

**AEG**



**DATENVERARBEITUNG**

---

**TR 440**  
**Große Befehlsliste**

Lfd.Nr.	Datum	Seite	Befehl	Spalte	Berichtigung
---------	-------	-------	--------	--------	--------------

<b>6</b>	10.2.70	2		3	Bei einem Teil der letzten Auflage ist unter "12 Takte" der Text verstümmelt. Es muß in der 2. Zeile heißen: Sind unter "Werk" Befehls- und Rechenwerk aufgeführt, so gilt die Zeit für beide Werke. Tragen Sie als letzte Zeile ein: Für genaue Zeitberechnung siehe "TR 440 Befehls-Lexikon".
		4	BSS	Bemerkungen	In der 2. Zeile muß es heißen: s, ist ohne Bedeutung, muß aber angegeben werden.
		5	CNZ	Wirkung	In diese Spalte ist vor der ersten Zeile einzutragen: Voraussetzung: <A> = 0, f Bits
		6	QBR QCR }	Bemerkungen	Bitte eintragen: (nicht zugreifbares Register)
		13	PDP	Werk	Der Befehl belegt auch das Befehlswerk; bitte in dieser Spalte zusätzlich ein B eintragen.
		15	MABI	Bemerkungen	statt 2 Sternchen bitte 4 Sternchen eintragen: unter den Fußnoten einkleben: <div style="border: 1px dashed black; padding: 5px; width: fit-content; margin: 5px auto;">**** für c die Sprungbefehle von Seite 12 u. 13 (mit Ausnahme der Befehle NULL, PDP, MDP und WB) bei erfüllter Sprungbedingung ist ein Sprung in eine andere Großseite möglich.</div>
		19	Zentral- code	Feld 107  Feld 35 u.36	In dieses Feld ist das Zeichen "inverser Schrägstrich" (\) einzutragen.  Die Zeichen "T1" und "T2" sind zu tilgen; die Felder bleiben leer.
		23	TRX	im Text	Es muß in der ersten Zeile heißen: Von den Bits 9 - 12 darf <u>höchstens</u> eines = L sein, ...
		29 Umschlag innen	ZTO M MFU XB XC XCN RX MRX }	Code  Int.	Der Befehlscode wird ZTO geschrieben (Ziffer 0 statt Buchstabe O) Auf Anregung eines Benutzers ist zu empfehlen, in der Spalte Int. (Interncode) bei diesen Befehlen ein Sternchen zu machen und als Fußnote einzukleben: <div style="border: 1px dashed black; padding: 5px; width: fit-content; margin: 5px auto;">*Diese Befehle benutzen gemeinsame Interncodes. Unterschiede liegen im Adressenteil siehe Seiten 22/23 "Spezifikationen-Intern".</div>



# TR 440 Große Befehlsliste

zum Programmieren von  
Operatoren in der  
Programmiersprache TAS

Einleitung

Dem Programmierer des Digital-Rechners TR 440 soll mit dieser Liste eine handliche Arbeitsunterlage zum Programmieren von Operatoren gegeben werden. In knapper Tabellenform stehen hier die wesentlichen Informationen zur Verfügung. Auf weitergehende Einzelheiten ist, um eine gute Übersichtlichkeit zu erzielen, verzichtet worden. Die hier nicht angegebenen Einzelheiten sind dem " TR 440 - Befehlslexikon " zu entnehmen.

Diese erste Ausgabe soll erweitert werden und zu einem späteren Zeitpunkt weitere Informationen und Arbeitshilfen bieten.

Inhaltsverzeichnis	Seite
Bedeutung der Spalten	2
Erklärung der Zeichen	3
Transportbefehle	4-6
Festkomma-Arithmetik	7
Gleitkomma-Arithmetik	8
Boolesche Operationen	9
Halbwort-Arithmetik	9
Teilwort-Arithmetik	9
Index-Arithmetik	10
Setzen und Löschen	10, 11
Sprünge	12, 13
Modifizieren	14
Ersetzen (und modifizieren)	15
Aufbereitung	16, 17
Tabelle durchsuchen	17
Zentralcode	19
Potenzen von 2	20
Konvertierungstafel	21
Interncode - Externcode	22
Internspezifikationen	23
Wortstruktur	24
Blockschaltbild	25
Alphabetische Liste der Befehle	29



