mikrofunktion 1 oder 2

Es gibt insgesamt 19 Leseverstärker

N.E.F

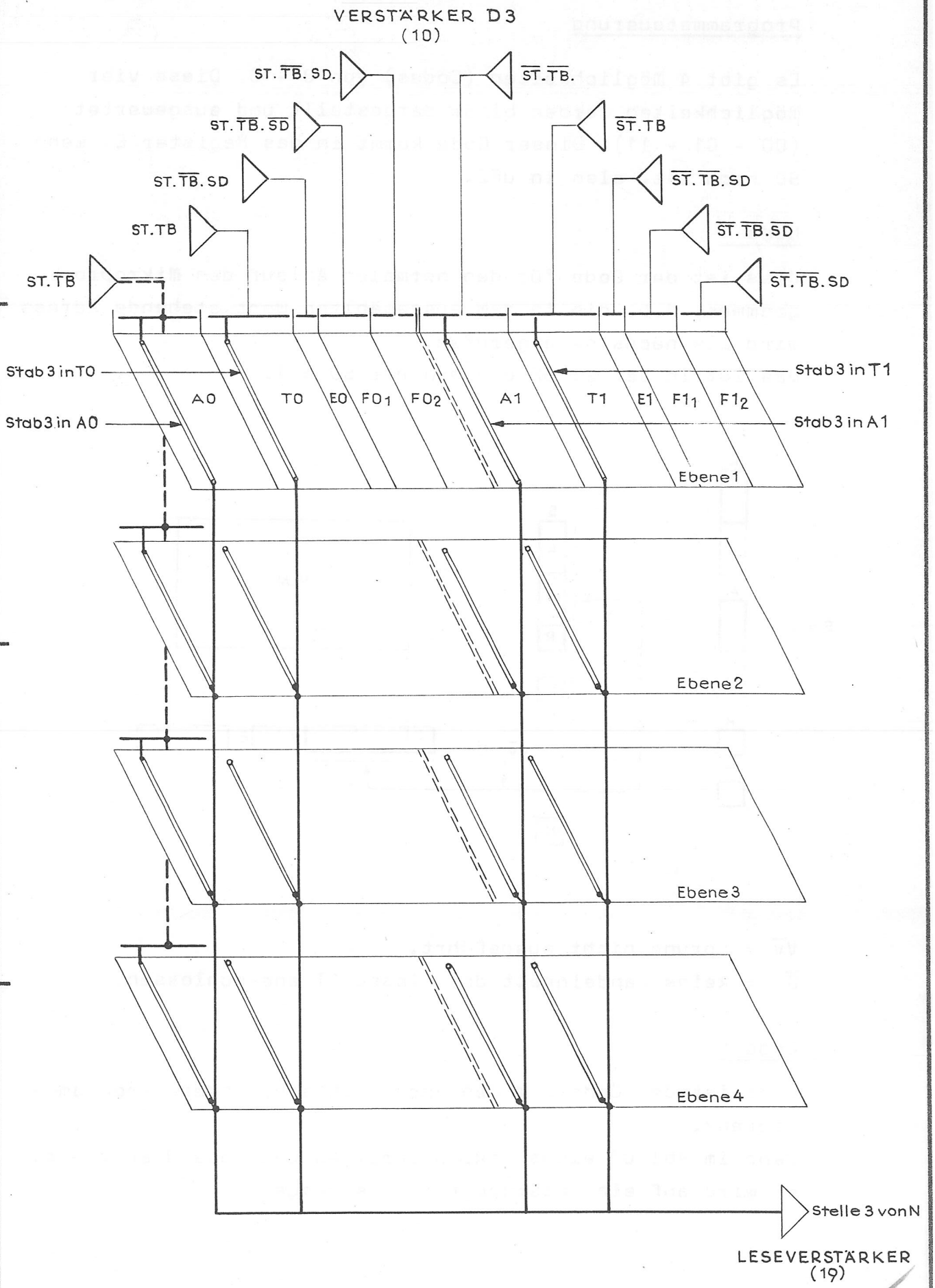
Leseauswahl:

Diese Auswahl wird gesteuert:

- durch das Flip-Flops ST, welches das eine oder andere Wort der Adresse freigibt.
- durch die Flip-Flops TB und SD, welche wählen zwischen uF1, uF2 oder Tafeln.

Diese Flip-Flops steuern 10 Verstärker D3, welche die Lesung auf verschiedene Bitdrähte festlegen.

~~Es gibt außerdem noch 2 Verstärker D3 spezial.~~

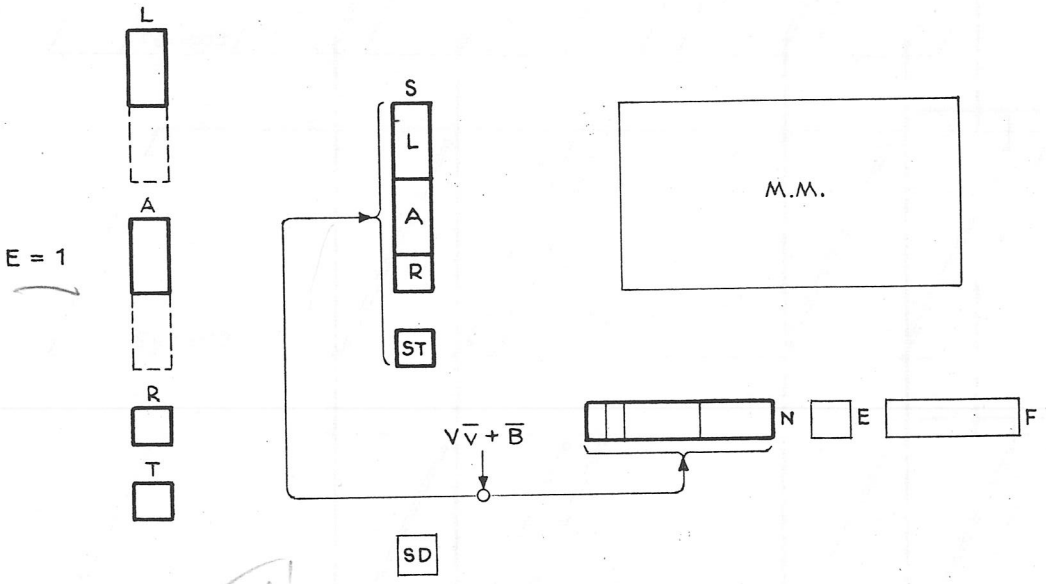

VERSTÄRKER D3
(10)
ST.$\overline{TB}$.SD.
$\overline{ST}$.$\overline{TB}$.
ST.$\overline{TB}$.$\overline{SD}$
$\overline{ST}$.TB
ST.$\overline{TB}$.SD
$\overline{ST}$.$\overline{TB}$.SD
ST.TB
$\overline{ST}$.$\overline{TB}$.$\overline{SD}$
ST.$\overline{TB}$
$\overline{ST}$.$\overline{TB}$.SD
Stab 3 in T0
Stab 3 in T1
A0
T0
E0
F0$_1$
F0$_2$
A1
T1
E1
F1$_1$
F1$_2$
Stab 3 in A0
Stab 3 in A1
Ebene 1
Ebene 2
Ebene 3
Ebene 4
Stelle 3 von N
LESEVERSTÄRKER
(19)

Programmsteuerung

Es gibt 4 Möglichkeiten (Codes) von 0 - 3. Diese vier Möglichkeiten werden binär dargestellt und ausgewertet (00 - 01 - 11). Dieser Code kommt in das Register E, wenn SD hoch ist, also in uF2.

Code 1

Dies ist der Code für den normalen Ablauf des Mikroprogrammes, d.h. die in dem ausgewählten Wort stehende Adresse, wird als nächstes angerufen.
Das ist in der 2. uF und man hat SD = 1.



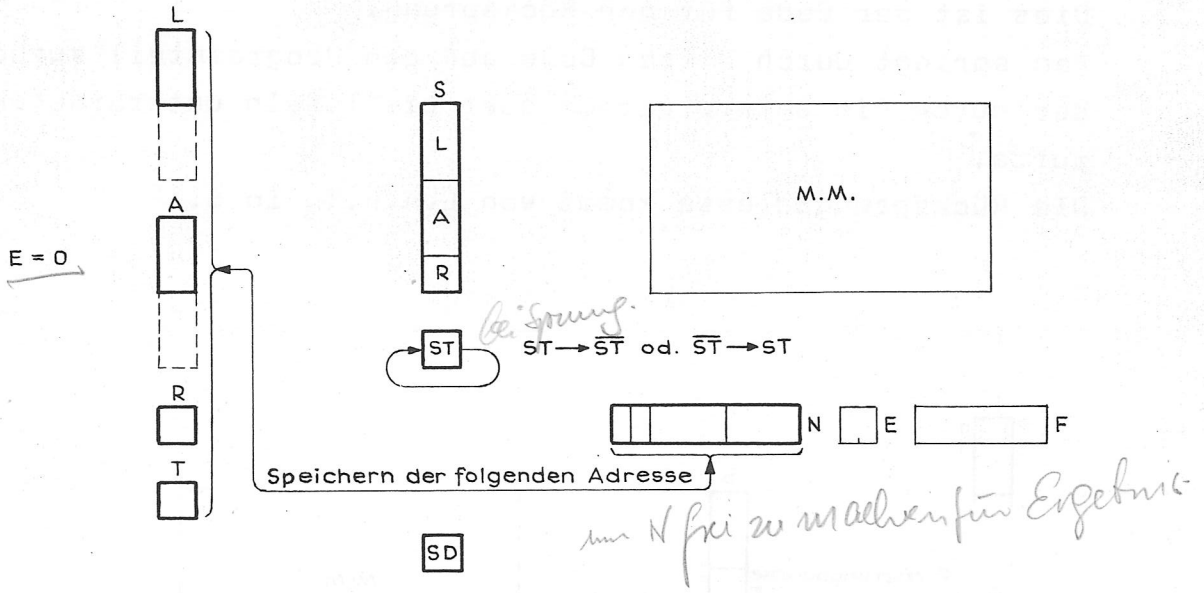
Variante validé

$V\overline{V}$ = Sprung nicht ausgeführt.
$\overline{B}$ = keine Randeinheit der Klasse II angeschlossen.

Code 0

Dies ist der Code für den Sprung in eine andere Programmsequenz.
Wenn im Ablauf eines Mikroprogrammes der Code 0 auftritt, so wird auf ein Unterprogramm gesprungen.

Dieser Code entspricht dem Sprungbefehl im Hauptprogramm.
Folgende Übertragungen werden ausgeführt:

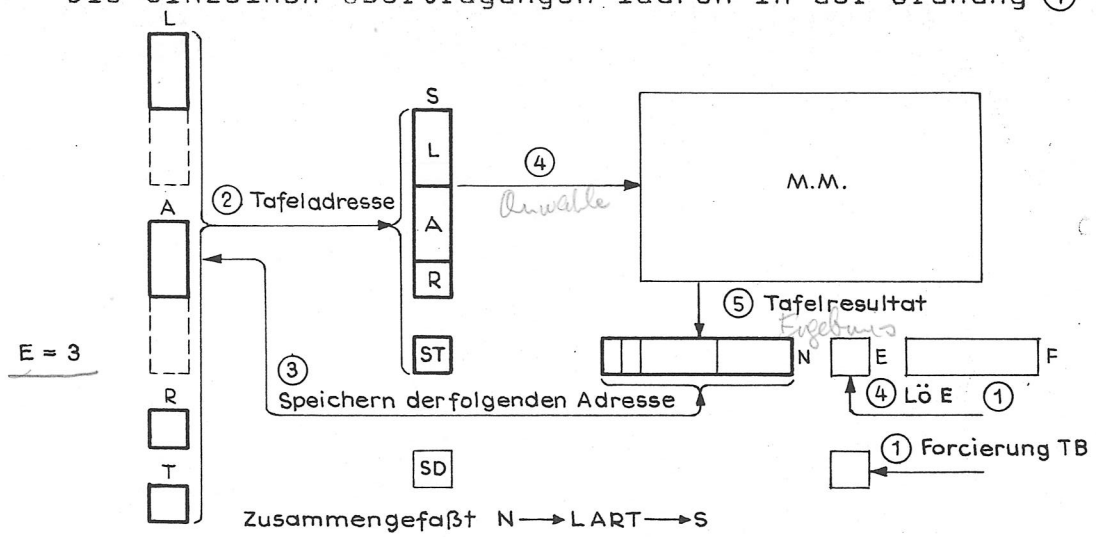


Der Sprung in die andere Sequenz wird ausgeführt, indem man ST umkehrt. Die, durch die Umkehrung von ST entstandene Adresse nennt man die Verbindungsadresse.

Z.B. aus 0 0 07 12 wird 1 0 07 12
oder aus 1 0 07 12 wird 0 0 07 12

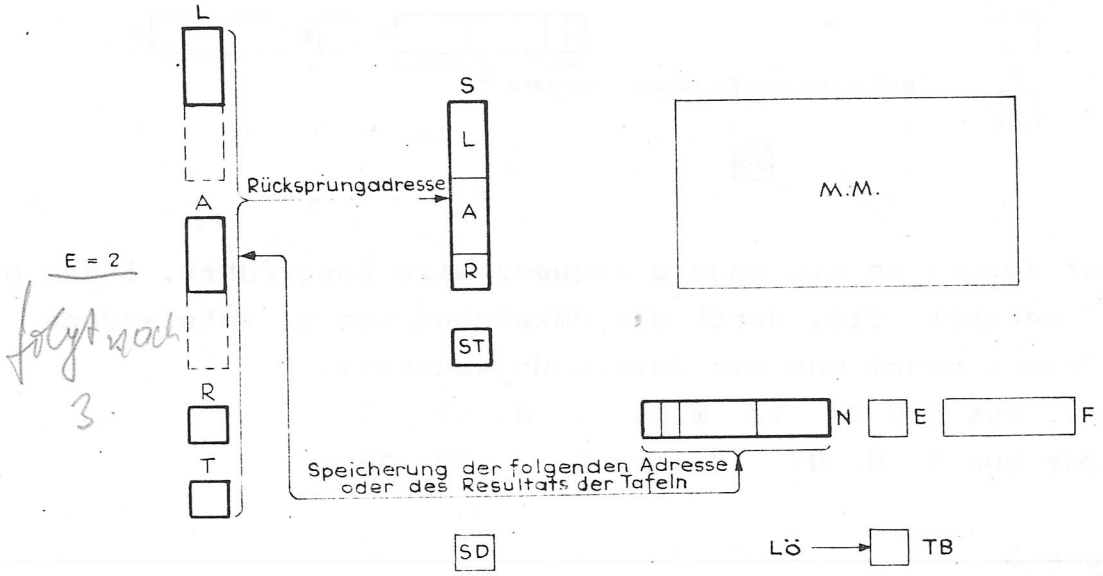
Code 3

Dieser Code gibt für den nächsten Zyklus die Rechen- und Übersetzungstafeln des Festspeichers frei.
Die Adressierung der Tafelzone ist durch L.A.R.T. gegeben. (s. Kapitel Tafeln)
Die Werte (z.B. einer Addition) stehen in L.A.R.T.
Die einzelnen Übertragungen laufen in der Ordnung ① ② ...ab.



Code 2

Dies ist der Code für den Rücksprung.
Man springt durch diesen Code auf den Programmteil zurück, der durch ein Unterprogramm oder die Tafeln unterbrochen wurde.
Die Rücksprungadresse kommt von L.A.R.T. in S.



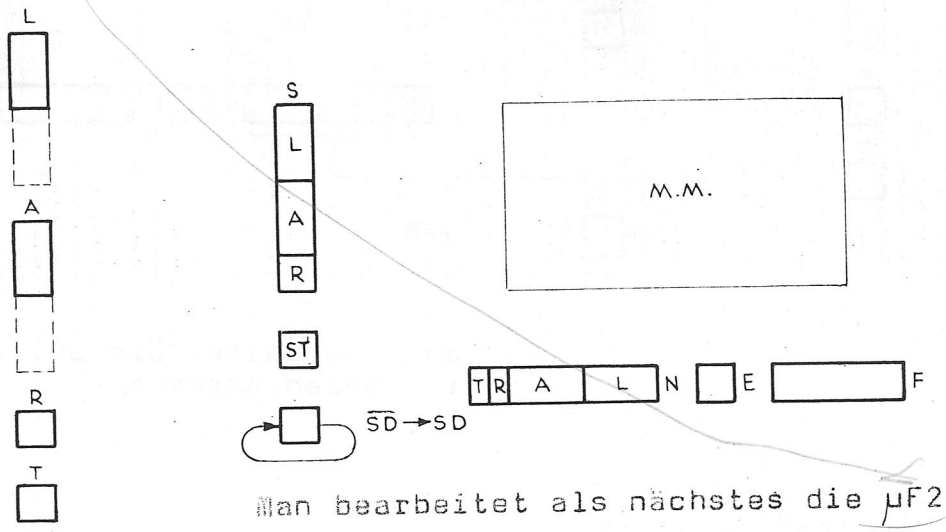
Sprünge

Vv = Sprung ist ausgeführt.

$V\overline{v}$ = Sprung ist nicht ausgeführt.

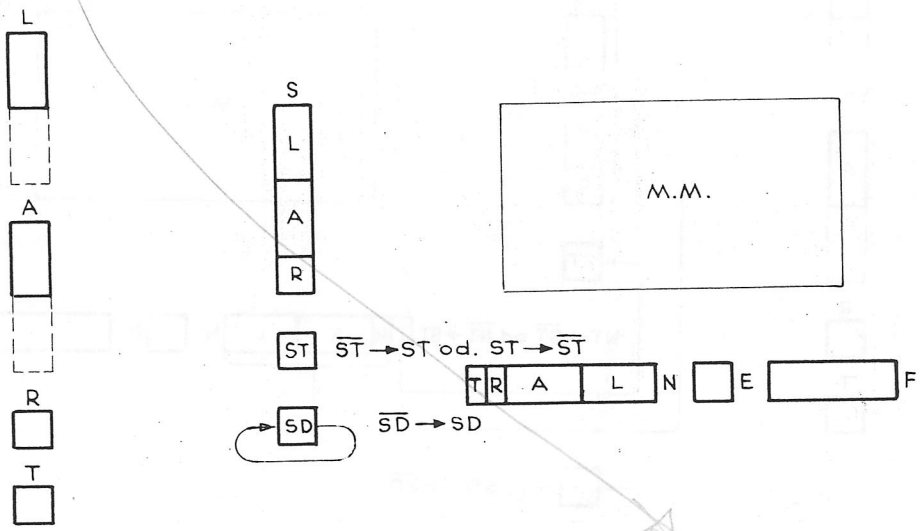
Wenn man einen Sprung in µF2 hat, ist immer Enchaînement 1.

