

03. Juli 1967

Ku



Tr

Fk
Z
L

TR 4

Gk
D. 2
Log. A.

Unterlagen für den Programmierer

SE
Aufber.

Ab

Int.
Ext-
Code

Sp

2

0,5

C

LS

Lk

EA

Vorwort

Die vorliegende Sammlung soll dem **angehenden Programmierer**, der sich mit Hilfe des Programmierhandbuches 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.

Inhalt	Inhalt
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 ₁ 2 Log. A _x
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
Konvertierungstabellen · 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).

Tr

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

Fk

Gruppe 1: Transportbefehle

Z
L

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

Gk

D, 2
Log. A

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

SE

- N = Adresse einer ganzen Arbeitsspeicherzelle
- N² = Adresse einer halben Arbeitsspeicherzelle
- N* = Wahladresse

Außer

- I bzw. J = Adresse einer Indexspeicherzelle, im Erst- bzw. Zweitadreßteil
- I^P bzw. J^P = Zahlwert (Parameter) im Erst- bzw. Zweitadreßteil
- I^S bzw. J^S = Spezifikation im Erst- bzw. Zweitadreßteil
- J^C = Zweitcode.

Ab

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

Int.
Ext-
Code

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.

Sp

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

2

Bedeutung der verwendeten Zeichen:

0,5

- ⌈ ⌋ rechtsbündig bzw. linksbündig
- <...> <...>₂ <...>₃ Inhalt von, Halbwort bzw. Drittelwort aus (= Adreßteil aus...)
- n i j Inhalt einer Speicherzelle N bzw. Indexspeicherzelle I oder J
- a h d q b Inhalt von AC, HR, MD, MQ bzw. BA
- M_N bzw. M_C Marke in der Speicherzelle N bzw. in CR
- t_N t_A t_H t_D t_Q TK in Speicherzelle N, in AC, HR, MD bzw. MQ
- t_{max} größte beteiligte TK
- ⇒ ⇔ geht nach, ergibt, Vertauschung von...

C

LS

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

Lk

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.

$\langle AC \rangle = 1, V, n_3^7$ 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 ²	Bringe zwei Halbwörter	$\langle N \rangle_2, \langle N+1 \rangle_2 \rightarrow AC; 0 \rightarrow YZ;$ wenn N ungerade, $q \rightarrow MD$		+	$t_{N, \langle N \rangle_2, \langle N+1 \rangle_2}$		$(t_{q,q})$		
1.12	B2	N ²	Bringe Halbwort	$0, n_2 \rightarrow AC$		+	$t_{N,0,n_2}$				
1.13	B2V	N ²	Bringe Halbw. m. Vorzeich.	$V, n_2 \rightarrow AC$		+	t_{N,V,n_2}				
1.14	B3	N ²	Bringe Drittelwort	$0, n_3 \rightarrow AC$		+	$t_{N,0,n_3}$				
1.15	B3V	N ²	Bringe Drittelw. m. Vorzch.	$V, n_3 \rightarrow AC$		+	t_{N,V,n_3}				
1.16	BA	N*	Bringe Adreßteil	$0, N \rightarrow AC$			$t_{N,N}$				siehe 2.01
1.17	BAV	N*	Bringe Adr.teil m. Vorzch.	$V, N \rightarrow AC$			$t_{N,V,N}$				siehe 2.02
1.18	BAR	N*	Bringe Adr.teil u. reserviere	$a \rightarrow HR; 0, N \rightarrow AC$			$t_{N,N}$	$t_{A,a}$			
1.19	BAP	N*	Br. Adr.teil m. Vz. u. reserv.	$a \rightarrow HR; V, N \rightarrow AC$			$t_{N,V,N}$	$t_{A,a}$			
1.20	BC	N	Bringe und Speichere	$a \leftarrow n; M_N$ bleibt erhalten, wenn $t_A \leq 1$ und $t_N \leq 1$	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_D, 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		$t_{N,h}$				
1.28	CZ	N	Speichere zwei Wörter	$a \rightarrow N; q \rightarrow N+2$	t_A, a						$\langle N+2 \rangle =$ t_Q, q
1.29	C2	N ²	Speichere Halbwort	$a_2 \rightarrow n_2$				$t_{N,n_{neu}}$			
1.30	C3	N ²	Speichere Drittelwort	$a_3 \rightarrow n_3$				$t_{N,n_{neu}}$			
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, a						Nur $t_A \leq 1$
1.33	CMC	N	Spch. m. Marke aus Spch.	$a \rightarrow N; M_N$ bleibt erhalten	t_A, M_N, a						Nur $t_A \leq 1$ und $t_N \leq 1$
1.34	RT	I ⁸	Register tauschen	Vertauschung der zwei durch $A = AC, H = HR, D = MD$ oder $Q = MQ$ bezeichneten Register; alle anderen Register bleiben erhalten.							
1.35	XB	I	Index: Bringe	$i \rightarrow BA$						i	siehe 2.15
1.36	XBA	N*	Index: Bringe Adreßteil	$N \rightarrow BA$						N	
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	$i \rightarrow J; i \rightarrow BA$				i			
1.40	TTX	J: I	Tausch-Transport im Indexspeicher	$i \rightarrow J; i \rightarrow BA$				i			
1.41	TXA	I	Trspt. a. Indexsp. nach AC	$0, i \rightarrow AC; i \rightarrow BA$				$t_{N,i}$	i		$i = 0$ immer als $-0!$
1.42	TXV	I	Trspt. a. Indexsp. n. AC m. Vorz.	$V, i \rightarrow AC; i \rightarrow BA$				$t_{N,V,i}$	i		
1.43	TAX	I	Trspt. a. AC nach Indexsp.	$a_2 \rightarrow I; a_2 \rightarrow BA$			a_2			a_2	
1.44	TCB	N ²	Trspt. a. Speicher nach BA	$n_3 \rightarrow BA$						n_3	
1.45	TBC	N ²	Trspt. a. BA nach Speicher	$b \rightarrow n_3$							
1.46	CWL	N ²	Speichere WS und ML	$WS, ML \rightarrow n_3; 0 \rightarrow ML$							
1.47	WTD	JP: I	Wortgruppen-Transport mit Dehnung	$\langle I \rangle + 2k \rightarrow \langle I+1 \rangle + 2k$ 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	JP: I	Wortgruppen-Transport mit Komprimierung	$\langle I \rangle + 2k \rightarrow \langle I+1 \rangle + 2k$ 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; $1 \leq J \leq 127$						$\langle I \rangle$ und $\langle I+1 \rangle$ bleiben unverändert	
1.49	WTZ	JP: I	Wortgruppen-Transport mit Zählung	$\langle I \rangle + 2k \rightarrow \langle I+1 \rangle + 2k$ für $k = 0, 1, 2, \dots (J-1)$; $\langle I+1 \rangle + 2 \cdot (J-1) \rightarrow BA$; $1 \leq J \leq 255$							
1.50	HB	N	Höhere Genauigkeit: Bringe	$\langle N+2 \rangle \rightarrow X 17; + 2k$ 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	$\langle X 17 \rangle + 2k \rightarrow N + 2k$ für $k = 0, 1, 2, \dots, m$; Beendigung wie WTD, evtl. TK-Alarm; m ist im allg. die aus dem linken Adreßteil von $\langle X 17 \rangle$ zu entnehmende Mantissenlänge; $L \rightarrow CR$, wenn $\langle X 17 \rangle$ markiert ist; $\langle BZ \rangle \rightarrow BA; 1 \leq m \leq 15$							Nur TK0
1.52	QBR	N	Bequemes Bringen aller Register	$\langle N \rangle \rightarrow HR;$ $\langle N-2 \rangle \rightarrow MD;$ $\langle N-4 \rangle \rightarrow MQ;$ $\langle N-6 \rangle \rightarrow AC;$							
1.53	QCR	N	Bequemes Speichern aller Register	$2 + \langle CR \rangle \rightarrow TK$ von n ; $\langle YZ \rangle \rightarrow$ link. Op.teil von n ; $\langle BZ \rangle \rightarrow$ link. Adr.teil von n ; $\langle WZ \rangle \rightarrow$ recht. Op.teil von n ; $\langle BA \rangle \rightarrow$ recht. Adr.teil von n ;							
Beim Transport in oder aus den vier großen Rechenwerksregistern werden alle 48 Informationsbits unverändert übertragen, auch wenn die $TK \leq 1$ ist.											