# Erklärung der Zeichen

<u>Bezeichnung der Register im Rechenwerk:</u>		<u>Zeichen und ihre Bedeutung:</u>	<u>Indizes für Teile eines Wortes:</u>												
A	Akkumulator	:= Die links stehende Zielgröße ergibt sich aus der rechts stehenden Quellengröße Beispiel: $\langle A \rangle := \langle n \rangle$ Der Inhalt von A ergibt sich aus dem Inhalt von n.	$\langle \rangle_v$ Vorzeichen vom Inhalt $\langle \rangle_t$ Typenkennung vom Inhalt $\langle \rangle_m$ Marke vom Inhalt $\langle A \rangle_1$ Bit 1 im Register A $\langle A \rangle_{41-48}$ Bits 41 bis 48 im Register A $\langle n \rangle_{9-24}$ Bits 9 bis 24 in der Speicherzelle (Drittelwort) $\langle n \rangle_{1,2}$ Bits 1 und 2 in der Speicherzelle zur Zählung der Bits siehe Seite Wortstruktur												
Q	Quotientenregister														
D	Multiplizierenregister														
H	Hilfsregister														
Y	Schifftzähler	::: Links- und rechtsstehende Größen werden miteinander vertauscht. Beispiel: $\langle A \rangle ::= \langle H \rangle$ Die Inhalte von A und H werden vertauscht.													
M	Markenregister														
A,Q	doppeltlanges Register <u>A</u> und <u>Q</u>														
H,Q	doppeltlanges Register <u>H</u> und <u>Q</u>														
<u>Bezeichnung der Register im Befehlswerk:</u>			<u>Logische Verknüpfungen:</u>												
B	Bereitadressenregister	$t_x$ ; Typenkennung im Register x oder in der Speicherzelle x	$\wedge$ UND-Verknüpfung (Konjunktion)												
F	Befehlsfolgeregister	$1,1$ ; Typenkennung in beiden Registern eines doppeltlangen Registers													
X	Indexbasisregister		$0,$ der linke Teil des Registers ist mit Null aufgefüllt	$a := b \wedge c$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>L</td></tr> <tr><td>0</td><td>L</td><td>0</td></tr> <tr><td>L</td><td>L</td><td>L</td></tr> </table>	0	0	0	0	0	L	0	L	0	L	L
0	0	0													
0	0	L													
0	L	0													
L	L	L													
K	Merklicherregister	$v,$ der linke Teil des Registers ist vorzeichen gleich aufgefüllt	$\vee$ ODER-Verknüpfung (Disjunktion)												
U	Unterprogrammregister														
mod1	Modifikator erster Art	$\langle \rangle$ Inhalt eines Registers, einer Speicher- oder Indexzelle Beispiel: $\langle A \rangle$ Inhalt des Registers A (einschließlich Typenkennung)	$\nabla$ Antivalenz-Verknüpfung (exklusives ODER)												
mod2	Modifikator zweiter Art														
op	Operationscode eines Befehls	$\langle \rangle, \langle \rangle$ Inhalt von zwei getrennten Registern Beispiel: $\langle A \rangle, \langle Q \rangle$ Inhalt von A und Q	$a := b \nabla c$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>L</td><td>0</td><td>L</td></tr> <tr><td>L</td><td>L</td><td>0</td></tr> <tr><td>0</td><td>L</td><td>L</td></tr> </table>	0	0	0	L	0	L	L	L	0	0	L	L
0	0			0											
L	0	L													
L	L	0													
0	L	L													
adr	Inhalt des Adressenteils eines Befehls	$\langle , \rangle$ Inhalt zweier Register, die zusammengefaßt sind Beispiel: $\langle A, Q \rangle$ Inhalt der zu einem doppelt langen Register vereinigten Register A und Q	$\neg$ Negation												
n	Speicheradresse eines Ganzwortes														
m	Speicheradresse eines Halbwortes	$   $ Betrag einer Größe, Beispiel: $ \langle n \rangle $ Betrag vom Inhalt der Speicherzelle n	$a := \neg b$ <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>L</td></tr> <tr><td>L</td><td>0</td></tr> </table>	0	L	L	0								
0	L														
L	0														
z	Zahl oder Operand														
i	Indexadresse														
p	Parameter														
s	Spezifikation														
c	Zweitcode (Code für den Zweitbefehl)														