$V\overline{v}$ in µF1



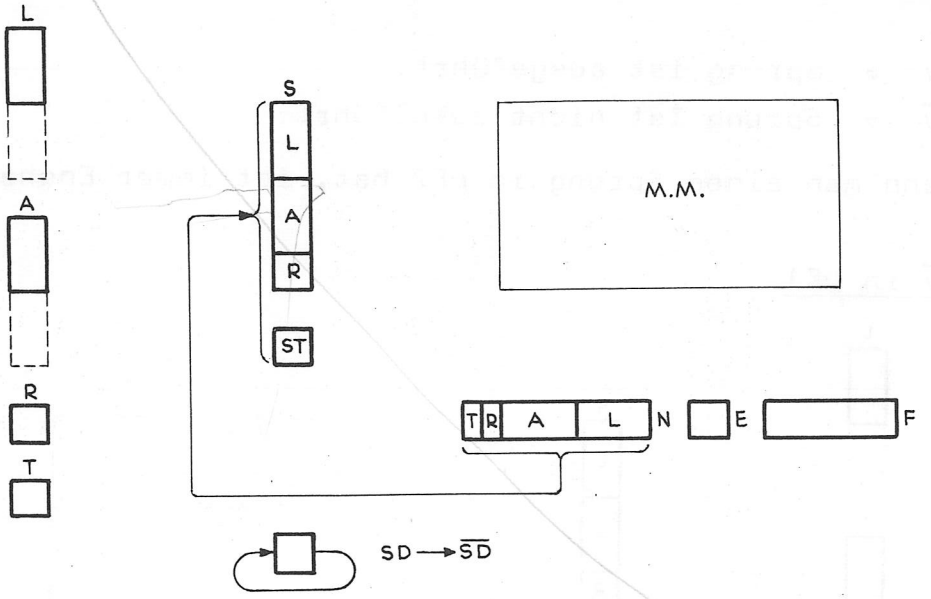
Man bearbeitet als nächstes die µF2 dieser Adresse.

Vv in µF1



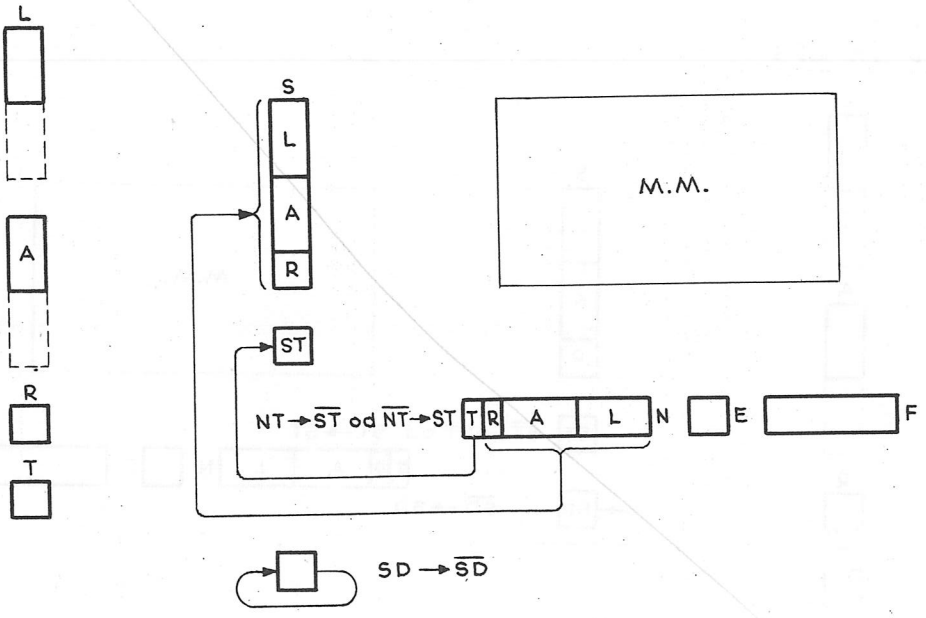
Man bearbeitet die µF2 der Sprungadresse.

$V\overline{v}$ in µF2



Man bearbeitet die µF1 der folgenden Adresse.

Vv in µF2



Man bearbeitet die µF1 der Sprungadresse

Allgemeines

Jedes Wort des Festspeichers enthält 36 Bits, von denen 10 für die Tafeln reserviert sind.
Zur Benutzung der Tafeln muß man in Mikrofunktion 2 und Programmsteuercode 3 sein. Der Code 3 setzt das Flip-Flop TB, welches die Tafeln freigibt. Die Auswahladresse für die Tafeln steht in LART. Das Resultat steht im Register N, während Programmsteuercode 2.
Das Flip-Flop SD wird bis zu diesem Zeitpunkt gehalten.

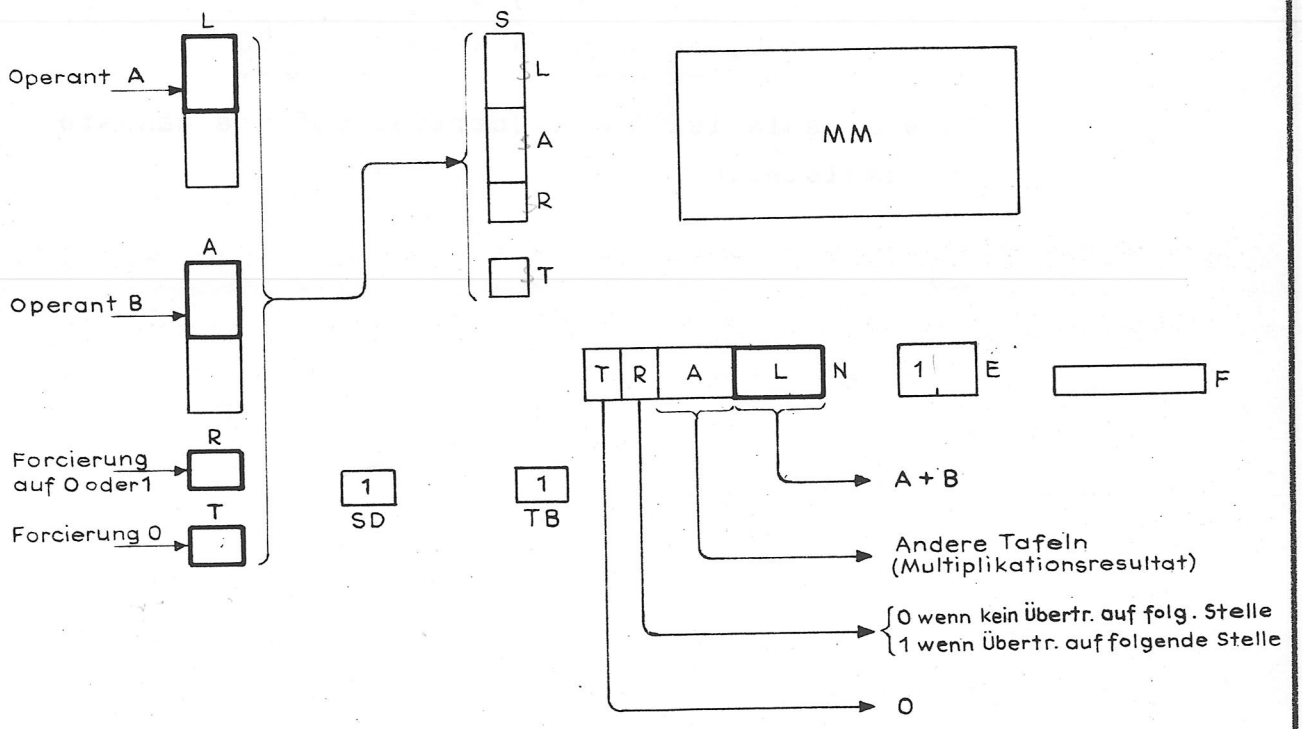
Rechentafeln

Addition:

Es gibt 2 Additionstafeln, die beide mit ST = 0 adressiert sind.

SR = 0 - Additionstafel ohne Übertrag auf die folgende Dezimalstelle.

SR = 1 - Additionstafel mit Übertrag auf die folgende Dezimalstelle.



Tafelauszug

ST	SR
0	0

SA → 06 ohne Übertrag 07 08

SL =												
0	0	0	00	06	0	0	00	07	0	0	00	08
1	0	0	06	07	0	0	07	08	0	0	08	09
2	0	0	02	08	0	0	04	09	0	1	06	00

T R A L N T R A L T R A L N

ST	SR
0	1

SA → 06 07 08

SL =												
0	0	0	00	07	0	0	00	08	0	1	00	09
1	0	0	00	08	0	0	00	09	0	1	00	00
2	0	0	01	09	0	1	01	00	0	1	01	01

T R A L

z.B.: 2 + 8; die Tafel wird ausgewählt durch

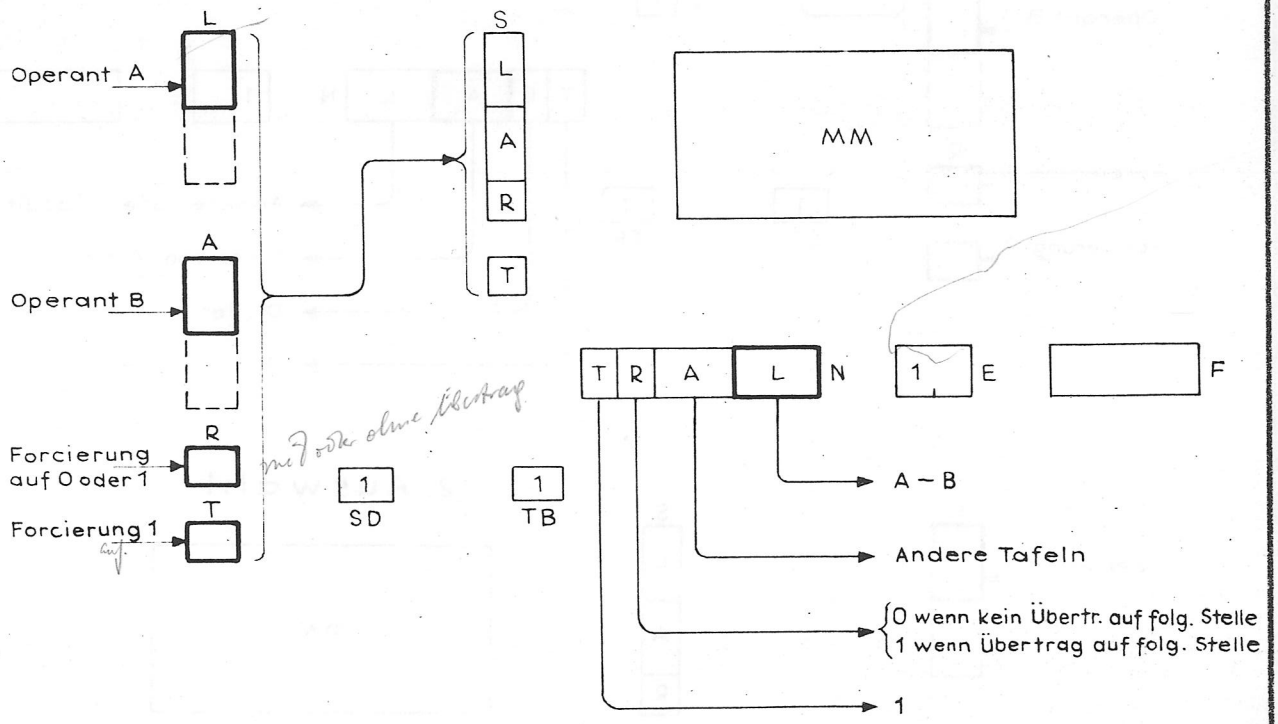
SL = 2, SA = 8, SR = 0, ST = 0.

NT = 0, NR = 1, NA = 06, NL = 00

Das Ergenis ist 0 mit Übertrag auf die nächste Dezimalstelle = 10.

Subtraktionstafeln

Beide Tafeln sind mit ST = 1 adressiert. Ansonsten das gleiche Prinzip wie die Additionstafeln.

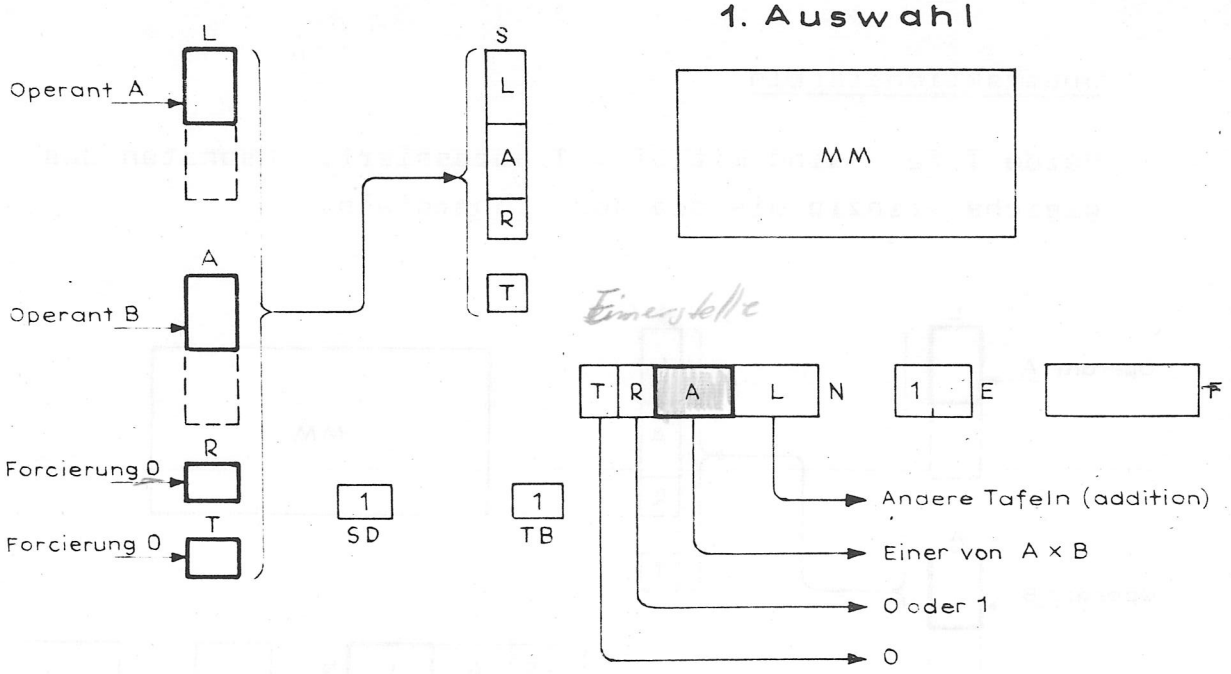


Multiplikationstafeln

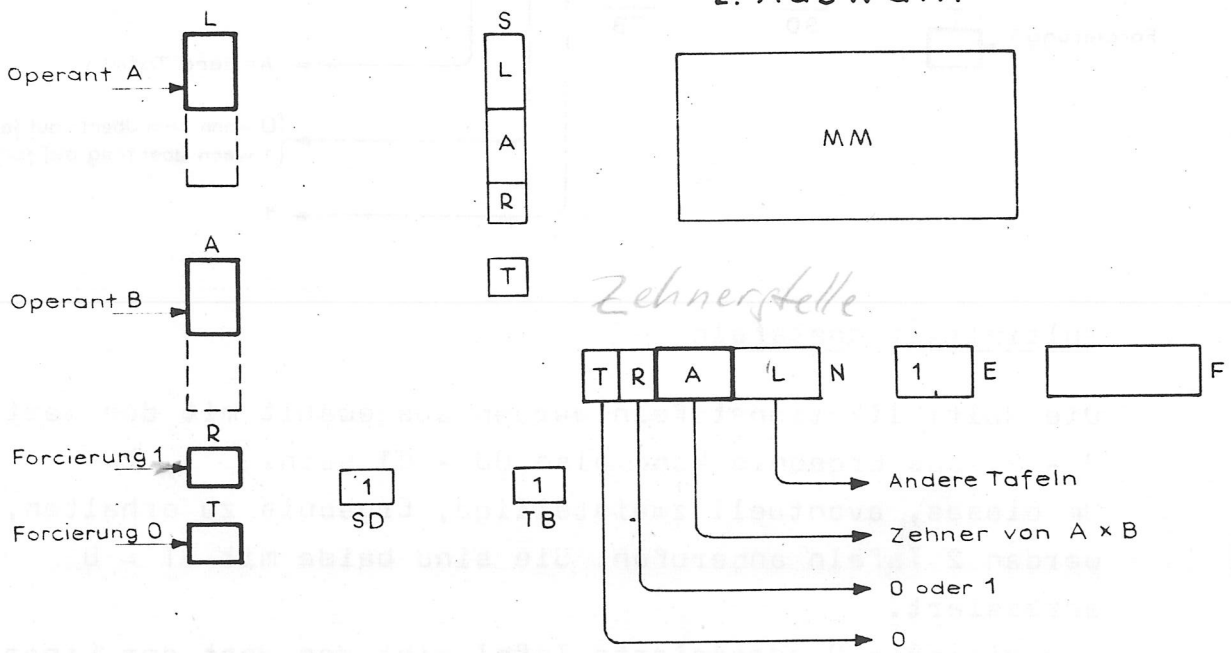
Die Multiplikationstafeln werden ausgewählt mit den Werten 0 - 9. Das Ergebnis kann also 00 - 81 sein.
Um dieses, eventuell zweistellige, Ergebnis zu erhalten, werden 2 Tafeln angerufen. Sie sind beide mit ST = 0 adressiert.
Die mit SR = 0 adressierte Tafel gibt den Wert der Einerstelle des Produktes an.
Die mit SR = 1 adressierte Tafel gibt den Wert der Zehnerstelle des Produktes an.

Grundsätzlich werden beide Tafeln bei Multiplikation gebraucht.

1. Auswahl



2. Auswahl



Beisp.: Siehe Auszug der Additionstafel (S.112) 7 · 2

1. Auswahl: SL = 2, SA = 7, SR = 0, ST = 0
   dies ergibt: NT = 0, NR = 0, NA = 04, NL = 09
   Das Resultat der Einerstelle ist 4.

2. Auswahl: SL = 2, SA = 7, SR = 1, ST = 0
   dies ergibt: NT = 0, NR = 1, NA = 01, NL = 00
   Das Resultat der Zehnerstelle ist 1.

Tafeln für Verdichten und Entdichten

Wiederholung:

Format 1 : 1 Zeichen in Code ISO pro Byte. (~~Entverdichten~~ verdichten)

Format 2 : 2 Zeichen in Hexadezimal pro Byte.

Format 1 ⟶ Format 2 = Verdichten

Format 2 ⟶ Format 1 = Entdichten.

Dieser Formatwechsel kann nur mit den ersten 6 Buchstaben, also A, B, C, D, E und F oder den Zahlen 0 - 9 gemacht werden.