Beim Transport in oder aus den vier großen Rechenwerksregistern werden alle 48 Informationsbits unverändert übertragen, auch wenn die $TK \leq 1$ ist.

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 ^S	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 Q = MQ. Die angegebene TK wird dem angegebenen Register erteilt. Die 48 Informationsbits bleiben erhalten.	
2.04	L	I ^S	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 ^S	Lösche im AC	I besteht aus genau einer Bezeichnung F = erste 40 Inf.-Bits = Mantissenteil 2 = erste 24 Inf.-Bits = linkes Halbwort E = letzte 8 Inf.-Bits = Exponententeil 3 = letzte 16 Inf.-Bits = rechter Adreßteil (Drittelwort) 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	Pseudobefehl! L → CR: MD wird verändert.	
2.07	ZT0	N	Setze TK0 (im Speicher)	N erhält TK 0;	Die 48 Informationsbits von N bleiben erhalten.
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;	
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 $t_N \leq 1$
2.13	LMC	N	Lösche Marke im Speicher	In N wird Marke gelöscht, falls vorhanden; TK und übrige Informationsbits bleiben erhalten.	Nur $t_N \leq 1$
2.14	ZMC	N	Setze Marke im Speicher	In N wird Marke gesetzt, falls noch nicht vorhanden; TK und übrige Informationsbits bleiben erhalten.	Nur $t_N \leq 1$
2.15	XBA	N*	Index: Bringe Adreßteil	N → BA	Siehe 1.36
2.16	ZX	J ^P ; I	Setze Index	J → I; J → BA; J ≤ 127	J = ± 0 nur als - 0!
2.17	ZL	I ^S	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 ^S	Lösche Merklicht	I wie bei ZL (2.17). Die bezeichneten ML werden gelöscht, alle übrigen bleiben erhalten. I = 0: Leerbefehl!	
2.19	NL	I ^S	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.cd.	Adr.	Bezeichnung	Wirkung	$\langle N \rangle$	$\langle CR \rangle$	$\langle AC \rangle$	$\langle MD \rangle$	$\langle MQ \rangle$	$\langle YZ \rangle$	Bemerkgn.
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		+	$1, a:n$	$1, 0$	$1, \text{Rest}$	0	
3.14	FDVI	N	Dividiere invers	$n : a \rightarrow AC$; Rest $\rightarrow MQ$; Rest hat gleiches Vorz. w. Dividend		+	$1, n:a$	$1, 0$	$1, \text{Rest}$	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.			$t_A, a + N$ $2, a + N$	$1, 0, N$ $1, 0, N$			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.			$t_A, a - N$ $2, a - N$	$1, 0, N$ $1, 0, N$			Siehe 3.43 TK3 mit Vorbehalt
3.18	AQ	N	Addiere in MQ	$a, q + n \cdot 2^{-48} \rightarrow AC, MQ$		+	$1, \Sigma_I$	$1, n$	$1, \Sigma_{II}$	0	
3.19	SBQ	N	Subtrahiere in MQ	$a, q - n \cdot 2^{-48} \rightarrow AC, MQ$		+	$1, \Delta_I$	$1, n$	$1, \Delta_{II}$	0	
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	Multipl. negativ mit Rundung	$-a \cdot n \rightarrow AC$		+	$1, -a \cdot n$	$1, n$	$1, 0$	0	
3.22	MAR	N	Multipl. akkumul. m. Rundung	$h + a \cdot n \rightarrow AC$		+	$1, \Sigma$	$1, n$	$1, 0$	0	
3.23	MAP	N	Multipl. 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		+	$1, \text{Quotient}$	$1, 0$	$1, \text{Rest}$	0	
3.25	FPOL	N	Berechne Polynom	$\sum_{k=0}^p \langle N + 2k \rangle \cdot a^{N-k} \rightarrow AC$; $a \rightarrow HR$; $\langle N + 2p \rangle \rightarrow MD$; $0 \rightarrow MQ$; $L \rightarrow CR$; $0 \rightarrow YZ$; $N + 2p \rightarrow BA$; Beendigung durch Marke in $\langle N + 2p \rangle$							Nur TK1
3.26	FMSK	JP: I	Multipliziere skalar	$\sum_{k=0}^p \langle \langle I \rangle + 2k \rangle \cdot \langle \langle I + 1 \rangle + 2k \rangle \rightarrow AC, MQ$; $\sum_{k=0}^{p-1} \rightarrow HR$; $\langle \langle I \rangle + 2p \rangle \rightarrow MD$; CR wird durch Marke in $\langle \langle I + 1 \rangle + 2k \rangle$ gesetzt; $0 \rightarrow YZ$; $\langle \langle I \rangle + 2p + 2 \rangle \rightarrow BA$; Beendigung durch Marke in $\langle \langle I \rangle + 2p \rangle$; $1 \leq J \leq 127$							

TK ≥ 2
nur mit Vorbehalt

Nur TK1

Siehe 3.43
TK3 mit Vorbehalt

Nur TK1

Fk
Z
L

Gk
D, 2
Log. A

SE
Aufber.

