

H. J. Schilling  
7750 Konstanz  
Moosbruggerstr. 17 • Tel. 5709

Inhalt

03. Juli 1967

Ku



Tr

Fk  
z  
L

# TR 4

Unterlagen für den Programmierer

Gk  
D<sub>2</sub>  
Log. A<sub>x</sub>

SE  
Aufbau,

Ab

Int.  
Ext.-  
Code

Sp

2

0,5

C

LS

Lk

E/A

Ku

Tr

**Vorwort**

Die vorliegende Sammlung soll dem **angehenden Programmierer**, der sich mit Hilfe des Programmierhandbuchs in die Programmierung der Digital-Rechenanlage TR 4 einarbeitet, schnell zur notwendigen **Übersicht** verhelfen und dem **ausgebildeten Programmierer** alle zur Arbeit benötigten grundlegenden **Informationen zugriffsbereit zur Verfügung stellen**. Zu diesem Zweck wird auf weniger wichtige Einzelheiten bewußt verzichtet, um die wesentlichen Dinge um so deutlicher in Erscheinung treten zu lassen. Insbesondere werden alle Sachverhalte weggelassen, die sich bei grundsätzlicher Kenntnis der Programmierung und des Aufbaus des Rechners von selbst verstehen. Der Stoff ist in Tabellenform zusammengefaßt, soweit es irgendwie angängig ist. Verbale Formulierungen, die sich nicht umgehen ließen, sind so knapp wie möglich gehalten.

Fk  
z  
LGk  
D.  
Log. ASE  
Aufber.

Ab

Int.  
Ext.-  
Code

Sp

2

0,5

C

LS

Lk

EA

Kurzbeschreibung der Befehle · Bedeutung der verwendeten Zeichen

Ku

Transportbefehle

Tr

Setz- und Löschbefehle · Arithmetische Befehle (dual, Festkomma)

Fk  
Z  
L

(dual, Gleitkomma) · (logisch, dezimal, Halbwort, Index)

Gk  
D. 2  
Log. A.

Aufbereitungsbefehle/Sprungbefehle

Tabellenbefehle/Ersetz- und Modifizierbefehle

SE  
Aufber.

Alphabetische Liste der Befehle und allgemeine Eigenschaften

Ab

Interncode - Externcode · Zulässige Zweitcodes

Int.  
Ext.-  
Code

Intern-Spezifikationen

Sp

Konvertierungstafeln · Potenzen von 2

2

Potenzen von 0,5

0,5

TR 4 - Hexadencode · Lochstreifencodes

C

Internationaler Fernschreibcode CCIT Nr. 2  
Friden - Flexowriter Code NC-1

LS

Lochkartencodes · Übertragung Lochkarte ↔ Speicher

Lk

Verkehr mit dem Verteilerprogramm

EA

## Kurzbeschreibung der Befehle

Ku

Diese Liste enthält die Befehle in möglichst systematischer Anordnung; soweit sich Überschneidungen ergeben, treten einige Befehle mit Rückverweis doppelt auf (BA, BAV, AA und SBA).

Der gesamte TR 4-Befehlsvorrat (mit Ausnahme der dem Vertellerprogramm vorbehaltenen Spezialbefehle, insbesondere Startbefehle) ist gegliedert in:

### Gruppe 1: Transportbefehle

- .. 2: Setz- und Löschenbefehle
- .. 3: Arithmetische Befehle
- .. 4: Aufbereitungsbefehle
- .. 5: Sprungbefehle
- .. 6: Tabellenbefehle
- .. 7: Modifizier-, Ersetz- und Adressierbefehle.

In der Spalte „**Adr.**“ ist der Adressencharakter folgendermaßen bezeichnet:

N = Adresse einer ganzen Arbeitsspeicherzelle

N<sup>2</sup> = Adresse einer halben Arbeitsspeicherzelle

N\* = Wahladresse

I bzw. J = Adresse einer Indexspeicherzelle, im Erst- bzw. Zweitadreßteil

I<sup>P</sup> bzw. J<sup>P</sup> = Zahlwert (Parameter) im Erst- bzw. Zweitadreßteil

I<sup>S</sup> bzw. J<sup>S</sup> = Spezifikation im Erst- bzw. Zweitadreßteil

J<sup>C</sup> = Zweitcode.

Außerdem ist ersichtlich, ob Erst- und Zweitadreßteil im EXTERNCODE durch Doppelpunkte voneinander zu trennen sind, nämlich bei allen Doppeladreßbefehlen außer SH und Doppelcodebefehlen.

In der Spalte „**Wirkung**“ sind alle Register oder Speicherzellen angegeben, deren Inhalt durch den betreffenden Befehl verändert wird oder verändert werden kann. Nur in besonderen Fällen wird ausdrücklich erwähnt, was erhalten bleibt. Es werden nur die Wirkungen mit Alternativen erwähnt, die bei normaler Benutzung der Befehle eine Rolle spielen.

Die angegebenen Einschränkungen werden im Sinne einer möglichst übersichtlichen Darstellung geringfügig verschärft, soweit dadurch die Bewegungsfreiheit des Programmierers nicht eingeschränkt wird.

### Bedeutung der verwendeten Zeichen:

l r	rechtsbündig bzw. linksbündig	0,5
<...> <...> <sub>2</sub> <...> <sub>3</sub>	Inhalt von, Halbwort bzw. Drittewort aus (= Adreßteil aus...)	
n i j	Inhalt einer Speicherzelle N bzw. Indexspeicherzelle I oder J	C
a h d q b	Inhalt von AC, HR, MD, MQ bzw. BA	
M <sub>N</sub> bzw. M <sub>C</sub>	Marke in der Speicherzelle N bzw. in CR	
t <sub>N</sub> t <sub>A</sub> t <sub>H</sub> t <sub>D</sub> t <sub>Q</sub>	TK in Speicherzelle N, in AC, HR, MD bzw. MQ	LS
t <sub>max</sub>	größte beteiligte TK	
→ ⇒ ⇌	geht nach, ergibt, Vertauschung von...	

Die angegebenen Inhalte und TK geben im allgemeinen den Zustand vor Ausführung des Befehls an, insbesondere wenn sie am Anfang eines

Tr

Fk  
ZL

Gk  
D. 2  
Log. Ax

SE  
Aufba

Ab

Int.  
Ext.-  
Code

Sp

2

C

LS

Lk

EA

Pfeiles stehen. Am Ende eines Pfeiles stellen sie dagegen immer den Zustand nach Ausführung des Befehls dar.

Im Zweifelsfall wird ein Index „alt“ bzw. „neu“ hinzugefügt.

$\Pi$   $\Sigma$   $\Delta$  Produkt, Summe bzw. Differenz als Ergebnis einer Rechenoperation.

I II Index zur Bezeichnung des vorderen bzw. hinteren Teiles einer doppelt langen Zahl.

0 bedeutet immer +0; —0 wird immer ausdrücklich angegeben, +0 nur dann, wenn es besonders betont werden soll.

„+“ in der Spalte „<CR>“ bedeutet, daß CR durch ein markiertes Zahlwort aus dem Speicher gesetzt wird.

„s“ in der Spalte „<YZ>“ bedeutet die Anzahl der Binärschifte bei einer Normalisierung nach links.

$\langle AC \rangle = 1, V, n_3$  bedeutet TK 1, erste 32 Informationsbits vorzeichengleich (= 33. Bit) besetzt, letzte 16 Bits dem Adreßteil einer Halbzelle entnommen.

Sondersymbole für die Gruppe 7 (Modifizier-, Ersetz- und Adressierbefehle) sind im Programmierhandbuch erläutert.

Verschiedene Wirkungen eines Befehls werden durch „;“ voneinander abgesetzt.

Bei komplizierten Befehlen werden die Wirkungen durch vorangeschickte, ebenfalls durch „;“ abgesetzte Definitionen erläutert.

## Kurzbeschreibung der Befehle

### 1. Transportbefehle

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung	$\langle N \rangle$	$\langle CR \rangle$	$\langle AC \rangle$	$\langle HR \rangle$	$\langle MD \rangle$	$\langle MQ \rangle$	Bemerkungen
1.01	B	N	Bringe (nach AC)	$n \rightarrow AC$	+	$t_N, n$					
1.02	BH	N	Bringe nach HR	$n \rightarrow HR$	+			$t_N, n$			
1.03	BD	N	Bringe nach MD	$n \rightarrow MD$	+			$t_N, n$			
1.04	BQ	N	Bringe nach MQ	$n \rightarrow MQ$	+					$t_N, n$	
1.05	BN	N	Bringe negativ	$-n \rightarrow AC$	+	$t_N, -n$					
1.06	BB	N	Bringe Betrag	$ n  \rightarrow AC$	+	$t_N,  n $					
1.07	BR	N	Bringe und reserviere	$a \rightarrow HR; n \rightarrow AC$	+	$t_N, n$		$t_A, a$			
1.08	BNR	N	Bringe negativ u. reserv.	$a \rightarrow HR; -n \rightarrow AC$	+	$t_N, -n$		$t_A, a$			
1.09	BQB	N	Bringe nach MQ u. bringe	$n \rightarrow AC; n \rightarrow MQ$	+	$t_N, n$				$t_N, n$	
1.10	BZ	N	Bringe zwei Wörter	$\langle N \rangle \rightarrow AC; \langle N+2 \rangle \rightarrow MQ$	+	$t_N, n$				$t_{N+2}, \langle N+2 \rangle$	
1.11	BZ2	N <sup>2</sup>	Bringe zwei Halbwörter	$\langle N_2, N+1 \rangle \rightarrow AC; O \rightarrow YZ;$ wenn N ungerade, q → MD	+	$t_N, \langle N_2, N+1 \rangle_2$				(t <sub>O, q</sub> )	
1.12	B2	N <sup>2</sup>	Bringe Halbwort	$0, n_2 \rightarrow AC$	+	$t_N, 0, n_2$					
1.13	B2V	N <sup>2</sup>	Bringe Halbw. m. Vorzeich.	$V, n_2 \rightarrow AC$	+	$t_N, V, n_2$					
1.14	B3	N <sup>2</sup>	Bringe Drittelwort	$0, n_3 \rightarrow AC$	+	$t_N, 0, n_3$					
1.15	B3V	N <sup>2</sup>	Bringe Drittelw. m. Vorzh.	$V, n_3 \rightarrow AC$	+	$t_N, V, n_3$					
1.16	BA	N*	Bringe Adreßteil	$0, N \rightarrow AC$			$t_A, a$				siehe 2.01
1.17	BAV	N*	Bringe Adr.teil m. Vorzh.	$V, N \rightarrow AC$			$t_A, a$				siehe 2.02
1.18	BAR	N*	Bringe Adr.teil u. reserviere	$a \rightarrow HR; 0, N \rightarrow AC$			$t_A, a$				
1.19	BAP	N*	Br. Adr.teil m. Vz. u. reserv.	$a \rightarrow HR; V, N \rightarrow AC$			$t_A, a$				
1.20	BC	N	Speichere und Speichere	$a \leftarrow n; M_N$ bleibt erhalten,	$t_A, a$	+	$t_N, n$				
1.21	C	N	Speichere (aus AC)	$a \rightarrow N$		$t_A, a$					
1.22	CH	N	Speichere aus HR	$h \rightarrow N$		$t_H, h$					
1.23	CD	N	Speichere aus MD	$d \rightarrow N$		$t_M, d$					
1.24	CQ	N	Speichere aus MQ	$q \rightarrow N$		$t_Q, q$					
1.25	CN	N	Speichere negativ	$-a \rightarrow N$		$t_A, -a$					
1.26	CB	N	Speichere Betrag	$ a  \rightarrow N$		$t_A,  a $					
1.27	CR	N	Speichere u. bringe Reserve	$a \rightarrow N; h \rightarrow AC$		$t_A, a$		$t_H, h$			
1.28	CZ	N	Speichere zwei Wörter	$a \rightarrow N; o \rightarrow N+2$			$t_A, a$			$t_O, q$	$\langle N+2 \rangle =$
1.29	C2	N <sup>2</sup>	Speichere Halbwort	$a_2 \rightarrow n_2$							
1.30	C3	N <sup>2</sup>	Speichere Drittelwort	$a_3 \rightarrow n_3$							
1.31	CMT	N	Speichere markiert	$a$ mit Marke $\rightarrow N$		$t_A, M, a$					Nur $t_A \leq 1$
1.32	CMR	N	Spch. m. Marke aus Reg.	$CR, AC \rightarrow N$		$t_A, M_C, a$					Nur $t_A \leq 1$
1.33	CMC	N	Spch. m. Marke aus Sach.	$a \rightarrow N; M_N$ bleibt erhalten		$t_A, M_N, a$					Nur $t_A \leq 1$ und $t_N \leq 1$
1.34	RT	I*	Register tauschen	Vertauschung der zwei durch A = AC, H = HR, D = MD oder Q = MQ bezeichneten Register; alle anderen Register bleiben erhalten.							
					$\langle N \rangle$	$\langle I \rangle$	$\langle J \rangle$	$\langle AC \rangle$	$\langle BA \rangle$		
1.35	XB	I	Index: Bringe	$i \rightarrow BA$							
1.36	XBA	N*	Index: Bringe Adreßteil	$N \rightarrow BA$							
1.37	XC	I	Index: Speichere	$b \rightarrow I$			$b$				
1.38	XCN	I	Index: Speichere negativ	$-b \rightarrow I$			$-b$				
1.39	TXX	J : I	Transport aus Indexspeicher in Indexspeicher	$i \rightarrow j; i \rightarrow BA$							
1.40	TTX	J : I	Tausch-Transport im Indexspeicher	$i \rightarrow j; j \rightarrow BA$							
1.41	TXA	I	Transpt. a. Indexsp. nach AC	$0, i \rightarrow AC; i \rightarrow BA$							
1.42	TXV	I	Trsp. a. lbsp. n. AC m. Vorz.	$V, i \rightarrow AC; i \rightarrow BA$							
1.43	TAX	I	Trsp. a. AC nach Indexsp.	$a_2 \rightarrow i; a_2 \rightarrow BA$							
1.44	TCB	N <sup>2</sup>	Trsp. a. Speicher nach BA	$n_3 \rightarrow BA$							
1.45	TBC	N <sup>2</sup>	Trsp. a. BA nach Speicher	$b \rightarrow n_2$							
1.46	CWL	N <sup>2</sup>	Speichere WS und ML	$WS, ML \rightarrow n_3; 0 \rightarrow ML$							
1.47	WTD	J <sup>2</sup> : I	Wortgruppen-Transport mit Dehnung	$ I + 2k  \rightarrow  I + 1  + 2k$ für $k = 0, 1, 2, \dots$ bis zum ersten Wort anderer TK oder mit Marke einschließlich; die Adresse der Zelle, in die dieses letzte Wort abgelegt wird, erscheint in BA; $1 \leq  J  \leq 127$							
1.48	WTK	J <sup>2</sup> : I	Wortgruppen-Transport mit Komprimierung	$ I + 2k  \rightarrow  I + 1  + 2k$ für $k = 0, 1, 2, \dots$ bis zum ersten Wort anderer TK oder mit Marke einschließlich; die Adresse der Zelle, in die dieses letzte Wort abgelegt wird, erscheint in BA; $1 \leq  J  \leq 127$							
1.49	WTZ	J <sup>2</sup> : I	Wortgruppen-Transport mit Zählung	$ I + 2k  \rightarrow  I + 1  + 2k$ für $k = 0, 1, 2, \dots (J-1);$ $ I + 1 + 2 \cdot (J-1)  \rightarrow BA;$ $1 \leq J \leq 255$							
1.50	HB	N	Höhere Genauigkeit: Bringe	$\langle N + 2k \rangle \rightarrow X17 \rightarrow N + 2k$ für $k = 0, 1, 2, \dots m$ ; Beendigung wie WTD, evtl. TK-Alarm; m ist im allg. die aus dem linken Adreßteil von $\langle N \rangle$ zu entnehmende Mantissenlänge; $L \rightarrow CR$ , wenn $\langle N \rangle$ markiert ist; $\langle BZ \rangle \rightarrow BA; 1 \leq m \leq 15$							Nur TK0
1.51	HC	N	Höhere Genauigkeit: Speichere	$\langle X17 + 2k \rangle \rightarrow N + 2k$ für $k = 0, 1, 2, \dots m$ ; Beendigung wie WTD, evtl. TK-Alarm; m ist im allg. die aus dem linken Adreßteil von $\langle X17 \rangle$ zu entnehmende Mantissenlänge; $L \rightarrow CR$ , wenn $\langle X17 \rangle$ markiert ist;							Nur TK0
1.52	QBR	N	Bequemes Bringen aller Register	$\langle N \rangle \rightarrow HR; \quad 2. Bit der TK von \langle N - 8 \rangle \rightarrow CR;$ $\langle N - 2 \rangle \rightarrow MD; \quad \text{link. Op.teil von } \langle N - 8 \rangle \rightarrow YZ;$ $\langle N - 4 \rangle \rightarrow MQ; \quad \text{recht. Op.teil von } \langle N - 8 \rangle \rightarrow WZ;$ $\langle N - 6 \rangle \rightarrow AC; \quad \text{recht. Adr.teil von } \langle N - 8 \rangle \rightarrow BA;$							
1.53	QCR	N	Bequemes Speichern aller Register	$2 + \langle CR \rangle \rightarrow TK \text{ von } n;$ $\langle YZ \rangle \rightarrow \text{link. Op.teil von } n;$ $\langle BZ \rangle \rightarrow \text{link. Adr.teil von } n;$ $\langle WZ \rangle \rightarrow \text{recht. Op.teil von } n;$ $\langle BA \rangle \rightarrow \text{recht. Adr.teil von } n;$ $\langle HR \rangle \rightarrow N + 8;$							Beim Transport in oder aus den vier großen Rechenwerksregistern werden alle 48 Informationsbits unverändert übertragen, auch wenn die TK $\leq 1$ ist.