# Transportbefehle

Bezeichnung	de	adr	Wirkung	mod2	R	<A>	<Q>	<H>	<D>	<Y>	<B>	<M>	Werk	Takte	Int.	Bemerkungen
Bringe (nach A)	B	n	<A> := <n>	+	x	$t_n; \langle n \rangle$						x	R	2	70	
Bringe <u>un</u> verändert	BU	n	<A> := <n>	+	x	$t_n; \langle n \rangle$							R	2	D3	unverändert bei jeder TK
Bringe und <u>res</u> erviere	BR	n	<H> := <A> <A> := <n>	+	x	$t_n; \langle n \rangle$		$t_A; \langle A \rangle$				x	R	5	76	
Bringe <u>neg</u> ativ	BN	n	<A> := -<n>	+	x	$t_n; -\langle n \rangle$						x	R	5	75	
Bringe <u>neg</u> ativ und <u>res</u> erviere	BNR	n	<H> := <A> <A> := -<n>	+	x	$t_n; -\langle n \rangle$		$t_A; \langle A \rangle$				x	R	5	77	
Bringe <u>B</u> etrag	BB	n	<A> :=  <n>	+	x	$t_n;  \langle n \rangle $						x	R	5	74	
Bringe nach <u>Q</u>	BQ	n	<Q> := <n>	+	x		$t_n; \langle n \rangle$					x	R	3	72	
Bringe nach <u>D</u>	BD	n	<D> := <n>	+	x				$t_n; \langle n \rangle$			x	R	3	71	
Bringe nach <u>H</u>	BH	n	<H> := <n>	+	x			$t_n; \langle n \rangle$				x	R	3	73	
Bringe nach <u>Q</u> und <u>br</u> inge (nach A)	BQB	n	<A> := <n> <Q> := <n>	+	x	$t_n; \langle n \rangle$	$t_n; \langle n \rangle$					x	R	2	DA	
Bringe <u>z</u> wei Wörter	BZ	n	<A> := <n> <Q> := <n+2>	+		$t_n; \langle n \rangle$	$t_{n+2}; \langle n+2 \rangle$					x	B R	15	D9	
Bringe <u>z</u> wei Wörter <u>neg</u> ativ	BZN	n	<A> := -<n> <Q> := -<n+2>	+		$t_n; -\langle n \rangle$	$t_{n+2}; -\langle n+2 \rangle$					x	B R	15	D1	
Bringe <u>T</u> eilwort	BT	n	<A> <sub>x</sub> := <n> <sub>x</sub> falls <Q> <sub>x</sub> = 0 <A> <sub>x</sub> := 0 falls <Q> <sub>x</sub> = L	+	x	$t_n; \dots$ (s. Wirk.) <A> wird um p Stellen rechts geschiftet, 0 nachgezogen							R	7 +2p	F6	p: Anzahl der L, die rechtsbündig in Q stehen. x: 1,2,...,48
Bringe <u>n</u> ächstes <u>Z</u> eichen	BNZ	$i_L \ i_R$	<A> := 0, Zeichen gemäß a und b rechtsb. <A> <sub>t</sub> := <a> <sub>t</sub> b := b + 1 wenn b < d b := +0 <i_R> := <i_R> + 2 } wenn b = d		sp	$t_n; 0, \dots$ (s. Wirk.)			undefiniert	$\langle i_L \rangle_{a1t}$			B R	41	E6	a: <i_R> + mod2 lauf. Adr. eines Wortes der Liste b: <i_L> <sub>17-24</sub> lauf. Nummer eines Zeichens im Wort (0,1,...) d: (48/f)-1 max. Zeichennummer f: <i_L> <sub>9-12</sub> Anzahl der Bits pro Zeich. (4,6,8 oder 12) andere Bits von <i_L> bedeutungslos
Bringe <u>z</u> wei <u>H</u> albwörter	BZ2	m	<A> := <m,m+1>	+		$t_n; \langle m, m+1 \rangle$							B R	10	D8	
Bringe <u>H</u> albwort	B2	m	<A> := <m>	+	x	$t_n; 0, \langle m \rangle$							R	9	6E	
Bringe <u>H</u> albwort mit <u>V</u> orzeichen	B2V	m	<A> := <m>	+	x	1; v, <m>							R	10	6F	
Bringe <u>H</u> albwort mit <u>V</u> orzeichen <u>neg</u> ativ	B2VN	m	<A> := -<m>	+	x	1; -(v, <m>)							R	12	67	
Bringe <u>D</u> rittelwort	B3	m	<A> := <m> <sub>9-24</sub>	+	x	1; 0, <m> <sub>9-24</sub>							R	9	6C	
Bringe <u>D</u> rittelwort mit <u>V</u> orzeichen	B3V	m	<A> := <m> <sub>9-24</sub>	+	x	1; v, <m> <sub>9-24</sub>							R	10	6D	