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	CR	AC	MD	MQ	YZ	Bemerkgn.
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	$n + a \rightarrow N$; 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	GABB	N	Subtrahiere Betrag	$a - n \rightarrow AC$		+	$0, a - n $	$0, n$	$0, 0$	s	
3.32	GSBC	N	Subtrahiere im Speicher	$n - a \rightarrow N$; 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	$d - n \rightarrow AC$		+	$0, d - n$		$0, 0$	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	$h - a \cdot n \rightarrow AC$		+	$0, \Sigma$	$0, n$	$0, 0$	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: $a \cdot 16^{-1} \rightarrow AC$; $I = \text{letzte 8 Bits von } N; I \leq 127$			$0, a \cdot 16^{-1}$	$1, 0, I$			Siehe 3.16
3.43	SBA	N*	Subtrahiere Adreßteil	TK0: $a \cdot 16^{-1} \rightarrow AC$; $I = \text{letzte 8 Bits von } N; I \leq 127$			$0, a \cdot 16^{-1}$	$1, 0, I$			Siehe 3.17
3.44	AU	N	Addiere unnormalisiert	$a + n \rightarrow AC$; unnormalisiert		+	$0, a + n$	$0, n$	$0, 0$	0	
3.45	SBU	N	Subtrahiere unnormalisiert	$a - n \rightarrow AC$; 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	$\sum_{k=0}^p N + 2k \cdot a^{p-k} \rightarrow AC$; $a \rightarrow HR$; $\langle N + 2p \rangle \rightarrow MD$; $0 \rightarrow MQ$; $L \rightarrow CR$; $s \rightarrow YZ$ (Addition von $\langle N + 2p \rangle$); $N + 2p \rightarrow BA$; Beendigung durch Marke in $\langle N + 2p \rangle$							
3.48	GMSK	J ² ; I	Multipliziere skalar	$\sum_{k=0}^p \langle I + 2k \rangle \cdot \langle I + 1 + 2k \rangle J \rightarrow AC$; $\sum_{k=0}^{p-1} I \rightarrow HR$; $\langle I \rangle + 2p \rightarrow MD$; $0 \rightarrow MQ$; CR wird durch Marke in $\langle I + 1 + 2k \rangle J$ gesetzt; $s \rightarrow YZ$ (Addition des letzten Produkts); $\langle I \rangle + 2p + 2 \rightarrow BA$; Beendigung durch Marke in $\langle I \rangle + 2p$; $1 \leq J \leq 127$							
3.49	HA	N	Höhere Genauigkeit: Addiere	$\langle X 17 \rangle_{rr} + \langle N \rangle_{rr}$ normalisiert $\rightarrow \langle X 17 \rangle_{rr}$; $\langle X 17 \rangle_{rr}$ legt Mantissenlänge fest; TK0, $+0 \rightarrow AC$, $\rightarrow HR$, $\rightarrow MD$ und $\rightarrow MQ$; CR wird durch Marke im Kopfwort des Speicheroperanden gesetzt; $0 \rightarrow YZ$; $\langle BZ \rangle$ oder $\langle X 17 \rangle \rightarrow BA$							Nur TK0
3.50	HAU	N	Höhere Genauigkeit: Addiere unnormalisiert	$\langle X 17 \rangle_{rr} + \langle N \rangle_{rr}$ unnormalisiert $\rightarrow \langle X 17 \rangle_{rr}$; $\langle X 17 \rangle_{rr}$ legt Mantissenlänge fest; TK0, $+0 \rightarrow AC$, $\rightarrow HR$, $\rightarrow MD$ und $\rightarrow MQ$; CR wird durch Marke im Kopfwort des Speicheroperanden gesetzt; $0 \rightarrow YZ$; $\langle BZ \rangle$ oder $\langle X 17 \rangle \rightarrow BA$							
3.51	HSB	N	Höhere Genauigkeit: Subtrahiere	$\langle X 17 \rangle_{rr} - \langle N \rangle_{rr}$ normalisiert $\rightarrow \langle X 17 \rangle_{rr}$; $\langle X 17 \rangle_{rr}$ legt Mantissenlänge fest; TK0, $+0 \rightarrow AC$, $\rightarrow HR$, $\rightarrow MD$ und $\rightarrow MQ$; CR wird durch Marke im Kopfwort des Speicheroperanden gesetzt; $0 \rightarrow YZ$; $\langle BZ \rangle$ oder $\langle X 17 \rangle \rightarrow BA$							
3.52	HSU	N	Höhere Genauigkeit: Subtrahiere unnormalisiert	$\langle X 17 \rangle_{rr} - \langle N \rangle_{rr}$ unnormalisiert $\rightarrow \langle X 17 \rangle_{rr}$; $\langle X 17 \rangle_{rr}$ legt Mantissenlänge fest; TK0, $+0 \rightarrow AC$, $\rightarrow HR$, $\rightarrow MD$ und $\rightarrow MQ$; CR wird durch Marke im Kopfwort des Speicheroperanden gesetzt; $0 \rightarrow YZ$; $\langle BZ \rangle$ oder $\langle X 17 \rangle \rightarrow BA$							
3.53	HML	N	Höhere Genauigkeit: Multipliziere	$\langle X 17 \rangle_{rr} \cdot \langle N \rangle_{rr}$ normalisiert $\rightarrow \langle X 17 \rangle_{rr}$; $\langle X 17 \rangle_{rr}$ legt Mantissenlänge fest; TK0, $+0 \rightarrow AC$, $\rightarrow HR$, $\rightarrow MD$ und $\rightarrow MQ$; CR wird durch Marke im Kopfwort des Speicheroperanden gesetzt; $0 \rightarrow YZ$; $\langle BZ \rangle$ oder $\langle X 17 \rangle \rightarrow BA$							

Kurzbeschreibung der Befehle

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

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung	<N>	<CR>	<AC>	<HR>	<MD>	<MQ>	<YZ>	Bemerk.
3.54	VEL	N	Vel	$a \vee n \rightarrow AC$		+	$t_{max}, a \vee n$		$t_{N,n}$			
3.55	AUT	N	Aut	$a \oplus n \rightarrow AC$		+	$t_{max}, a \oplus n$		$t_{N,n}$			
3.56	ET	N	Et	$a \& n \rightarrow AC$		+	$t_{max}, a \& n$		$t_{N,n}$			
3.57	ZUS	N	Setze zusammen	$(h \& a) \vee (h \& n) \rightarrow AC$		+	$t_{max}(AC-N),$ Zus		$t_{N,n}$			
3.58	VLA	N*	Vel Adreßteil	$h \vee (0, N) \rightarrow AC$			$t_{H_i},$ $h \vee (0, N)$		1,0,N			
3.59	ATA	N*	Aut Adreßteil	$h \oplus (0, N) \rightarrow AC$			$t_{H_i},$ $h \oplus (0, N)$		1,0,N			
3.60	ETA	N*	Et Adreßteil	$h \& (0, N) \rightarrow AC$			$t_{H_i},$ $h \& (0, N)$		1,0,N			
3.61	DA	N	Dezimale Addition	$a + n \rightarrow AC$		+	$1, a + n$	1,0	1,0	1,0		
3.62	DAC	N	Dezimale Addition im Speicher	$n + a \rightarrow N;$ M_N bleibt erhalten	1, n + a	+	$1, n + a$	1,0	1, n + a	1,0		
3.63	DSB	N	Dezimale Subtraktion	$a - n \rightarrow AC$		+	$1, a - n$	1,0	1,0	1,0		
3.64	DSC	N	Dezimale Subtraktion im Speicher	$n - a \rightarrow N;$ M_N bleibt erhalten	1, n - a	+	$1, n - a$	1,0	1, n - a	1,0		
3.65	DML	N	Dezimale Multiplikation	$a \cdot n \rightarrow AC, MQ$		+	$1, \Pi_I$	1,0	1, n	$1, \Pi_{II}$	0	
3.66	DMN	N	Dezimale Multiplikation negativ	$-a \cdot n \rightarrow AC, MQ$		+	$1, -\Pi_I$	1,0	1, n	$1, -\Pi_{II}$	0	Nur TK1
3.67	DDV	N	Dezimale Division	$a : n \rightarrow AC; \text{Rest} \rightarrow MQ;$ Rest hat gleiches Vorz. w. Divid.		+	$1, a : n$	1,0	1,0	1, Rest	0	
3.68	DDD	N	Dezimale Division doppelt lang	$\langle AC, MQ \rangle : n \rightarrow AC; \text{Rest} \rightarrow MQ;$ Rest hat gleiches Vorz. w. Divid.		+	1, Quotient	1,0	1,0	1, Rest	0	
3.69	A2	N ²	Addiere Halbwort	$a_2^{-1} + n_2 \rightarrow a_2^{-1}$		+	$t_{max},$ $V, a_2 + n_2$		$t_{A,0}$			TK ≥ 2 nur mit Vor-behalt
3.70	SB2	N ²	Subtrahiere Halbwort	$a_2^{-1} - n_2 \rightarrow a_2^{-1}$		+	$t_{max},$ $V, a_2 - n_2$		$t_{A,0}$			
3.71	M2	N ²	Multipliziere Halbwort	$a \cdot n_2^{-1} \rightarrow AC$		+	$1, a \cdot n_2^{-1}$	1,0	1,0	0		
3.72	M2N	N ²	Multipliziere Halbwort negativ	$-a \cdot n_2^{-1} \rightarrow AC$		+	$1, -a \cdot n_2^{-1}$	1,0	1,0	0		
3.73	M2R	N ²	Multipliziere Halbwort mit Rundung	$\lceil a \cdot n_2 \rceil \rightarrow AC$		+	$1, \lceil a \cdot n_2 \rceil$	1,0	1,0	0		Nur TK1
3.74	M2P	N ²	Multipliziere Halbwort negativ mit Rundung	$\lceil -a \cdot n_2 \rceil \rightarrow AC$		+	$1, \lceil -a \cdot n_2 \rceil$	1,0	1,0	0		
3.75	V	N*	Vergleiche	$b - N \rightarrow BA$								
3.76	VX	I	Vergleiche mit Indexspeicher	$b - i \rightarrow BA$								
3.77	VC	N ²	Vergleiche mit Speicher	$b - n_2^{-1} \rightarrow BA$								
3.78	H	J ² ; I	Erhöhe	$i + j \rightarrow I; i + j \rightarrow BA$ $ j \leq 127$								
3.79	HX	J : I	Erhöhe aus Indexspeicher	$i + j \rightarrow I; i + j \rightarrow BA$								
3.80	HXN	J : I	Erhöhe aus Indexspeicher negativ	$i - j \rightarrow I; i - j \rightarrow BA$								

