



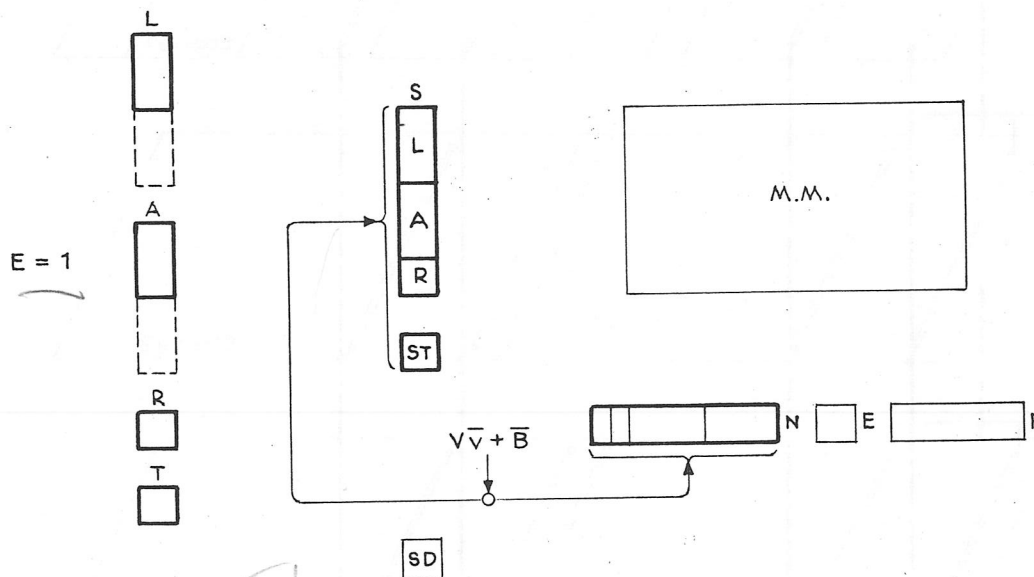
### Programmsteuerung

Es gibt 4 Möglichkeiten (Codes) von 0 - 3. Diese vier Möglichkeiten werden binär dargestellt und ausgewertet (00 - 01 - 10 - 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 valide*

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

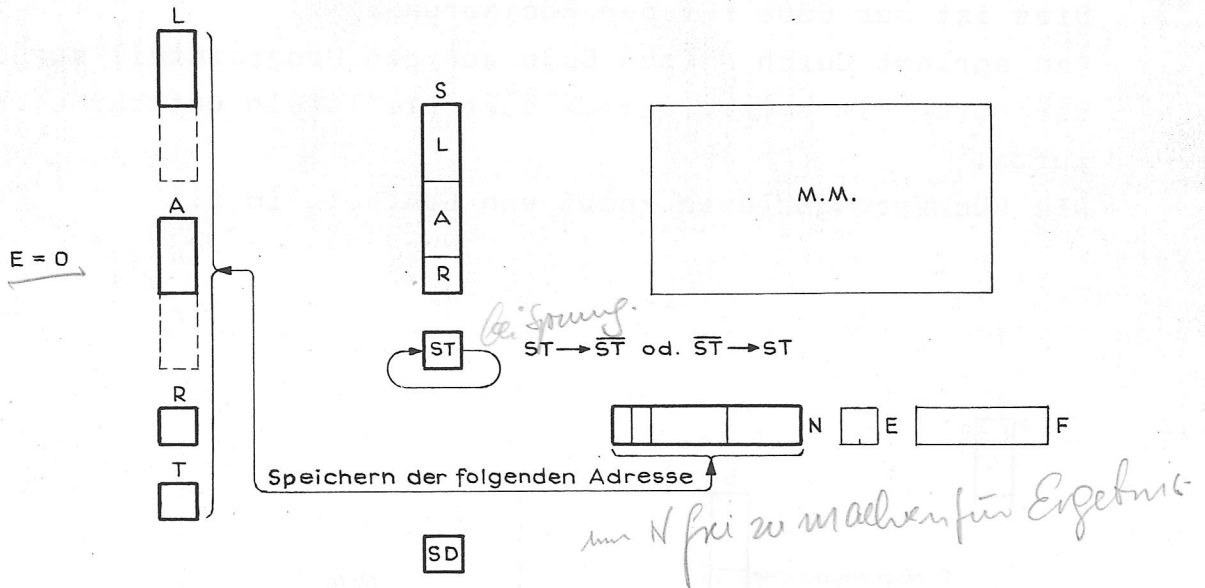
$\overline{B}$  = keine Randeinheit der Klasse II angeschlossen.