/A)

Bringe aus <u>L</u> eitblock	BLEI	p	<A> := <<BL> · 2 <sup>p</sup> + p>	+		$t; \langle \langle BL \rangle \cdot 2^p + p \rangle$							B R	13	BB	p: +0...255 BL = Leitadressenregister
Bringe (und setze) <u>S</u> teuerbits (und <u>S</u> perren)	BSS	$a_L \ a_R$	<A> := Steuerbits falls $a_0=L$			2; Steuerbits falls $a_0=L$							B R	3	FB	$a_L$ : Bits $a_0$ bis $a_{16}$ $a_R$ : Bits $a_{17}$ bis $a_{24}$

Bezeichnung	Code	adr	Wirkung	mod2	R	$\langle n \rangle_n$	$\langle n \rangle$	$\langle n+2 \rangle$	$\langle A \rangle$	$\langle M \rangle$	BÜ	TK	Werk	lakte	Int.	Bemerkungen
Bringe und lösche	BL	n	$\langle A \rangle := \langle n \rangle$ $\langle n \rangle := +0$	+					$t_n; \langle n \rangle$				B R	17	B0	unverändert bei jeder TK
Bringe und speichere	BC	n	$\langle A \rangle := \langle n \rangle$	+		$\langle n \rangle_n$	$t_A; \langle A \rangle$		$t_n; \langle n \rangle$	x	$\geq$		B R	13	A3	

									$\langle X \rangle$	$\langle U \rangle$						
Bringe und speichere Indexbasis	BCI	n	$\langle X \rangle := \langle n \rangle_{3-24}$ $\langle U \rangle := \langle n \rangle_{41-48}$				$3; 0, \langle X \rangle, 0, \langle U \rangle$		$\langle n \rangle_{3-24}$	$\langle n \rangle_{41-48}$			B	$17x+66$	B6	x = Anzahl der zurückgespeicherten Indexreg.
Speichere	C	n	$\langle n \rangle := \langle A \rangle$	+		0	$t_A; \langle A \rangle$				$\geq$		B R	8	80	
Speichere unverändert	CU	n	$\langle n \rangle := \langle A \rangle$	+		$\langle A \rangle_1$	$t_A; \langle A \rangle$				$\geq$		B R	8	D0	unverändert bei jeder TK
Speichere und bringe Reserve	CR	n	$\langle n \rangle := \langle A \rangle$ $\langle A \rangle := \langle H \rangle$	+		0	$t_A; \langle A \rangle$	$t_H; \langle H \rangle$			$\geq$		B R	9	81	
Speichere negativ	CN	n	$\langle n \rangle := -\langle A \rangle$	+		0	$t_A; -\langle A \rangle$				$\geq$		B R	9	84	
Speichere Betrag	CB	n	$\langle n \rangle :=  \langle A \rangle $	+		0	$t_A;  \langle A \rangle $				$\geq$		B R	9	85	
Speichere markiert	CMT	n	$\langle n \rangle := \langle A \rangle$	+		L	$t_A; \langle A \rangle$				$\geq$	2,3	B R	8	82	
Speichere mit Marke aus Register	CMR	n	$\langle n \rangle := \langle A \rangle$	+		$\langle M \rangle$	$t_A; \langle A \rangle$				$\geq$	2,3	B R	8	83	
Speichere mit Marke aus Speicher	CMC	n	$\langle n \rangle := \langle A \rangle$	+		$\langle n \rangle_n$	$t_A; \langle A \rangle$				$\geq$	2,3	B R	13	A2	
Speichere aus Q	CQ	n	$\langle n \rangle := \langle Q \rangle$	+		0	$t_Q; \langle Q \rangle$				$\geq$		B R	9	87	
Speichere aus D	CD	n	$\langle n \rangle := \langle D \rangle$	+		0	$t_D; \langle D \rangle$				$\geq$		B R	9	86	
Speichere aus H	CH	n	$\langle n \rangle := \langle H \rangle$	+		0	$t_H; \langle H \rangle$				$\geq$		B R	9	8F	
Speichere zwei Wörter	CZ	n	$\langle n \rangle := \langle A \rangle$ $\langle n+2 \rangle := \langle Q \rangle$	+		0	$t_A; \langle A \rangle$	$t_Q; \langle Q \rangle$			$\geq$		B R	16	DB	