Gk
D, 2
Log. A_x

SE
Aufber.

Ab

Int.
Ext.-
Code

Sp

2

0,5

C

LS

Lk

EA

Kurzbeschreibung der Befehle

4. Aufbereitungsbefehle

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung	Bemerkungen
4.01	SH	J ^S I ^P	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	I ^S	Normalisiere	Normalisierung gemäß der Spezifikation I: (nur tetradenweise nach links) F = Festkommazahl (s = Anzahl der Binärschifte → YZ); F4 = wie F, aber s = Anzahl der Tetradenschifte; G = Gleitkommazahl (s = Anzahl der Binärschifte → YZ); FG = Festkommazahl, linksbündig, in Gleitkommazahl (0 → YZ).	
4.03	VGf	keine	Verwandle Gleitkommazahl in Festkommazahl	Pseudobefehl! Verwandlung der in AC gegebenen Gleitkommazahl x in Festkommazahl $16 \cdot x \rightarrow AC, MQ$; <MD> und <BA> werden verändert. Der Bezugsexponent e muß im Adreßteil des folgenden (Leer-) Befehls stehen. WZ muß vorher mit ZU gesetzt worden sein!	
4.04	VAQ	keine	Vorzeichenangleich zwischen AC und MQ	Angleichung des Vorzeichens 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 → HR, → MD und → MQ; Adreßteil N bedeutungslos.	Nur TK1
4.06	KVH	I ^P	Konvertiere hin	<AC, MQ> dezimal, höchstens I-stellig → AC, dual, exakt, mit TK1; TK1, + 0 → HR, → MD und → MQ; $0 \leq I \leq 13$; nur für positive ganze Zahlen.	
4.07	KVZ	I ^P	Konvertiere zurück	<AC, dual, mit TK0 oder TK1 → AC, MQ> dezimal, abgerundet, mit TK1, I-stellig; Konvertierungsrest → MD mit TK1; $0 \leq I \leq 13$; nur für positive echte Brüche.	Nur TK ≤ 1
4.08	KBH	keine	Konvertiere Bruch hin	Pseudobefehl! <AC, MQ> dezimal, 13-stellig → <AC, dual, normal gerundet, mit TK1; TK1, + 0 → HR und → MQ; TK1, $2^{24} \cdot 10^{-13} = 384B\ 84D0\ 92ED$ → MD; $0 \rightarrow YZ$; <BZ> → BA; nur für positive echte Brüche.	
4.09	KGZ	keine	Konvertiere ganze Zahl zurück	Pseudobefehl! <AC> dual, mit TK1 → <AC, MQ> dezimal, exakt, mit TK1; TK1, $10^{13} \cdot 1 = 0918\ 4E72\ 9FFF$ → HR; bedeutungsloser Konvertierungsrest → MD; $0 \rightarrow YZ$; <BZ> → BA; nur für positive ganze Zahlen.	Nur TK1
4.10	KOH	keine	Konvertiere Gleitkommazahl (TK0) hin;	Pseudobefehl! Verwandlung eines im Lebenspeicher befindlichen string in eine im AC befindliche duale Gleitkommazahl. Näheres siehe Programmbeschreibung; Dauer etwa 3700 Mikrosekunden.	
4.11	KOZ	keine	Konvertiere Gleitkommazahl (TK0) zurück;	Pseudobefehl! Verwandlung einer im AC befindlichen dualen Gleitkommazahl in einen im Lebenspeicher befindlichen string. Näheres siehe Programmbeschreibung; Dauer etwa 1700 Mikrosekunden.	
4.12	VRK	I	Verkürze	Verkürzung von höchstens 13 Ziffernhexaden (≤ 15), beginnend mit den ersten <I> Hexaden von <I + 1>, unter Wegfall der Füllhexaden (40 bis 43) zu Tetraden → AC, MQ, mit TK1; + 0, TK1 → HR und → MD; Anzahl der Ziffern → YZ; Grenzhexade in ihrer Speicherzelle an das linke Ende geschiftet; Adresse der Zelle, der die letzte Hexade entnommen wurde, → I + 1; neuer Hexadenzählerstand (0 bis 7) → BA.	Nur TK3
4.13	VRL	I ^P	Verlängere	Verlängerung der letzten 13 Tetraden aus AC, MQ (erste 11 Tetraden = + 0!) zu Ziffernhexaden → AC, MQ, mit TK3, mit Einfügung des (Dezimal-)Punkts nach der I. Ziffer; vor dem Punkt wenigstens 1 Ziffer; sonst <I> anstelle führender Nullen vor dem Punkt; <I> auch anstelle Punkt am Ende; TK3, + 0 → HR und → MD; $0 \leq I \leq 13$.	TK beliebig
4.14	LK1	N	Lochkartenbefehl 1	Hineinschreiben der in der ersten Tetraden von MD stehenden Zeilennummer (0 bis 12) von links in die Zellen des bei N beginnenden halben Spaltenbildes gemäß der in den ersten 40 Bits von AC befindlichen Halbzeile; TK3, + 0 → AC und → MQ; $0 \leq YZ$.	Nur TK3
4.15	LK2	J : I	Lochkartenbefehl 2	Aufbau des bei I beginnenden Zellenbildes einer Lochkarte vor dem Ausstanzen aus dem bei J beginnenden Spaltenbild; TK3, + 0 → AC, → MD und → MQ; $0 \leq YZ$; $I + 160 \rightarrow BA$. AC → X 17, mit I Mantissenzellen; $1 \leq I \leq 15$; TK0, + 0 → AC; $0 \rightarrow YZ$; <X 17> → BA.	
4.16	HWH	I ^P	Höhere Genauigkeit: Wandle hin	<N> _{rr} → AC; <N> → MD; $0 \rightarrow YZ$.	
4.17	HWZ	N	Höhere Genauigkeit: Wandle zurück	— <N> _{rr} → N _{rr} durch Invertierung des Vorzeichenbits im Kopfwort.	Nur TK0
4.18	HN	N	Höhere Genauigkeit: Negiere	Normalisierung von <N> _{rr} ; TK0, + 0 → AC, → HR, → MD und → MQ; $0 \rightarrow YZ$; N → BA.	
4.19	HNM	N	Höhere Genauigkeit: Normalisiere		