#### 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

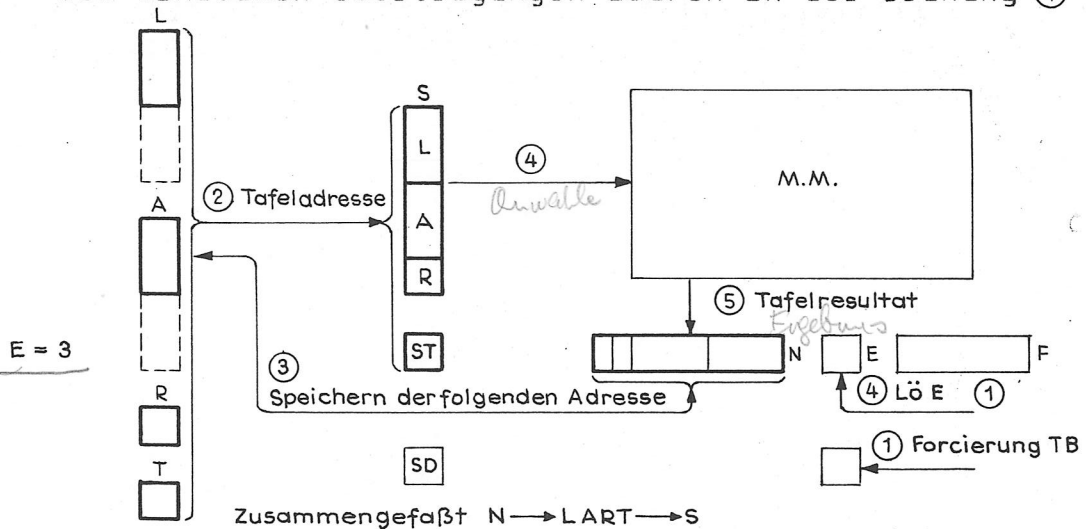
Code 3

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.

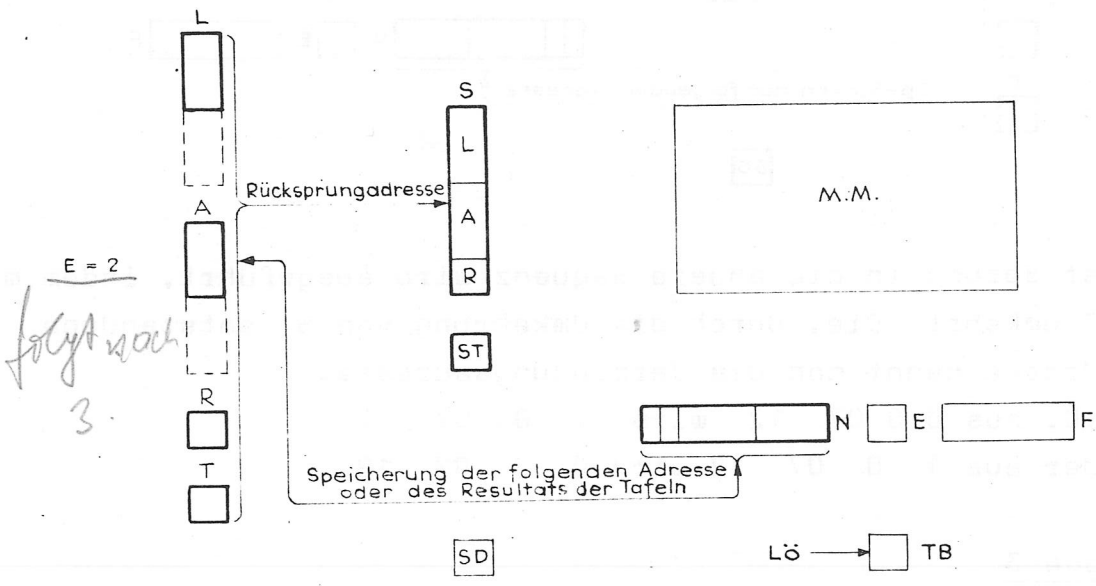


Code 2

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.



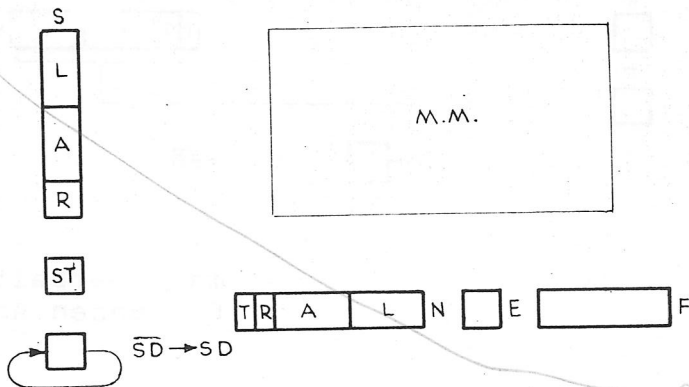
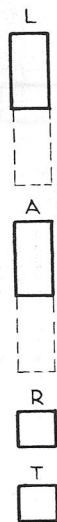
Sprünge

$V\bar{v}$  = Sprung ist ausgeführt.