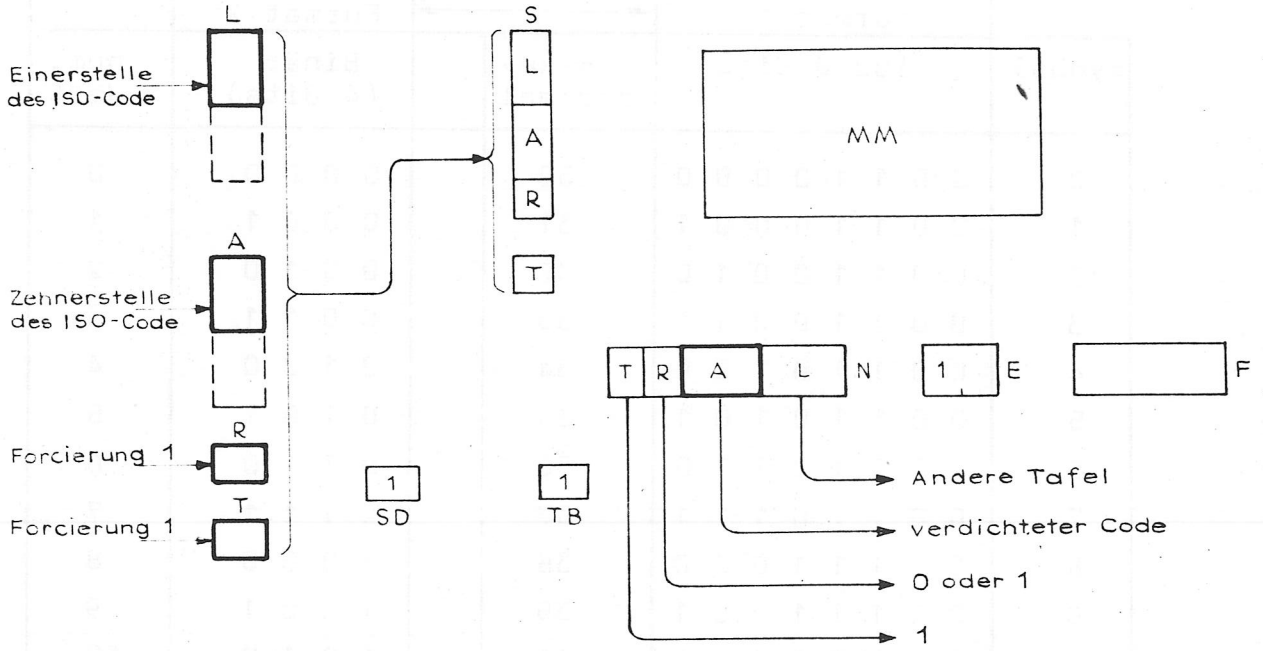
	Format 1	⟷	Format 2	
Symbol	ISO 8 Bits	Hexa-dezimal	Binär (4 Bits)	num. Wert
0	0 0 1 1 0 0 0 0	30	0 0 0 0	0
1	0 0 1 1 0 0 0 1	31	0 0 0 1	1
2	0 0 1 1 0 0 1 0	32	0 0 1 0	2
3	0 0 1 1 0 0 1 1	33	0 0 1 1	3
4	0 0 1 1 0 1 0 0	34	0 1 0 0	4
5	0 0 1 1 0 1 0 1	35	0 1 0 1	5
6	0 0 1 1 0 1 1 0	36	0 1 1 0	6
7	0 0 1 1 0 1 1 1	37	0 1 1 1	7
8	0 0 1 1 1 0 0 0	38	1 0 0 0	8
9	0 0 1 1 1 0 0 1	39	1 0 0 1	9
A	0 1 0 0 0 0 0 1	41	1 0 1 0	10
B	0 1 0 0 0 0 1 0	42	1 0 1 1	11
C	0 1 0 0 0 0 1 1	43	1 1 0 0	12
D	0 1 0 0 0 1 0 0	44	1 1 0 1	13
E	0 1 0 0 0 1 0 1	45	1 1 1 0	14
F	0 1 0 0 0 1 1 0	46	1 1 1 1	15

Man kann auch den Code für Spaltensprung verdichten. Das ergibt 20 in Hexadezimal und 0 in Binär.

Verdichtungstafel

Sie wandelt die internen Codes (8-Bit-Code) für die Buchstaben A - F und die Ziffern 0 - 9 um, in Hexadezimal.

Diese Tafel ist adressiert mit ST = 1 und SR = 1.
Die Auswahl erfolgt mit SL und SA.
SL = enthält die Einerstelle des ISO-Code (0 - 9)
SA = enthält die Zehnerstelle des ISO-Code (2 - 4) 120 - 40
Das verdichtete Resultat befindet sich in NA.



Tafelauszug

SA	SR
1	1

SA	03				04				05			
SL = 0	1	1	00	06	1	1	08	05	1	1	00	04
1	1	1	01	07	1	1	10	06	1	1	01	05
2	1	1	02	08	1	1	11	07	1	1	02	06

Beispiel: Verdichten von A

A in ISO = 0100 0001 also 41 in Hexadezimal.

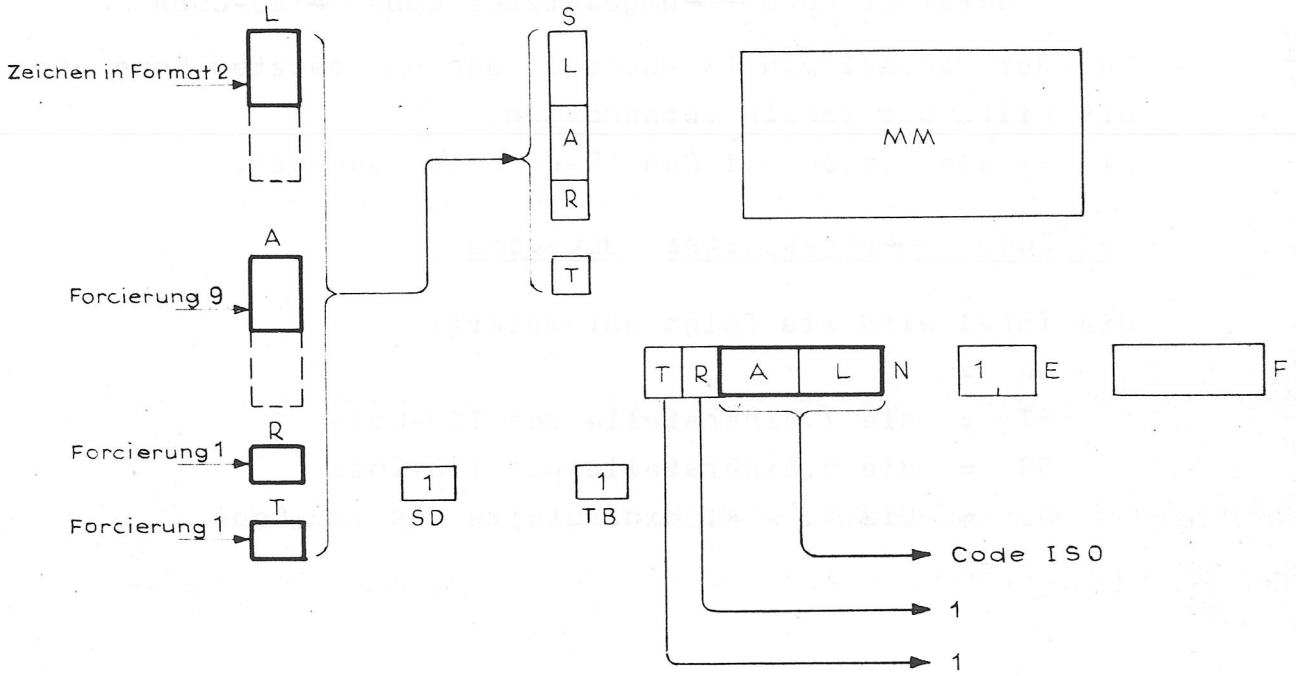
1. 4 ⟶ A
2. 1 ⟶ L
3. Forcierung von T = 1 und R = 1
4. LART ⟶ S

Als Ergebnis erhalten wir aus der Tabelle in Register NA =10.

Entdichtungstafel

Diese Tafel setzt die 4-Bit-Code 0 - 15 um, in den ISO-Code (7Bits).
Sie ist adressiert mit ST = 1, SR = 1 und SA = 9.
In SL stehen die Werte 0 - 15, welche den auszugebenden Code bestimmen.
Die Ziffer 0 ergibt immer 30, also niemals wieder Spaltensprung (20) wie im umgekehrten Fall.
Das Ergebnis steht in NA und NL.

2 Halbbyt = 1 Byte



Tafelauszug

ST	SR	SA	07	08	09
1	1	SL =0	11 08 02	11 15 01	11 03 00
		1	11 08 03	11 14 02	11 03 01
		2	11 08 04	11 13 03	11 03 02

Beispiel: Umwandeln einer 2 in ISO-Code.

1. 2 ⟶ L und Forcierung von A = 9, R = 1 und T = 1
2. LART ⟶ S

In NA und NL erhalten wir als Resultat 32.

Übersetzungstafeln

Wiederholung: Es wird übersetzt, ISO-Code in IBM- oder BULL-Code und umgekehrt.
Diese beiden Übersetzungsfälle werden in 2 Zügen vorgenommen. (s.Seite 16-18)

ISO-Code ⟶ umgesetzter Code ⟶ externer Code
externer Code ⟶ umgesetzter Code ⟶ ISO-Code

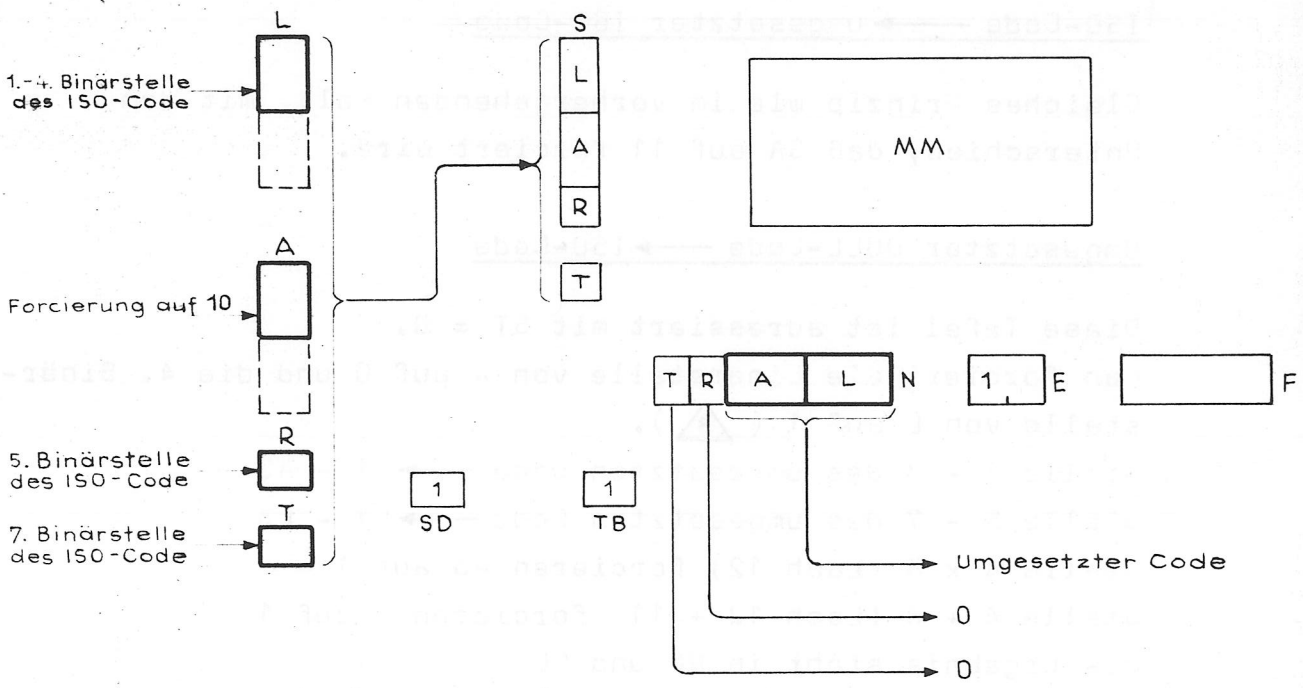
Nur der Wechsel von ISO-Code in den umgesetzten Code wird mit Hilfe der Tafeln vorgenommen.
Die Tafeln werden bei den TR-Befehlen benutzt.

Transcodage

ISO-Code ⟶ umgesetzter BULL-Code

Die Tafel wird wie folgt adressiert:

SA = 10
ST = die 7.Binärstelle des ISO-Code
SR = die 5.Binärstelle des ISO-Code
SL = die 1. - 4. Binärstelle des ISO-Code



Tafelauszug

ST	SR
1	0

SA		09				10				11			
SL	1	1	1	05	02	0	0	11	08	0	0	04	01
	2	1	1	06	03	0	0	03	01	0	0	04	02
	3	1	1	06	04	0	0	03	02	0	0	04	03

Beispiel: Übersetzen des Zeichen C.

C in ISO-Code = 43 = 0100 0011

Dies ergibt als Adresse:

ST = 1, SR = 0, SA = 10 und SL = 3

Das Ergebnis in NA und NL ist 32.

Dies ist der umgesetzte BULL-Code für C (s.S. 16)

ISO-Code ⟶ umgesetzter IBM-Code

Gleiches Prinzip wie im vorhergehenden Fall, mit dem Unterschied, daß SA auf 11 forciert wird.

Umgesetzter BULL-Code ⟶ ISO-Code

Diese Tafel ist adressiert mit ST = 0.
Man forciert die Einerstelle von A auf 0 und die 4. Binärstelle von L auf 1 (⚠8).
Stelle 1 - 3 des umgesetzten Code ⟶ A1 - A3
Stelle 5 - 7 des umgesetzten Code ⟶ L1 - L3
Stelle 4 x 8 (Loch 12) forcieren A3 auf 1
Stelle 4 + 8 (Loch 12 + 11) forcieren R auf 1
Das Ergebnis steht in NA und NL

Tafelauszug:

ST	SR
0	0

SL \ SA	04			05			06		
10	00	02	02	00	02	05	00	02	06
11	00	04	05	00	04	06	00	04	07
12	00	02	11	00	05	13	00	05	12
13	00	04	14	00	05	00	00	05	01

Beispiel: Im umgesetzten BULL-Code ergibt P = 55.
Die Tafel wird also ausgewählt mit:

ST = 0, SR = 0, SA = 5 und SL = 13

Das Ergebnis in NA und NL ergibt 50.
Dies ist der hexadezimale Wert für P im ISO-Code.

Umgesetzter IBM-Code ⟶ ISO-Code

Gleiches Prinzip wie im vorhergehenden Fall, mit dem Unterschied, daß ST auf 1 forciert wird.

Tafel für das 15-Komplement

Diese Tafel ist adressiert mit ST = 1, SR = 1 und SA = 8.
In SL steht der Wert, für den das Komplement gesucht wird.
Das Ergebnis steht in NA. s. Logische Funktionen. 01,

Rechentafel für die Realadresse der Register

Wiederholung:
Die Register sind numeriert von 00 - 99. Man erhält die Realadresse, indem man die Registernummer mit 5 multipliziert und 100 zuaddiert. (5 · n) + 100.
Diese Tafel bestimmt nur die Zehnerstelle der Realadresse und ist adressiert mit ST = 1 und SR = 0.
SA enthält die Zehnerstelle der Registernummer.
SL enthält die Einerstelle der Registernummer.
Resultat im Register NA.

Tafelauszug

ST	SR
1	0

SL \ SA	0			1			2		
0	10	00	00	11	05	09	11	00	08
1	10	00	01	10	05	00	11	00	09
2	10	01	02	10	06	01	10	01	00

Beispiel: Realadresse von Register 21.

- Errechnen der Hunderterstelle durch Multiplikation:

  5 · 20 = 0100 + 100 = 0200

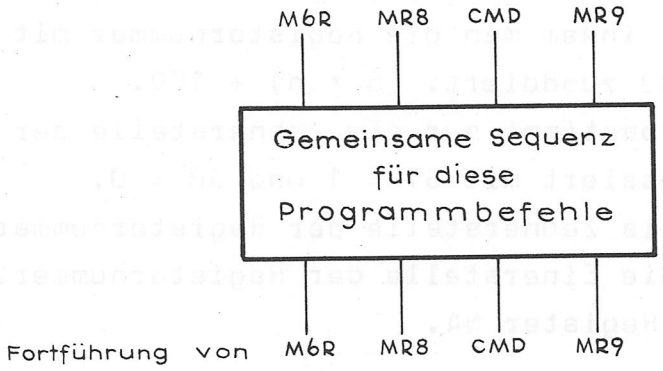
- Errechnen der Zehnerstelle durch die Tafel:
  ST = 1, SR = 0, SA = 2 und SL = 1 ergibt in NA = 00.

- Bestimmen der Einerstelle:
  wenn Registernummer gerade, Einerstelle = 0
  wenn Registernummer ungerade, Einerstelle = 5

Das ergibt für Register 21, die Realadresse 205.

Tafel für die Rücksprungadresse der Mikroprogrammsequenzen.

Der Festspeicher ist in Mikroprogrammsequenzen aufgeteilt. Jede dieser Sequenzen hat eine besondere Aufgabe (z.B. Vergleich, Addition, Subtraktion usw.)
Die Bearbeitung eines Programmbefehls, besteht aus mehreren Mikroprogrammsequenzen.
Diese Sequenz wird meistens von mehreren Befehlen benutzt. Z.B.



Das Problem ist, der Rücksprung in den Befehlsablauf, der durch diese Sequenz unterbrochen wurde.
Zu diesem Zweck wird während der Bearbeitung des Befehls, ein für diesen Befehl charakteristisches Zeichen, in die Speicherstelle 0000 übertragen.
Am Ende der Sequenz wird diese Speicherstelle gelesen und entsprechend ihrem Inhalt eine Adresse in L.A.R.T. eingegeben.
Durch diese Adresse wird die Tafel ausgewählt, welche die Rücksprungadresse auf die nächste Mikrofunktion des unterbrochenen Befehls ausgibt.
Während des Ablaufes einer Sequenz kann man auf eine Untersequenz springen. Die Rücksprungadresse wird in diesem Fall, in Form einer Ziffer, in Register G gespeichert.
Die Adressierung ist immer mit SA = 15 (ST, SR und SL sind veränderlich). Die uF2, die einem Programmsteuercode 3 folgt, ist immer eine NULL. sequenz. zum Rücksprung Adresse v. LART → S
= uF mdts. Inhalt Reg ≠ 0
Es gibt außerdem noch Tafeln für die Umsetzung des Anschlußcode der Randeinheiten.

Verbindung zwischen Zentraleinheit und Randeinheit

Die Randeinheiten werden über Kanäle (Zeichenkanäle) an die Zentraleinheit angeschlossen.
Man unterscheidet zwei verschiedene Typen:

1. Die langsamen Kanäle (Kanäle Klasse III) zum Anschluß von:
   - numerische Tastatur und Pufferspeicher.
   - Kartenstanzer P112 A.
   - Drucker MB1 (Olivetti).
   - Alphatastatur.

2. Die schnellen Kanäle (Kanäle Klasse II) zum Anschluß von:
   - Magnetbandstationen.
   - Magnettrommel.
   - Drucker I 41 (BULL).

Aufbau der Kanäle Klasse III.