5. Sprungbefehle

Lfd.Nr.	Ext.cd.	Adr.	Bezeichnung	Wirkung	Bemerkungen
5.01	0	N*	Leerbefehl	Keine (außer der selbstverständlichen <BZ> + 1 → BZ); Adreßteil N bedeutungslos.	
5.02	S	N ²	Springe	N → BZ	
5.03	SZB	N ²	Springe mit BZ → BA	<BZ> _{neu} → BA; N → BZ	
5.04	SU	N ²	Springe in Unterprogramm	<WZ> + 1 → WZ; <BZ> _{neu} → <WZ> _{neu} ; N → BZ	
5.05	T	N ²	Tu!	Ausführung des Befehls n ₂ .	
5.06	EA	(N ²)	Ein- und Ausgabe	Unterprogramm sprung in den durch externe symbolische Adresse bezeichneten Zweig des Verteilerprogramms; im Verteilerprogramm bleiben nur WZ und ML unverändert.	
5.07	HLT	keine	Halt	Pseudobefehl! Sprung in speziellen Zweig des Verteilerprogramms zur Beendigung des Programms.	
5.08	SIO	N ²	Springe, wenn identisch 0	N → BZ, w. <AC> = 0;	Die Ausführung des Vergleichs wird von der TK von <AC> bestimmt. a - MD; Die Ausführung des Vergleichs wird von der höheren der beiden TK von <AC> und <HR> bestimmt; ist t _A = t _H = 0, so wird <HR> normalisiert; dabei 0 → YZ.
5.09	SNO	N ²	Springe, wenn nicht ident. 0	N → BZ, w. <AC> ≠ 0;	
5.10	SGO	N ²	Springe, wenn größer als 0	N → BZ, w. <AC> > 0;	
5.11	SKO	N ²	Springe, wenn kleiner als 0	N → BZ, w. <AC> < 0;	
5.12	SI	N ²	Springe, wenn identisch	N → BZ, w. <AC> = <HR>;	
5.13	SN	N ²	Springe, wenn nicht identisch	N → BZ, w. <AC> ≠ <HR>;	
5.14	SG	N ²	Springe, wenn größer	N → BZ, w. <AC> > <HR>;	
5.15	SK	N ²	Springe, wenn kleiner	N → BZ, w. <AC> < <HR>;	
5.16	SCB	N ²	Springe, w. größerer Betrag	N → BZ, w. <AC> > <HR>;	
5.17	SKB	N ²	Springe, w. kleinerer Betrag	N → BZ, w. <AC> < <HR>;	
5.18	SR	N ²	Springe, wenn rechtes Bit L	N → BZ, wenn das letzte Bit in AC den Wert L hat.	
5.19	SRN	N ²	Springe, w. rech. Bit nicht L	N → BZ, wenn das letzte Bit in AC den Wert 0 hat.	
5.20	SRM	N ²	Springe, wenn Marke	N → BZ, wenn <CR> = L, und 0 → CR	
5.21	SRM	N ²	Springe, wenn Marke nicht	N → BZ, wenn <CR> = 0, sonst 0 → CR	
5.22	SAAN	N ²	Springe, w. Alarm: arithmet.	N → BZ, wenn A-Alarmmeldung, und diese löschen	
5.23	SAT	N ²	Springe, w. Alarm: Typenkenn.	N → BZ, wenn TK-Alarmmeldung, und diese löschen	
5.24	SXI	N ²	Springe, w. Index identisch	N → BZ, wenn <BA> = 0; Das erste Bit von <BA> wird von diesen	
5.25	SXN	N ²	Springe, w. Index nicht ident.	N → BZ, wenn <BA> ≠ 0; Befehlen als Vorzeichen interpretiert;	
5.26	SXG	N ²	Springe, wenn Index größer	N → BZ, wenn <BA> > 0; <BA> ≤ 32 767;	
5.27	SXK	N ²	Springe, wenn Index kleiner	N → BZ, wenn <BA> < 0; 0 in BA nur als -0.	
5.28	SEG	J ^P ; I ^P	Springe, w. Exponent größer	Nur TK0 in AC; I ≤ 127; Exponent von <AC> ≤ 127; <BZ> _{alt} + J → BZ, wenn Exponent von <AC> ≥ I; a → MD;	
5.29	ST	J ^P ; I ^S	Springe, w. Typenkennung	I wie bei ZTR (2.03); <BZ> _{alt} + J → BZ, wenn angegebene TK in angegebenem Register;	
5.30	STN	J ^P ; I ^S	Springe, w. Typenkenn. nicht	I wie bei ZTR (2.03); <BZ> _{alt} + J → BZ, wenn angegebene TK nicht in angegebenem Register;	
5.31	SL	J ^P ; I ^S	Springe, wenn Merklicht	I wie bei ZL (2.17); <BZ> _{alt} + J → BZ, wenn mindestens ein bezeichnetes ML gesetzt;	J ≤ 127
5.32	SLL	J ^P ; I ^S	Springe, w. Merkll. u. lösche	I wie bei ZL (2.17); <BZ> _{alt} + J → BZ, wenn mindestens ein bezeichnetes ML gesetzt, und diese löschen;	
5.33	SLN	J ^P ; I ^S	Springe, w. Merklicht nicht	I wie bei ZL (2.17); <BZ> _{alt} + J → BZ, wenn alle bezeichneten ML gelöscht;	
5.34	SNL	J ^P ; I ^S	Springe, wenn Merklicht nicht, sonst lösche	I wie bei ZL (2.17); <BZ> _{alt} + J → BZ, wenn alle bezeichneten ML gelöscht, sonst diese löschen;	
5.35	SW	J ^P ; I ^S	Springe, wenn Wahlschalter	I besteht aus den WS-Bezeichnungen 1...8 in beliebiger Reihenfolge; <BZ> _{alt} + J → BZ, 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 ± 2 oder ± 4 usw. und mit $k = 0, 1, 2, \dots$, soweit dort $TK = t_D$;

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: $(h \& \langle N + 2k \rangle) = h \& d$, 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: $(h \& \langle N + bk \rangle) = h \& d$, 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 $\rightarrow AC$