Tr  
Fk  
ZL

Gk  
D, 2  
Log. Ax

Int.  
Ext.-  
Code

Sp  
2

C

Lk  
EA

## Kurzbeschreibung der Befehle

### 2. Setz- und Löschbefehle

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung	Bemerkungen
2.01	BA	N*	Bringe Adreßteil	0,N mit TK 1 → AC	Siehe 1.16
2.02	BAV	N*	Bringe Adr.teil m. Vorzch.	V,N mit TK 1 → AC	Siehe 1.17
2.03	ZTR	I <sup>s</sup>	Setze TK im Register	I besteht aus einer TK-Bezeichnung 0, 1, 2 oder 3 und einer darauffolgenden Registerbezeichnung A = AC, H = HR, D = MD oder O = MQ. Die angegebene TK wird dem angegebenen Register erteilt. Die 48 Informationsbits bleiben erhalten.	
2.04	L	I <sup>s</sup>	Lösche (Register)	I wie bei ZTR (2.03), jedoch sind ein bis vier Registerbezeichnungen in beliebiger Reihenfolge erlaubt. Die angegebene TK wird unter gleichzeitigem Löschen auf +0 den angegebenen Registern erteilt.	
2.05	LA	I <sup>s</sup>	Lösche im AC	I besteht aus genau einer Bezeichnung F = erste 40 Inf.-Bits = Mantisseite 2 = erste 24 Inf.-Bits = linkes Halbwort E = letzte 8 Inf.-Bits = Exponententeil 3 = letzte 16 Inf.-Bits = rechter Adreßteil (Drittelfwort) H = erste 42 Inf.-Bits = ganzes Wort ohne rechte Hexade T = erste 44 Inf.-Bits = ganzes Wort ohne rechte Tetrade V = erste 2 Inf.-Bits = Vorzeichenstellen M = Markenregister; Die bezeichneten Binärstellen von AC bzw. CR werden auf +0 gelöscht, unabhängig von der TK in AC, und ohne die TK zu verändern.	
2.06	ZMR	keine	Setze Marke im Register	<b>Pseudobefehl!</b> L → CR; MD wird verändert.	
2.07	ZT0	N	Setze TK0 (im Speicher)	N erhält TK 0;	
2.08	ZT1	N	Setze TK1 (im Speicher)	N erhält TK 1;	
2.09	ZT2	N	Setze TK2 (im Speicher)	N erhält TK 2;	
2.10	ZT3	N	Setze TK3 (im Speicher)	N erhält TK 3;	Die 48 Informationsbits von N bleiben erhalten.
2.11	LC	N	Lösche Speicher	+ 0 → N; TK und Marke bleiben erhalten.	
2.12	LMT	N	Lösche markiert	+ 0 markiert → N; TK bleibt erhalten	Nur t <sub>N</sub> ≤ 1
2.13	LMC	N	Lösche Marke im Speicher	In N wird Marke gelöscht, falls vorhanden; TK und übrige Informationsbits bleiben erhalten.	Nur t <sub>N</sub> ≤ 1
2.14	ZMC	N	Setze Marke im Speicher	In N wird Marke gesetzt, falls noch nicht vorhanden; TK und übrige Informationsbits bleiben erhalten.	Nur t <sub>N</sub> ≤ 1
2.15	XBA	N*	Index: Bringe Adreßteil	N → BA	Siehe 1.36
2.16	ZX	J <sup>p</sup> : I	Setze Index	J → I; J → BA;  J  ≤ 127	J = ± 0 nur als -0!
2.17	ZL	I <sup>s</sup>	Setze Merklicht	I besteht aus den ML-Bezeichnungen 1...8 in beliebiger Reihenfolge. Die bezeichneten ML werden gesetzt, alle übrigen bleiben erhalten. I = 0: Leerbefehl!	
2.18	LL	I <sup>s</sup>	Lösche Merklicht	I wie bei ZL (2.17). Die bezeichneten ML werden gelöscht, alle übrigen bleiben erhalten. I = 0: Leerbefehl!	
2.19	NL	I <sup>s</sup>	Negiere Merklicht	I wie bei ZL (2.17). Die bezeichneten ML werden negiert (invertiert), alle übrigen bleiben erhalten. I = 0: Leerbefehl!	
2.20	ZU	I	Setze Unterprogramm-Ordnungszähler	In WZ wird die Indexspeicheradresse I eingesetzt.	

## Kurzbeschreibung der Befehle

### 3. Arithmetische Befehle (dual, Festkomma)

Lfd.Nr.	Ext.ccd.	Adr.	Bezeichnung	Wirkung	$\langle N \rangle$	$\langle CR \rangle$	$\langle AC \rangle$	$\langle MD \rangle$	$\langle MQ \rangle$	$\langle YZ \rangle$	Bemerkn.
3.01	FA	N	Addiere	$a + n \rightarrow AC$		+	$t_{\max}, a + n$		$t_N, n$		
3.02	FAB	N	Addiere Betrag	$a +  n  \rightarrow AC$		+	$t_{\max}, a +  n $		$t_N, n$		
3.03	FAC	N	Addiere im Speicher	$n + a \rightarrow N; M_N$ bleibt erhalten	$t_{\max}, n + a$	+			$t_{\max}, n + a$		
3.04	FSB	N	Subtrahiere	$a - n \rightarrow AC$		+	$t_{\max}, a - n$		$t_N, n$		
3.05	FSBB	N	Subtrahiere Betrag	$a -  n  \rightarrow AC$		+	$t_{\max}, a -  n $		$t_N, n$		
3.06	FSBC	N	Subtrahiere im Speicher	$n - a \rightarrow N; M_N$ bleibt erhalten	$t_{\max}, n - a$	+			$t_{\max}, n - a$		
3.07	FSBI	N	Subtrahiere invers	$n - a \rightarrow AC$		+	$t_{\max}, n - a$		$t_N, n$		
3.08	FSBD	N	Subtrahiere von MD	$d - n \rightarrow AC$		+	$t_{\max}, d - n$				
3.09	FML	N	Multipliziere	$a \cdot n \rightarrow AC, MQ$		+	1, $\Pi_I$	1, n	1, $\Pi_{II}$	0	
3.10	FMLN	N	Multipliziere negativ	$-a \cdot n \rightarrow AC, MQ$		+	1, $-\Pi_I$	1, n	1, $-\Pi_{II}$	0	
3.11	FMLA	N	Multipliziere akkumulierend	$h, q + a \cdot n \rightarrow AC, MQ$		+	1, $\Sigma_I$	1, n	1, $\Sigma_{II}$	0	
3.12	FMAN	N	Multipl. akkumul. negativ	$h, q - a \cdot n \rightarrow AC, MQ$		+	1, $\Delta_I$	1, n	1, $\Delta_{II}$	0	
3.13	FDV	N	Dividiere	$a : n \rightarrow AC; Rest \rightarrow MQ$ Rest hat gleiches Vorz. w. Dividend		+	1, $a:n$	1, 0	1, Rest	0	
3.14	FDVI	N	Dividiere invers	$n : a \rightarrow AC; Rest \rightarrow MQ$ Rest hat gleiches Vorz. w. Dividend		+	1, $n:a$	1, 0	1, Rest	0	
3.15	FW	N	Ziehe Wurzel	$ n  \rightarrow AC$ $n < 0$ : arithmet. Alarmmeldung		+	1, $ n $	1, 0	1, 0	0	
3.16	AA	N*	Addiere Adreßteil	TK1 oder TK3: $a + N \rightarrow AC$ TK2: $a_3 + N \rightarrow a_3$ mit Einerrücklauf Die ersten 32 Inf.-Bits AC bleiben erh.			$t_A, a + N$	1, 0, N			Siehe 3.42
3.17	SBA	N*	Subtrahiere Adreßteil	TK1 oder TK3: $a - N \rightarrow AC$ TK2: $a_3 - N \rightarrow a_3$ mit Einerrücklauf Die ersten 32 Inf.-Bits AC bleiben erh.			$t_A, a - N$	1, 0, N			Siehe 3.43 TK3 mit Vorbehalt
3.18	AQ	N	Addiere in MQ	$a, q + n \cdot 2^{-46} \rightarrow AC, MQ$		+	1, $\Sigma_I$	1, n	1, $\Sigma_{II}$		
3.19	SBQ	N	Subtrahiere in MQ	$a, q - n \cdot 2^{-46} \rightarrow AC, MQ$		+	1, $\Delta_I$	1, n	1, $\Delta_{II}$		
3.20	MLR	N	Multipliziere mit Rundung	$a \cdot n \rightarrow AC$		+	1, $a \cdot n$	1, n	1, 0	0	
3.21	MNR	N	Mult. negativ mit Rundung	$-a \cdot n \rightarrow AC$		+	1, $-a \cdot n$	1, n	1, 0	0	
3.22	MAR	N	Mult. akkumul. m. Rundung	$h + a \cdot n \rightarrow AC$		+	1, $\Sigma$	1, n	1, 0	0	
3.23	MAP	N	Mult. akkumul. negat. m. Rdg.	$h - a \cdot n \rightarrow AC$		+	1, $\Delta$	1, n	1, 0	0	
3.24	DVD	N	Dividiere doppelt lang	$\langle AC, MQ \rangle : n \rightarrow AC; Rest \rightarrow MQ$ Rest hat gleiches Vorz. w. Dividend		+	1, Quotient	1, 0	1, Rest	0	
3.25	FPOL	N	Berechne Polynom	$\sum_{k=0}^p \langle N + 2k \rangle \cdot a^{p-k} \rightarrow AC; a \rightarrow HR; \langle N + 2p \rangle \rightarrow MD; 0 \rightarrow MQ;$ $L \rightarrow CR; 0 \rightarrow YZ; N + 2p \rightarrow BA;$ Beendigung durch Marke in $\langle N + 2p \rangle$							Nur TK1
3.26	FMSK	J <sup>P</sup> : I	Multipliziere skalar	$\sum_{k=0}^p \langle \langle I \rangle + 2k \rangle \cdot \langle \langle I + 1 \rangle + 2k \rangle \rightarrow AC, MQ; \sum_{k=0}^{p-1} \langle \langle I \rangle + 2p \rangle \rightarrow MD;$ CR wird durch Marke in $\langle \langle I + 1 \rangle + 2k \rangle$ gesetzt; $0 \rightarrow YZ; \langle \langle I \rangle + 2p + 2 \rightarrow BA$ ; Beendigung durch Marke in $\langle \langle I \rangle + 2p \rangle$ ; $1 \leq  I  \leq 127$							