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

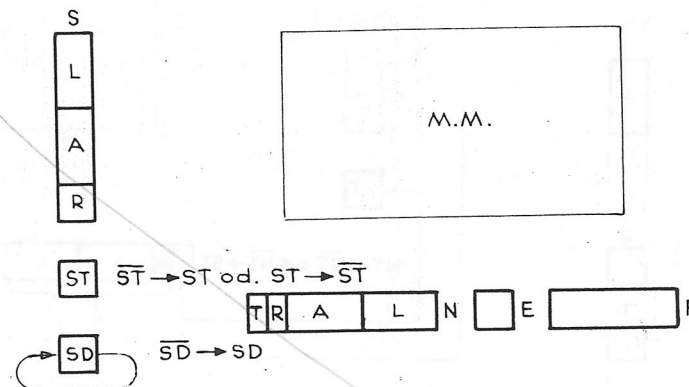
Wenn man einen Sprung in  $\mu F2$  hat, ist immer Enchainement 1.

$\bar{V}v$  in  $\mu F1$



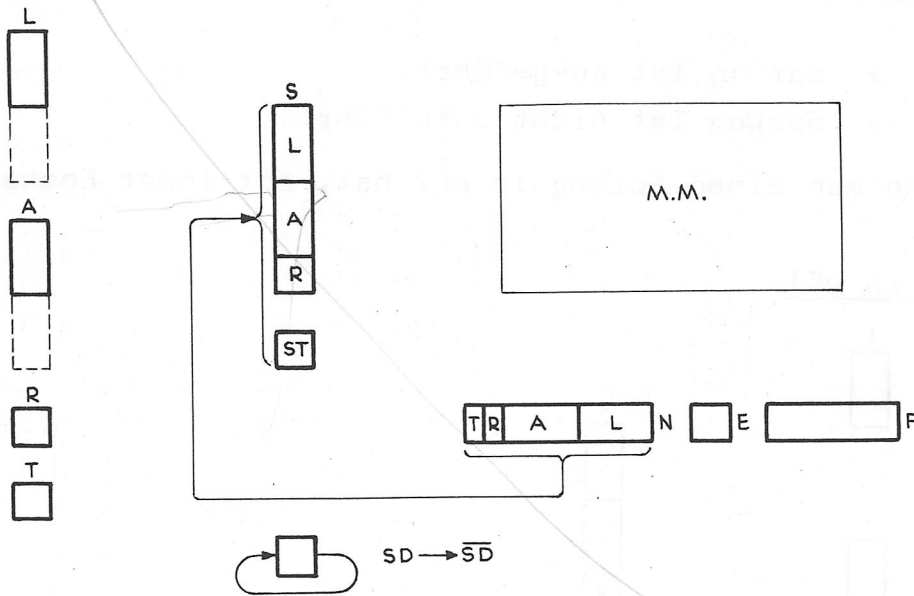
Man bearbeitet als nächstes die  $\mu F2$  dieser Adresse.

$Vv$  in  $\mu F1$



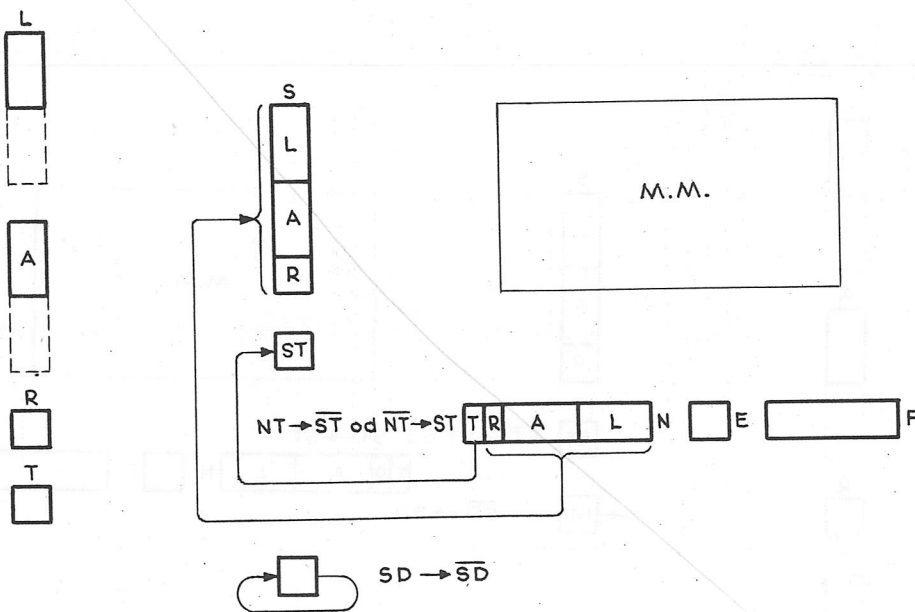
Man bearbeitet die  $\mu F2$  der Sprungadresse.