7. Modifizier-, Ersetz- und Adressierbefehle

7.01	MF	I	Modifiziere folgenden Adreßteil	$N_X + i^+ \rightarrow AM$; nach der Modifizierung ist $\langle BA \rangle = i$; $N_X = i$
7.02	MCF	N ²	Modifiziere 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	Modifiziere	$N_{FX} + i^+ \rightarrow N_{FE}$; nach der Modifizierung ist $\langle BA \rangle = i^+$; $N_X = i$
7.04	MH	J ² : I	Modifiziere und erhöhe	$i + J \rightarrow I$; $N_{FX} + i + J \rightarrow N_{FE}$; nach der Modifizierung ist $\langle BA \rangle = i + J$; keine N_X ; $ J \leq 127$
7.05	MD	J: I	Modifiziere doppelt	$(N_F + i^+)_{X+J} \rightarrow N_{FE}$; nach der Modifizierung ist $\langle BA \rangle = j$; $N_X = i$
7.06	MM	I	Modifiziere mehrfach	$N_{FX} + i^+ \rightarrow AM$; 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 ²	Modifiziere aus Speicher	$N_{FX} + n_3^+ \rightarrow N_{FE}$; nach der Modifizierung ist $\langle BA \rangle = n_3^+$; $N_X = n_3$
7.08	MCE	N ²	Modifiziere aus Speicher nach Ersetzungen	$N_{FX} + \langle \dots \langle N_3 \dots \rangle_3^+ \rangle \rightarrow N_{FE}$; Ersetzkette, bis Op-teil mit L endet, einschließlich; nach der Modifizierung ist $\langle BA \rangle = \langle \dots \langle N_3 \dots \rangle_3^+ \rangle$; $N_X = \langle \dots \langle N \dots \rangle \rangle_3$
7.09	MA	N*	Modifiziere mit Adreßteil	$N_{FX} + N^+ \rightarrow N_{FE}$; nach der Modifizierung ist $\langle BA \rangle = N^+$; $N_X = N$
7.10	MB	N*	Modifiziere mit BA	$N_{FX} + b^+ \rightarrow N_{FE}$; BA^+ bleibt nach der Modifizierung erhalten; $N_X = b$; Adressenteil N bedeutungslos
7.11	MAB	J ² : I ²	Modifiziere Adreßteil mit BA	$J \rightarrow OP$; $i + b \rightarrow AM$; $i + b \rightarrow BA$; keine N_X ; $ i \leq 127$
7.12	E	J ² : I	Ersetze	$J \rightarrow OP$; $i_N^+ \rightarrow N_E$; BA^+ bleibt nach der Ersetzung erhalten; $N_X = i_N$
7.13	EZ	J ² : I	Ersetze zählend	$J \rightarrow OP$; $i + 2 \rightarrow AM$; $i + 2 \rightarrow I$; BA ; $N_X = i + 2$
7.14	ENZ	J ² : I	Ersetze negativ zählend	$J \rightarrow OP$; $i^+ \rightarrow AM$; $i - 2 \rightarrow I$; BA ; $N_X = i$
7.15	EMB	J ² : I	Ersetze und modifiziere mit BA	$J \rightarrow OP$; $i_N + b \rightarrow N_E$; BA^+ bleibt nach der Modifizierung erhalten; keine N_X
7.16	MU	J ² : I ²	Modifiziere über Unterprogrammordnungsähler	$J \rightarrow OP$; $i + \langle \langle WZ \rangle \rangle \rightarrow AM$; falls J ein auszuführender Sprungbefehl mit Ganzadresse ist, $\langle WZ \rangle - 1 \rightarrow WZ$; keine N_X ; $ i \leq 127$
7.17	EMU	J ² : I ²	Ersetze nach Modifizierung über Unterpr.ordn.zähler	$J \rightarrow OP$; $\langle i + \langle \langle WZ \rangle \rangle_{2N} \rangle \rightarrow N_E$; $N_X = \langle i + \langle \langle WZ \rangle \rangle_{2N} \rangle$; $ i \leq 127$
7.18	RL	J ² : I ²	Relative Adressierung	$J \rightarrow OP$; $(i + \langle \langle BZ \rangle \rangle_{2N})^+ \rightarrow N_E$; $N_X = (i + \langle \langle BZ \rangle \rangle_{2N})$; $ i \leq 127$
7.19	R	J ² : I ²	Register-Adressierung	$J \rightarrow OP$, soweit zugelassener Zweitcode; als Operand wird der Inhalt des in I bezeichneten Registers verwendet: $A = AC, H = HR, D = MD, Q = MQ, L = \text{links}, Z = BZ \text{ (neu)}, W = WZ, B = BA, Y = YZ; N_X = N$ Gegebenenfalls Ausschnitt rechts, Auffüllen mit $+0$ links, Hinzufügen $TK1$

Alphabetische Liste der Befehle und allgemeine Eigenschaften

Spalte	Erläuterungen
Externcode	enthält alle externen Befehlscodes alphabetisch geordnet, auch die der Pseudobefehle sowie die Pseudocodes ($0 < \dots < 9 < A < \dots < Z$)
lfd. Nr.	laufende Nummer aus der Kurzbeschreibung der Befehle.
Interncode	enthält die internen Befehlscodes in Tetraden- und technischer Notierung.
belegt	gibt an, welche Werke in der Ausführungsphase belegt sind: B nur das Befehlswerk (und oft auch das Speicherwerk) R nur das Rechenwerk BR beide Werke (und oft auch das Speicherwerk). 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.
Nennadresse	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). 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. 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.
Zweitcode bei R	+ kennzeichnet die Befehle, die als Zweitcode bei R zugelassen sind. (+) kennzeichnet die Befehle, bei denen es zwar technisch möglich, aber sinnlos ist, sie als Zweitcode bei R zu verwenden. — kennzeichnet die Befehle, die grundsätzlich nicht als Zweitcode bei R auftreten dürfen.
AZ	Ausführungszeit in Mikrosekunden

Ausführungszeiten der TR4-Befehle

Abrufzeit: 6 Mikrosekunden

wenn ein Modifizierbefehl vorausgeht: 8,5 Mikrosekunden

Die Abrufphase 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	Sp
WTD, WTK	„ „ transportierten Wörter	
LK1, LK2	„ „ Löcher in der Lochkartenzeile bzw. Lochkarte	
VRK	„ „ Zellen	2
NRM	„ „ Tetradschiffe	

p bedeutet bei:

VRK	Anzahl der Hexaden, einschließlich Grenzhexade	0,5
-----	--	-----

C

LS

Lk

Alphabetische Liste der Befehle und allgemeine Eigenschaften

Externcode	lfd. 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	9 8 460	R	=	—	11,5
AQ	3.18	7 E 372	R	=	+	7,5
ATA	3.59	8 9 421	R	=	—	7,5
AU	3.44	4 9 221	R	=	+	15
AUT	3.55	6 9 321	R	=	+	2,5
B	1.01	7 0 340	R	=	+	3
B2	1.12	6 E 332	R	=	+	4
B2V	1.13	6 F 333	R	=	+	4,5
B3	1.14	6 C 330	R	=	+	6,5
B3V	1.15	6 D 331	R	=	+	7
BA	1.16	8 E 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	7 4 350	R	=	+	3
BC	1.20	A 3 503	B R	=	—	8,5
BD	1.03	7 1 341	R	=	+	3
BH	1.02	7 3 343	R	=	+	3
BN	1.05	7 5 351	R	=	+	3
BNR	1.08	7 7 353	R	=	+	3
BQ	1.04	7 2 342	R	=	+	3
BQB	1.09	DA 662	R	=	+	3
BR	1.07	7 6 352	R	=	+	3
BZ	1.10	D 9 661	B R	=	—	9
BZ2	1.11	D 8 660	B R	=	—	7
C	1.21	8 0 400	B R	=	—	6,5
C2	1.29	A 0 500	B R	=	—	9
C3	1.30	A 1 501	B R	=	—	9
CB	1.26	8 5 411	B R	=	—	6,5
CD	1.23	8 6 412	B R	=	—	6,5
CH	1.22	8 F 433	B R	=	—	7,5
CMC	1.33	A 2 502	B R	=	—	8,5
CMR	1.32	8 3 403	B R	=	—	6,5
CMT	1.31	8 2 402	B R	=	—	6,5
CN	1.25	8 4 410	B R	=	—	6,5
CQ	1.24	8 7 413	B R	=	—	6,5
CR	1.27	8 1 401	B R	=	—	6,5
CWL	1.46	0 6 012	B	#	—	9
CZ	1.28	DB 663	B R	=	—	12
DA	3.61	C 6 612	R	=	+	10,5
DAC	3.62	C 4 610	B R	=	—	16,5
DDD	3.68	C 3 603	R	=	+	420
DDV	3.67	C 2 602	R	=	+	420
DML	3.65	C 0 600	R	=	+	240
DMN	3.66	C 1 601	R	=	+	240
DR	4.05	DE 672	R	=	(+)	7,5