Es gibt 3 dieser Kanäle. Jeder Kanal enthält:

- Ein Ein-Ausgangs- (quantitatives) Register, genannt K1, K2 oder K3. Jedes Register hat 9 Flip-Flops. Diese erlauben die vorübergehende Speicherung eines Zeichens, während des Dialogs Zentraleinheit - Randeinheit.
- Ein Steuer- (qualitatives) Register, genannt V1, V2 oder V3. In den Steuerregistern ist für jede anzuschließende Randeinheit ein Flip-Flop reserviert. (s. Hauptschema).

Jedes Flip-Flop eines V-Registers verbindet also die ihm entsprechende Randeinheit mit dem Register K.

Auswahl der Kanäle Klasse III

Die Kanäle der Klasse III werden durch das Register J

ausgewählt. Dieses Register besteht aus 4 Flip-Flops.

- J1 wählt den Kanal 1 aus, d.h. K1 und V1
- J2 wählt den Kanal 2 aus, d.h. K2 und V2
- J3 wählt den Kanal 3 aus, d.h. K3 und V3
- J4 wird im Moment noch nicht verwendet.

Dialog in Klasse III

Es gibt 2 verschiedene Dialoge. Einmal ist die Randeinheit das aufnehmende und ein andermal das abgebende Element.

Zur Randeinheit (s. Schema Seite 126)

Die Mikrofunktion TLJ wählt den Kanal aus.
Das Flip-Flop, welches den Kanal freigibt, wird bis zur nächsten TLJ gehalten.

In dem Fall, daß mehrere Randeinheiten pro Kanal angeschlossen werden, muß vor Anschluß einer Randeinheit gesichert sein, daß der Kanal frei ist. (Keine andere Randeinheit ist angeschlossen.) Dies wird geprüft durch die beiden uF, TVL und VLZ. Wenn der Kanal frei ist, sind alle Flip-Flops von Register V auf Null und der Sprung VLZ wird ausgeführt.
Man kann also nun ein Zeichen nach Register L und von hier in das Register K übertragen. (Reg.K entsprechend dem angewählten Kanal)
Durch die uF TLV wird jetzt die Randeinheit angeschlossen, die das Zeichen verarbeiten soll.
Ist dieses Zeichen bearbeitet, wird das Register V auf Null gesetzt und gibt so den Kanal für die Bearbeitung des nächsten Zeichens frei.
Bei der Übertragung von L nach K wird der Zustand 1 sowie der Zustand 0 übertragen, so daß es eine automatische Löschung des alten Registerinhaltes gibt.

Von Randeinheit (s. Schema S. 127)

Die Funktionsweise ist, bis zum Signal "Kanal frei", gleich dem vorherigen Fall (zur Randeinheit).

Durch die uF, TLV wird jetzt das Flip-Flop V gesetzt und somit die Randeinheit angeschlossen. Anschließend gibt es Ablauf IOC.
Während dieser Zeit wird K gelöscht, ein Zeichen von der Randeinheit nach K übertragen (hier wird nur der Zustand 1 übertragen) und V durch die Randeinheit gelöscht. Der Kanal ist also wieder frei.
Beim nächsten Ablauf wird das 1. Zeichen nach L und dann in den Speicher übertragen.
Anschließend wird V, für das 2. Zeichen, erneut gesetzt.

Anmerkung: Auf den Seiten 127 und 128 sind nur die Hauptfunktionen für den Dialog dargestellt.
Der genaue Ablauf ist in Band 2 der Technischen Beschreibung dargestellt.

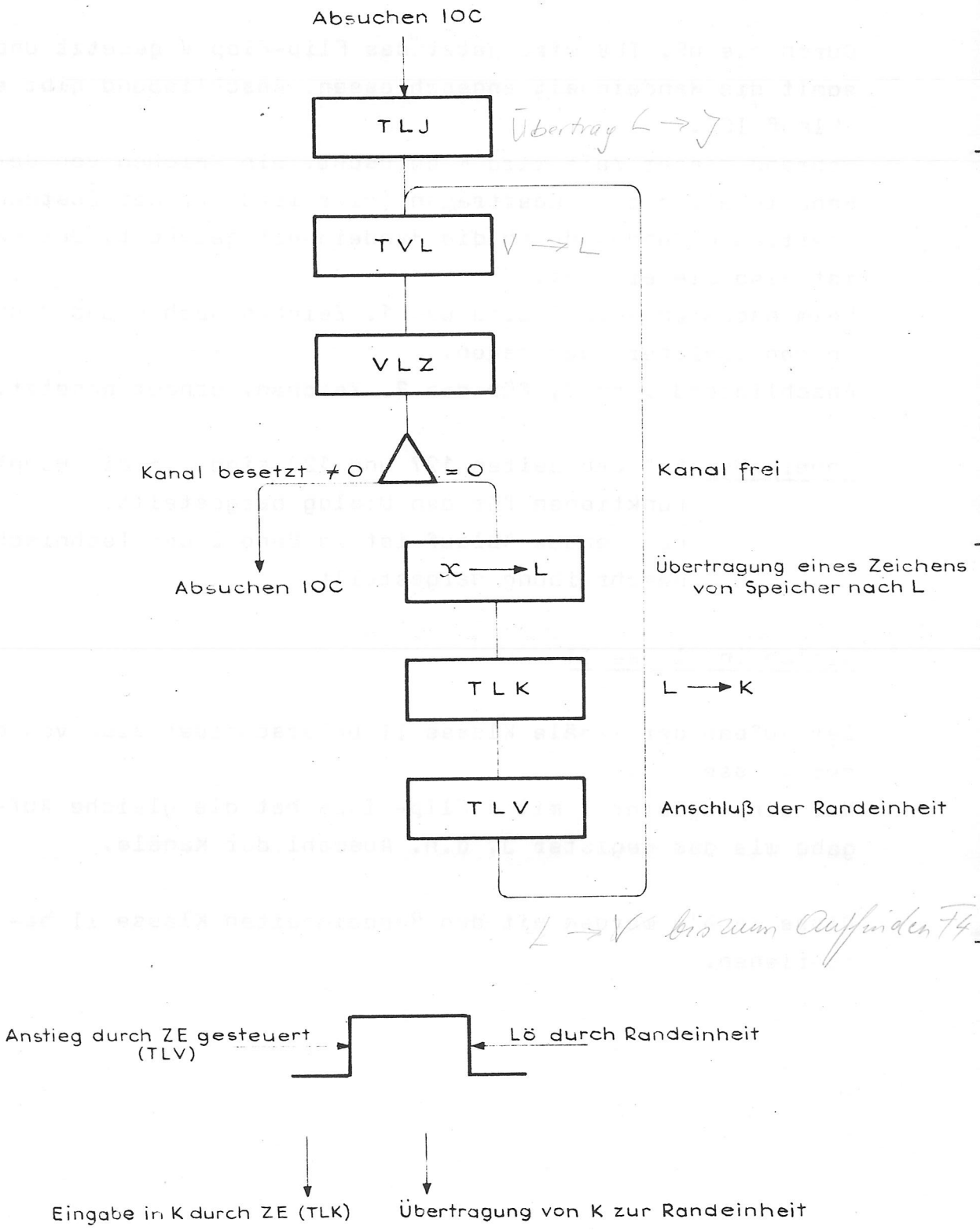
Dialog in Klasse II

Der Aufbau der Kanäle Klasse II unterscheidet sich von dem der Klasse III.
Nur ein Register I mit 4 Flip-Flops hat die gleiche Aufgabe wie das Register J, d.h. Auswahl der Kanäle.

Diese Kanäle werden mit den Randeinheiten Klasse II beschrieben.

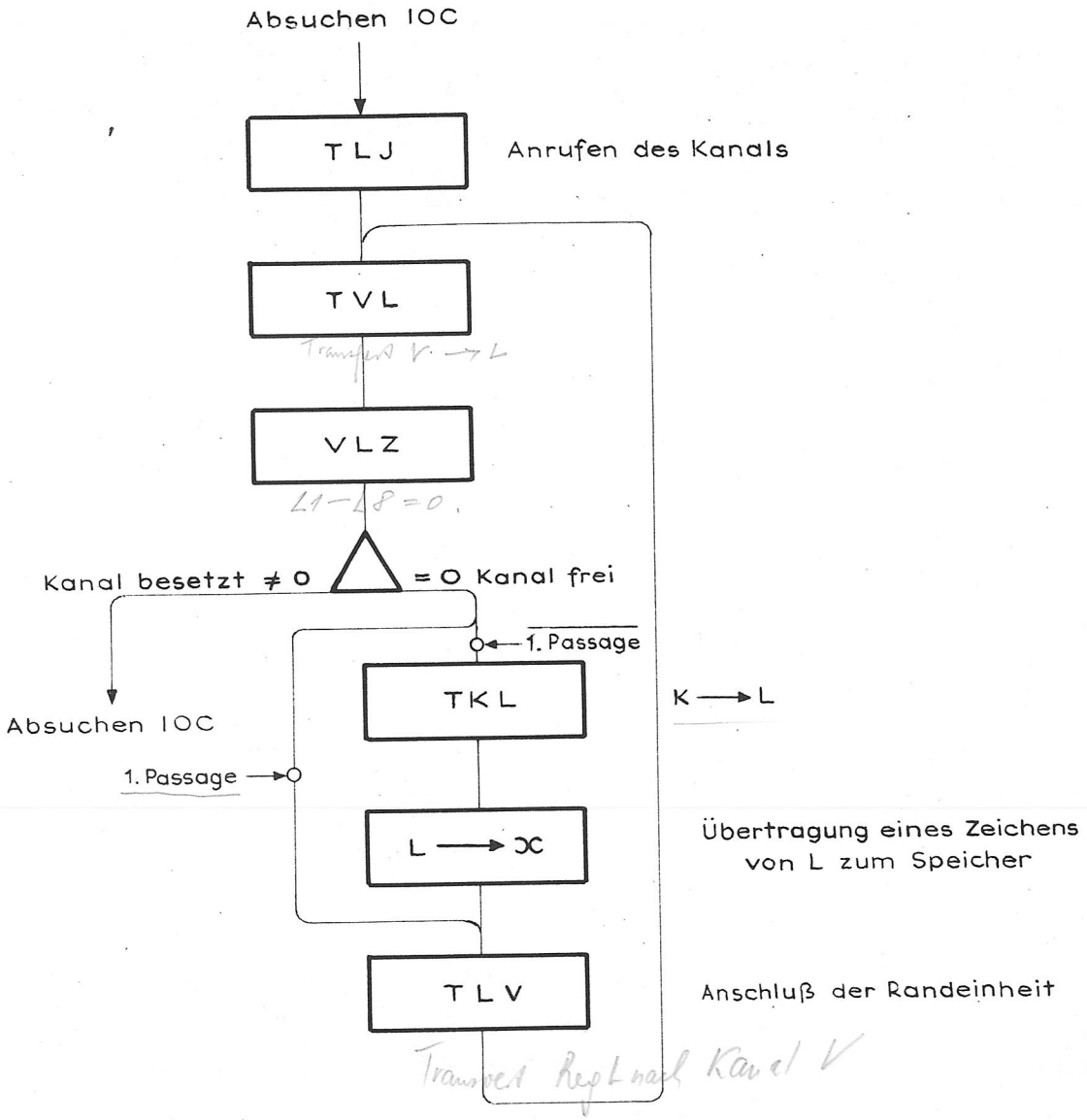
Dialog Zentraleinheit ⟶ Randeinheit (MM04)

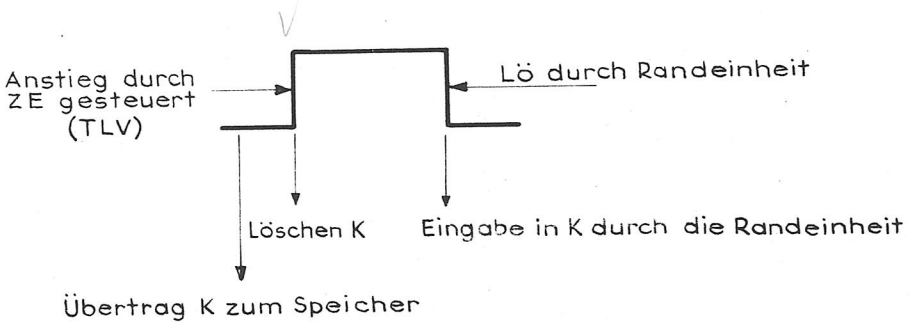
PRINZIP



Dialog Randeinheit ⟶ Zentraleinheit (MM04)

PRINZIP



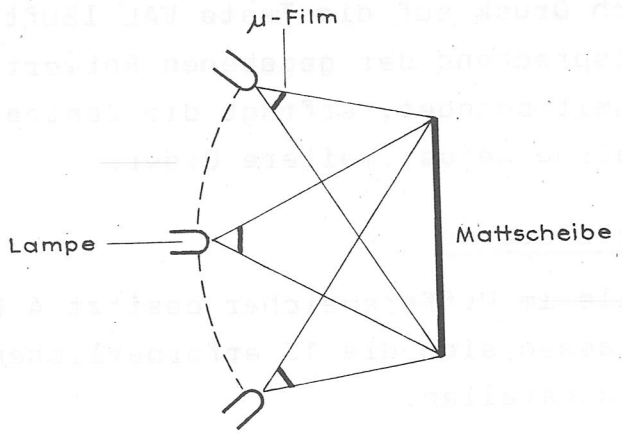


Pufferspeicher und numerische Tastatur

Die numerische Tastatur besitzt 12 Tasten für die Werte 0 - 9, s.w. 11 (:) und 12 (;). Außerdem eine Taste Spaltensprung, eine Taste für den Übertrag in den Pufferspeicher, eine Taste zum Löschen des Pufferspeichers und eine Taste für den Kartenleser L617. L150
Die mechanische Funktion der Tastatur ist in Band 3 der technischen Beschreibung dargestellt.

Die numerische Tastatur ist nur mit dem Pufferspeicher durch einen IOC-Befehl verbunden.

Der Pufferspeicher besitzt ~~8~~ 6 Stellen, zu je 4 Bits. Jede Eingabe geht auf die 1. Stelle. Durch jede Eingabe wird der Inhalt des Pufferspeichers um eine Stelle versetzt (Versetzregister), ohne daß die letzte Stelle wieder auf der ersten Stelle erscheint. D.h., daß der Wert in der letzten Stelle verloren geht.
Der Pufferspeicher ist direkt mit einer Leuchtanzeige verbunden, die zu jeder Zeit den Inhalt des Pufferspeichers angibt.
Jede Stelle der Leuchtanzeige hat für jeden Wert eine Lampe mit einem entsprechenden Mikrofilm. Es gibt also insgesamt ~~8~~ 6 · 12 = 96 Lampen.
Wenn eine Lampe leuchtet, projiziert sie ihr Zeichen auf eine Mattscheibe, die an der Vorderfront der Zentraleinheit sitzt.



Der Pufferspeicher kann:

- einen Wert von der numerischen Tastatur empfangen. Dieser Wert kann durch die Taste RAZ gelöscht werden.
- einen, von der numerischen Tastatur empfangenen Wert, zum Zentralspeicher übertragen. Das geschieht durch Druck auf die Taste "Übertragen" (VAL).
- einen 8-stelligen Wert vom Zentralspeicher empfangen und für die Bedienung sichtbar machen. Die Bedienung zeigt, durch Druck auf die Taste "Löschen" (RAZ) an, daß sie die Information gelesen hat und das Programm weitergehen kann.
- von der Zentraleinheit einen Servicecode empfangen, durch welchen sie unterscheidet, ob die Daten vom Zentralspeicher oder von der Tastatur kommen.
- einen Dialog zwischen Zentraleinheit und Bedienung ermöglichen.

Z.B.: Die Zentraleinheit stellt, zwecks Programmorientierung, eine Frage an die Bedienung über die Leuchtanzeige. Nach Druck auf die Taste RAZ, wird von der Zentraleinheit die numerische Tastatur angeschlossen. Der Anschluß der Tastatur wird durch die Lampe "N" angezeigt.
Die Bedienung gibt ihre Antwort über die numerische Tastatur.
Nach Druck auf die Taste VAL läuft das Programm, entsprechend der gegebenen Antwort ab. Ist diese Arbeit beendet, erfragt die Zentraleinheit auf die gleiche Weise, weitere Order.

Pufferspeichercode

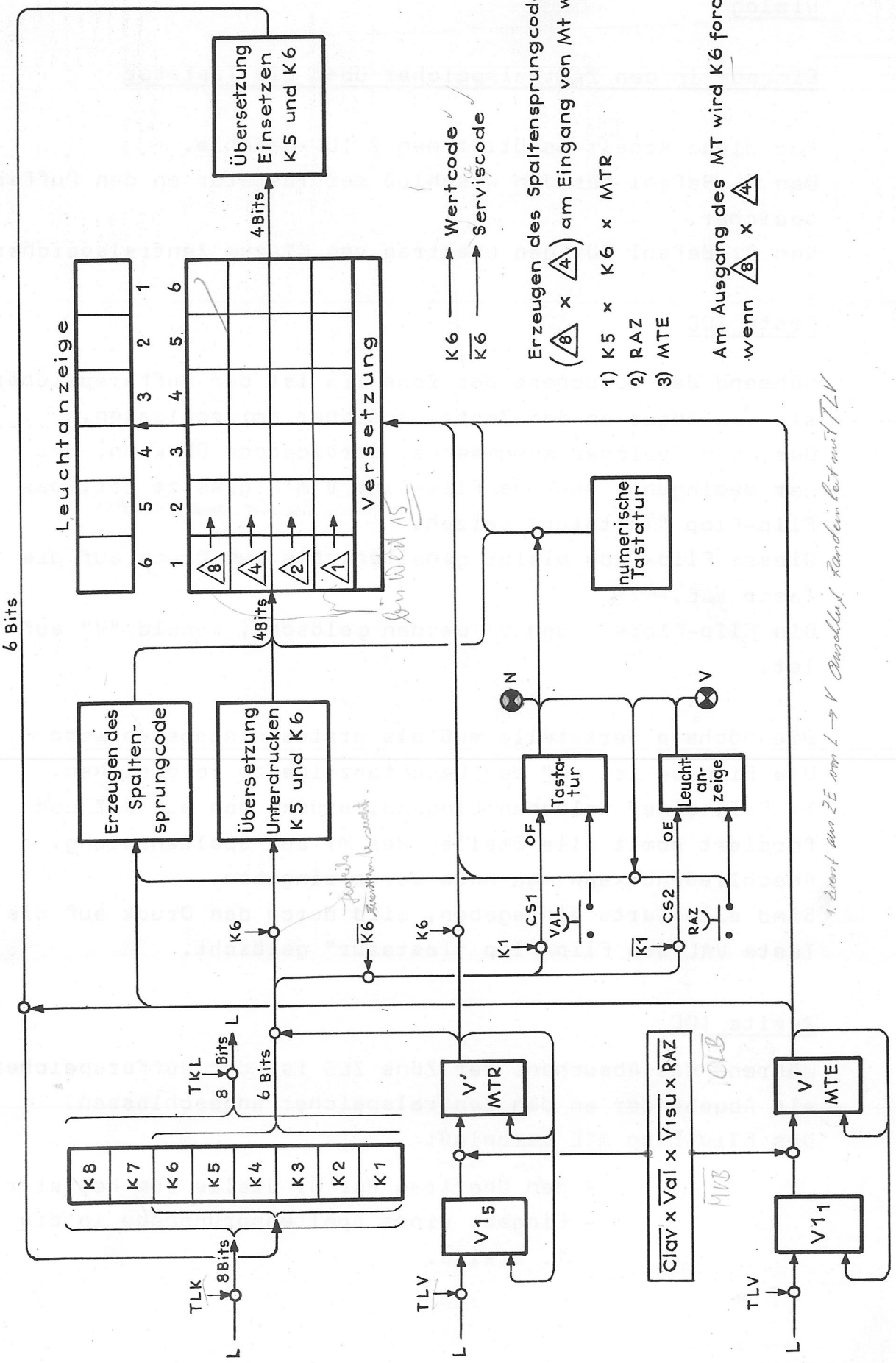
Jede Stelle im Pufferspeicher besitzt 4 Binärpositionen. Hiermit lassen sich die 13 erforderlichen Kombinationen des Code darstellen.