$\bar{V}v$  in  $\mu F2$



Man bearbeitet die  $\mu F1$  der folgenden Adresse.

$Vv$  in  $\mu F2$



Man bearbeitet die  $\mu F1$  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 <sup>E3</sup> 3 sein. Der Code 3 setzt das Flip-Flop <sup>zur Tafelgerichte</sup> 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.

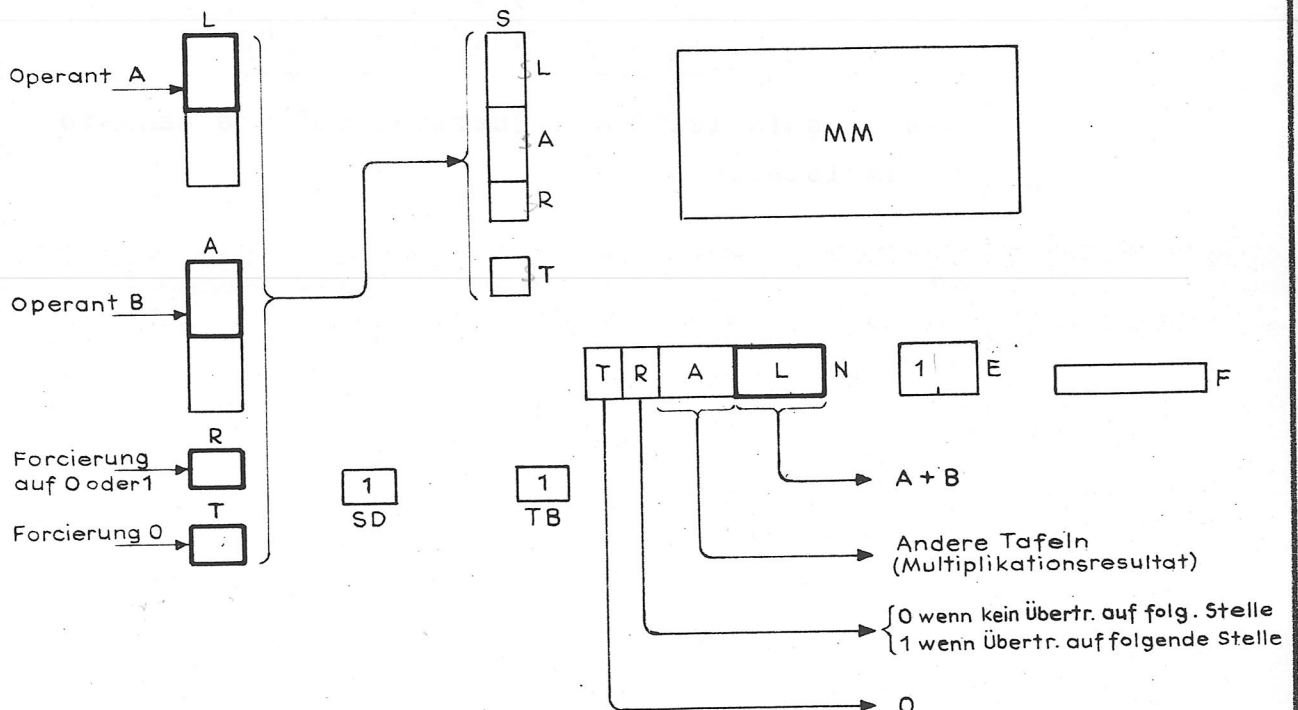
Rechentafeln

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  $\rightarrow$  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  $\rightarrow$  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