Alphabetische Liste der Befehle und allgemeine Eigenschaften

Externcode	lfd. Nr.	Interncode	belegt	Nennadresse	Zweitcode bei R	AZ
DSB	3.63	C 7 613	R	=	+	10,5
DSC	3.64	C 5 611	B R	=	—	16,5
DVD	3.24	6 1 301	R	=	+	157
E	7.12	2 9 121	B	i _N	—	4,5
EA	5.06	3 B 163	B	=	—	0,5
EMB	7.15	2 8 120	B	II #	—	9,5
EMU	7.17	0 4 010	B	<I + <<WZ>>> _{3N}	—	20,5
ENZ	7.14	2 A 122	B	i	—	9
ET	3.56	6 A 322	R	=	+	2,5
ETA	3.60	8 A 422	R	=	—	7,5
EZ	7.13	2 B 123	B	i + 2	—	8
F A	3.01	4 2 202	R	=	+	6
F AB	3.02	4 0 200	R	=	+	6,5
F AC	3.03	4 3 203	B R	=	—	12
F DV	3.13	6 0 300	R	=	+	145
F DVI	3.14	6 2 302	R	=	+	145
F MAN	3.12	5 A 262	R	=	+	32,5
F ML	3.09	5 4 250	R	=	+	29,5
F MLA	3.11	5 6 252	R	=	+	32,5
F MLN	3.10	5 8 260	R	=	+	29,5
F MSK	3.26	E 6 712	B R	#	—	16 + 39,5 m
F POL	3.25	E 4 710	B R	#	—	37,5 m
F SB	3.04	4 6 212	R	=	+	6,5
F SBB	3.05	4 1 201	R	=	+	6,5
F SBC	3.06	4 7 213	B R	=	—	12,5
F SBD	3.08	4 5 211	R	=	+	6,5
F SBI	3.07	4 4 210	R	=	+	6,5
F W	3.15	5 0 240	R	=	+	150
G A	3.27	4 B 223	R	=	+	16
G AB	3.28	5 2 242	R	=	+	16
G AC	3.29	4 A 222	B R	=	—	22
G DV	3.39	6 4 310	R	=	+	120
G DVI	3.40	6 6 312	R	=	+	120
G MAN	3.38	5 D 271	R	=	+	49
G ML	3.35	5 E 272	R	=	+	31
G MLA	3.37	5 F 273	R	=	+	49
G MLN	3.36	5 C 270	R	=	+	31
G MSK	3.48	E 7 713	B R	#	—	16 + 56 m
G POL	3.47	E 5 711	B R	#	—	49 m
G SB	3.30	4 F 233	R	=	+	16
G SBB	3.31	5 3 243	R	=	+	16
G SBC	3.32	4 E 232	B R	=	—	22
G SBD	3.34	4 C 230	R	=	+	16
G SBI	3.33	4 8 220	R	=	+	16
G W	3.41	5 1 241	R	=	+	150
H	3.78	2 C 130	B	#	—	8,5
HA	3.49	F 0 740	B R	#	—	62 + 20 m

Int.
Ext.-
Code

Sp

2

0,5

C

LS

Lk

Alphabetische Liste der Befehle und allgemeine Eigenschaften

Externcode	lfd. Nr.	Interncode	belegt	Nennadresse	Zweitcode bei R	AZ
HALT	—	9D 471	B R	#	(+)	1,5
HAU	3.50	F 2 742	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 ²
HN	4.18	CD 631	B	#	—	8,5
HNM	4.19	F 8 760	B R	#	—	18+12 m
HSB	3.51	F 1 741	B R	#	—	62+20 m
HSU	3.52	F 3 743	B R	#	—	62+20 m
HWH	4.16	FB 763	B R	#	—	12+ 6 m
HWZ	4.17	F 9 761	B R	#	—	12
HX	3.79	2 E 132	B	#	—	14
HXN	3.80	2 F 133	B	#	—	14,5
KBH	4.08	Pseudobefehl!				107
KGZ	4.09	Pseudobefehl!				153
KVH	4.06	9 4 450	R	=	—	12+3 · 1
KVZ	4.07	9 5 451	R	=	—	5+4 · 1
L	2.04	9 A 462	B R	=	—	2
LA	2.05	8 B 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	10 040	B	#	—	1,5
LMC	2.13	3 1 141	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	7 A 362	R	=	+	23
M2N	3.72	7 8 360	R	=	+	23
M2P	3.74	7 9 361	R	=	+	20
M2R	3.73	7 B 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	5 B 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 ₃	+	3
MCE	7.08	1 7 053	B	II <...<N> ₃ ...> ₃	+	-2+6 m
MCF	7.02	1 6 052	B	n ₃	+	3
MD	7.05	0 9 021	B	(II) i	—	7,5
MF	7.01	0 B 023	B	i	—	1,5
MH	7.04	2 D 131	B	II #	—	8,5
MLR	3.20	5 5 251	R	=	+	29,5
MM	7.06	0 A 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	lfd. Nr.	Interncode	belegt	Nennadresse	Zweitcode bei R	AZ	wenn nicht AZ 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	$(I + \langle BZ \rangle)_N$	—	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	BA 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	5.85	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
SXK	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_{3N}	+	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

Alphabetische Liste der Befehle und allgemeine Eigenschaften

Externcode	lfd. Nr.	Interncode	belegt	Nennadresse	Zweitcode bei R	AZ
TCB	1.44	3 9 161	B	=	+	3,5
TDM	6.06	E A 722	B R	#	—	6,5+6,5m
TL	6.01	E F 733	B R	#	—	3 + 8 m
TLB	6.02	E D 731	B R	#	—	3 + 8 m
TLD	6.05	E B 723	B R	#	—	7 +8,5m
TLI	6.03	E C 730	B R	#	—	3 + 6 m
TLM	6.04	E E 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+3,5m+1,5p
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	—	B C 570	B R	=	—	18,5
YB	—	3 C 170	B	=	+	3
YC	—	3 D 171	B	=	—	6
YF	—	3 7 153	B	=	—	0
YL	—	B D 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

Int.
Ext.-
Code

Sp

LS

Lk

FA

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

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	60	DH	30
AD	A 0	QH	50		
AH	90				

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)

ST, STN, ZTR und L (siehe auch folgende Gruppe!)

Spezifikation im Erstadreßteil

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

L (siehe auch vorige Gruppe!)

Spezifikation im Erstadreßteil

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

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)

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
.
.
.