							Wirkung	$\langle D \rangle$								
Speichere Teilwort	CT	n	$\langle n \rangle_x := \langle A \rangle_x$	+			$\langle A \rangle$ um p Stellen links im Kreis geschiftet $\langle n \rangle_x := \langle A \rangle_x$ für $\langle Q \rangle_x = 0$ $\langle n \rangle_x := \langle n \rangle_x$ für $\langle Q \rangle_x = L$ $\langle A \rangle$ im Kreis zurückgeschiftet	$t_n; \langle n \rangle$			$\geq$		B R	22	F7	p: Anzahl der L, die rechtsbündig im Reg. Q stehen x: 1,2,...,48

							$\langle A \rangle$	$\langle n \rangle$	$\langle B \rangle$	$\langle Y \rangle$	$\langle Q \rangle$	$\langle D \rangle$					
Speichere nächstes Zeichen	CNZ	$i_l$ $i_r$	rechte f Bits von $\langle A \rangle$ in a gemäß b eingesetzt $\langle A \rangle := \langle a \rangle$ $\langle A \rangle_t := \langle a \rangle_t$ b := b + 1 wenn b < d  b := +0 $\langle i_r \rangle := \langle i_r \rangle + 2$ } wenn b = d	sp			$t_a; \langle a \rangle$	$\langle i_l \rangle_{a1t}$	undef.	abzusp. Zeichen aus A, gem. b geschiftet; andere Bits=0	undef.			B R	55	E7	a: $\langle i_r \rangle + \text{mod}2$ lauf. Adr. eines Wortes der Liste b: $\langle i_l \rangle_{17-24}$ lauf. Nummer eines Zeichens im Wort (0,1,...) d: $(48/f) - 1$ max. Zeichennummer f: $\langle i_l \rangle_{9-12}$ Anzahl der Bits pro Zeich. (4,6,8 oder 12) andere Bits $\langle i_l \rangle$ bedeutungslos

							$\langle m \rangle$										
Speichere Halbwort	C2	m	$\langle m \rangle := \langle A \rangle_{25-48}$	+			$\langle A \rangle_{25-48}$							B R	15	A0	Das Halbwort wird unverändert eingesetzt

							$\langle m \rangle_{9-24}$										
Speichere Drittelwort	C3	m	$\langle m \rangle_{9-24} := \langle A \rangle_{33-48}$	+			$\langle A \rangle_{33-48}$							B R	15	A1	Das Drittelwort wird unverändert eingesetzt



Transportiere Oktaden wenn $\langle H \rangle_4 = 3$ wenn $\langle H \rangle_4 = 2$	TOK	z	$\langle a+x \rangle := \langle q+x \rangle$ Quellgeb. z Okt $(q+6) \geq a$		3; Inhalt d. letzten Ganzwortes im Zielgebiet	3;0	letzte Halbw. adr. i. Zielg.	x	B	R	min. 191 min. 209	FD	z: 1...65535 a: $\langle H \rangle_{1-24} =$ Okt.adr.Zielgebiet q: $\langle H \rangle_{25-48} =$ Okt.adr.Quellgebiet x: 0,1,2,...,z-1 y: 0,1,2,3,4,5, 0, 1, 2, ...
---	-----	---	--	--	---	-----	------------------------------	---	---	---	----------------------	----	--