Dieser Code entspricht den 1. vier Stellen des ISO-Code. Eine Ausnahme ist der Spaltensprung. Er ist codiert mit $\overline{1}$, $\overline{2}$, 4, 8, zur Unterscheidung mit dem Code für Null $\overline{1}$, $\overline{2}$, $\overline{4}$, $\overline{8}$. Bei Spaltensprung bleibt die Leuchtanzeige auf dieser Stelle dunkel.
Ist der Pufferspeicher gelöscht, so sind alle Stellen auf Spaltensprung forciert.

Tastatur	Leuchtanzeiger	Code MT 8 4 2 1	Code Register K 8 7 6 5	4 3 2 1	Hexadez. Darstell.
Esp.		1 1 0 0	0 0 1 0	0 0 0 0	20
0	0	0 0 0 0	0 0 1 1	0 0 0 0	30
1	1	0 0 0 1	0 0 1 1	0 0 0 1	31
2	2	0 0 1 0	0 0 1 1	0 0 1 0	32
3	3	0 0 1 1	0 0 1 1	0 0 1 1	33
4	4	0 1 0 0	0 0 1 1	0 1 0 0	34
5	5	0 1 0 1	0 0 1 1	0 1 0 1	35
6	6	0 1 1 0	0 0 1 1	0 1 1 0	36
7	7	0 1 1 1	0 0 1 1	0 1 1 1	37
8	8	1 0 0 0	0 0 1 1	1 0 0 0	38
9	9	1 0 0 1	0 0 1 1	1 0 0 1	39
11 (:)	•	1 0 1 0	0 0 1 1	1 0 1 0	3A
12 (;)	•	1 0 1 1	0 0 1 1	1 0 1 1	3B
Serviscode	numerische Tastatur		0 0 0 0	1 1 1 1	0F
	Leuchtanzeiger		0 0 0 0	1 1 1 0	0E

Die Service-Codes werden nicht in den Pufferspeicher eingegeben.
0F setzt das Flip-Flop "Tastatur", zum Anschluß der numerischen Tastatur.
0E setzt das Flip-Flop "Leuchtanzeige", um eine Zone des Zentralspeichers sichtbar zu machen.

Wenn man vom Pufferspeicher zum Zentralspeicher überträgt, muß der Wert in den ISO-Code umgesetzt werden.
Bei den Werten von 0 - 12 werden nur die Binärstellen 5 und 6 auf 1 gesetzt. Bei Spaltensprung wird die Binärstelle 6 auf 1 und alle anderen auf 0 forciert.


Übersetzung Einsetzen K5 und K6
4 Bits
Leuchtanzeige
Versetzung
K6 → Wertcode
K̄6 → Serviscode
Erzeugen des Spaltensprungcode
(8 × 4) am Eingang von Mt wenn
1) K5 × K6 × MTR
2) RAZ
3) MTE
Am Ausgang des MT wird K6 forciert
wenn 8 × 4
numerische Tastatur
6 Bits
4 Bits
Erzeugen des Spalten-sprungcode
Übersetzung Unterdrucken K5 und K6
N
V
Tasta-tur
Leucht-an-zeige
OF
OE
CS1
CS2
VAL
RAZ
K6
K̄6
K1
K̄1
TKL
8 Bits
L
6 Bits
K8
K7
K6
K5
K4
K3
K2
K1
V' MTR
V1 5
Clav × Val × Visu × RAZ
V' MTE
V1 1
TLK
8 Bits
TLV
TLV
L
L
L

Dialog

Eingabe in den Zentralspeicher über die Tastatur

Für diese Arbeit benötigt man 2 IOC-Befehle.
Den 1. Befehl für den Anschluß der Tastatur an den Pufferspeicher.
Den 2. Befehl für den Übertrag vom MT zum Zentralspeicher.

Erste IOC

Während des Absuchens der Zone ZES ist der Pufferspeicher als Empfänger an den Zentralspeicher angeschlossen.
Der, vom Speicher ankommende, Servicecode OF kann, unter der Bedingung, daß das Flip-Flop V'MTR gesetzt ist, das Flip-Flop "Tastatur" setzen.
Mémoire Tampon Réceptrice
Dieses Flip-Flop bleibt gehalten, bis zum Druck auf die Taste VAL. MVB
Die Flip-Flops V und V' werden gelöscht, sobald "V" auf 1 ist.

Die höchste Wertstelle muß als erstes eingegeben werden.
Die Eingabe ist auf der Leuchtanzeige zu vergleichen.
Im Fall einer falschen Eingabe, drückt man auf ~~RAZ~~ CLB und forciert somit alle Stellen des MT auf Spaltensprung.
Anschließend kann man neue Werte eingeben.
Sind alle Werte eingegeben, wird durch den Druck auf die Taste ~~VAL~~ MVB das Flip-Flop "Tastatur" gelöscht.

Zweite IOC

Während des Absuchens der Zone ZES ist der Pufferspeicher als Abgebender an den Zentralspeicher angeschlossen.
Das Flip-Flop MTE veranlaßt:
Mémoire Tampon Emettrice

- den Übertrag der 8. Stelle zum Register K.
- Eingabe eines Spaltensprungcode in die 1. Stelle.

Verschiebung nach links.

- Versetzung der Zeichen im Pufferspeicher um eine Stelle.
- Löschen der Flip-Flops V und V'

Das Flip-Flop V wird für jedes Zeichen wieder angerufen, bis die Trennmarke F4 aufgefunden wird.

Wenn beim ersten Anruf von V, die Eingabe in die Tastatur noch nicht beendet ist (das ist im allgemeinen der Fall), wird das Flip-Flop V' nicht gesetzt und V gehalten.
V gesetzt, ergibt beim Absuchen der IOC-Zone, die Entscheidung nicht frei für diesen Kanal, bis die Taste "Validation" gedrückt wird.

<u>Beispiel:</u> 1, IOC 53-01-90-12

	0	F	F	4		0	1	5	5	Rg.12
01 55										

Anschluß der Tastatur-Eingabe von 1, 2, 3, 4, 5 und 6 -
Übertragung (Taste VAL)
Freigabe Tastatur IOC mit MKB

2. IOC 53-01-95-10

					X	X	X	X	Rg.10

Übertragung zum Zentralspeicher

	3	1	3	2	3	3	3	4	3	5	3	6	F	4
XXXX														

Anzeige einer Speicherzone auf der Leuchtanzeige

Diese Arbeit kann auf zwei verschiedene Arten ausgeführt werden:

- mit einem IOC-Befehl oder
- mit zwei IOC-Befehlen

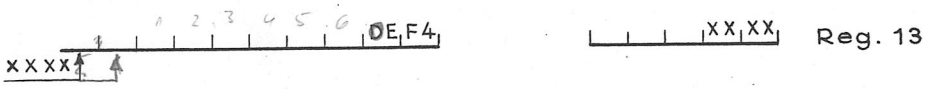
Mit einem IOC-Befehl

Während des Absuchens der IOC-Zone (ZES) ist der Pufferspeicher als Empfänger angeschlossen. Das Flip-Flop V1 5 steuert V'MTR.
Der Service-Code OE ist direkt vor der Trennmarke F4 in der entsprechenden Speicherzone plaziert.
Wenn der Code OE gelesen wird, geht das Flip-Flop VISU auf 1 und bleibt gehalten bis zum Druck auf die Taste RAZ.

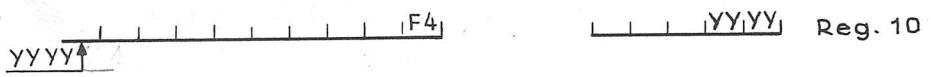
Mit zwei IOC-Befehlen

Während des Ablaufs der 1. IOC wird die Speicherzone zum Pufferspeicher übertragen.
Wenn die Trennmarke F4 gelesen wird, startet das Programm die 2. IOC.
Diese 2. IOC ruft ein Register an, in dem OE und F4 gespeichert sind. OE setzt das Flip-Flop VISU, welches gehalten wird, bis die Taste RAZ gedrückt wird.
Die Übertragung vom Flip-Flop V nach V' ist in diesem Fall durch VISU oder RAZ unterbunden, d.h. daß erst am Ende von RAZ übertragen wird.

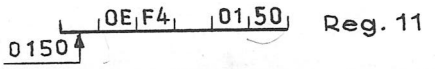
Beispiel: 1. Fall 53 01 9D 13



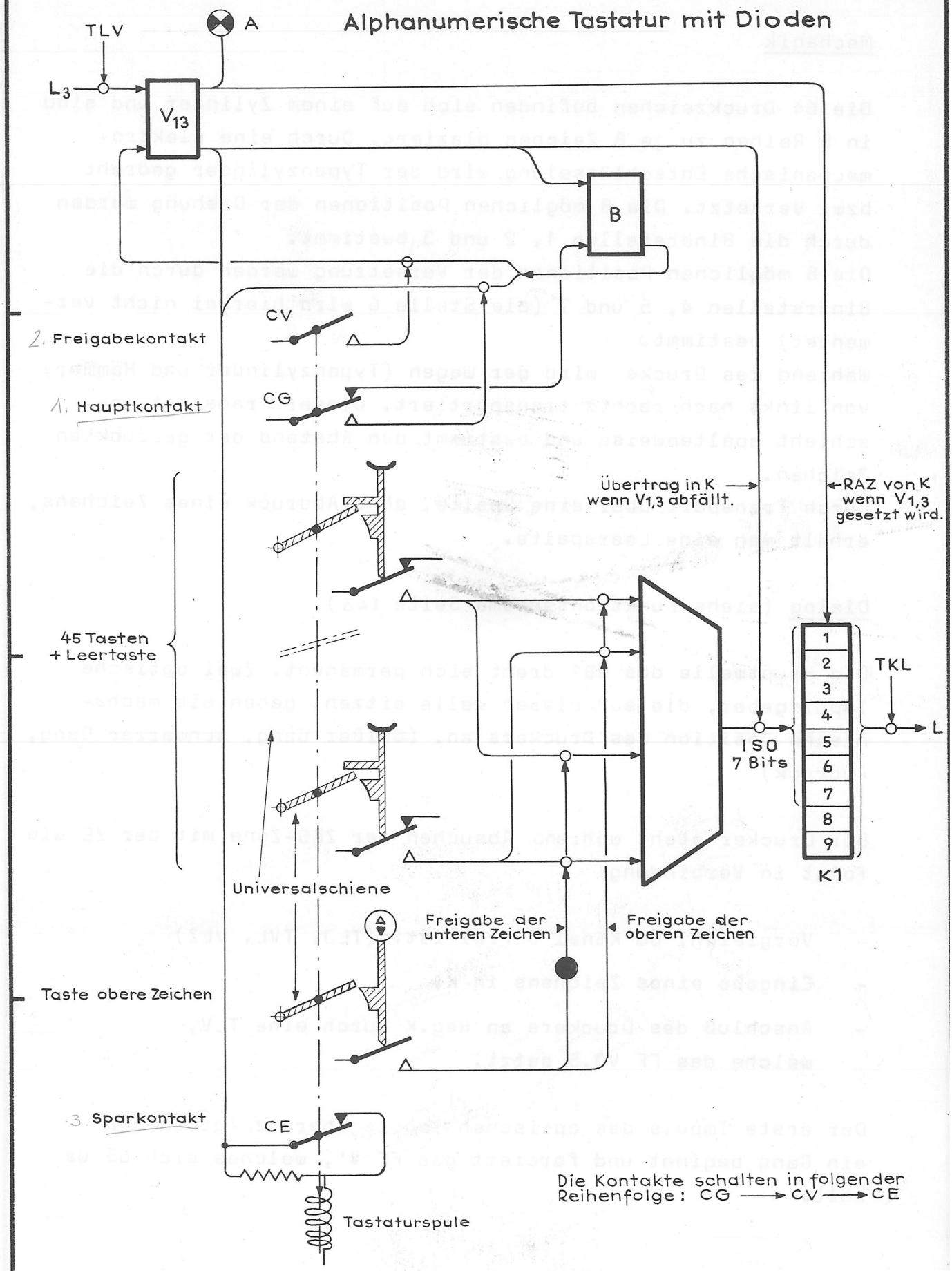
2. Fall 53 01 9D 10



53 01 9D 11



Alphanumerische Tastatur mit Dioden



Die Kontakte schalten in folgender Reihenfolge: CG ⟶ CV ⟶ CE

Mechanik

Die 64 Druckzeichen befinden sich auf einem Zylinder und sind in 8 Reihen zu je 8 Zeichen plaziert. Durch eine elektromechanische Entschlüsselung wird der Typenzylinder gedreht bzw. versetzt. Die 8 möglichen Positionen der Drehung werden durch die Binärstellen 1, 2 und 3 bestimmt.
Die 8 möglichen Positionen der Versetzung werden durch die Binärstellen 4, 5 und 7 (die Stelle 6 wird hierbei nicht verwendet) bestimmt.
Während des Drucks wird der Wagen (Typenzylinder und Hammer) von links nach rechts transportiert. Dieser Transport geschieht spaltenweise und bestimmt den Abstand der gedruckten Zeichen.
Durch Transport über eine Spalte, ohne Abdruck eines Zeichens, erhält man eine Leerspalte.

Dialog (siehe Funktionsschema Seite 143)

Die Hauptwelle des MB1 dreht sich permanent. Zwei optische Impulsgeber, die auf dieser Welle sitzen, geben die mechanische Position des Druckers an. (Weißer Gang, schwarzer Gang, Abdruck)

Der Drucker steht während Absuchen der ZES-Zone mit der ZE wie folgt in Verbindung:

- Vergleich, ob Kanal 3 frei ist. (TLJ, TVL, VLZ)
- Eingabe eines Zeichens in K.
- Anschluß des Druckers an Reg.K durch eine TLV, welche das FF V3,5 setzt.

Der erste Impuls des optischen Impulsgebers zeigt an, daß ein Gang beginnt und forciert das FF V', welches sich 65 us hält.

Unter der Bedingung ($\overline{\text{Papierfehler}}$ X $\overline{\text{Sperren RC}}$ X $\overline{\text{Sperren INT}}$) erfolgt durch das FF V',

- das Löschen von V3,5 (nach 50 us)
  die Freigabe der Arbeitssteuerung entsprechend dem Inhalt von Register K.

Das heißt:

Im ersten Fall: K3 enthält den Code für ein Zeichen.
($\overline{\text{Servis-Code}}$ X $\overline{\text{Spaltensprung}}$)

Das FF V' steuert

- den Übertrag des Zeichens in den Pufferspeicher.

und 50 us später

- die Speicher-FF für Abdruck und Spaltensprung.
- einen Impuls von 11 ms zur Erregung der Codierungsspulen.
- das FF V3,5 auf Log. 0.

V' wird 15 µs nach V3,5 auf Log. 0 gestellt.
Der Druckimpuls (optischer Unterbrecher) setzt

- das FF "Spaltensprung" (Avance) zur Steuerung des seitlichen Wagentransportes.
- das FF "Abdruck" (Frappe), welches im folgenden Gang den Abdruck bringt.
  Im gleichen Gang können auch die Codierungsspulen für das nächste Zeichen schon erregt werden. Liegt kein neues Zeichen vor, so steht im Pufferspeicher noch der alte Wert und die Komplementspulen werden durch das FF "Abdruck" angesteuert. Hierdurch wird die Codierungsmechanik neutralisiert, so daß der Typenzylinder bis zum nächsten Zeichen in Ruhestellung bleibt.

Im zweiten Fall: K3 enthält das Zeichen für Spaltensprung (20).

Das FF V' setzt in diesem Fall nur das Speicher-FF für Spaltensprung. Die Ausführung ist dann wie im vorhergehenden Fall.

Im dritten Fall: K3 enthält den Servis-Code 0A (Zeilensprung).

Durch das FF V' wird nur das Speicher-FF "Zeilensprung" gesetzt. Dieses FF setzt 15 µs später das FF "Sperren Zeilensprung" (blocage interligne), welches sich über 3 Gänge hält. Der Druckimpuls setzt das FF "Zeilensprung", welches einen Gang gehalten wird und durch Erregen der Kupplungsspule den Zeilensprung auslöst.
Die mechanische Ausführung des Zeilensprungs dauert 2 1/2Gänge (1/2 Gang als Sicherheit für die Kupplung und 2 Gänge für den Sprung).
Das FF "Sperren Zeilensprung" verhindert die Auswertung des FF V' während des Zeilensprungs.

Im vierten Fall: K3 enthält den Servis-Code 0C (Papiersprung)

V' setzt das Speicher-FF "Sprung", welches gehalten wird, bis die Sprungbürste mit Masse in Berührung kommt (durch ein Loch im Steuerstreifen). Dieses FF steuert permanent die Gruppe Zeilensprung an.

Dieser Sprung kann auch durch die Taste "Sprung und Wagenrücklauf" (SCR) gestartet werden.

Im fünften Fall: K3 enthält den Servis-Code 0D (Wagenrücklauf)

V' setzt das FF "Wagenrücklauf", welches einen Gang gehalten wird. 15 µs später wird das FF "Sperren Wagenrücklauf" gesetzt.
Durch das FF "Wagenrücklauf"wird eine Spule erregt, welche die mechanische Verbindung zwischen Wagen und Spaltensprungantrieb löst. Der Wagen wird nun durch Federkraft in seine Ausgangsstellung gebracht. (Spalte 1). Der Wagen geht bei diesem Vor-

gang über seine Ausgangsstellung zurück und schließt einen Kontakt. Dieser Kontakt steuert sofort einen Spaltensprung, durch den der Wagen in die Spalte 1 gebracht wird.
Das hat den Vorteil, daß der Wagen immer in der gleichen Position beginnt. Bei einfachem Anschlag könnte das, durch die Prellung, nicht gewährleistet werden.
Durch diesen Spaltensprung wird auch das FF "Sperren Wagenrücklauf" gelöscht.
Dieser Wagenrücklauf kann auch durch die Taste "Sprung und Wagenrücklauf" (SCR) gestartet werden.