Intern-Spezifikationen in je 2 hexadekadischen Ziffern

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

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	×	ψ	,	"	(>	?
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

Z# Zeilenende

leer Leerzeichen; wird ignoriert

ψ 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 (einzeln)

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

2. Tetrade (paarweise)

2. Tetrade (einzeln)

0	0	G	16	W	32	#	48	0
1	1	H	17	X	33	;	49	1
2	2	I	18	Y	34	"	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	×	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	ψ	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 (einzeln) 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

L00000: Formularende, hat nur Bedeutung im off-line-Betrieb:

Magnetband kann formularweise vor- und zurückgespult werden.

oL0000: 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):

oo0000: kein Vorschub;

oo000L: Vorschub bis zur Lochung im Kanal 2;

oo00LO: Vorschub bis zur Lochung im Kanal 3;

oo00LL: Vorschub bis zur Lochung im Kanal 4;

oo0L00: Vorschub bis zur Lochung im Kanal 5;

oo0L0L: Vorschub bis zur Lochung im Kanal 6;

oo0LL0: Vorschub bis zur Lochung im Kanal 7;

oo0LLL: Vorschub bis zur Lochung im Kanal 8;

ooL000: Vorschub um 1 Zeile

ooL00L: Vorschub um 2 Zeilen

ooL0LO: Vorschub um 3 Zeilen

ooL0LL: Vorschub um 4 Zeilen

ooLL00: Vorschub um 5 Zeilen

ooLL0L: Vorschub um 6 Zeilen

ooLLL0: Vorschub um 7 Zeilen

ooLLLL: Vorschub um 8 Zeilen

Ein- und Ausgabe über EA-Geräte mit Ausnahme der Kontrollschreibmaschine SM

Startbefehle (davor obligatorisch **EA P**)

- | | | | |
|-------------|--|---|--|
| EA S | Normaler Start eines EA-Gerätes; | $\langle AC \rangle_3 = \rangle STW \langle ;$ | Abschluß mit EA Z
(Löschen mit EA A) |
| EA F | Fortstart eines EA-Gerätes, wenn möglich,
sonst normaler Start; | $\langle AC \rangle_3 = \rangle STW \langle ;$ | Abschluß mit EA Z
(Löschen mit EA A) |
| | Startsatz: STW s g s = Spezifikation (s. u.); g = externe Gerätenummer (vorläufig 0 bis 19)
VW1 f: a f = ausgewähltes Fach (0 oder 1, nur bei Lochkarten)
a = Anfangsadresse des EA-Speichers (Endadresse bei MB-Eingabe rückw.)
VW2 z z = Länge des EA-Speichers (Anzahl der Ganzwörter, $2 \leq z \leq 4095$) | | |
| | Spezifikation s: | S = Lochstreifeneingabe
K = Lochkarteneingabe
B = Magnetbandeingabe vorwärts
BR = Magnetbandeingabe rückwärts | SA = Lochstreifenabgabe
KA = Lochkartenabgabe
BA = Magnetbandabgabe
D = Abgabe auf on-line-Schnelldrucker |
| EA T | Teilblockstart, d. h. Ein- bzw. Ausgabe mehrerer gleich langer Blöcke, hauptsächlich bei MB-Abgabe für off-line-Schnelldrucker
Startsatz: wie bei EA S und EA F , nur VW2 t: z , wobei t = Anzahl der Blöcke ($2 \leq t \leq 64$) | $\langle AC \rangle_3 = \rangle STW \langle ;$ | Abschluß mit EA Z
(Löschen mit EA A) |
| EA L | Magnetband stückweise löschen
Startsatz: STW s g s = Spezifikation, (nach Belieben B oder BA);
g = externe Gerätenummer (vorläufig 0 bis 19)
VW2 z z = Länge der Blocklücke (Anzahl der Ganzwörter, $2 \leq z \leq 4095$) | $\langle AC \rangle_3 = \rangle STW \langle$ | |
| EA U | Magnetband umspulen;
Startsatz: STW s g s = Spezifikation (s. u.);
g = externe Gerätenummer (vorläufig 0 bis 19) | $\langle AC \rangle_3 = STW$ | |
| | 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;
Startsatz: STW s g | $\langle AC \rangle_3 = STW$
s = Spezifikation, wie beim zugehörigen Startbefehl;
g = externe Gerätenummer (vorläufig 0 bis 19) |
| | Rückgabe der Regie nicht vor Blockendmeldung des ältesten offenen Startbefehls zur gleichen Gerätenummer, im allgemeinen mit $\langle CR \rangle = 0$; wenn $\langle CR \rangle = L$, dann Bandendmeldung. | |
| EA A | Aufgabe, d. h. Löschen aller offenen Startbefehle für ein Gerät;
Startsatz: wie bei EA Z | $\langle AC \rangle_3 = STW$ |
| EA E | Erneuerung, d. h. Wiederholung aller offenen Startbefehle für ein Gerät, nachdem sie wegen einer Gerätestörung gesperrt werden mußten;
Startsatz: wie bei EA Z | $\langle AC \rangle_3 = STW$ |
| EA R | Reservierung, d. h. vorläufige Stilllegung aller offenen Startbefehle für ein Gerät, nachdem sie wegen einer Gerätestörung gesperrt werden mußten, um unter Umgehung der Sperre vor EA E andere Startbefehle zu geben (nur geringe Bedeutung);
Startsatz: wie bei EA Z | $\langle AC \rangle_3 = STW$ |

Ein- und Ausgabe über die Kontrollschreibmaschine SM

- EA M** Zuerst Ausgabe, dann Eingabe über SM; $\langle AC \rangle_3 = \text{Startsatz}$
 Startsatz: $lz: e \quad z = \text{Anzahl der einzugebenden Ganzwörter } (0 \leq z \leq 15)$
 $e = \text{Anfangsadresse für Eingabe (im Fall } z = 0 \text{ irrelevant)}$
 $0 \quad a \quad a = \text{Anfangsadresse für Ausgabe (höchstens 32 Wörter, soweit TK3)}$
 Rückgabe der Regie frühestens nach vollständiger Erledigung der Ausgabe bzw. Aus- und Eingabe.
- EA N** Ausgabe über SM; $\langle AC \rangle_3 = a = \text{Anfangsadresse für Ausgabe}$
 Rückgabe der Regie sofort. (höchstens 32 Wörter, soweit TK3)
- 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; $\langle AC \rangle_3 = \text{Sprungadresse } s$;
 $\langle HR \rangle_3 = \text{rechtsbündige Intern- darstellung des internen Wahlschalters.}$
 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; $\langle AC \rangle_3 = f$;
 $\langle HR \rangle_3 = \text{STW}$
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: $0 \quad p \quad p = \text{Programmnummer } (1 \leq p \leq 8)$
 $0 \quad s \quad s = \text{Startadresse}$
 lwH: $a \quad w = \text{intern dargestellter interner Wahlschalter;}$
 $a = \langle AC \rangle_3 \text{ beim Start des Programms}$
 Das Nebenprogramm muß mit **HLT** enden.
- EA G** Programmzieladresse setzen; $\langle AC \rangle_3 = p \quad (1 \leq p \leq 8)$
 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

Adreßteil:	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 Zweitadreßteil, die Kanaladresse $2n-1$ (n = Kanalnummer 1 bis 8) dual rechtsbündig im Erstadreßteil.

- VW1:** Operationsteil: extern: 0: 1:
intern: 00 80 (nur bei Lochkarten, sonst immer 00)
- Adreßteil: Anfangs- (bzw. End-) Adresse des EA-Speichers, dual, einschließlich Translations-
adresse
- VW2:** Operationsteil: nur für **EA T**: ($-t+1$) dual rechtsbündig, sonst immer 00
Adreßteil: $F000 + z$ dual rechtsbündig.

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