Fk  
zGK  
D, 2  
Log. AxSE  
Außer

Ab

Int.  
Ext.-  
Code

Sp

2

0,5

C

LS

Lk

## Kurzbeschreibung der Befehle

### 3. Arithmetische Befehle (dual, Gleitkomma)

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung	N	$\langle CR \rangle$	$\langle AC \rangle$	$\langle MD \rangle$	$\langle MQ \rangle$	$\langle YZ \rangle$	Bemerkn.
3.27	GA	N	Addiere	$a + n \rightarrow AC$		+	$0.a + n$	0,n	0,0	s	
3.28	GAB	N	Addiere Betrag	$a +  n  \rightarrow AC$		+	$0.a +  n $	0,n	0,0	s	
3.29	GAC	N	Addiere im Speicher	$n + a : N; M_N$ bleibt erhalten	0,n+a	+		$0.n + a$	0,0	s	
3.30	GSB	N	Subtrahiere	$a - n \rightarrow AC$		+	$0.a - n$	0,n	0,0	s	
3.31	GSBB	N	Subtrahiere Betrag	$a -  n  \rightarrow AC$		+	$0.a -  n $	0,n	0,0	s	
3.32	GSBC	N	Subtrahiere im Speicher	$n - a : N; M_N$ bleibt erhalten	0,n-a	+		$0.n - a$	0,0	s	
3.33	GSBI	N	Subtrahiere invers	$n - a \rightarrow AC$		+	$0.n - a$	0,n	0,0	s	
3.34	GSBD	N	Subtrahiere von MD	$d - n \rightarrow AC$		+	$0.d - n$	0,n	0,0	s	
3.35	GML	N	Multipliziere	$a \cdot n \rightarrow AC$		+	$0.a \cdot n$	0,n	0,0	s	
3.36	GMLN	N	Multipliziere negativ	$-a \cdot n \rightarrow AC$		+	$0.-a \cdot n$	0,n	0,0	s	
3.37	GMLA	N	Multipliziere akkumulierend	$h + a \cdot n \rightarrow AC$		+	$0.\Sigma$	0,n	0,0	s	
3.38	GMAN	N	Multipliziere akkumul. negativ	$h - a \cdot n \rightarrow AC$		+	$0.\Delta$	0,n	0,0	s	
3.39	GDV	N	Dividiere	$a : n \rightarrow AC$		+	$0.a : n$	0,0	0,0	0	
3.40	GDVI	N	Dividiere invers	$n : a \rightarrow AC$		+	$0.n : a$	0,0	0,0	0	
3.41	GW	N	Ziehe Wurzel	$ n  \rightarrow AC; n < 0:$ arithmet. Alarm-Meldung		+	$0. n $	0,0	0,0	0	
3.42	AA	N*	Addiere Adreßteil	TK0: $a \cdot 16^l \rightarrow AC; l = \text{letzte } 8 \text{ Bits von } N;  l  \leq 127$			$0.a \cdot 16^l$	1,0,l			Siehe 3.16
3.43	SBA	N*	Subtrahiere Adreßteil	TK0: $a \cdot 16^{-l} \rightarrow AC; l = \text{letzte } 8 \text{ Bits von } N;  l  \leq 127$			$0.a \cdot 16^{-l}$	1,0,l			Siehe 3.17
3.44	AU	N	Addiere unnormalisiert	$a + n \rightarrow AC; \text{unnormalisiert}$		+	$0.a + n$	0,n	0,0	0	
3.45	SBU	N	Subtrahiere unnormalisiert	$a - n \rightarrow AC; \text{unnormalisiert}$		+	$0.a - n$	0,n	0,0	0	
3.46	REZ	N	Bilde reziproken Wert	$1:n \rightarrow AC$		+	$0.1:n$	0,0	0,0	0	
3.47	GPOL	N	Berechne Polynom	$\sum_{k=0}^{p-1} N + 2k \cdot a^{p-k} \rightarrow AC; a \rightarrow HR; \langle N + 2p \rangle \rightarrow MD; 0 \rightarrow MQ; L \rightarrow CR; s \rightarrow YZ (\text{Addition von } \langle N + 2p \rangle); N + 2p \rightarrow BA; \text{Beendigung durch Marke in } \langle N + 2p \rangle.$							
3.48	GMSK	JP: I	Multipliziere skalar	$\sum_{k=0}^{p-1} \langle 1 + 2k \rangle \cdot \langle 1 + 1 + 2k \rangle \rightarrow AC; \sum_{k=0}^{p-1} \langle 1 + 2k \rangle \rightarrow HR; \langle \langle 1 + 2p \rangle \rightarrow MD; 0 \rightarrow MQ; CR \text{ wird durch Marke in } \langle 1 + 1 + 2k \rangle \text{ gesetzt}; s \rightarrow YZ (\text{Addition des letzten Produkts}); \langle 1 + 2p + 2 \rightarrow BA; \text{Beendigung durch Marke in } \langle 1 + 2p \rangle; 1 \leq  J  \leq 127$							
3.49	HA	N	Höhere Genauigkeit: Addiere	$\langle X 17 \rangle_{fr} + \langle N_{fr} \rangle \text{ normalisiert} \rightarrow \langle X 17 \rangle_{fr}; \langle X 17 \rangle_{fr} \text{ legt Mantissenlänge fest}; TK0, + 0 \rightarrow AC, \rightarrow HR, \rightarrow MD \text{ und } \rightarrow MQ; CR \text{ wird durch Marke im Kopfwort des Speicheroperanden gesetzt}; 0 \rightarrow YZ; \langle BZ \rangle \text{ oder } \langle X 17 \rangle \rightarrow BA$							Nur TK0
3.50	HAU	N	Höhere Genauigkeit: Addiere unnormalisiert	$\langle X 17 \rangle_{fr} + \langle N_{fr} \rangle \text{ unnormalisiert} \rightarrow \langle X 17 \rangle_{fr}; \langle X 17 \rangle_{fr} \text{ legt Mantissenlänge fest}; TK0, + 0 \rightarrow AC, \rightarrow HR, \rightarrow MD \text{ und } \rightarrow MQ; CR \text{ wird durch Marke im Kopfwort des Speicheroperanden gesetzt}; 0 \rightarrow YZ; \langle BZ \rangle \text{ oder } \langle X 17 \rangle \rightarrow BA$							
3.51	HSB	N	Höhere Genauigkeit: Subtrahiere	$\langle X 17 \rangle_{fr} - \langle N_{fr} \rangle \text{ normalisiert} \rightarrow \langle X 17 \rangle_{fr}; \langle X 17 \rangle_{fr} \text{ legt Mantissenlänge fest}; TK0, + 0 \rightarrow AC, \rightarrow HR, \rightarrow MD \text{ und } \rightarrow MQ; CR \text{ wird durch Marke im Kopfwort des Speicheroperanden gesetzt}; 0 \rightarrow YZ; \langle BZ \rangle \text{ oder } \langle X 17 \rangle \rightarrow BA$							
3.52	HSU	N	Höhere Genauigkeit: Subtrahiere unnormalisiert	$\langle X 17 \rangle_{fr} - \langle N_{fr} \rangle \text{ unnormalisiert} \rightarrow \langle X 17 \rangle_{fr}; \langle X 17 \rangle_{fr} \text{ legt Mantissenlänge fest}; TK0, + 0 \rightarrow AC, \rightarrow HR, \rightarrow MD \text{ und } \rightarrow MQ; CR \text{ wird durch Marke im Kopfwort des Speicheroperanden gesetzt}; 0 \rightarrow YZ; \langle BZ \rangle \text{ oder } \langle X 17 \rangle \rightarrow BA$							
3.53	HML	N	Höhere Genauigkeit: Multipliziere	$\langle X 17 \rangle_{fr} \cdot \langle N_{fr} \rangle \text{ normalisiert} \rightarrow \langle X 17 \rangle_{fr}; \langle X 17 \rangle_{fr} \text{ legt Mantissenlänge fest}; TK0, + 0 \rightarrow AC \rightarrow HR, \rightarrow MD \text{ und } \rightarrow MQ; CR \text{ wird durch Marke im Kopfwort des Speicheroperanden gesetzt}; 0 \rightarrow YZ; \langle BZ \rangle \text{ oder } \langle X 17 \rangle \rightarrow BA$							

## Kurzbeschreibung der Befehle

### 3. Arithmetische Befehle (logisch, dezimal, Halbwort, Index)

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung	$\langle N \rangle$	$\langle CR \rangle$	$\langle AC \rangle$	$\langle HR \rangle$	$\langle MD \rangle$	$\langle MQ \rangle$	$\langle YZ \rangle$	Bemerk.
3.54	VEL	N	<b>Vel</b>	$a \vee n \rightarrow AC$	+		$t_{\max}, a \vee n$			$t_{N,n}$		
3.55	AUT	N	<b>Aut</b>	$a \oplus n \rightarrow AC$	+		$t_{\max}, a \oplus n$			$t_{N,n}$		
3.56	ET	N	<b>Et</b>	$a \& n \rightarrow AC$	+		$t_{\max}, a \& n$			$t_{N,n}$		
3.57	ZUS	N	<b>Setze zusammen</b>	$(h \& a) v (h \& n) \rightarrow AC$	+		$t_{\max} (AC - s), Zus$			$t_{N,n}$		
3.58	VLA	N*	<b>Vel Adreßteil</b>	$h \vee (0, N) \rightarrow AC$			$t_H, h \vee (0,N)$			1,0,N		
3.59	ATA	N*	<b>Aut Adreßteil</b>	$h \oplus (0, N) \rightarrow AC$			$t_H, h \oplus (0,N)$			1,0,N		
3.60	ETA	N*	<b>Et Adreßteil</b>	$h \& (0, N) \rightarrow AC$			$t_H, h \& (0,N)$			1,0,N		

3.61	DA	N	<b>Dezimale Addition</b>	$a + n \rightarrow AC$	+		$1,a + n$	1,0	1,0	1,0		
3.62	DAC	N	<b>Dezimale Addition im Speicher</b>	$n + a \rightarrow N; M_N$ bleibt erhalten		$1, n + a$	$+ 1,n + a$	1,0	$1,n + a$	1,0		
3.63	DSB	N	<b>Dezimale Subtraktion</b>	$a - n \rightarrow AC$	+		$1,a - n$	1,0	1,0	1,0		
3.64	DSC	N	<b>Dezimale Subtraktion im Speicher</b>	$n - a \rightarrow N; M_N$ bleibt erhalten		$1, n - a$	$+ 1,n - a$	1,0	$1,n - a$	1,0		
3.65	DML	N	<b>Dezimale Multiplikation</b>	$a \cdot n \rightarrow AC, MQ$	+		$1,\Pi_1$	1,0	1,n	$1,\Pi_{11}$	0	
3.66	DMN	N	<b>Dezimale Multiplikation negativ</b>	$-a \cdot n \rightarrow AC, MQ$	+		$1,-\Pi_1$	1,0	1,n	$1,-\Pi_{11}$	0	
3.67	DDV	N	<b>Dezimale Division</b>	$a : n \rightarrow AC; Rest \rightarrow MQ; Rest hat gleiches Vorz. w. Divid.$	+		$1,a:n$	1,0	1,0	1,Rest	0	
3.68	DDD	N	<b>Dezimale Division doppelt lang</b>	$\langle AC, MQ \rangle : n \rightarrow AC; Rest \rightarrow MQ; Rest hat gleiches Vorz. w. Divid.$	+		1, Quotient	1,0	1,0	1,Rest	0	

3.69	A2	N <sup>2</sup>	<b>Addiere Halbwort</b>	$a_2 + n_2 \rightarrow a_2$	+		$t_{\max}, V.a_2 + n_2$		$t_{A,0}$			<b>TK<math>\geq 2</math></b>
3.70	SB2	N <sup>2</sup>	<b>Subtrahiere Halbwort</b>	$a_2 - n_2 \rightarrow a_2$	+		$t_{\max}, V.a_2 - n_2$		$t_{A,0}$			<b>nur mit Vorbehalt</b>
3.71	M2	N <sup>2</sup>	<b>Multipliziere Halbwort</b>	$a \cdot n_2 \rightarrow AC$	+		$1,a \cdot n_2$	1,0	1,0	0		
3.72	M2N	N <sup>2</sup>	<b>Multipliziere Halbwort negativ</b>	$-a \cdot n_2 \rightarrow AC$	+		$1,-a \cdot n_2$	1,0	1,0	0		
3.73	M2R	N <sup>2</sup>	<b>Multipliziere Halbwort mit Rundung</b>	$r_a \cdot n_2 \rightarrow AC$	+		$1,r_a \cdot n_2$	1,0	1,0	0		<b>Nur TK1</b>
3.74	M2P	N <sup>2</sup>	<b>Multipliziere Halbwort negativ mit Rundung</b>	$r_a \cdot n_2 \rightarrow AC$	+		$1,r_a \cdot n_2$	1,0	1,0	0		

3.75	V	N*	<b>Vergleiche</b>	$b == N \rightarrow BA$								
3.76	VX	I	<b>Vergleiche mit Indexspeicher</b>	$b - i \rightarrow BA$								
3.77	VC	N <sup>2</sup>	<b>Vergleiche mit Speicher</b>	$b - n_2 \rightarrow BA$								
3.78	H	J <sup>2</sup> : I	<b>Erhöhe</b>	$i + J \rightarrow I; i + J \rightarrow BA$ $ J  \leq 127$								
3.79	HX	J : I	<b>Erhöhe aus Indexspeicher</b>	$i + j \rightarrow I; i + j \rightarrow BA$								
3.80	HXN	J : I	<b>Erhöhe aus Indexspeicher negativ</b>	$i - j \rightarrow I; i - j \rightarrow BA$								

Gk  
D, 2  
Log. Ax

SE  
Aufber.