Im sechsten Fall: K3 enthält den Servis-Code 0B (Wagenrücklauf und Zeilensprung).

V' ruft in diesem Fall die FF "Wagenrücklauf" und "Zeilensprung" an.
Die Ausführung ist simultan (s.Fall 4 und 5).

Papierfehler

Dies ist der einzige Fehlerfall, der angezeigt wird.
Man unterscheidet 2 Fälle:

1. Ende Papiervorrat.

Dieser Kontakt sitzt auf der Papiervorratsauflage und schließt, sobald kein Papier mehr aufliegt.
Das FF "Fehler Papier" (Incident papier) wird erst gesetzt, wenn die Sprungbürste mit Masse in Verbindung kommt, also ein Formular fertig geschrieben ist.

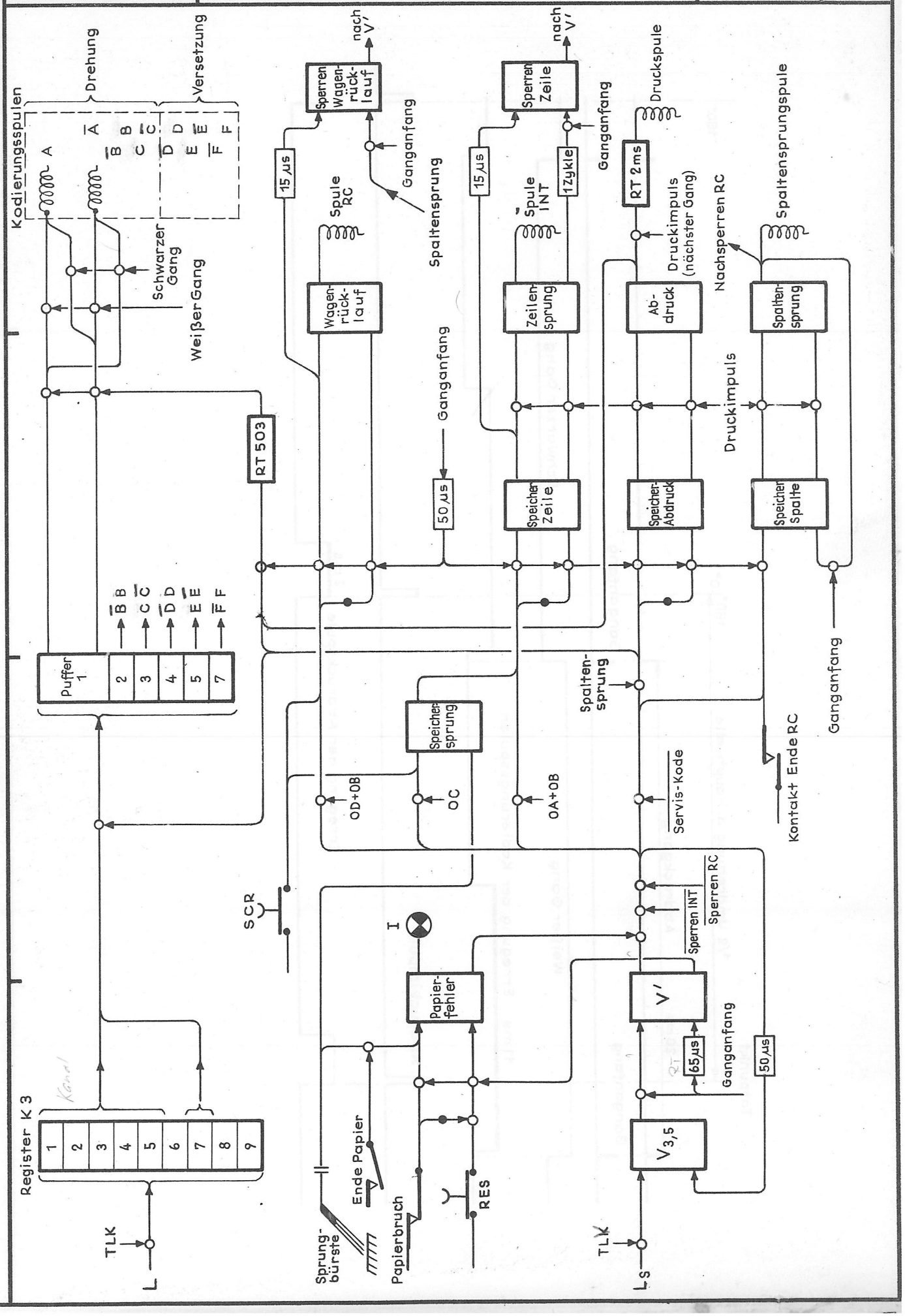
Durch die Taste "Löschen Papierfehler" (RES) kann das FF gelöscht werden, um ein weiteres Formular zu schreiben.
Am Ende dieses Formulars wird aber "Fehler Papier" wieder gesetzt.

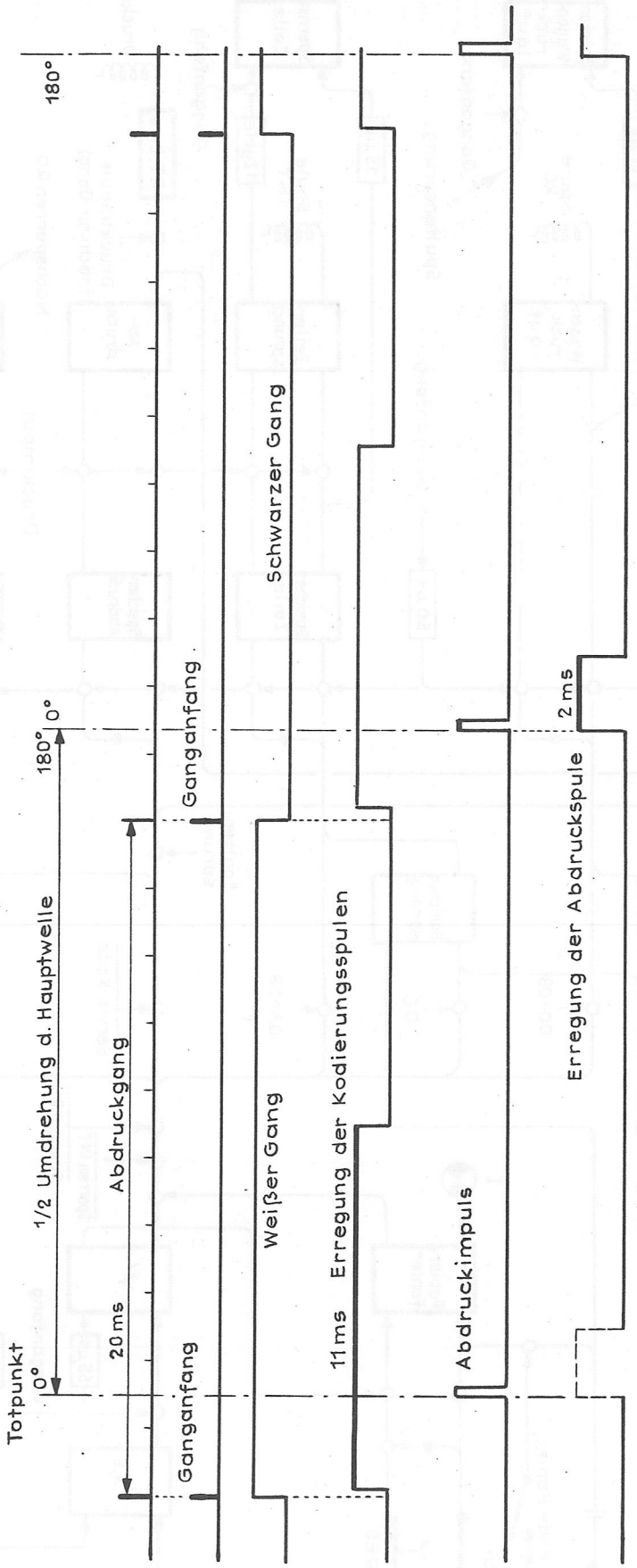
2. Papierbruch

Dieser Kontakt schließt, sobald die Spannung des transportierten Papiers nachläßt, das Papier also gerissen ist.

Der Kontakt wirkt sofort auf das FF "Papierfehler" unter der Bedingung V'.

Das FF "Papierfehler" kann in diesem Fall nur gelöscht werden, wenn der Kontakt "Papierbruch" wieder geöffnet ist, der Fehler also behoben ist.


Kodierungsspulen
Drehung
Versetzung
A
B
C
D
E
F
Schwarzer Gang
Weißer Gang
RT 503
Puffer 1
2
3
4
5
7
Register K 3
1
2
3
4
5
6
7
8
9
TLK
L
Sprung-bürste
Ende Papier
Papierbruch
RES
SCR
I
Papier-fehler
OD+OB
OC
OA+OB
Speicher sprung
Wagen-rück-lauf
Spule RC
15 µs
Sperren Wagen-rück-lauf
nach V'
Ganganfang
Spaltensprung
50 µs
Zeilen-sprung
Speicher Zeile
Spule INT
1Zykle
Sperren Zeile
Ab-druck
Speicher-Abdruck
Druckspule
RT 2ms
Druckimpuls (nächster Gang)
Druckimpuls
Spalten-sprung
Speicher Spalte
Nachsperren RC
Spaltensprungspule
Servis-Kode
Kontakt Ende RC
Sperren INT
Sperren RC
V'
65 µs
V3,5
TLK
LS


180°
Schwarzer Gang
Ganganfang
180° 0°
2 ms
1/2 Umdrehung d. Hauptwelle
Abdruckgang
Weißer Gang
Erregung der Kodierungsspulen
Erregung der Abdruckspule
20 ms
11 ms
Abdruckimpuls
Totpunkt
0°
Ganganfang

Allgemeines:

Die Lesung erfolgt durch Photozellen mit einer Geschwindigkeit von 150 Karten/min.
Die Zeit für die Lesung einer Karte beträgt 275 ms.
Zwischen dem ersten TLV und der Lesung der ersten Spalte liegen 110 ms.
Die Maschine arbeitet Karte für Karte ohne Stop zwischen den einzelnen Spalten.
Wenn der Leser angeschlossen wird, kommt eine Karte aus dem Magazin unter die Lesestation und wird, durch die permanent drehenden Rollen der Bahn, bis in das Ablagefach transportiert.
Das Magazin enthält 500 Karten.
Der Leser verarbeitet auch versetzte Löcher. Es soll aber wenigstens ein trennender Steg zwischen 2 Lochungen bestehen.
Die abgelegten Karten liegen in der gleichen Ordnung wie im Magazin.
Ein perforiertes Steuerband, das durch die Karte transportiert wird, synchronisiert automatisch die Karte mit dem Taktgeber, der ebenfalls in das Steuerband gelocht ist.

Drucktasten

Es gibt 4 Tasten:

- eine Taste Ablage (EJ)
- eine Taste ~~MAN~~ Reset
- eine Taste mit Verriegelung SRD welche:
  - in eingedrücktem Zustand die Freigabe der nächsten Karte von der numerischen Tastatur (Taste SRD) abhängig macht.
  - in Ruhezustand die Steuerung der nächsten Karte von der Zentraleinheit abhängig macht.
- eine mechanische Taste zur nachträglichen Eingabe kleinerer Kartenmengen (1 - 10 Stk.)

Funktionsweise

1. Taste SRD in Ruhe

Die Lesung einer Karte wird durch die Zentraleinheit angesteuert. Mit dem 1. TLV wird die Zufuhr erregt und die Karte verläßt das Magazin. Sobald der erste Impuls des Taktgebers gelesen wird, geht die Zufuhr in Ruhe und die Transportrollen steuern die weitere Bewegung der Karte. Das FF V bleibt gehalten.
Der 2. Impuls des Taktgebers gibt die Lesung der 1. Spalte frei.
Das FF V wird gelöscht und mit der ahsteigenden Flanke des Ausgangs 2 wird der gelesene Wert in das Register K übertragen. Der TLV (neuer Anschluß) für die Spalte 2 soll vor dem 3. Impuls des Taktgebers kommen (d.h. spätestens 3,4 ms nach dem 2. Impuls).
Alle TLV sollen also aufeinander folgen. Zu diesem Zweck ist während der Lesung der dauernde Ablauf der ZES-Zone erforderlich.
Somit ist auch der simultane Ablauf eines Programmes ausgeschlossen.
Der 81. Impuls markiert die Lesung der Spalte 80 und liefert nach 3,9 ms einen Impuls "Ende Karte".

Single Card Read

2. Taste SRD in Arbeit

Ist die Taste SRD auf der numerischen Tastatur in Ruhe, so wird keine Karte gelesen, wenn auch das FF V gesetzt ist.
Ein Druck auf die Taste SRD (num. Tastatur) steuert eine Zufuhr. Die Karte wird gestartet mit dem ersten TLV.
Nach der Zufuhr ist die Bearbeitung gleich der im vorherigen Fall.

3. Ablage

Ein Druck auf diese Taste führt eine Karte zu, die an der Lesestation vorbei läuft. Es wird aber kein Wert gelesen, selbst wenn während der Ablage ein TLV ankommt.

4. Lesen des Code FF in einer Ladekarte

Wird in einer Karte der Code FF gelesen, so läuft in der Zentraleinheit eine Mikroprogrammsequenz ab, ~~welche den nächsten TLV verzögert~~. Hierdurch wird die Spalte die FF enthält, wie die letzte Spalte behandelt und das Flip-Flop "Ende Lesung" gesetzt. Dieses Flip-Flop verhindert das Löschen von V und somit die Eingabe eines neuen Zeichens in Register K.

Fehlerfälle

- volles Fach
- Zellenkontrolle
- fehlende Impulse des Taktgebers.

Die Fehlerfälle bringen 1 sec. nach ihrer Auffindung die Fehlerlampe und verhindern eine neue Zufuhr.
Die Zellenkontrolle ist am Ende einer Karte noch nicht ausgeführt. Durch die Verzögerung von 1 sec. wird die Fehlerlampe aber nicht angerufen, da inzwischen die Kontrolle ausgeführt wurde.
Nach einem Fehlerhalt durch die Zellenkontrolle kann man nur durch Ablage oder Aus- und Einschalten der Gesamtspannung weiterarbeiten.

Ein Stau bei der Kartenzuführung macht sich dadurch bemerkbar, daß 1 sec. nach Start einer Karte kein Impuls gekommen ist.
Die Zufuhr wird dann gelöscht und die Fehlerlampe leuchtet.

Verwendung der Taste ~~MAN~~ RESET

1. Bei Stau im Magazin

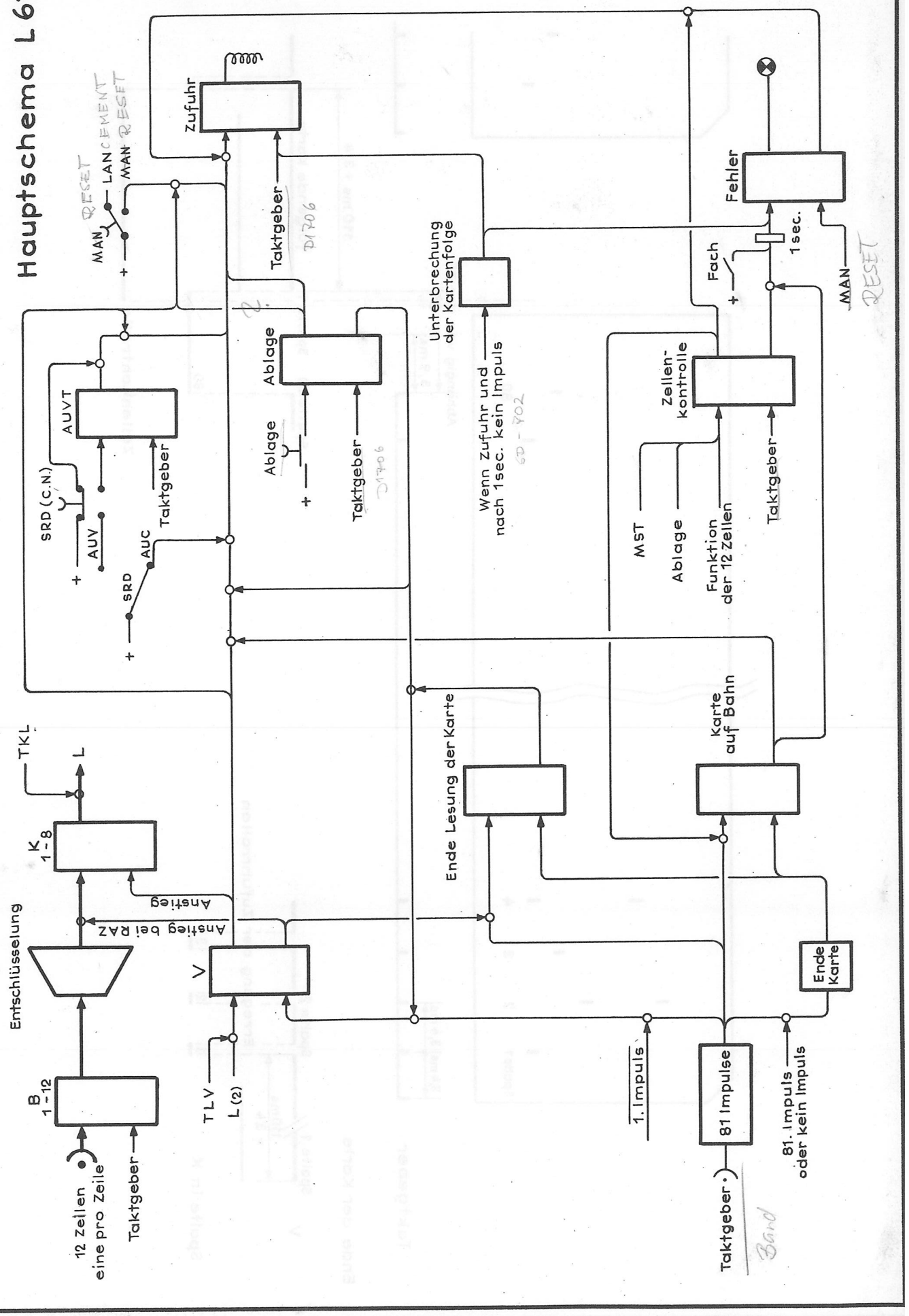
Das FF V, das für die Karte angerufen wurde, welche den Stau verursacht hat, bleibt gehalten, bis zum Impuls des Taktgebers, der in diesem Fall ausbleibt.