T R A L

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

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

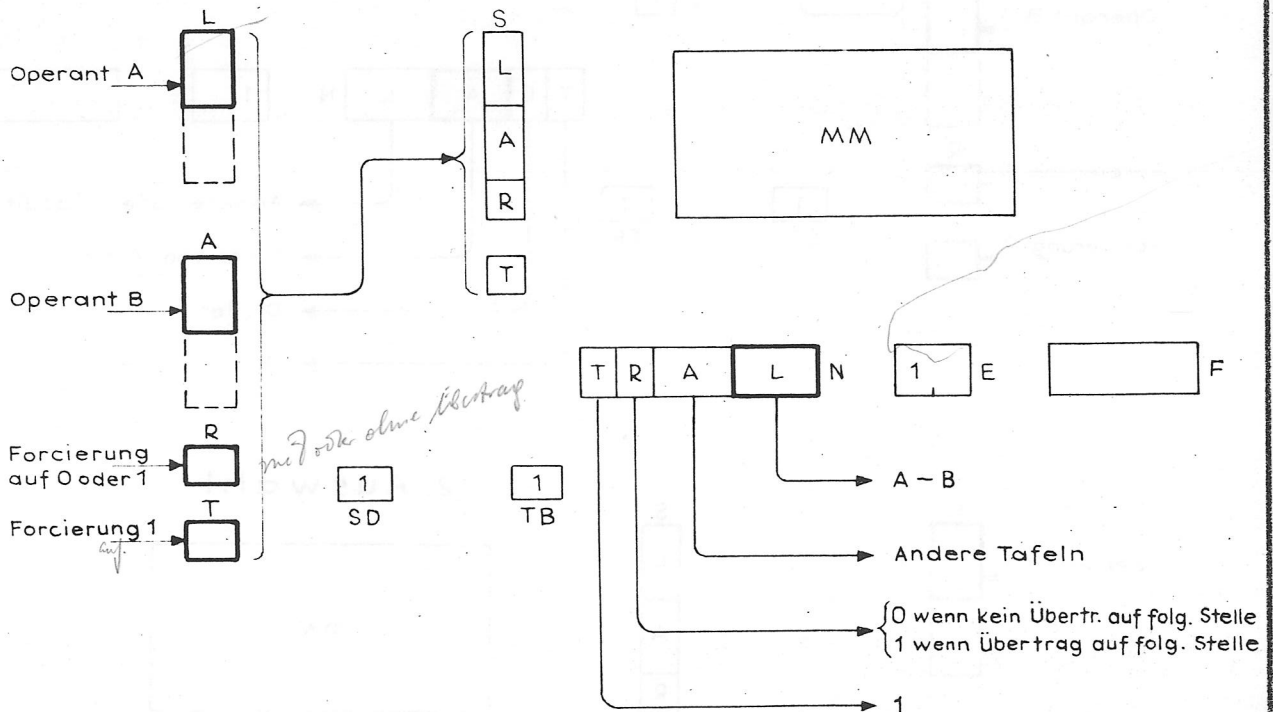
NT = 0, NR = 1, NA = 06, NL = 00

Das Ergebnis ist <sup>Repr.</sup> 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.

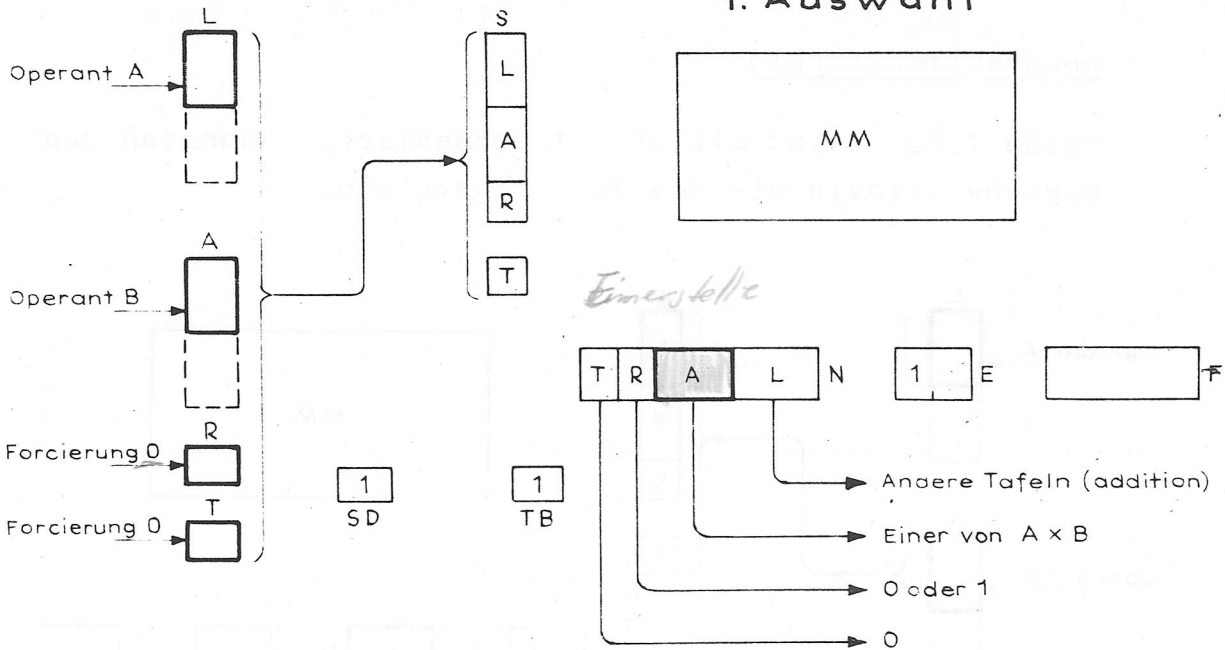
Um dieses, eventuell zweistellige, Ergebnis zu erhalten, werden 2 Tafeln angerufen. Sie sind beide mit ST = 0 adressiert.

Die mit SR = 0 adressierte Tafel gibt den Wert der Einerstelle des Produktes an.