Ab

Int.  
Ext.-  
Code

Sp

2

0,5

C

LS

Lk

## **Kurzbeschreibung der Befehle**

## 4. Aufbereitungsbefehle

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung	Bemerkungen
4.01	SH	JS   <sup>P</sup>	Schifte	Schift um I Stellen gemäß der Spezifikation J: A = AC, Q = MQ, Z = AC und MQ zusammen (doppelt lang), L = nach links, K = im Kreis (Ringschift), D = dezimal (tetradenweise), R = mit Rundung; $0 \leq I \leq 127$ , TK wird nicht geschiftet; <HR>, <MD> und <YZ> bleiben erhalten.	
4.02	NRM	S	Normalisiere	Normalisierung gemäß der Spezifikation I: (nur tetradenweise nach links) F = Festkommazahl (s = Anzahl der Binärschiffe $\rightarrow$ YZ); F4 = wie F, aber s = Anzahl der Tetradschiffe; G = Gleitkommazahl (s = Anzahl der Binärschiffe $\rightarrow$ YZ); FG = Festkommazahl, linksbündig, in Gleitkommazahl (0 $\rightarrow$ YZ).	
4.03	VGF	keine	Verwandle Gleitkommazahl in Festkommazahl	<b>Pseudobefehl!</b> Verwandlung der in AC gegebenen Gleitkommazahl x in Festkommazahl $\lceil 16^{-e} \cdot x \rceil \rightarrow AC, MQ, MD, BA$ ; werden verändert. Der Bezugsexponent e muß im Adreßteil des folgenden (Leer-) Befehls stehen. WZ muß vorher mit <b>ZU</b> gesetzt worden sein!	
4.04	VAQ	keine	Vorzeichenangleich zwischen AC und MQ	Angleichung des Vorzeichen der beiden Teile einer doppelt langen dualen Festkomma-Zahl in AC, MQ unter Erhaltung ihres Wertes.	Nur TK1
4.05	DR	N*	Dezimale Rundung	Rundung einer doppelt langen dezimalen Festkommazahl in AC, MQ am Ende von AC; TK1, + 0 $\rightarrow$ HR, $\rightarrow$ MD $\rightarrow$ MQ; Adresse Teil N bedeutungslos.	Nur TK1
4.06	KVH	P	Konvertiere hin	$AC, MQ^*$ , dezimal, höchstens I-stellig $\rightarrow$ AC*, dual, exakt, mit TK1; TK1, + 0 $\rightarrow$ HR, $\rightarrow$ MD und $\rightarrow$ MQ; 0 $\rightarrow$ YZ; $1 \leq I \leq 13$ ; nur für positive ganze Zahlen.	
4.07	KVZ	P	Konvertiere zurück	$AC^*$ , dual, mit TK0 oder TK1 $\rightarrow$ AC, MQ*, dezimal, abgerundet, mit TK1, I-stellig; Konvertierungsrest $\rightarrow$ MD mit TK1; 0 $\rightarrow$ YZ; $1 \leq I \leq 13$ ; nur für positive echte Brüche.	Nur $TK \leq 1$
4.08	KBH	keine	Konvertiere Bruch hin	<b>Pseudobefehl!</b> $AC, MQ^*$ , dezimal, 13-stellig $\rightarrow$ AC, dual, normal gerundet, mit TK1; TK1, + 0 $\rightarrow$ HR und $\rightarrow$ MQ; TK1, $2^{43} \cdot 10^{-13} = 384B\ 84D0\ 92ED \rightarrow MD$ ; 0 $\rightarrow$ YZ; <BZ> $\rightarrow$ BA; nur für positive echte Brüche.	
4.09	KGZ	keine	Konvertiere ganze Zahl zurück	<b>Pseudobefehl!</b> $\langle AC \rangle^*$ , dual, mit TK1 $\rightarrow$ AC, MQ*, dezimal, exakt, mit TK1; TK1, $10^{13-1} = 0918\ 4E72\ 9FFF \rightarrow HR$ ; bedeutungsloser Konvertierungsrest $\rightarrow$ MD; 0 $\rightarrow$ YZ; <BZ> $\rightarrow$ BA; nur für positive ganze Zahlen.	Nur TK1
4.10	KOH	keine	Konvertiere Gleitkommazahl (TKO) hin;	<b>Pseudobefehl!</b> Verwandlung eines im Lebensspeicher befindlichen string in eine im AC befindliche duale Gleitkommazahl. Näheres siehe Programmbeschreibung; Dauer etwa 3700 Mikrosekunden.	
4.11	KOZ	keine	Konvertiere Gleitkommazahl (TKO) zurück;	<b>Pseudobefehl!</b> Verwandlung einer im AC befindlichen dualen Gleitkommazahl in einen im Lebensspeicher befindlichen string. Näheres siehe Programmbeschreibung; Dauer etwa 1700 Mikrosekunden.	
4.12	VRK	I	Verkürze	Verkürzung von höchstens 13 Ziffernhexaden ( $\leq 15$ ), beginnend mit den ersten $\langle I \rangle$ Hexaden von $\langle I+1 \rangle$ , unter Wegfall der Füllhexaden (40 bis 43) zu Tetraden $\rightarrow AC, MQ^*$ , mit TK1; + 0, TK1 $\rightarrow$ HR und $\rightarrow$ MD; Anzahl der Ziffern $\rightarrow$ YZ; Grenzhexade in ihrer Speicherzelle an das linke Ende geschiftet; Adresse der Zeile, der die letzte Hexade entnommen wurde, $\rightarrow I+1$ ; neuer Hexadenzählerstand (0 bis 7) $\rightarrow$ BA.	Nur TK3
4.13	VRL	P	Verlängere	Verlängerung der letzten 13 Tetraden aus $AC, MQ^*$ (erste 11 Tetraden = + 0!) zu Ziffernhexaden $\rightarrow AC, MQ^*$ , mit TK3, mit Einfügung des (Dezimal-)Punkts nach der I. Ziffer; vor dem Punkt wenigstens 1 Ziffer; sonst $\langle I \rangle$ anstelle führender Nullen vor dem Punkt; $\langle I \rangle$ auch anstelle Punkt am Ende; TK3, + 0 $\rightarrow$ HR und $\rightarrow$ MD; 0 $\rightarrow$ YZ; $0 \leq I \leq 13$ .	TK beliebig
4.14	LK1	N	Lochkartenbefehl 1	Hineinschiften der in der ersten Tetrade von MD stehenden Zeilennummer (0bis12) von links in die Zeilen des bei N beginnenden halben Spaltenbildes gemäß der in den ersten 40 Bits von AC befindlichen Halbzelle; TK3, + 0 $\rightarrow$ AC; 0 $\rightarrow$ YZ.	Nur TK3
4.15	LK2	J : 1	Lochkartenbefehl 2	Aufbau des bei J beginnenden Zeilenbildes einer Lochkarte vor dem Ausstanzen aus dem bei j beginnenden Spaltenbild; TK3, + 0 $\rightarrow$ AC, $\rightarrow$ MD und $\rightarrow$ MQ; 0 $\rightarrow$ YZ; j + 160 $\rightarrow$ BA.	
4.16	HWI	P	Höhere Genauigkeit: Wandle hin	$AC = X17_{16}$ mit $I \leq I \leq 15$ ; TK0, + 0 $\rightarrow$ AC; 0 $\rightarrow$ YZ; <X17> $\rightarrow$ BA.	
4.17	HWZ	N	Höhere Genauigkeit: Wandle zurück	$\langle N \rangle_{ff} \rightarrow AC; N \rightarrow MD; 0 \rightarrow YZ$ .	
4.18	HN	N	Höhere Genauigkeit: Negiere	$- N_{ff} \rightarrow N_{ff}$ durch Invertierung des Vorzeichenbits im Kopfwort.	Nur TK0
4.19	HNM	N	Höhere Genauigkeit: Normalisiere	Normalisierung von $\langle N \rangle_{ff}; TK0, + 0 \rightarrow AC, \rightarrow HR, \rightarrow MD$ und $\rightarrow MQ; 0 \rightarrow YZ; N \rightarrow BA$ .	

## 5. Sprungbefehle

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung
5.01	0	N*	Leerbefehl	Keine (außer der selbstverständlichen $\wedge BZ_1 + 1 \rightarrow BZ$ ); Adreßteil N bedeutungslos.
5.02	S	N <sup>2</sup>	Springe	$N \rightarrow BZ$
5.03	SZB	N <sup>2</sup>	Springe mit BZ ... BA	$\wedge BZ_{>neu} \cdot \cdot \cdot BA; N \rightarrow BZ$
5.04	SU	N <sup>2</sup>	Springe in Unterprogramm	$\wedge WZ_1 + 1 \rightarrow WZ; \wedge BZ_{>neu} \cdot \cdot \cdot <WZ>_{neu}; N \rightarrow BZ$
5.05	T	N <sup>2</sup>	T!	Ausführung des Befehls n <sub>2</sub> .
5.06	EA	(N <sup>2</sup> )	Ein- und Ausgabe	Unterprogrammsprung in den durch externe symbolische Adresse bezeichneten Zweig des Verteilerprogramms; im Verteilerprogramm bleiben nur WZ und ML unverändert.
5.07	HLT	keine	Halt	<b>Pseudobefehl!</b> Sprung in speziellen Zweig des Verteilerprogramms zur Beendigung des Programms.
5.08	SIO	N <sup>2</sup>	Springe, wenn identisch 0	$N \rightarrow BZ, w. \langle AC \rangle = 0;$
5.09	SNO	N <sup>2</sup>	Springe, wenn nicht ident. 0	$N \rightarrow BZ, w. \langle AC \rangle \neq 0;$ Die Ausführung des Vergleichs wird von der TK von $\langle AC \rangle$ bestimmt.
5.10	SG0	N <sup>2</sup>	Springe, wenn größer als 0	$N \rightarrow BZ, w. \langle AC \rangle > 0;$
5.11	SK0	N <sup>2</sup>	Springe, wenn kleiner als 0	$N \rightarrow BZ, w. \langle AC \rangle < 0;$
5.12	SI	N <sup>2</sup>	Springe, wenn identisch	$N \rightarrow BZ, w. \langle AC \rangle \equiv \langle HR \rangle;$
5.13	SN	N <sup>2</sup>	Springe, wenn nicht identisch	$N \rightarrow BZ, w. \langle AC \rangle \neq \langle HR \rangle;$ Die Ausführung des Vergleichs wird von der höheren der beiden TK von $\langle AC \rangle$ und $\langle HR \rangle$ bestimmt;
5.14	SG	N <sup>2</sup>	Springe, wenn größer	$N \rightarrow BZ, w. \langle AC \rangle > \langle HR \rangle;$ a - MD; ist $t_A = t_H = 0$ , so wird $\langle HR \rangle$ normalisiert;
5.15	SK	N <sup>2</sup>	Springe, wenn kleiner	$N \rightarrow BZ, w. \langle AC \rangle < \langle HR \rangle;$ dabei 0 $\rightarrow YZ.$
5.16	SGB	N <sup>2</sup>	Springe, w. größerer Betrag	$N \rightarrow BZ, w. \langle AC \rangle \text{MAXIMALEN} \langle HR \rangle;$
5.17	SKB	N <sup>2</sup>	Springe, w. kleinerer Betrag	$N \rightarrow BZ, w. \langle AC \rangle \text{MINIMALEN} \langle HR \rangle;$
5.18	SR	N <sup>2</sup>	Springe, wenn rechtes Bit L	$N \rightarrow BZ, w. \text{das letzte Bit in } AC \text{ den Wert L hat.}$
5.19	SRN	N <sup>2</sup>	Springe, w. rechtes Bit nicht L	$N \rightarrow BZ, w. \text{das letzte Bit in } AC \text{ den Wert 0 hat.}$
5.20	SM	N <sup>2</sup>	Springe, wenn Marke	$N \rightarrow BZ, w. \langle CR \rangle = L_1 \text{ und } 0 \rightarrow CR$
5.21	SMN	N <sup>2</sup>	Springe, wenn Marke nicht	$N \rightarrow BZ, w. \langle CR \rangle = 0, \text{ sonst } 0 \rightarrow CR$
5.22	SAA	N <sup>2</sup>	Springe, w. Alarm: arithmet.	$N \rightarrow BZ, w. \text{A-Alarmmeldung, und diese löschen}$
5.23	SAT	N <sup>2</sup>	Spr., w. Alarm: Typenkenn.	$N \rightarrow BZ, w. \text{TK-Alarmmeldung, und diese löschen}$
5.24	SXI	N <sup>2</sup>	Springe, w. Index identisch	$N \rightarrow BZ, w. \langle BA \rangle = 0;$ Das erste Bit von $\langle BA \rangle$ wird von diesen Befehlen als Vorzeichen interpretiert:
5.25	SXN	N <sup>2</sup>	Springe, w. Index nicht ident.	$N \rightarrow BZ, w. \langle BA \rangle \neq 0;$ $  \langle BA \rangle   \leq 32767;$
5.26	SXG	N <sup>2</sup>	Springe, wenn Index größer	$N \rightarrow BZ, w. \langle BA \rangle > 0;$ 0 in BA nur als -0.
5.27	SXK	N <sup>2</sup>	Springe, wenn Index kleiner	$N \rightarrow BZ, w. \langle BA \rangle \leq 0;$ Nur TK0 in $\langle AC \rangle;   \langle AC \rangle   \leq 127;$
5.28	SEG	J <sup>1</sup> : I <sup>1</sup>	Springe, w. Exponent größer	$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn Exponent von } \langle AC \rangle \geq 1; a \rightarrow MD;$
5.29	ST	J <sup>1</sup> : I <sup>1</sup>	Springe, w. Typenkennung	$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn bei ZTR (2.03);}$
5.30	STN	J <sup>1</sup> : I <sup>1</sup>	Springe, w. Typenkenn. nicht	$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn bei ZTR (2.03);}$
5.31	SL	J <sup>1</sup> : I <sup>1</sup>	Springe, wenn Merklicht	$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn angegebene TK nicht in angegebenem Register;}$
5.32	SLL	J <sup>1</sup> : I <sup>1</sup>	Springe, w. Merkl. u. lösche	$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn mindestens ein bezeichnetes ML gesetzt;}$
5.33	SLN	J <sup>1</sup> : I <sup>1</sup>	Springe, w. Merklicht nicht	$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn bei ZL (2.17);}$
5.34	SNL	J <sup>1</sup> : I <sup>1</sup>	Springe, wenn Merklicht nicht, sonst lösche	$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn bei ZL (2.17);}$
5.35	SW	J <sup>1</sup> : I <sup>1</sup>	Springe, wenn Wahlschalter	$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn alle bezeichneten ML gelöscht;}$
				$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn alle bezeichneten ML gelöscht, sonst diese löschen;}$
				$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn aus den WS-Bezeichnungen 1...8 in beliebiger Reihenfolge;}$
				$\wedge BZ_{>alt} + J \rightarrow BZ, \text{ wenn mindestens ein bezeichneter WS gesetzt.}$

## Kurzbeschreibung der Befehle

### 6. Tabellenbefehle

Erläuterungen zu den Tabellenbefehlen 6.01 bis 6.06:

„Suchwort“ =  $\langle MD \rangle = d$ ;

Für 6.01 bis 6.04: „Tabelle“ =  $\langle N + 2k \rangle$  mit  $k = 0, 1, 2, \dots$ , soweit dort  $TK = t_D$ ;

Für 6.05 und 6.06: „Tabelle“ =  $\langle N + bk \rangle$  mit geradem positivem oder negativem „Dehnungswert“  $\langle BA \rangle = b \neq 0$ , also  $\pm 2$  oder  $\pm 4$  usw. und mit  $k = 0, 1, 2, \dots$ , soweit dort  $TK = t_B$ ;

Jeder Tabellenbefehl liefert in BA die Adresse  $N + 2k$  bzw.  $N + bk$  des „gefundenen Wortes“, d. h. entweder die des ersten Wortes aus der Tabelle, das der Suchbedingung genügt, oder die des ersten nicht mehr zur Tabelle gehörigen Wortes, dann verbunden mit der TK-Alarm-Meldung.