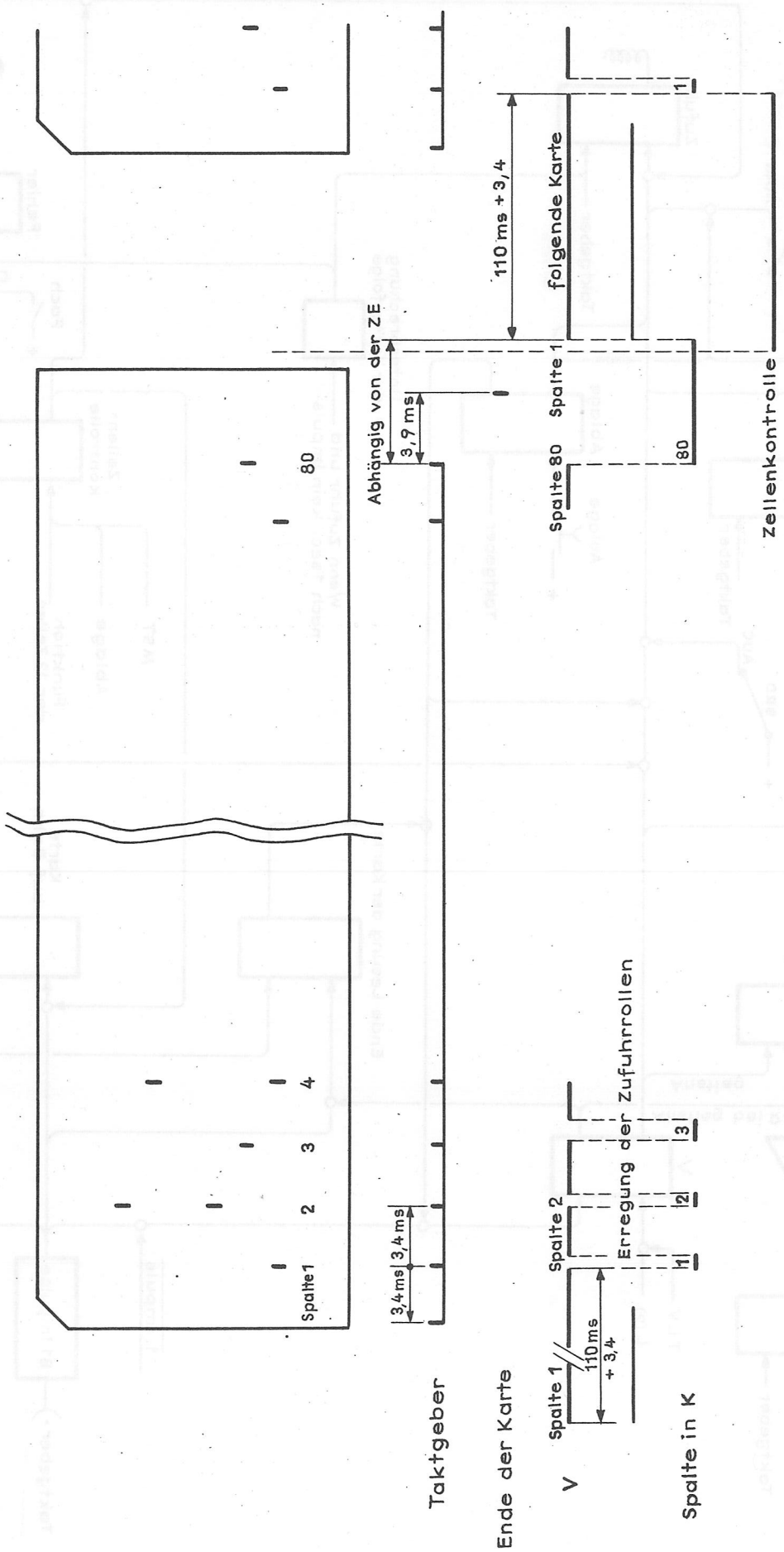
Nach Zurückziehen dieser Karte kann man diese wieder in das Magazin legen und durch die Taste MAN zuführen. Das noch gehaltene FF V überträgt die erste Spalte. Der folgende Ablauf der Lesung ist wieder normal.
Der Druck auf MAN löscht auch das FF "Fehler".

2. Bei vollem Fach

Bei vollem Fach wird eine neue Zufuhr verhindert und das FF V für die nächste Karte gehalten.
Nach der Entleerung des Ablagefaches drückt man die Taste MAN. Hierdurch wird das FF "Fehler" gelöscht und die nächste Karte zugeführt.

Hauptschema L 617




Spalte 1
2
3
4
80
3,4 ms
3,4 ms
Taktgeber
Ende der Karte
V
Spalte 1
110 ms + 3,4
Spalte 2
1
2
3
Erregung der Zufuhrrollen
Spalte in K
Abhängig von der ZE
3,9 ms
Spalte 80
Spalte 1
80
110 ms + 3,4
folgende Karte
1
Zellenkontrolle

Stanzer P112 A

Sehen Sie hierzu das Hauptschema sowie das Ablaufschema auf den folgenden Seiten.

Das zu stanzende Zeichen steht in Register K2.
Durch eine TLV drehen sich auch die Nocken C1, C2 und C3.

Außerdem wird V' unter der Bedingung $\overline{\text{Ablage}}$ X $\overline{\text{Verhindern Stanzen}}$, gesetzt.
Der Zeitvergleich erfolgt durch die Nocken C1 und C2:

- Das FF "Anfang Zeitvergleich" wird, wenn V' anwesend ist, durch C1 gesetzt.
- Das FF "Ende Zeitvergleich" wird unter der Bedingung FF "Anfang Zeitvergleich", durch C2 gesetzt.

Während des Zeitvergleichs, der von Anfang C1 bis Anfang C2 dauert:

- wird die Spaltensprungspule sofort angesteuert.
- empfangen, wenn $\overline{\text{OD}}$, die Stanzspulen die Signale von K2. (Der Inhalt des Registers wird, entsprechend der Codierungsplatte, in den T- oder H-Code umgeschlüsselt.)

Zwischen dem Inhalt von K2 und den Kontrollzellen findet ein Vergleich statt.
Das Vergleichsergebnis wird durch die Nocke C3 ausgewertet (höchster Punkt der Stanznocke).
Die Rückstellung von V2,5 erfolgt durch Gleichheit und V' auf 1, bei Ende Zeitvergleich.

V2,5 in Ruhe bringt dann V', wodurch die FF "Anfang Zeitvergleich" und "Ende Zeitvergleich" gelöscht werden.

Stanzfehler:

Wenn der Vergleich zwischen K2 und den Kontrollzellen Ungleichheit ergibt, kann V2,5 nicht gelöscht werden. Hierdurch bleibt V' auf 1 und "Anfang Zeitvergleich" und "Ende Zeitvergleich" können nicht zurückgestellt werden.
Die Maschine hält in der nächsten Spalte und die Lampe PC brennt.
Das zu stanzende Zeichen ist weiterhin in K2.
Zur Wiederaufnahme der Arbeit muß IC gedrückt werden.

Ablage

Die Ablage kann durch den Servis-Code OD oder die Taste "Ablage" gestartet werden.

Ablage durch den Servis-Code OD

Der Servis-Code OD kommt über das Register K2 in L. Mit TLV und L5 wird das FF V2,5 gesetzt.
V2,5 bringt V' und die Warnerkupplung, wodurch die Nocken C1, C2 und C3 drehen. Mit V' werden "Anfang Zeitvergleich" und "Ende Zeitvergleich" durch die Nocken C1 bzw. C2 gesetzt.

Während des Zeitvergleiches:

- wird die Spaltensprungspule, wie beim Stanzen, direkt erregt. Die Erregung der Stanzspulen wird durch den Code OD verhindert.
- wird durch den Code OD das FF "mech. Ablage" gesetzt, das bis zur letzten Spalte, durch $\overline{CT2}$ gehalten bleibt.

Am Ende des Zeitvergleiches:

- wird der Servis-Code OD in den Speicher OD übertragen.
- wird durch den Anstieg des Speichers OD das FF V2,5 und dann "Anfang Zeitvergleich" und "Ende Zeitvergleich" gelöscht.

Ein weiterer Zeitvergleich während der Ablage ist unmöglich, da die Übertragung von V2,5 nach V' vom OD-Speicher abhängig ist und dieser bis in die letzte Spalte (CT2) gehalten wird.

Ablage durch Taste

Die Ablage der Karte durch Taste kann erfolgen:

a) bei Beginn der Arbeit

b) am Ende des Stanzens

c) während des Stanzens.

a) Bei Beginn der Arbeit wird obligatorisch die Taste Ablage gedrückt, um zu gewährleisten, daß eine Karte mit der 1. Spalte unter der Stanzstation liegt.

b) Bei Beendigung der Stanzarbeit, um die zuletzt gestanzte Karte in das Ablagefach zu transportieren.
Durch Druck auf die Taste wird das FF "Tastenspeicher" gesetzt, das bei CT2 (letzte Spalte) gelöscht wird.
Wird die Taste losgelassen, steuert das Signal $\overline{\text{Taste}}$ die Warnerkupplung an, wodurch die Nocken C1, C2 und C3 drehen.
Gleichzeitig wird V' gesetzt, welches mit C1 den Zeitvergleich startet.

Während des Zeitvergleiches wird:

- Die Spaltensprungspule direkt angesteuert.
- das FF "mech. Ablage" gesetzt, das bis CT 2 gehalten wird. Während dieser Zeit ist die Tabulationsspule erregt.

Der Zeitvergleich findet nur einmal zu Beginn statt, da V' durch den Tastenspeicher bis CT2 gehalten wird. Die FF "Anfang Zeitvergleich" und "Ende Zeitvergleich" bleiben hierdurch auf 1. Während der gesamten Ablage brennt die Lampe PC.

c) Während des Stanzens

Das Zeichen, das sich in diesem Moment in K2 befindet, wird in die erste Spalte der neuen Karte gestanzt. Darum ist es verboten, während des Stanzens die Ablagetaste zu drücken.

Stanzverbot I.P.

Das FF "Stanzverbot" (I.P.) wird gesetzt durch:

- Druck auf die Taste ON (S.T.)

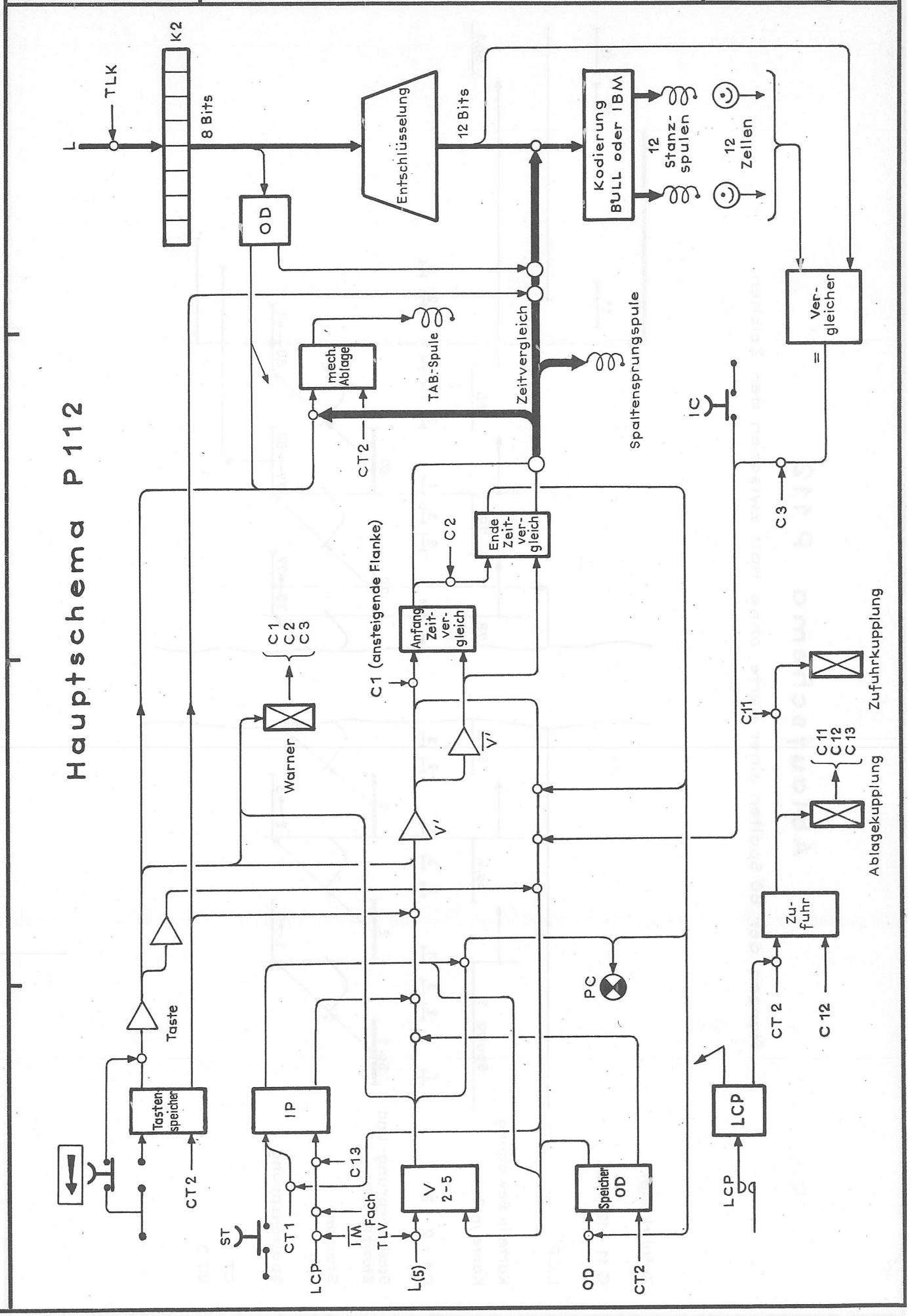
- CT (Trommelkontakt) in der letzten Spalte.

Es wird mit C13 gelöscht, wenn kein Fehler vorliegt. (IM, volles Fach, keine Zufuhr).

Wenn das FF "Stanzverbot" gesetzt ist, wird der Übertrag von V2,5 nach V' verhindert.

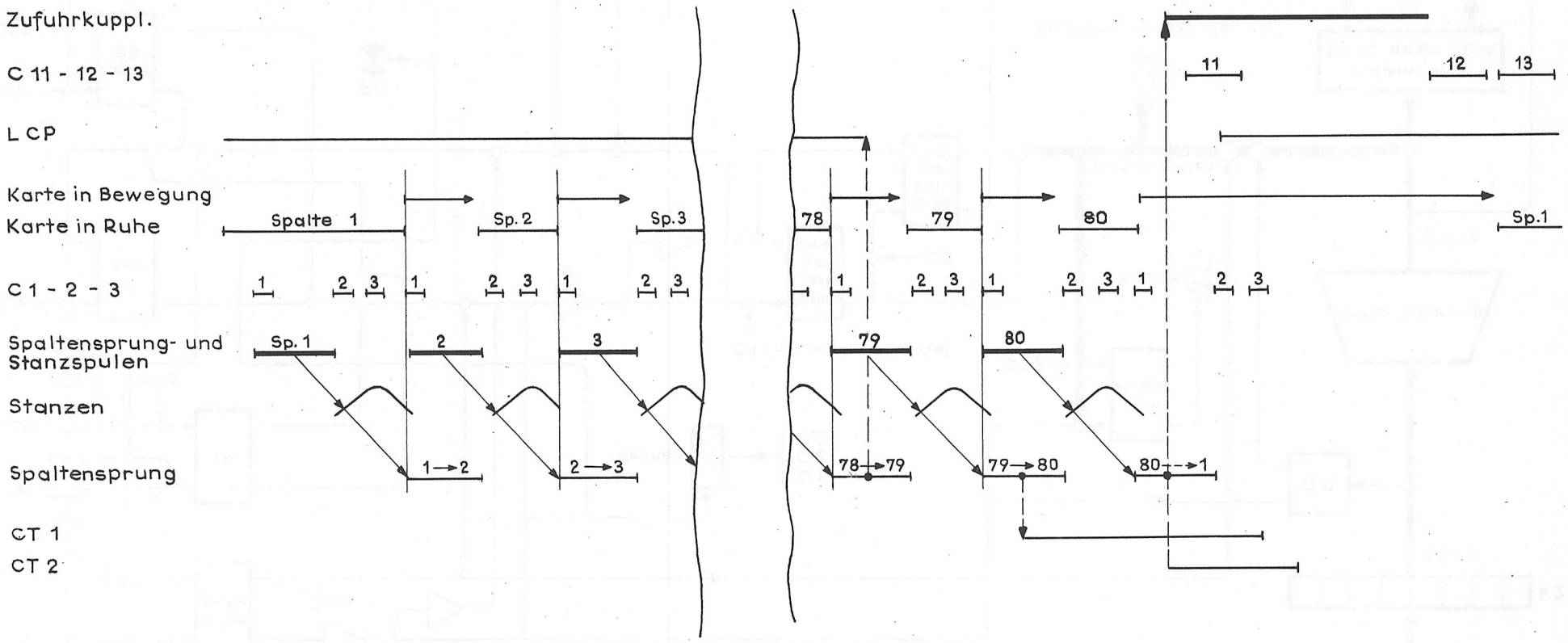
Im ersten Fall wird durch diese Einrichtung gewährleistet, daß zu Beginn der Arbeit (Einschalten der Maschine) eine Ablage von Hand erfolgen muß. Wodurch mit Sicherheit eine Karte mit Spalte 1 unter der Stanzstation liegt.

Im Falle eines vollen Ablagefaches oder leeren Magazins, wird die Karte, die gerade bearbeitet wird, nicht gestört. Die Maschine hält unmittelbar nach der nächsten Zufuhr. Im letzten Fall brennt die Lampe CP nur in Spalte 1.