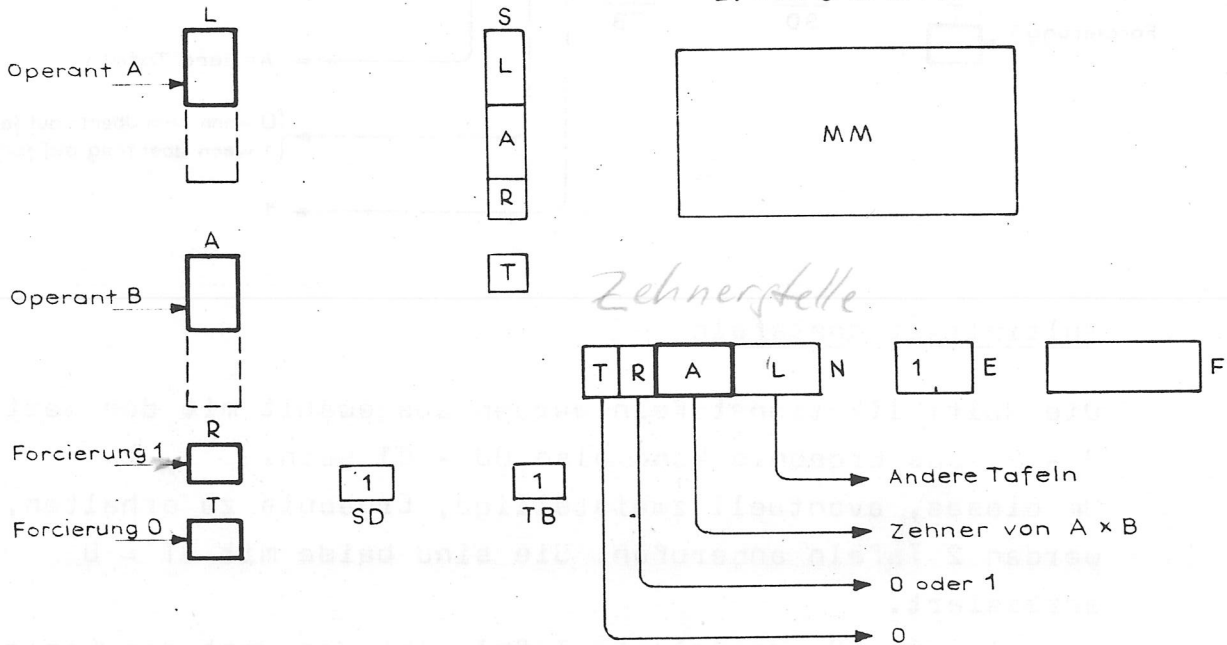
Die mit SR = 1 adressierte Tafel gibt den Wert der Zehnerstelle des Produktes an.

*bei Multiplikation grundsätzlich werden beide Tafeln gebraucht.*

1. Auswahl



2. Auswahl



Beisp.: Siehe Auszug der Additionstafel (S.112)  $7 \cdot 2$

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. (*Unverdichten*  
*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.

Symbol	Format 1		Format 2		num. Wert
	ISO 8 Bits		Hexa- dezimal	Binär (4 Bits)	
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.

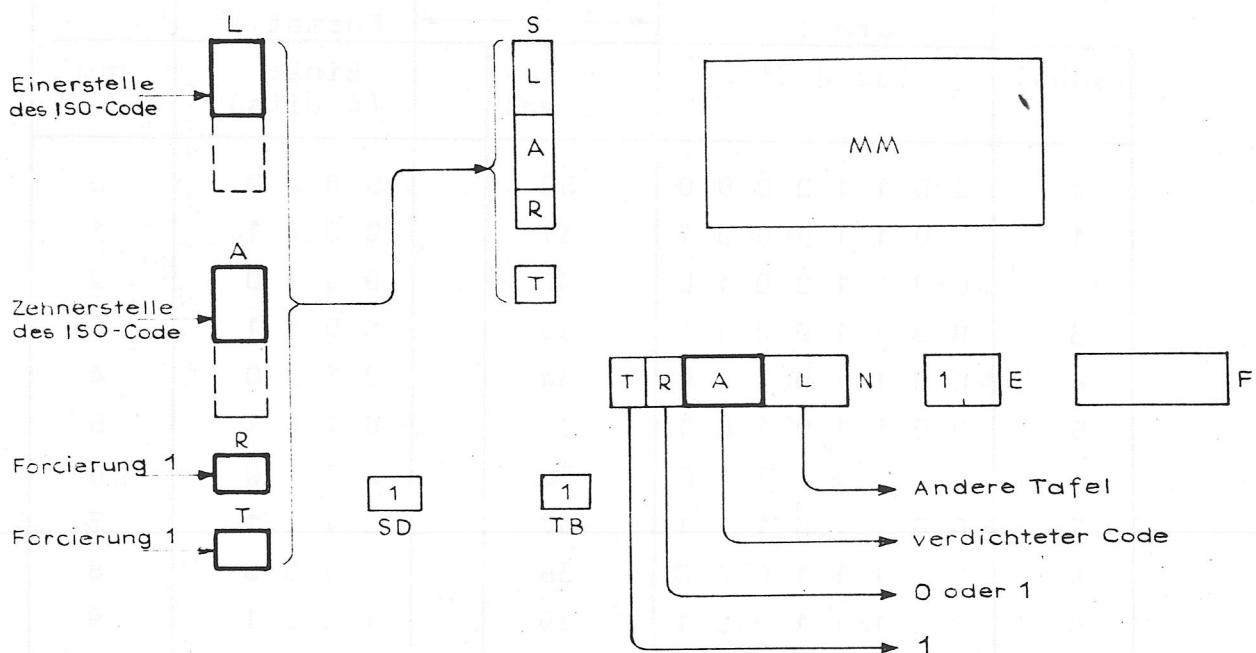
Diese Tafel ist adressiert mit ST = 1 und SR = 1.