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung
6.01	TL	N	Tabelle lesen	Suchbedingung: $\langle N + 2k \rangle \geq d$ gemäß $t_D$ ; $N + 2k \rightarrow BA$ ; $d \rightarrow AC$ (evtl. normalisiert); wenn $t_D = 0$ , dann Normalisierung von $d$ und $+0 \rightarrow YZ$
6.02	TLB	N	Tabelle lesen: Betrag	Suchbedingung: $\langle N + 2k \rangle \geq d$ gemäß $t_D$ ; $N + 2k \rightarrow BA$ ; $d \rightarrow AC$ (evtl. normalisiert); wenn $t_D = 0$ , dann Normalisierung von $d$ und $+0 \rightarrow YZ$
6.03	TLI	N	Tabelle lesen: Identität	Suchbedingung: $\langle N + 2k \rangle = d$ gemäß $t_D$ ; $N + 2k \rightarrow BA$ ; $d \rightarrow AC$ (evtl. normalisiert); wenn $t_D = 0$ , dann Normalisierung von $d$ und $+0 \rightarrow YZ$
6.04	TLM	N	Tabelle lesen mit Maske	Suchbedingung: $(h \& \langle N + 2k \rangle) = \bar{h} \& d$ , also Übereinstimmung in dem von der „Maske“ $HR = h$ festgelegten „Nullfeld“; $N + 2k \rightarrow BA$ ; $t_{1H}$ ( $h \& \langle N + 2k \rangle$ ) = „Einsfeld“ des gefundenen Wortes $\rightarrow AC$
6.05	TLD	N	Tabelle lesen mit Dehnung	Suchbedingung: $\langle N + bk \rangle \geq d$ gemäß $t_D$ ; $N + bk \rightarrow BA$ ; $d \rightarrow AC$ (evtl. normalisiert); wenn $t_D = 0$ , dann Normalisierung von $d$ und $+0 \rightarrow YZ$
6.06	TDM	N	Tabelle lesen mit Dehnung und Maske	Suchbedingung: $(h \& \langle N + bk \rangle) = \bar{h} \& d$ , also Übereinstimmung in dem von der „Maske“ $HR = h$ festgelegten „Nullfeld“; $N + bk \rightarrow BA$ ; $t_{1H}$ ( $h \& \langle N + bk \rangle$ ) = „Einsfeld“ des gefundenen Wortes $\rightarrow AC$

SE

Aufba

## 7. Modifizier-, Ersetz- und Adressierbefehle

Ab

7.01	MF	I	Modifizierte folgenden Adreßteil	$N_F + i^+ \rightarrow AM$ ; nach der Modifizierung ist $\langle BA \rangle = i$ ; $N_X = i$
7.02	MCF	N*	Modifizierte aus Speicher folgenden Adreßteil	$N_F + n_3^- \rightarrow AM$ ; nach der Modifizierung ist $\langle BA \rangle = n_3^+$ ; $N_X = n_3$
7.03	M	I	Modifizierte	$N_{FN} + i^- \rightarrow N_{FE}$ ; nach der Modifizierung ist $\langle BA \rangle = i^+$ ; $N_X = i$
7.04	MH	J <sup>P</sup> ; I	Modifizierte und erhöhe	$i + J \rightarrow I$ ; $N_{FN} + i + J \rightarrow N_{FE}$ ; nach der Modifizierung ist $\langle BA \rangle = i + J$ ; $k_s$ ne $N_X$ ; $ J  \leq 127$
7.05	MD	J : I	Modifizierte doppelt	$(N_F + i^-)_X + j > N_{FE}$ ; nach der Modifizierung ist $\langle BA \rangle = j$ ; $N_X = i$
7.06	MM	I	Modifizierte mehrfach	$N_{FE,F} + i^+ \rightarrow AM$ ; Modifizierung aller folgenden Befehle bis vor MA und jeden Befehl mit $N_X \neq N$ sowie bis nach TCB, R, SZB und EA; nach der Modifizierung ist $\langle BA \rangle = i^+$ ; $N_X = i$
7.07	MC	N <sup>2</sup>	Modifizierte aus Speicher	$N_{FN} + n_3^+ \rightarrow N_{FE}$ ; nach der Modifizierung ist $\langle BA \rangle = n_3^+$ ; $N_X = n_3$
7.08	MCE	N <sup>2</sup>	Modifizierte aus Speicher nach Ersetzungen	$N_{FN} + \langle \langle N_3, \dots \rangle \rangle^+ \rightarrow N_{FE}$ ; Ersatzkette, bis Op-Teil mit L endet, einschließlich; nach der Modifizierung ist $\langle BA \rangle = \langle \langle N_3, \dots \rangle \rangle^+$ ; $N_X = \langle \langle N_3, \dots \rangle \rangle_3$
7.09	MA	N*	Modifizierte mit Adreßteil	$N_{FN} + N^+ \rightarrow N_{FE}$ ; nach der Modifizierung ist $\langle BA \rangle = N^+$ ; $N_X = N$
7.10	MB	N*	Modifizierte mit $\langle BA \rangle$	$N_{FN} + b^+ \rightarrow N_{FE}$ ; $\langle BA \rangle$ bleibt nach der Modifizierung erhalten; $N_X = b$ ; Adressenteil N bedeutungslos
7.11	MAB	J <sup>C</sup> I <sup>P</sup>	Modifizierte Adreßteil mit $\langle BA \rangle$	$J \rightarrow OP$ ; $I + b \rightarrow AM$ ; $I + b \rightarrow BA$ ; keine $N_X$ ; $ I  \leq 127$
7.12	E	J <sup>C</sup> I	Ersetze	$J \rightarrow OP$ ; $i_N^- \rightarrow NE$ ; $\langle BA \rangle$ bleibt nach der Ersetzung erhalten; $N_X = i_N$
7.13	EZ	J <sup>C</sup> I	Ersetze zählend	$J \rightarrow OP$ ; $i + 2 \rightarrow AM$ ; $i + 2 \rightarrow I$ ; $\rightarrow BA$ ; $N_X = i + 2$
7.14	ENZ	J <sup>C</sup> I	Ersetze negativ zählend	$J \rightarrow OP$ ; $i^- \rightarrow AM$ ; $i - 2 \rightarrow I$ ; $\rightarrow BA$ ; $N_X = i$
7.15	EMB	J <sup>C</sup> I	Ersetze und modifizierte mit $\langle BA \rangle$	$J \rightarrow OP$ ; $i_N + b \rightarrow NE$ ; $\langle BA \rangle$ bleibt nach der Modifizierung erhalten; keine $N_X$
7.16	MU	J <sup>C</sup> I <sup>P</sup>	Modifizierte über Unterprogrammordnungszählern	$J \rightarrow OP$ ; $I + \langle \langle WZ \rangle \rangle \rightarrow AM$ ; falls J ein auszuführender Sprungbefehl mit Ganzadresse ist, $\langle WZ \rangle - 1 = WZ$ ; keine $N_X$ ; $ I  \leq 127$
7.17	EMU	J <sup>C</sup> I <sup>P</sup>	Ersetze nach Modifizierung über Unterpr.ordn.zähler	$J \rightarrow OP$ ; $\langle I + \langle \langle WZ \rangle \rangle \rangle_{2N} \rightarrow NE$ ; $N_X = \langle I + \langle \langle WZ \rangle \rangle \rangle_{2N}$ ; $ I  \leq 127$
7.18	RL	J <sup>C</sup> I <sup>P</sup>	Relative Addressierung	$J \rightarrow OP$ ; $(I + \langle \langle BZ \rangle \rangle_{2N}) \rightarrow NE$ ; $N_X = (I + \langle \langle BZ \rangle \rangle_{2N})_{2N}$ ; $ I  \leq 127$
7.19	R	J <sup>C</sup> I <sup>S</sup>	Register-Adressierung	$J \rightarrow OP$ , soweit zugelassener Zweitcode; als Operand wird der Inhalt des in I bezeichneten Registers verwendet; A = AC, H = HR, D = MD, Q = MQ, L = links, Z = BZ (neu), W = WZ, B = BA, Y = YZ; $N_X = N$ Gegebenenfalls Ausschnitt rechts, Auffüllen mit +0 links, Hinzufügen TK1

Int.  
Ext-  
Code

Sp

2

0,5

C

LS

Lk

## Alphabetische Liste der Befehle und allgemeine Eigenschaften

<b>Spalte</b>	<b>Erläuterungen</b>
Externcode	enthält alle externen BefehlsCodes alphabetisch geordnet, auch die der Pseudobefehle sowie die Pseudocodes ( $0 < \dots < 9 < A < \dots Z$ )
Ifd. Nr.	laufende Nummer aus der Kurzbeschreibung der Befehle.
Interncode	enthält die internen BefehlsCodes in Tetraden- und technischer Notierung.
belegt	<p>gibt an, welche Werke in der Ausführungsphase belegt sind:</p> <p>B nur das Befehlswerk (und oft auch das Speicherwerk)      R nur das Rechenwerk      BR beide Werke (und oft auch das Speicherwerk).</p> <p>Zur zeitlichen Optimierung von Programmen lässt man nach Möglichkeit reine Befehlswerksbefehle auf länger laufende reine Rechenwerksbefehle (z. B. W, DV, ML, KVH, GA, SH) folgen.      Reine Befehlswerksbefehle laufen auch dann noch ab, wenn ein vorausgehender Befehl eine A- oder TK-Alarmmeldung erzeugt hat. Außerdem läuft SAA bei bestehender TK-Alarmmeldung und SAT bei bestehender A-Alarmmeldung normal ab.</p>
Nennadresse	<p>II kennzeichnet die Befehle, die eine Modifizierung 2. Art bewirken. (Bei MD ist das nur die Modifizierung mit j).      = kennzeichnet die Befehle, deren Adreßteil bei einer Modifizierung 2. Art ebenso wie bei einer solchen 1. Art unmittelbar modifiziert wird. (Nennadresse = Adreßteil.)        (hinter ,=') kennzeichnet die Befehle, bei denen zwar Nennadresse = Adreßteil ist, nach deren Ausführung aber eine von MM bewirkte Kette von Modifizierungen abbricht, während sie sonst nur vor jedem Befehl abbricht, bei dem nicht ,=' angegeben ist, also auch vor MA.      # kennzeichnet die Befehle, deren Adreßteil von einer Modifizierung 2. Art nicht berührt wird (Befehle ohne Nennadresse).</p> <p>Alle Befehle, bei denen weder ,=' noch ,#' angegeben ist, weisen statt dessen eine Größe auf, die aus dem Adreßteil durch einen diesen Befehlen jeweils eigentümlichen Ersetzungsprozeß hervorgeht und die „Nennadresse“ dieser Befehle heißt.</p> <p>Das Wesen der Modifizierung 2. Art besteht darin, daß vor der Modifizierung zuerst diese Ersetzung abläuft, so daß die Modifizierung nicht auf den ursprünglichen Adreßteil, sondern erst auf die Nennadresse ausgeübt wird.</p>
Zweitcode bei R	<p>+</p> <p>(+)</p> <p>-</p>
AZ	Ausführungszeit in Mikrosekunden

## Ausführungszeiten der TR4-Befehle

**Abrufzeit:** 6 Mikrosekunden

wenn ein Modifizierbefehl vorausgeht: 8,5 Mikrosekunden

Die Abrupphase jedes Befehls beginnt, sobald der vorhergehende Befehl das Befehlswerk freigegeben hat; daher läuft sie gleichzeitig mit der Ausführung reiner Rechenwerksbefehle ab.

**Doppelcodebefehle:** Zu den angegebenen Ausführungszeiten muß noch die Ausführungszeit des Zweitcodes hinzugezählt werden.