Hauptschema P112
L
TLK
K2
8 Bits
OD
Entschlüsselung
12 Bits
Kodierung BULL oder IBM
12 Stanz-spulen
12 Zellen
Ver-gleicher
=
mech. Ablage
TAB.-Spule
Zeitvergleich
Spaltensprungspule
CT2
IC
C3
Ende Zeit-ver-gleich
C2
C1 (ansteigende Flanke)
Anfang Zeit-ver-gleich
C1 C2 C3
Warner
V'
V'
C11
Zufuhrkupplung
C11 C12 C13
Ablagekupplung
Zu-führ
PC
CT 2
C 12
Taste
Tasten-speicher
CT 2
IP
C13
ST
CT1
LCP
TM
Fach
TLV
V 2-5
L(5)
Speicher OD
OD
CT2
LCP
LCP

Ablaufschema P 112

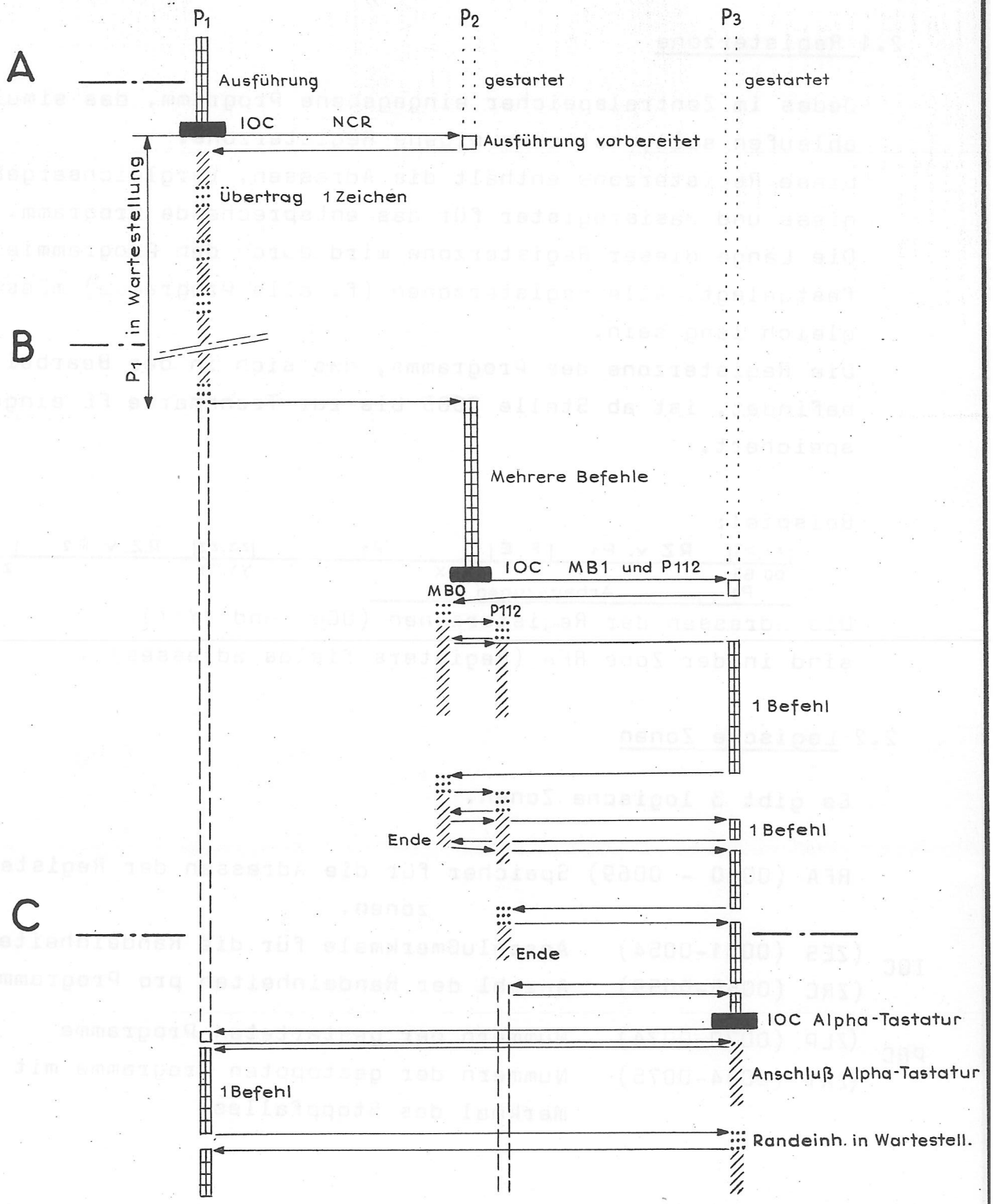
Stanzen der 80 Spalten einer Karte ohne Halt zwischen den Zeichen.



Kartenstanzer P112 A

1. Prinzip

3 Programme (P1, P2 und P3) sind eingespeichert. P1 steht in Stelle 0070, die beiden anderen sind ebenfalls gestartet. Ihre Nummern stehen in 0071 und 0072.



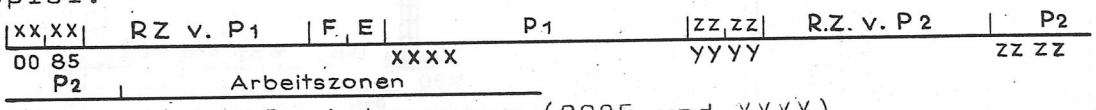
2. Spezialzonen im Zentralspeicher

Es gibt 2 Spezialzonen: die Registerzone und die logische Zone.

2.1 Registerzone

Jedes im Zentralspeicher eingegebene Programm, das simultan ablaufen soll, hat eine eigene Registerzone.
Diese Registerzone enthält die Adressen, Vergleichsergebnisse und Basisregister für das entsprechende Programm.
Die Länge dieser Registerzone wird durch den Programmierer festgelegt. Alle Registerzonen (f. alle Programme) müssen gleich lang sein.
Die Registerzone des Programms, das sich in der Bearbeitung befindet, ist ab Stelle 0085 bis zur Trennmarke FE eingespeichert.

Beispiel:



Die Adressen der Registerzonen (0085 und YYYY) sind in der Zone RFA (Registers fields adresses) .

2.2 Logische Zonen

Es gibt 3 logische Zonen.

RFA (0060 - 0069) Speicher für die Adressen der Registerzonen.

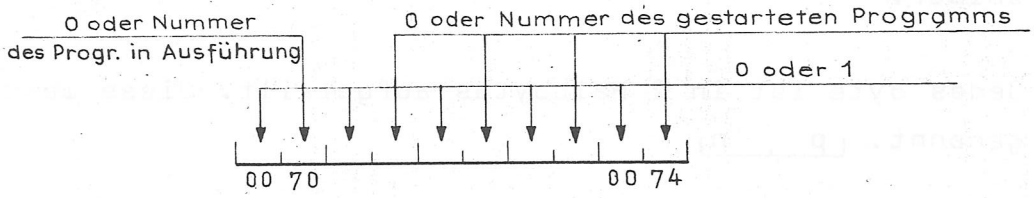
IOC (ZES (0031-0054) Anschlußmerkmale für die Randeinheiten.
(ZRC (0055-0059) Anzahl der Randeinheiten pro Programm.

PRC (ZLP (0070-0074) Nummern der gestarteten Programme
(ZAT (0084-0075) Nummern der gestoppten Programme mit dem Merkmal des Stoppfalles.

PRC-Zone (Programs control)

Diese Zone besteht aus zwei Teilen:

- 0070 - 0074 (ZLP) Die Programmstartzone besteht aus 5 Bytes. Im ersten Byte steht die Nummer des Programmes, das sich in Ausführung befindet. Im zweiten Byte (0071) steht die Nummer des nächst gestarteten Programmes.



Das linke Halbbyte enthält immer 1, wenn das entsprechende Programm das zuletzt gestartete oder wieder gestartete ist.

Beispiel: Schnitt A in Kap. 1.

0	1	0	2	1	3	0	0	0	0

00 70 ... 00 74

Schnitt B in Kap. 1

0	2	1	3	0	0	0	0	0	0

00 70 ... 00 74

Schnitt C in Kap. 1

0	3	1	1	0	0	0	0	0	0

Man kann max. 5 Programme starten und simultan verarbeiten.

- 0084-0075 die Programmwartezone besteht aus 10 Bytes. Diese Zone enthält die Nummern der Programme, die durch die Befehle PRD oder STOP gestoppt wurden. Diese Programme werden durch die Befehle FREE und START wieder gestartet.

Diese Zone kann bei Bedarf nach links erweitert werden, soweit in der dortigen Zone ZLP freie Stellen sind. Es muß aber wenigstens 1 Leerstelle zwischen dem letzten Zeichen ZLP und dem letzten Zeichen ZAT sein.

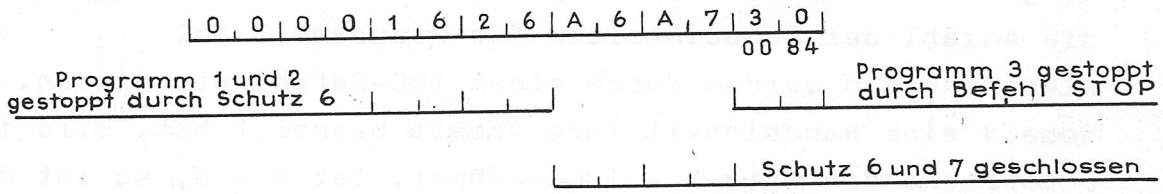
Die Benutzung der Zone ZAT erfolgt von rechts nach links. Die einzelnen Bytes haben keine spezielle Aufgabe, sie werden eines nach dem anderen besetzt, entsprechend dem Programmablauf.

Jedes Byte ist in 2 Halbbytes aufgeteilt. Diese werden p und n genannt. ⌊p , n⌋

Folgende Kombinationen sind möglich:

p = 0 + n = 0	Trennmarke Ende ZAT
p ≠ 0 + n = 0	p gibt die Nummer eines durch den Befehl STOP angehaltenen Programms an.
p = A + n = 0	n gibt die Nummer eines Schutzes an, der geschlossen ist.
p ≠ 0 + n = 0 ≠ A	n gibt die Nummer eines Schutzes an, durch den das Programm gestoppt wurde, dessen Nummer in p steht.

Beispiel:



Wenn die Leerstelle zwischen ZAT und ZLP in der Zone PRC nicht vorhanden ist, wird der laufende Befehl noch beendigt. Danach geht die Maschine in IP und das Fehlerprogramm läuft ab.

IOC-Zone (Input-Output-Control)

Diese Zone besteht aus 2 Teilen:

- 0031 - 0054 enthält die Merkmale für die Randeinheiten. (s.Technische Beschreibung, Band 2)

- 0055 - 0059 enthält die Anzahl der anzuschließenden Randeinheiten.
  Dieser Teil besteht aus 5 Bytes, von denen jedes mit einem Programm korrespondiert.
  Stelle 55 = Programm 1, Stelle 56 = Programm 2 usw..
  Für jedes Programm steht in diesen Bytes die Anzahl der Randeinheiten, deren Arbeit noch nicht beendet ist.
  Das linke Halbbyte enthält F, für die Programme, in denen wenigstens 1 Randeinheit angeschlossen wird.

N_1 gibt die Anzahl der Randeinheiten für Programm 1 und N_2 die Anzahl der Randeinheiten für Programm 2 an.
Diese Angaben werden durch einen IOC-Befehl eingegeben.
Sobald eine Randeinheit ihre Arbeit beendigt hat, wird in dem entsprechenden Byte N - 1 gerechnet. Ist N = 0, so ist der IOC-Befehl beendet und das entsprechende Programm wird wieder gestartet.

Beispiel: Schnitt C in Kap. 1

| F | 0 | F | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

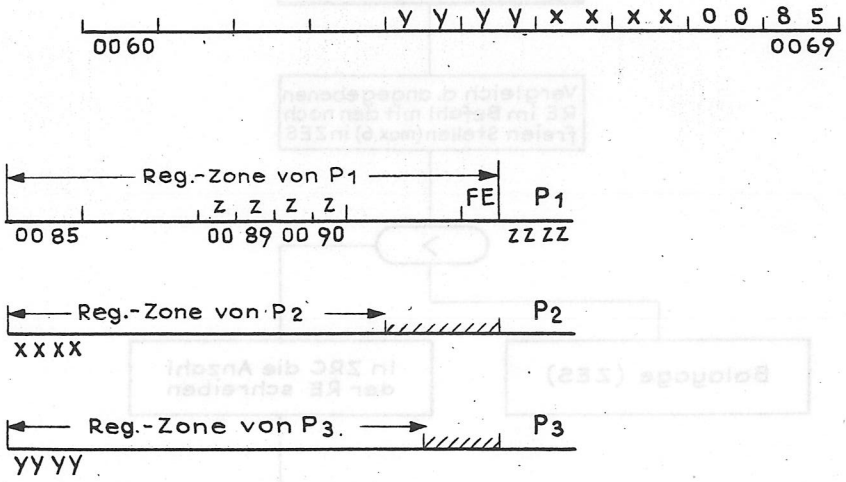
P1 hat keine Randeinheit mehr anzuschließen, es ist also wieder gestartet.
P2 hat noch eine Randeinheit, die nicht ihre Arbeit beendet hat.
P3, P4 und P5 haben keine Randeinheiten anzuschließen.

REA-Zone (Registers fields adresses)

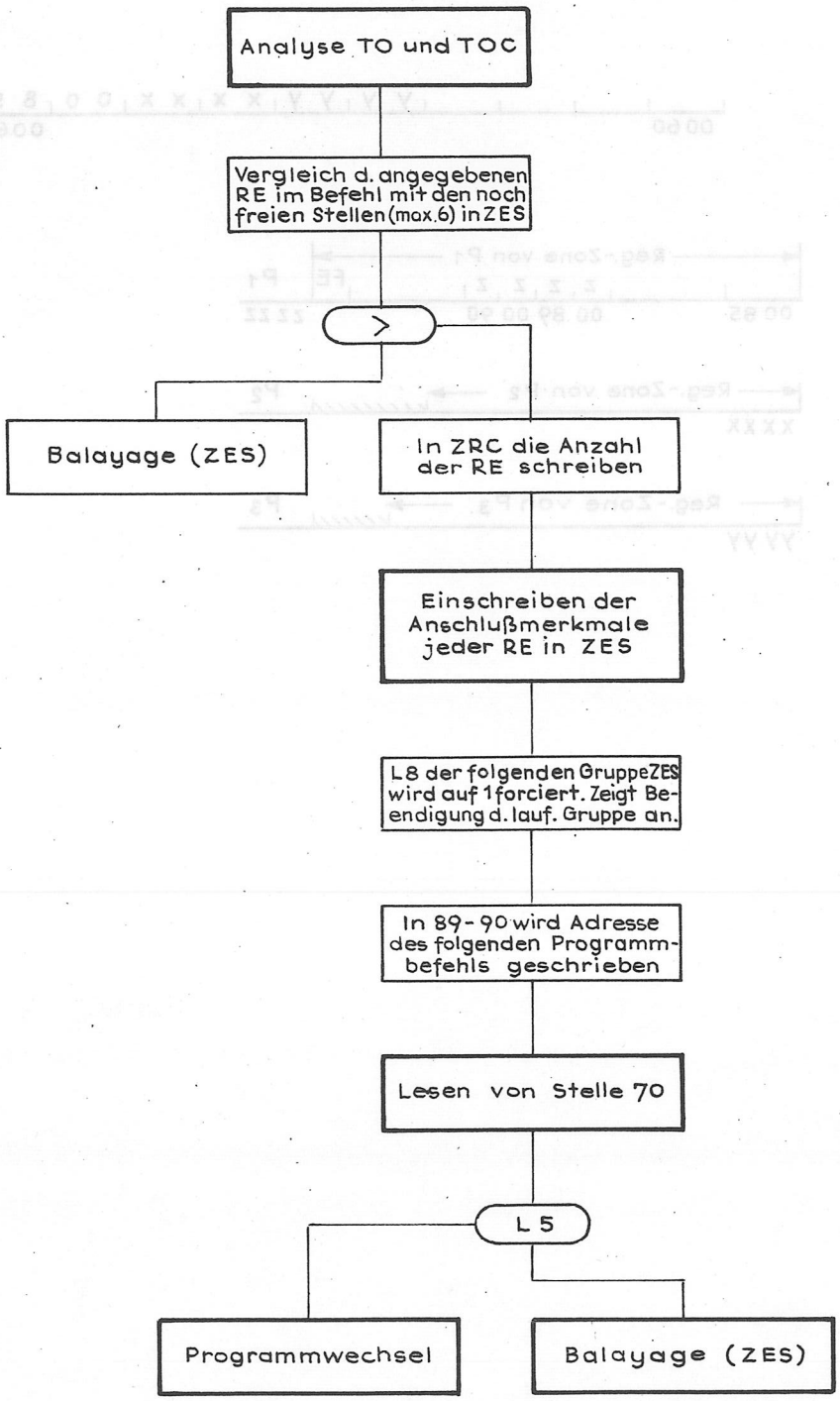
Diese Zone besteht aus 10 Bytes und enthält die Realadressen der Registerzonen für maximal 5 Programme .

Die Adresse des Programmes, das gerade bearbeitet wird, befindet sich in Stelle 68 - 69.

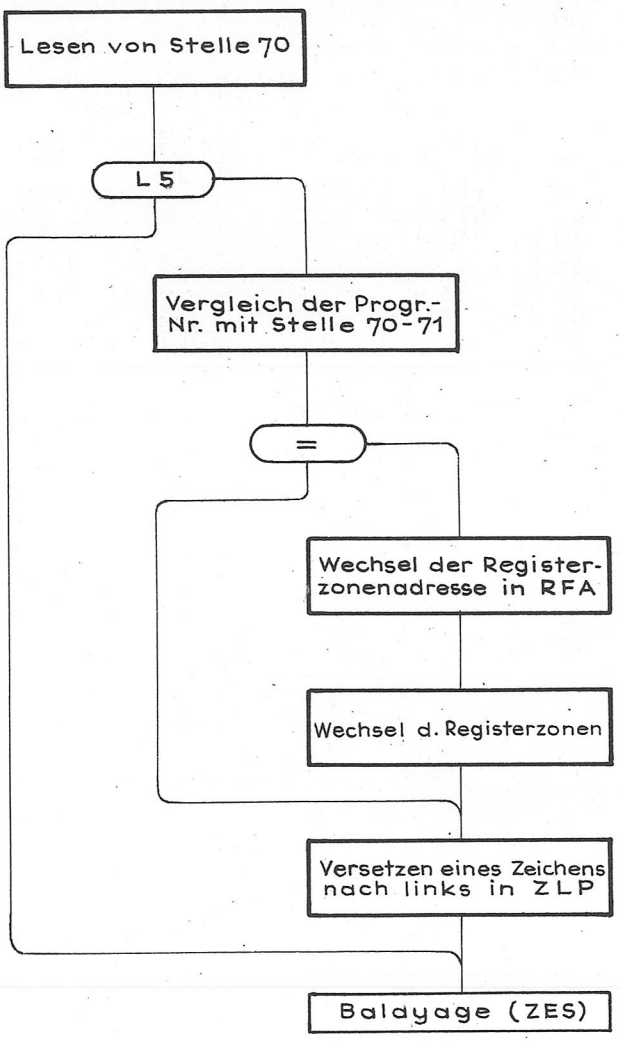
Punkt A Kapitel 1



IOC - Befehl

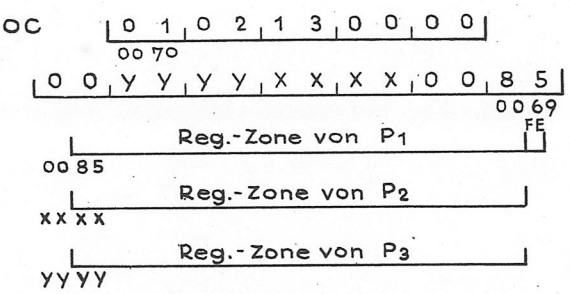

Analyse TO und TOC
Vergleich d. angegebenen RE im Befehl mit den noch freien Stellen (max.6) in ZES
>
Balayage (ZES)
In ZRC die Anzahl der RE schreiben
Einschreiben der Anschlußmerkmale jeder RE in ZES
L8 der folgenden Gruppe ZES wird auf 1 forciert. Zeigt Beendigung d. lauf. Gruppe an.
In 89-90 wird Adresse des folgenden Programmbefehls geschrieben
Lesen von Stelle 70
L 5
Programmwechsel
Balayage (ZES)

Programmwechsel



Beispiel: Kapitel 1 Punkt 1

Vor Bearbeitung der IOC



Nach Bearbeitung der IOC