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) (20-40)

Das verdichtete Resultat befindet sich in NA.



SA	SR
1	1

Tafelauszug

SA	SR	03	04	05
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).

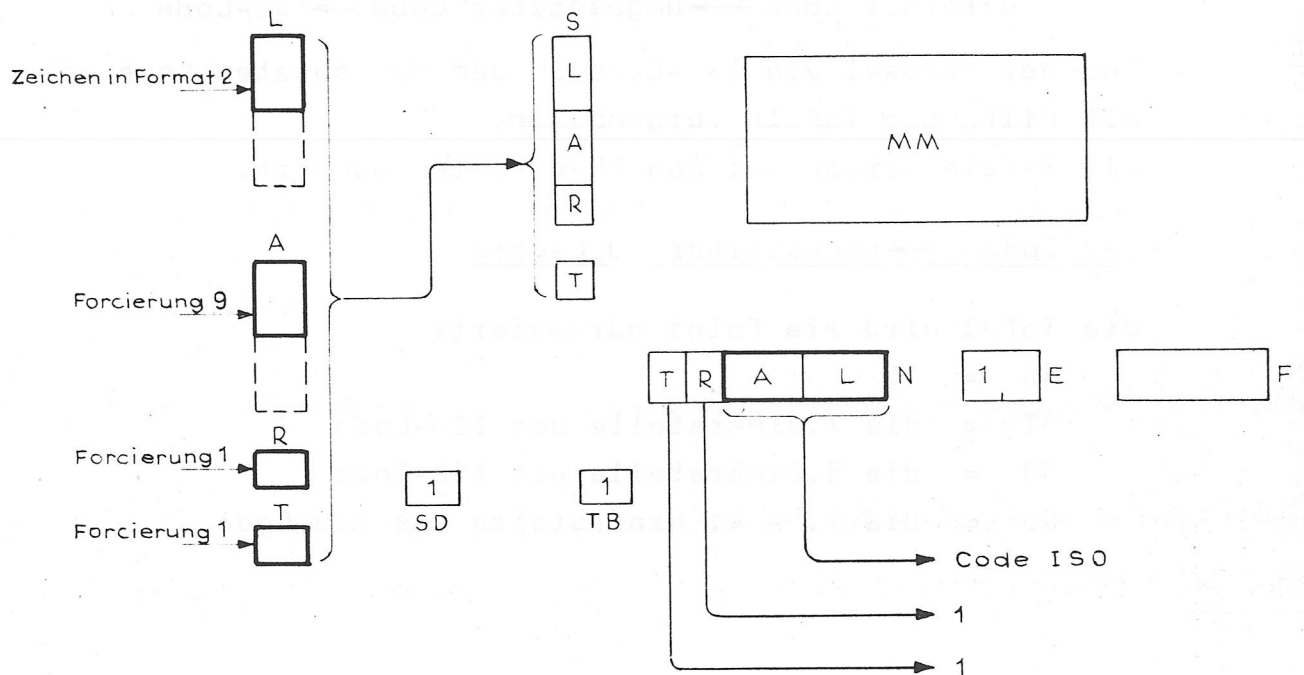
Sie ist adressiert mit ST = 1, SR = 1 und SA = 9.

In SL stehen die Werte 0 - 15, welche den auszugebenden Code bestimmen.

Die Ziffer 0 ergibt immer 30, also niemals wieder Spalten-sprung (20) wie im umgekehrten Fall.

Das Ergebnis steht in NA und NL.