**Gleitkommaoperationen:** Es werden normalisierte Operanden vorausgesetzt.

**m bedeutet bei:**

FPOL, GPOL	Grad des Polynoms	Ab
FMSK, GMSK	Anzahl der Vektorkomponenten	
HA, HAU, HB, HC, HML, HNM, HSB, HSU, HWH	" " Mantissenwörter	Int. Ext.- Code
MCE	" " Ersetzungen	
TDM, TL, TLB, TLD, TLI, TLM	" " Wörter bis zum gefundenen Wort	
WTD, WTK	" " transportierten Wörter	Sp
LK1, LK2	" " Löcher in der Lochkartenzeile bzw. Lochkarte	
VRK	" " Zellen	
NRM	" " Tetradschifte	2

**p bedeutet bei:**

VRK	Anzahl der Hexaden, einschließlich Grenzhexade
-----	--

0,5

C

LS

Lk

**Alphabetische Liste der Befehle  
und allgemeine Eigenschaften**

Externcode	Ifd. Nr.	Interncode	belegt	Nennadresse	Zweitcode bei R	AZ
0	5.01	00 000	B	#	(+)	1,5
A2	3.69	7C 370	R	=	+	9
AA	3.16	98 460	R	=	-	11,5
AQ	3.18	7E 372	R	=	+	7,5
ATA	3.59	89 421	R	=	-	7,5
AU	3.44	49 221	R	=	+	15
AUT	3.55	69 321	R	=	+	2,5
B	1.01	70 340	R	=	+	3
B2	1.12	6E 332	R	=	+	4
B2V	1.13	6F 333	R	=	+	4,5
B3	1.14	6C 330	R	=	+	6,5
B3V	1.15	6D 331	R	=	+	7
BA	1.16	8E 432	B R	=	-	6
BAP	1.19	DD 671	B R	=	-	6,5
BAR	1.18	DC 670	B R	=	-	6
BAV	1.17	DF 673	B R	=	-	6,5
BB	1.06	74 350	R	=	+	3
BC	1.20	A3 503	B R	=	-	8,5
BD	1.03	71 341	R	=	+	3
BH	1.02	73 343	R	=	+	3
BN	1.05	75 351	R	=	+	3
BNR	1.08	77 353	R	=	+	3
BQ	1.04	72 342	R	=	+	3
BQB	1.09	DA 662	R	=	+	3
BR	1.07	76 352	R	=	+	3
BZ	1.10	D9 661	B R	=	-	9
BZ2	1.11	D8 660	B R	=	-	7
C	1.21	80 400	B R	=	-	6,5
C2	1.29	A0 500	B R	=	-	9
C3	1.30	A1 501	B R	=	-	9
CB	1.26	85 411	B R	=	-	6,5
CD	1.23	86 412	B R	=	-	6,5
CH	1.22	8F 433	B R	=	-	7,5
CMC	1.33	A2 502	B R	=	-	8,5
CMR	1.32	83 403	B R	=	-	6,5
CMT	1.31	82 402	B R	=	-	6,5
CN	1.25	84 410	B R	=	-	6,5
CQ	1.24	87 413	B R	=	-	6,5
CR	1.27	81 401	B R	=	-	6,5
CWL	1.46	06 012	B	#	-	9
CZ	1.28	DB 663	B R	=	-	12
DA	3.61	C6 612	R	=	+	10,5
DAC	3.62	C4 610	B R	=	-	16,5
DDD	3.68	C3 603	R	=	+	420
DDV	3.67	C2 602	R	=	+	420
DML	3.65	C0 600	R	=	+	240
DMN	3.66	C1 601	R	=	+	240
DR	4.05	DE 672	R	=	(+)	7,5

**Alphabetische Liste der Befehle  
und allgemeine Eigenschaften**

Externcode	Ifd. Nr.	Interncode	belegt	Nennadresse	Zweitcode bei R	AZ
DSB	3.63	C7 613	R	=	+	10,5
DSC	3.64	C5 611	B R	=	-	16,5
DVD	3.24	61 301	R	=	+	157
E	7.12	29 121	B	i <sub>N</sub>	-	4,5
EA	5.06	3B 163	B	=	-	0,5
EMB	7.15	28 120	B	II #	-	9,5
EMU	7.17	04 010	B	<I + <<WZ>> <sub>3N</sub>	-	20,5
ENZ	7.14	2A 122	B	i	-	9
ET	3.56	6A 322	R	=	+	2,5
ETA	3.60	8A 422	R	=	-	7,5
EZ	7.13	2B 123	B	i + 2	-	8
F A	3.01	42 202	R	=	+	6
F AB	3.02	40 200	R	=	+	6,5
F AC	3.03	43 203	B R	=	-	12
F DV	3.13	60 300	R	=	+	145
F DVI	3.14	62 302	R	=	+	145
F MAN	3.12	5A 262	R	=	+	32,5
F ML	3.09	54 250	R	=	+	29,5
F MLA	3.11	56 252	R	=	+	32,5
F MLN	3.10	58 260	R	=	+	29,5
F MSK	3.26	E6 712	B R	#	-	16 + 39,5 m
F POL	3.25	E4 710	B R	#	-	37,5 m
F SB	3.04	46 212	R	=	+	6,5
F SBB	3.05	41 201	R	=	+	6,5
F SBC	3.06	47 213	B R	=	-	12,5
F SBD	3.08	45 211	R	=	+	6,5
F SBI	3.07	44 210	R	=	+	6,5
F W	3.15	50 240	R	=	+	150
G A	3.27	4B 223	R	=	+	16
G AB	3.28	52 242	R	=	+	16
G AC	3.29	4A 222	B R	=	-	22
G DV	3.39	64 310	R	=	+	120
G DVI	3.40	66 312	R	=	+	120
G MAN	3.38	5D 271	R	=	+	49
G ML	3.35	5E 272	R	=	+	31
G MLA	3.37	5F 273	R	=	+	49
G MLN	3.36	5C 270	R	=	+	31
G MSK	3.48	E7 713	B R	#	-	16 + 56 m
G POL	3.47	E5 711	B R	#	-	49 m
G SB	3.30	4F 233	R	=	+	16
G SBB	3.31	53 243	R	=	+	16
G SBC	3.32	4E 232	B R	=	-	22
G SBD	3.34	4C 230	R	=	+	16
G SBI	3.33	48 220	R	=	+	16
G W	3.41	51 241	R	=	+	150
H	3.78	2C 130	B	#	-	8,5
HA	3.49	F0 740	B R	#	-	62 + 20 m

Int.  
Ext.-  
Code

Sp

2

0,5

C

LS

Lk

**Alphabetische Liste der Befehle  
und allgemeine Eigenschaften**

Externcode	Ifd. Nr.	Interncode	belegt	Nennadresse	Zweitcode bei R	AZ
HALT	—	9D 471	B R	#	(+)	1,5
HAU	3.50	F2 742	B R	#	—	62+20 m
HB	1.50	FC 770	B R	#	—	15,5+12 m
HC	1.51	FD 771	B R	#	—	15,5+12 m
HLT	5.07	Pseudobefehl!				
HML	3.53	FA 762	B R	#	—	60 m +18m <sup>2</sup>
HN	4.18	CD 631	B	#	—	8,5
HNM	4.19	F8 760	B R	#	—	18+12 m
HSB	3.51	F1 741	B R	#	—	62+20 m
HSU	3.52	F3 743	B R	#	—	62+20 m
HWH	4.16	FB 763	B R	#	—	12+ 6 m
HWZ	4.17	F9 761	B R	#	—	12
HX	3.79	2E 132	B	#	—	14
HXN	3.80	2F 133	B	#	—	14,5
KBH	4.08	Pseudobefehl!				
KGZ	4.09	Pseudobefehl!				
KVH	4.06	9 4 450	R	=	—	12+3 · 1
KVZ	4.07	9 5 451	R	=	—	5+4 · 1
L	2.04	9A 462	B R	=	—	2
LA	2.05	8B 423	B R	=	—	3,5
LC	2.11	3 3 143	B	=	—	8,5
LK1	4.14	E 2 702	B R	#	—	33+10 m
LK2	4.15	E 3 703	B R	#	—	540+19 m
LL	2.18	1 0 040	B	#	—	1,5
LMC	2.13	31 141	B	=	—	8,5
LMT	2.12	3 2 142	B	=	—	8,5
M	7.03	0 8 020	B	II i	—	1,5
M2	3.71	7A 362	R	=	+	23
M2N	3.72	7 8 360	R	=	+	23
M2P	3.74	7 9 361	R	=	+	20
M2R	3.73	7B 363	R	=	+	20
MA	7.09	0 3 003	B	II N	—	1,5
MAB	7.11	2 0 100	B	#	—	8
MAP	3.23	5B 263	R	=	+	32,5
MAR	3.22	5 7 253	R	=	+	32,5
MB	7.10	0 2 002	B	II b	(+)	1,5
MC	7.07	1 4 050	B	II n <sub>3</sub>	+	3
MCE	7.08	1 7 053	B	II <..<N> <sub>3</sub> ..> <sub>3</sub>	+	-2+6 m
MCF	7.02	1 6 052	B	n <sub>3</sub>	+	3
MD	7.05	0 9 021	B	(II) i	—	7,5
MF	7.01	0B 023	B	i	—	1,5
MH	7.04	2D 131	B	II #	—	8,5
MLR	3.20	5 5 251	R	=	+	29,5
MM	7.06	0A 022	B	i	—	1,5
MNR	3.21	5 9 261	R	=	+	29,5
MU	7.16	0 5 011	B	#	—	14
NL	2.19	1 2 042	B	#	—	2

**Alphabetische Liste der Befehle  
und allgemeine Eigenschaften**

Externcode	Ifd. Nr.	Interncode	belegt	Nennadresse	Zweitcode bei R	AZ	wenn nicht gesprg.
NRM	4.02	9 F 473	R	=	—	F, F4: 4+m/2 G: 5,5+m/2 FG: 9+m/2	
QBR	1.52	F E 772	B R	#	—	31	
QCR	1.53	F F 773	B R	#	—	34	
R	7.19	9 6 452	B R	=	—	3	
REZ	3.46	6 5 311	R	=	+	120	
RL	7.18	3 F 173	B	(I + <BZ>) <sub>N</sub>	—	10	
RT	1.34	9 7 453	R	=	—	3,5	
S	5.02	3 6 152	B	=	—	0	
SAA	5.22	A 9 521	B R	=	—	0,5	3
SAT	5.23	A 8 520	B R	=	—	0,5	3
SB2	3.70	7 D 371	R	=	+	9,5	
SBA	3.17	9 9 461	R	=	—	11,5	
SBQ	3.19	7 F 373	R	=	+	7,5	
SBU	3.45	4 D 231	R	=	+	15	
SEG	5.28	9 3 443	B R	=	—	11,5	7
SG	5.14	A F 533	B R	=	—	8,5	11,5
SG0	5.10	A 6 512	B R	=	—	1	3
SGB	5.16	A B 523	B R	=	—	8,5	11,5
SH	4.01	9 B 463	R	=	—	binär: 11,5+I/8 dezimal: 9,5+I/2 Rundung: +1,5	
SI	5.12	A C 530	B R	=	—	8,5	9,5
SI0	5.08	A 4 510	B R	=	—	1	3
SK	5.15	A E 532	B R	=	—	8,5	11,5
SK0	5.11	A 5 511	B R	=	—	1	3
SKB	5.17	AA 522	B R	=	—	8,5	11,5
SL	5.31	1 E 072	B	#	—	10	3
SLL	5.32	1 F 073	B	#	—	10	3
SLN	5.33	1 C 070	B	#	—	10	3
SM	5.20	3 4 150	B	=	—	0	2,5
SMN	5.21	3 5 151	B	=	—	0	2,5
SN	5.13	AD 531	B R	=	—	8,5	9,5
SN0	5.09	A 7 513	B R	=	—	1	3
SNL	5.34	1 D 071	B	#	—	10	3
SR	5.18	B 8 560	B R	=	—	0,5	3
SRN	5.19	B A 562	B R	=	—	0,5	3
ST	5.29	9 0 440	B R	=	—	10	4,5
STN	5.30	9 1 441	B R	=	—	10	4,5
STW	—	Pseudocode!					
SU	5.04	3 8 160	B	=	—	2	
SW	5.35	1 B 063	B	#	—	10,5	3,5
SXG	5.26	2 5 111	B	#	—	1,5	3
SXI	5.24	2 4 110	B	#	—	1,5	3
SKK	5.27	2 6 112	B	#	—	1,5	3
SXN	5.25	2 7 113	B	#	—	1,5	3
SZB	5.03	3 A 162	B	=	—	0,5	
T	5.05	CC 630	B	n <sub>3,N</sub>	+	2,5	
TAX	1.43	9 C 470	B R	#	(+)	3,5	
TBC	1.45	0 7 013	B	#	—	8,5	

Int.  
Ext.-  
Code

Sp

2

0,5

C

LS

Lk

EA

**Alphabetische Liste der Befehle  
und allgemeine Eigenschaften**

Externcode	Ifd. Nr.	Interncode	belegt	Nennadresse	Zweitcode bei R	AZ
TCB	1.44	3 9 161	B	=	+	3,5
TDM	6.06	EA 722	B R	#	-	6,5 + 6,5 m
TL	6.01	E F 733	B R	#	-	3 + 8 m
TLB	6.02	ED 731	B R	#	-	3 + 8 m
TLD	6.05	EB 723	B R	#	-	7 + 8,5 m
TLI	6.03	EC 730	B R	#	-	3 + 6 m
TLM	6.04	EE 732	B R	#	-	3 + 6 m
TTX	1.40	0 D 031	B	#	-	11
TXA	1.41	8 C 430	B R	#	-	6,5
TXV	1.42	8 D 431	B R	#	-	7
TXX	1.39	0 C 030	B	#	-	5
V	3.75	1 3 043	B	#	-	5
VAQ	4.04	6 3 303	R	=	(+)	8,5
VC	3.77	1 5 051	B	#	+	6
VEL	3.54	6 8 320	R	=	+	2,5
VGF	4.03	Pseudobefehl!				185
VLA	3.58	8 8 420	R	=	-	7,5
VRK	4.12	E 1 701	B R	#	-	21,5 + 3,5 m + 1,5 p
VRL	4.13	9 E 472	R	=	-	32
VW1	-	Pseudocode!				-
VW2	-	Pseudocode!				-
VX	3.76	0 F 033	B	#	-	6
WTD	1.47	2 3 103	B	#	-	15 + 13 m
WTK	1.48	2 2 102	B	#	-	16 + 13 m
WTZ	1.49	2 1 101	B	#	-	10 + 12 J
XB	1.35	0 E 032	B	#	-	1,5
XBA	1.36	0 1 001	B	#	-	1,5
XC	1.37	1 8 060	B	#	-	2
XCN	1.38	1 9 061	B	#	-	3
Y	-	BC 570	B R	=	-	18,5
YB	-	3 C 170	B	=	+	3
YC	-	3 D 171	B	=	-	6
YF	-	3 7 153	B	=	-	0
YL	-	BD 571	B R	=	-	18,5
YP	-	B 9 561	B R	=	-	4,5
YU	-	B E 572	B R	=	-	6,5
YW	-	B F 573	B R	=	-	17,5
ZL	2.17	1 1 041	B	#	-	1,5
ZMC	2.14	3 0 140	B	=	-	8,5
ZMR	2.06	Pseudobefehl!				3
ZT0	2.07	C 8 620	B	=	-	8,5
ZT1	2.08	C 9 621	B	=	-	8,5
ZT2	2.09	CA 622	B	=	-	8,5
ZT3	2.10	CB 623	B	=	-	8,5
ZTR	2.03	9 2 442	R	=	-	3,5
ZU	2.20	3 E 172	B	=	-	2
ZUS	3.57	6 B 323	R	=	+	4,5
ZX	2.16	1 A 062	B	#	-	4

## Interncode → Externcode

1. Tetrade							2. Tetrade							1. Tetrade						
↓	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	↓			
0	0	XBA	MB	MA	EMU	MU	CWL	TBC	M	MD	MM	MF	TXX	TTX	XB	VX	0			
1	LL	ZL	NL	V	MC	VC	MCF	MCE	XC	XCN	ZX	SW	SLN	SNL	SL	SLL	1			
2	MAB	WTZ	WTK	WTD	SXI	SXG	SXK	SXN	EMB	E	ENZ	EZ	H	MH	HX	HXN	2			
3	ZMC	LMC	LMT	LC	SM	SMN	S	YF	SU	TCB	SZB	EA	YB	YC	ZU	RL	3			
4	FAB	FSBB	FA	FAC	FSBI	FSBD	FSB	FSBC	GSB	AU	GAC	GA	GSBD	SBU	GSBC	GSB	4			
5	FW	GW	GAB	gSBB	FML	MLR	FMLA	MAR	FMLN	MNR	FMAN	MAP	GMLN	GMAN	GML	GMLA	5			
6	FDV	DVD	FDV	VI	VAQ	GDV	REZ	GDVI	VEL	AUT	ET	ZUS	B3	B3V	B2	B2V	6			
7	B	BD	BQ	BH	BB	BN	BR	BNR	M2N	M2P	M2	M2R	A2	SB2	AQ	SBQ	7			
8	C	CR	CMT	CMR	CN	CB	CD	CQ	VLA	ATA	ETA	LA	TXA	TXV	BA	CH	8			
9	ST	STN	ZTR	SEG	KVH	KVZ	R	RT	AA	SBA	L	SH	TAX	HALT	VRL	NRM	9			
A	C2	C3	CMC	BC	SI0	SK0	SG0	SN0	SAT	SAA	SKB	SGB	SI	SN	SK	SG	A			
B								SR	YP	SRN		Y	YL	YU	YW	B				
C	DML	DMN	DDV	DDD	DAC	DSC	DA	DSB	ZT0	ZT1	ZT2	ZT3	T	HN		C				
D									BZ2	BZ	BQB	CZ	BAR	BAP	DR	BAV	D			
E		VRK	LK1	LK2	FPOL	gPOL	FMSK	gMSK			TDM	TLD	TLI	TLB	TLM	TL	E			
F	HA	HSB	HAU	HSU					HNM	HWZ	HML	HWH	HB	HC	QBR	QCR	F			
↑	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	↑			
1. Tetrade							2. Tetrade							1. Tetrade						

Int.  
Ext.-  
Code

Sp

## Zulässige Zweitcodes

Bei den Doppelcodebefehlen

**E, EZ, ENZ, EMB, MU, EMU, MAB und RL**

ist grundsätzlich jeder Befehl als Zweitcode zugelassen; praktisch kommen hauptsächlich solche Befehle in Frage, deren Adreßteil bei normalem Gebrauch die Adresse einer Speicherzelle oder der Operand ist.

2

Bei dem Doppelcodebefehl **R** sind als Zweitcode zugelassen:

B, BH, BD, BQ, BN, BB, BR, BNR, BQB, B2, B2V, B3, B3V, TCB,  
 VEL, AUT, ET, ZUS, DA, DSB, DML, DMN, DDV, DDD,  
 (F und G) A, AB, SB, SBB, SBI, SBD, ML, MLN, MLA, MAN, DV, DVI, W,  
 AQ, SBQ, MLR, MNR, MAR, MAP, DVD,  
 A2, SB2, M2, M2N, M2R, M2P, AU, SBU, REZ,  
 VC, MC, MCF, MCE, T

0,5

C

LS

Lk

## **Intern-Spezifikationen**

### **Intern-Spezifikationen in je 2 hexadekadischen Ziffern**

(Bedeutung der Extern-Spezifikationen siehe Beschreibung der Befehle im Programmierhandbuch)

#### **LA (Spezifikation im Erstadreßteil)**

extern:	F	2	E	3	H	T	V	M
intern:	80	40	20	10	08	04	02	01

#### **NRM (Spezifikation im Erstadreßteil)**

extern:	F	F4	FG	G
intern:	40	60	C0	80

#### **SH (Spezifikation im Zweitadreßteil)**

ext	int	ext	int	ext	int	ext	int
A	8 0	AL	A 0	AK	9 0	ALK	B 0
Q	4 0	QL	6 0	QK	5 0	QLK	7 0
AQ	C 0	AQL	E 0	AQK	D 0	AQLK	F 0
Z	0 8	ZL	2 8	ZK	1 8	ZLK	3 8
AD	8 2	ALD	A 2	AKD	9 2	ALKD	B 2
QD	4 2	QLD	6 2	QKD	5 2	QLKD	7 2
AQD	C 2	AQLD	E 2	AQKD	D 2	AQLKD	F 2
ZD	0 A	ZLD	2 A	ZKD	1 A	ZLKD	3 A
AR	8 4						
QR	4 4						
AQR	C 4						
ZR	0 C	ZLR	2 C				

#### **R (Spezifikation im Erstadreßteil)**

ext	int	ext	int	ext	int
A	81	AL	80	Z	89
Q	41	QL	40	W	49
D	21	DL	20	B	29
H	11	HL	10	Y	19

#### **RT (Spezifikation im Erstadreßteil)**

ext	int	ext	int	ext	int
AQ	C 0	QD	6 0	DH	30
AD	A 0	QH	5 0		
AH	9 0				

### **Intern-Spezifikationen in je 2 hexadekadischen Ziffern**

(Bedeutung der Extern-Spezifikationen siehe Beschreibung der Befehle im Programmierhandbuch)

## Intern-Spezifikationen

### Intern-Spezifikationen in je 2 hexadekadischen Ziffern

(Bedeutung der Extern-Spezifikationen siehe Beschreibung der Befehle im Programmierhandbuch)

<b>ST, STN, ZTR und L</b> (siehe auch folgende Gruppe!)		Spezifikation im Erstadreßteil							
ext	int	ext	int	ext	int	ext	int	ext	int
0A	20	1A	60	2A	A0	3A	E0		
0Q	10	1Q	50	2Q	90	3Q	D0		
0D	08	1D	48	2D	88	3D	C8		
0H	04	1H	44	2H	84	3H	C4		

<b>L</b> (siehe auch vorige Gruppe!)		Spezifikation im Erstadreßteil							
ext	int	ext	int	ext	int	ext	int	ext	int
0AQ	30	1AQ	70	2AQ	B0	3AQ	F0		
0AD	28	1AD	68	2AD	A8	3AD	E8		
0AH	24	1AH	64	2AH	A4	3AH	E4		
0QD	18	1QD	58	2QD	98	3QD	D8		
0QH	14	1QH	54	2QH	94	3QH	D4		
0DH	0C	1DH	4C	2DH	8C	3DH	CC		
0AQD	38	1AQD	78	2AQD	B8	3AQD	F8		
0AQH	34	1AQH	74	2AQH	B4	3AQH	F4		
0ADH	2C	1ADH	6C	2ADH	AC	3ADH	EC		
0QDH	1C	1QDH	5C	2QDH	9C	3QDH	DC		
0AQDH	3C	1AQDH	7C	2AQDH	BC	3AQDH	FC		

Sp

### **ZL, LL, NL, SL, SLL, SLN, SNL und SW** (Spezifikation im Erstadreßteil)

extern:	1	2	3	4	5	6	7	8	(0)
intern:	80	40	20	10	08	04	02	01	(00)

2

Bei den insgesamt 256 verschiedenen möglichen Kombinationen ergeben sich die Intern-Spezifikationen ebenfalls durch hexadekadische Addition, z. B.:

ext	int	ext	int	ext	int	ext	int	ext	int
12	C0	123	E0	1234	F0	12345	F8	123456	FC
13	A0	124	D0	1235	E8	12346	F4	123457	FA
14	90	125	C8	1236	E4	12347	F2	123458	F9
15	88	126	C4	1237	E2	12348	F1	123467	F6
16	84	127	C2	1238	E1	12356	EC	123468	F5
17	82	128	C1	1245	D8	12357	EA	123478	F3
18	81	134	B0	1246	D4	12358	E9	123567	EE
23	60	135	A8	1247	D2	12367	E6	123568	ED
24	50	136	A4	1248	D1	12368	E5	123578	EB
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

0,5

C

LS

**Intern-Spezifikationen in je 2 hexadekadischen Ziffern**  
(Bedeutung der Extern-Spezifikationen siehe Beschreibung der Befehle im Programmierhandbuch)

Lk

## TR 4 - Hexadencode

Einzelne und **paarweise** hexadekadische Verschlüsselung

### 1. Tetrade (paarweise)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4	8	C	G	K	O	S	W	[	leer	-	#	%	'	+
1	5	9	D	H	L	P	T	X	]	leer	=	;	&	*	:
2	6	A	E	I	M	Q	U	Y	x	¶	,	"	(	)	?
3	7	B	F	J	N	R	V	Z	WZ	Tab	.	/	)	<	!

### 1. Hexade (paarweise)

In der Tabelle rechts steht neben jedem Zeichen seine dezimale Verschlüsselung.

Bei Doppelangaben gilt die linke für die Schreibmaschine (wie bei der Tabelle oben), die rechte für den Schnelldrucker (Rechenzentrum TELEFUNKEN).

WZ Wagenrücklauf und Zeilensprung

ZE Zeilenende

leer Leerzeichen; wird ignoriert

¶ Leertaste bzw. Zwischenraum

Tab Tabulator

Beispiel für die einzelne Verschlüsselung:

Q = 26 = 0L L0L0 = 1A

Beispiel für die **paarweise** Verschlüsselung:

R + = 0L L0LL LL LL00 = 6FC

### 1. Tetrade (einzelne)

0	1	2	3
0	1	2	3
4	5	6	7
8	9	A	B
C	D	E	F

### 2. Tetrade (paarweise)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	G	16	W	32	#	48			0					
1	1	H	17	X	33		49			1					
2	2	I	18	Y	34	"	I	50		2					
3	3	J	19	Z	35	/	51			3					
4	4	K	20	[	36	%	52			4					
5	5	L	21	]	37	&	53			5					
6	6	M	22	X	38	(	54			6					
7	7	N	23	WZ	39	)	55			7					
8	8	O	24	leer	40	'	56			8					
9	9	P	25	leer	41	*	57			9					
A	10	Q	26	¶	42	>	58			A					
B	11	R	27	Tab	§	43	<	59		B					
C	12	S	28	-	44	+	60			C					
D	13	T	29	=	45	:	61			D					
E	14	U	30	,	46	?	62			E					
F	15	V	31	.	47	!	63			F					

Hexade (einzelne) bzw.  
2. Hexade (paarweise)

### 3. Tetrade (paarweise)

## Vorschubsteuerhexade für den Schnelldrucker

Der Papiervorschub erfolgt vor dem Drucken der Zeile; er wird von der ersten Hexade des ersten Wortes gesteuert. Das Wort kann mit Leercode (= Hexade 41) aufgefüllt werden.

### Aufbau der Steuerhexade

00000: Formularenden, hat nur Bedeutung im off-line-Betrieb:

Magnetband kann formularweise vor- und zurückgespult werden.

0L000: Listenende, hat nur Bedeutung im off-line-Betrieb:

Drucker hält an; Magnetband kann listenweise vor- und zurückgespult werden.