*2 Halbbyts = 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

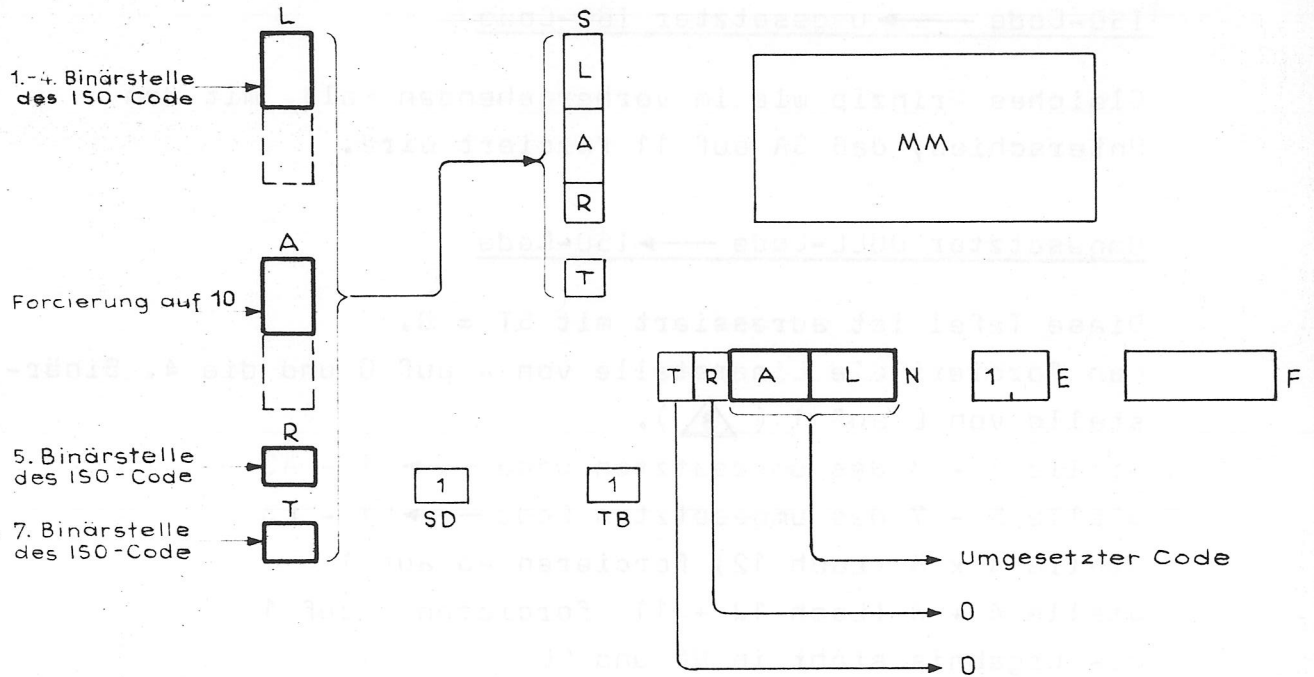
Nur der Wechsel von ISO-Code in den umgesetzten Code wird mit Hilfe der Tafeln vorgenommen.

Die Tafeln werden bei den TR-Befehlen benutzt.

*Transcode*  
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	SA	09	10	11
1	0				
	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 (  $\triangle 8$  ).

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	SA	04	05	06	07	08	09	0A	0B
0	0	10	00 02 02	00 02 05	00 02 06	00 04 07	00 04 08	00 05 09	00 05 0A	00 05 0B
	<u>SL</u>	11	00 04 05	00 04 06	00 04 07	00 04 08	00 04 09	00 05 0A	00 05 0B	00 05 0C
		12	00 02 11	00 05 13	00 05 14	00 05 15	00 05 16	00 05 17	00 05 18	00 05 19
		13	00 04 14	00 05 00	00 05 01	00 05 02	00 05 03	00 05 04	00 05 05	00 05 06

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. *J. Regiolo 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.  $(5 \cdot n) + 100$ .  
 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	SA	0	1	2
1	0				
	SL	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.

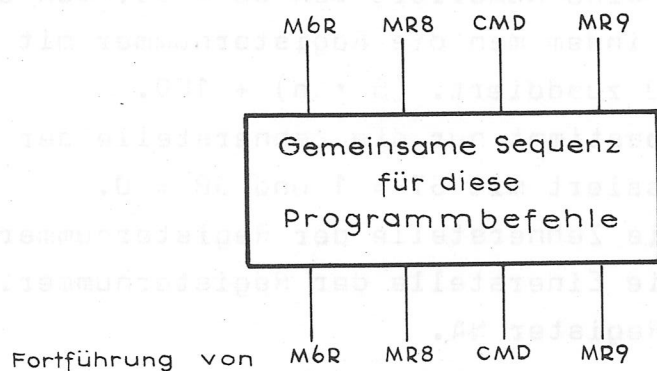
- Errechnen der Hunderterstelle durch Multiplikation:  
 $5 \cdot 20 = 0100 + 100 = 0200$
- Errechnen der Zehnerstelle durch die Tafel:  
 ST = 1, SR = 0, SA = 2, und SL = 1 ergibt in NA = 00.
- Bestimmen der Einerstelle:  
 wenn Registernummer gerade, Einerstelle = 0  
 wenn Registernummer ungerade, 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 *separanz zum Rückschreiber Adresse v. LART → S*

Es gibt außerdem noch *= uF makt Inhalt Reg F = 0* 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.