Der Papiervorschub kann unter Zuhilfenahme der Lochung in den Kanälen 2 bis 8 des Formatstreifens oder direkt durch Zeilenzählung gesteuert werden (auch gemischt):

000000: kein Vorschub;

0000L: Vorschub bis zur Lochung im Kanal 2;

000OL0: Vorschub bis zur Lochung im Kanal 3;

000OLL: Vorschub bis zur Lochung im Kanal 4;

000L00: Vorschub bis zur Lochung im Kanal 5;

000L0L: Vorschub bis zur Lochung im Kanal 6;

000LL0: Vorschub bis zur Lochung im Kanal 7;

000LLL: Vorschub bis zur Lochung im Kanal 8;

00L000: Vorschub um 1 Zeile

00L00L: Vorschub um 2 Zeilen

00L0L0: Vorschub um 3 Zeilen

00L0LL: Vorschub um 4 Zeilen

00LL00: Vorschub um 5 Zeilen

00LL0L: Vorschub um 6 Zeilen

00LLL0: Vorschub um 7 Zeilen

00LLL0: Vorschub um 8 Zeilen

# Verkehr mit dem Verteilerprogramm

## Ein- und Ausgabe über EA-Geräte mit Ausnahme der Kontrollschriftenmaschine SM

## **Startbefehle** (davor obligatorisch **EA P**)

- |             |   |   |
|-------------|---|---|
| <b>EA S</b> | Normaler Start eines EA-Gerätes;  | $\langle AC \rangle_3 = >STW<$ ; Abschluß mit <b>EA Z</b><br>(Löschen mit <b>EA A</b> ) |
| <b>EA F</b> | Fortstart eines EA-Gerätes, wenn möglich,<br>sonst normaler Start;  | $\langle AC \rangle_3 = >STW<$ ; Abschluß mit <b>EA Z</b><br>(Löschen mit <b>EA A</b> ) |
|             | Startsatz: <b>STW</b> s g    s = Spezifikation (s. u.); g = externe Gerätenummer (vorläufig 0 bis 19)                           |   |
|             | <b>VW1</b> f:a    f = ausgewähltes Fach (0 oder 1, nur bei Lochkarten)  |   |
|             | a = Anfangsadresse des EA-Speichers (Endadresse bei MB-Eingabe rückw.)  |   |
|             | <b>VW2</b> z    z = Länge des EA-Speichers (Anzahl der Ganzwörter, $2 \leq z \leq 4095$ )                                       |   |
|             | Spezifikation s:    S = Lochstreifeneingabe   | SA = Lochstreifenausgabe  |
|             | K = Lochkarteneingabe   | KA = Lochkartenausgabe  |
|             | B = Magnetbandeingabe vorwärts  | BA = Magnetbandausgabe  |
|             | BR = Magnetbandeingabe rückwärts  | D = Ausgabe auf on-line-Schnelldrucker  |
| <b>EA T</b> | Teilblockstart, d. h. Ein- bzw. Ausgabe mehrerer gleich langer Blöcke, hauptsächlich bei MB-Ausgabe für off-line-Schnelldrucker | $\langle AC \rangle_3 = >STW<$ ; Abschluß mit <b>EA Z</b><br>(Löschen mit <b>EA A</b> ) |
|             | Startsatz: wie bei <b>EA S</b> und <b>EA F</b> , nur <b>VW2 t:z</b> , wobei t = Anzahl der Blöcke ( $2 \leq t \leq 64$ )        |   |
| <b>EA L</b> | Magnetband stückweise löschen   | $\langle AC \rangle_3 = >STW<$  |
|             | Startsatz: <b>STW</b> s g    s = Spezifikation, (nach Belieben B oder BA);<br>g = externe Gerätenummer (vorläufig 0 bis 19)     |   |
|             | <b>VW2</b> z    z = Länge der Blocklücke (Anzahl der Ganzwörter, $2 \leq z \leq 4095$ )   |   |
| <b>EA U</b> | Magnetband umspulen;  | $\langle AC \rangle_3 = STW$  |
|             | Startsatz: <b>STW</b> s g    s = Spezifikation (s. u.);<br>g = externe Gerätenummer (vorläufig 0 bis 19)                        |   |
|             | Spezifikation s:    U = Umspulen ohne Rückmeldung; Kanal wird sofort frei, Gerät bleibt während des Umspulens belegt            |   |
|             | V = Umspulen mit Rückmeldung; Kanal und Gerät bleiben während des Umspulens belegt  |   |
|             | P = Umspulen mit Sperre; Kanal wird sofort frei, Gerät bleibt auch nach dem Umspulen belegt                                     |   |
|             | A = mit gleichzeitigem Löschen bis zum Bandende; sonst ohne Löschen   |   |
|             | R = rückwärts; sonst vorwärts   |   |
|             | Zulässige Kombinationen: U, V, P, UA, VA, PA, UR, VR, PR  |   |

### **Organisation zu den Startbefehlen**

- EA Z** Zieladresse setzen;  $\langle AC \rangle_3 = STW$   
 Startsatz: **STW** s g    s = Spezifikation, wie beim zugehörigen Startbefehl;  
                               g = externe Gerätenummer (vorläufig 0 bis 19)  
 Rückgabe der Regie nicht vor Blockendemeldung des ältesten offenen Startbefehls zur gleichen  
 Gerätenummer, im allgemeinen mit  $\langle CR \rangle = 0$ ; wenn  $\langle CR \rangle = L$ , dann Bandendemeldung.

**EA A** Aufgabe, d. h. Löschen aller offenen Startbefehle für  $\langle AC \rangle_3 = STW$   
 ein Gerät;  
 Startsatz: wie bei **EA Z**

**EA E** Erneuerung, d. h. Wiederholung aller offener Startbefehle  $\langle AC \rangle_3 = STW$   
 für ein Gerät, nachdem sie wegen einer Gerätestörung  
 gesperrt werden mußten;  
 Startsatz: wie bei **EA Z**

**EA R** Reservierung, d. h. vorläufige Stillegung aller offenen  $\langle AC \rangle_3 = STW$   
 Startbefehle für ein Gerät, nachdem sie wegen einer  
 Gerätestörung gesperrt werden mußten, um unter  
 Umgehung der Sperrre vor **EA E** andere Startbefehle zu  
 geben (nur geringe Bedeutung);  
 Startsatz: wie bei **EA Z**

## **Ein- und Ausgabe über die Kontrollschrreibmaschine SM**

- EA M** Zuerst Ausgabe, dann Eingabe über SM;  $\langle AC \rangle = \text{Startsatz}$   
Startsatz: Iz: e z = Anzahl der einzugebenden Ganzwörter ( $0 \leq z \leq 15$ )  
e = Anfangsadresse für Eingabe (im Fall z = 0 irrelevant)  
0 a a = Anfangsadresse für Ausgabe (höchstens 32 Wörter, soweit TK3)  
Rückgabe der Regie frühestens nach vollständiger Erledigung der Ausgabe bzw. Aus- und Eingabe.
- EA N** Ausgabe über SM;  $\langle AC \rangle_3 = a = \text{Anfangsadresse}$   
Rückgabe der Regie sofort. (höchstens 32 Wörter, soweit TK3) für Ausgabe
- EA X** Warten auf SM; Rückgabe der Regie frühestens nach Erledigung aller, auch von anderen Programmen, vorher angemeldeten EA-Vorgänge auf der SM.

## **Organisation des Programmablaufs**

- EA W** Sprung in Abhängigkeit vom internen Wahlschalter;  $\langle AC \rangle_3 = \text{Sprungadresse } s;$   
 $\langle HR \rangle = \text{rechtsbündige Intern-}$   
 $\text{darstellung des}$   
 $\text{internen Wahlschalters.}$   
Sprung nach s, wenn einer oder mehrere der im HR bezeichneten Wahlschalter gesetzt sind. (Die internen Wahlschalter können für jede Programmnummer vom Operateur gesetzt oder gelöscht werden.)
- EA P** Vorgabe der Fehleradresse f für ein EA-Gerät;  $\langle AC \rangle_3 = f;$   
 $\langle HR \rangle_3 = STW$   
**STW** wie bei **EA Z**. Die Fehleradresse muß vor dem ersten Startbefehl vorgegeben worden sein und darf später geändert werden. Falls bei dem im **STW** bezeichneten EA-Gerät eine vom Verteilerprogramm nicht behebbare Störung auftritt, wird aus dem Zieladressenprogramm (**EA Z**) nach f gesprungen
- EA V** Vorgabe der Haltadresse h;  $\langle AC \rangle_3 = h$   
Falls das Programm auf **EA H** läuft, wird mit  $\langle CR \rangle = 0$  nach h gesprungen; Falls das Programm auf Alarm läuft, wird mit  $\langle CR \rangle = L$  nach h gesprungen, nachdem alle Registerinhalte (BR, [CR, YZ, BZ, WZ, BA], AC, MQ, MD, HR) ab  $\langle h \rangle_3$  abgespeichert sind und das Fehlerwort im AC rechtsbündig bereitgestellt ist (2 letzte Tetraden):  
44 Überlauf WZ 08 Typenkennungsalarm  
20 Dreierprobenfehler im Vorrang 04 Befehlsfehler  
10 Arithmetischer Alarm 02 Dreierprobenfehler
- EA H** Sprung auf die von **EA V** vorgegebene Haltadresse h mit  $\langle CR \rangle = 0$ ;  
Ist keine Haltadresse vorgegeben worden, Ausführung von **HLT**.
- HLT** Löschen aller noch offenen Startbefehle, alle EA-Geräte betreffend, für die mit **EA P** eine Fehleradresse vorgegeben worden ist, wie durch **EA A**. Start des Nachfolgers mit gleicher Programmnummer, soweit vorhanden, sonst Regieabgabe.
- EA Y** Start eines Nebenprogramms;  $\langle AC \rangle_3 = >\text{Programmsatz}<$   
Programmsatz: 0 p p = Programmnummer ( $1 \leq p \leq 8$ )  
0 s s = Startadresse  
IwH: a w = intern dargestellter interner Wahlschalter;  
a =  $\langle AC \rangle_3$  beim Start des Programms  
Das Nebenprogramm muß mit **HLT** enden.
- EA G** Programmzieladresse setzen;  $\langle AC \rangle_3 = p (1 \leq p \leq 8)$   
Rückgabe der Regie frühestens nach Beendigung aller von diesem Programm unter der Programmnummer p mit **EA Y** gestarteten Nebenprogramme.

## Sonderbefehle (für das Betriebssystem)

- EA I** Abfrage der Belegung;  $\langle AC \rangle_3 = p$   
 Bei der Rückgabe der Regie gibt CR Auskunft, ob im Moment der Abfrage die Programmnummer p belegt war;  $\langle CR \rangle = L$  : belegt.
- EA O** Abbrechen;  $\langle AC \rangle_3 = p$   
 Das unter der Programmnummer p laufende Programm wird abgebrochen, als ob es auf **HLT** gelaufen wäre.
- EA Q** Unterbrechen;  $\langle AC \rangle_3 = p$   
 Das unter der Programmnummer p laufende Programm wird unterbrochen.
- EA C** Fortsetzen;  $\langle AC \rangle_3 = p$   
 Ein unter der Programmnummer p laufendes, aber durch **EA Q** unterbrochenes Programm wird fortgesetzt.
- BPN** Der interne Befehl SZB 0047 HU bringt die Programmnummer  
 Mit dem Pseudobefehl „Datum“ kann der Operateur zum Beginn eines Arbeitstages das Datum als Wort aus 8 Hexaden eingeben.  
**EA D** bringt das so eingegebene Datum nach AC

### Interndarstellungen (hexadekadisch)

**STW:** Operationsteil: 00

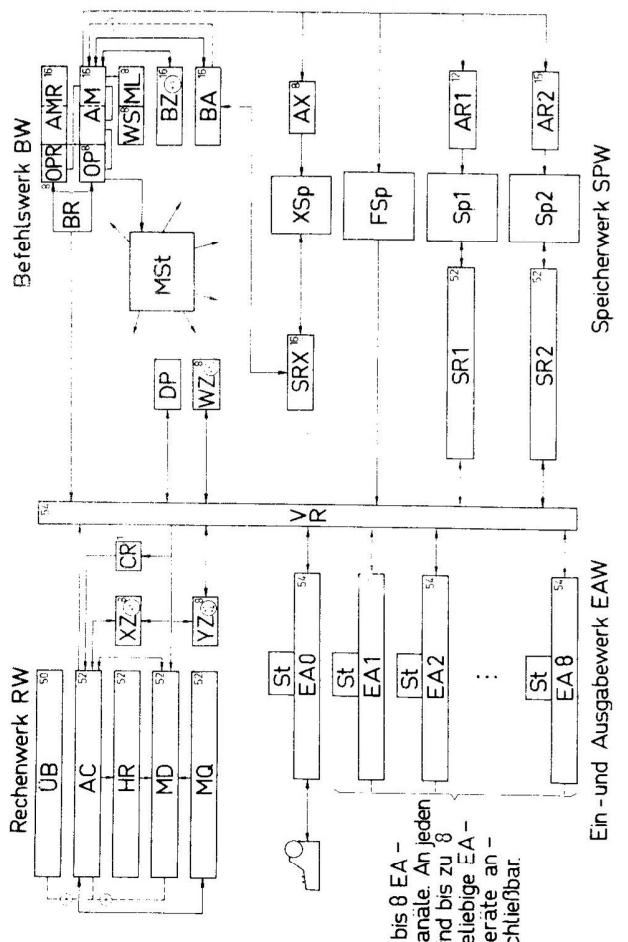
Adresse Teil:	extern	intern	extern	intern
S	0000	U	C000	
K	4000	V	4000	
B	C000	P	E000	
BR	D000	UA	C800	
SA	0800	VA	4800	
KA	4800	PA	E800	
BA	C800	UR	D000	
D	8800	VR	5000	
		PR	F000	

Die Gerätenummer (0 bis 7) steht außerdem dual rechtsbündig im Zweitadresse Teil, die Kanaladresse 2n-1 (n = Kanalnummer 1 bis 8) dual rechtsbündig im Erstdresse Teil.

- VW1:** Operationsteil: extern: 0: 1:  
 intern: 00 80 (nur bei Lochkarten, sonst immer 00) 02 mit LSE "gesch."  
 Adresse Teil: Anfangs- (bzw. End-) Adresse des EA-Speichers, dual, einschließlich Translationsadresse
- VW2:** Operationsteil: nur für **EA T**: (—t+1) dual rechtsbündig, sonst immer 00  
 Adresse Teil: F000 + z dual rechtsbündig.

Im Startzett für **EA M** steht z dual rechtsbündig im linken Operationsteil.

## BLOCKSCHAFTBILD



## Ein- und Ausgabewerk EAW

### Rechenwerk RW

- ÜB Übertragregister
- AC Akkumulator
- HR Hilfsregister
- MD Multiplikandenregister
- MQ Multiplikator-Quotientenregister
- CR Markenregister
- XZ Exponentenzähler
- YZ Schifftzähler

### Ein- Ausgabe-Werk

- EA0 Kanal f. Kontrollschrifbmasch.
- EA1...8 Ein- Ausgabe-Kanäle 1...8
- St Steuerelektronik
- VR Verteilerregister
- DP Dreierprobens-Prüfwerk
- MS Mikrogramm- Steuerwerk

## Speicherwerk SPW

- BR Befehlsregister
- OP Operationsteil
- OPR Operationsteil-Reserve
- AM Adreßteil
- AMR Adreßteil-Reserve
- WS Wahlschalterregister
- ML Merklicherregister
- BZ Befehlszähler
- BA Befehlsadressenregister
- WZ Unterprogramm-Ordnungszählz.

## Befehlswerk BW

- BR Befehlsregister
- OP Operationsteil
- OPR Operationsteil-Reserve
- AM Adreßteil
- AMR Adreßteil-Reserve
- WS Wahlschalterregister
- ML Merklicherregister
- BZ Befehlszähler
- BA Befehlsadressenregister
- WZ Unterprogramm-Ordnungszählz.

## WORTSTRUKTUR