# Festkomma-Arithmetik

Bezeichnung	Code	adr	Ung	mod2	R	A	Q	D	n	Y	AM	BÜ	TK	Werk	Takte	Int.	Bemerkungen	
Addiere	A	n	$\langle A \rangle := \langle A \rangle + \langle n \rangle$	+	x	$t_{max}; \langle A \rangle + \langle n \rangle$		$t_n; \langle n \rangle$			x	$\geq$		R	8	42		
Addiere Betrag	AB	n	$\langle A \rangle := \langle A \rangle +  \langle n \rangle $	+	x	$t_{max}; \langle A \rangle +  \langle n \rangle $		$t_n; \langle n \rangle$			x	$\geq$		R	8	40		
Addiere im Speicher	AC	n	$\langle n \rangle := \langle n \rangle + \langle A \rangle$	+				$t_{max}; \langle n \rangle + \langle A \rangle$	$t_{max}; \langle n \rangle + \langle A \rangle$		x	$\geq$		B	R	17	43	
Subtrahiere	SB	n	$\langle A \rangle := \langle A \rangle - \langle n \rangle$	+	x	$t_{max}; \langle A \rangle - \langle n \rangle$		$t_n; \langle n \rangle$			x	$\geq$		R	8	46		
Subtrahiere Betrag	SBB	n	$\langle A \rangle := \langle A \rangle -  \langle n \rangle $	+	x	$t_{max}; \langle A \rangle -  \langle n \rangle $		$t_n; \langle n \rangle$			x	$\geq$		R	8	41		
Subtrahiere im Speicher	SBC	n	$\langle n \rangle := \langle n \rangle - \langle A \rangle$	+				$t_{max}; \langle n \rangle - \langle A \rangle$	$t_{max}; \langle n \rangle - \langle A \rangle$		x	$\geq$		B	R	17	47	
Subtrahiere invers	SBI	n	$\langle A \rangle := \langle n \rangle - \langle A \rangle$	+	x	$t_{max}; \langle n \rangle - \langle A \rangle$		$t_n; \langle n \rangle$			x	$\geq$		R	8	44		
Subtrahiere von D	SBD	n	$\langle A \rangle := \langle D \rangle - \langle n \rangle$	+	x	$t_{max}; \langle D \rangle - \langle n \rangle$					x	$\geq$		R	11	45		
Multipliziere mit Rundung	MLR	n	$\langle A \rangle := \langle A \rangle \cdot \langle n \rangle$	+	x	$1; (\langle A \rangle \cdot \langle n \rangle)^*$	$1; +0$	$t_n; \langle n \rangle$		$+0$	x	$\neq 1$		R	57	55		
Multipliziere negativ mit Rundung	MNR	n	$\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle$	+	x	$1; (-\langle A \rangle \cdot \langle n \rangle)^*$	$1; +0$	$t_n; \langle n \rangle$		$+0$	x	$\neq 1$		R	57	59		
Multipliziere akkumulierend mit Rundung	MAR	n	$\langle A \rangle := \langle A \rangle \cdot \langle n \rangle + \langle H \rangle$	+	x	$1; (\langle A \rangle \cdot \langle n \rangle)^* + \langle H \rangle$	$1; +0$	$t_n; \langle n \rangle$		$+0$	x	$\geq$	$\neq 1$	R	67	57		
Multipliziere akkumulierend negativ m. Rund.	MANR	n	$\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle + \langle H \rangle$	+	x	$1; (-\langle A \rangle \cdot \langle n \rangle)^* + \langle H \rangle$	$1; +0$	$t_n; \langle n \rangle$		$+0$	x	$\geq$	$\neq 1$	R	67	5B		
Dividiere	DV	n	$\langle A \rangle := \langle A \rangle : \langle n \rangle$	+	x	$1; \langle A \rangle : \langle n \rangle$	$1; \text{Rest} \cdot 2^{48}$	$1; +0$		$+0$	x	**	$\neq 1$	R	220	60	Operanden, Quotient und Rest sind als echte Brüche betrachtet (Komma links)	
Dividiere invers	DVI	n	$\langle A \rangle := \langle n \rangle : \langle A \rangle$	+	x	$1; \langle n \rangle : \langle A \rangle$	$1; \text{Rest} \cdot 2^{48}$	$1; +0$		$+0$	x	**	$\neq 1$	R	222	62		
Dividiere doppelt lang	DVD	n	$\langle A \rangle := \langle A, Q \rangle : \langle n \rangle$	+	x	$1; \langle A, Q \rangle : \langle n \rangle$	$1; \text{Rest} \cdot 2^{48}$	$1; +0$		$+0$	x	**	$\neq 1$	R	228	61		

A, Q																	
Addiere in AQ	AQ	n	$\langle A, Q \rangle := \langle A, Q \rangle + \langle n \rangle$	+	x	$1, 1; \langle A, Q \rangle + \langle n \rangle$		$t_n; \langle n \rangle$			x	$\geq$	$\neq 1$	R	17	7E	Operand und Ergebnis: $\langle A \rangle_v$ kann $\neq \langle Q \rangle_v$
Subtrahiere in AQ	SBQ	n	$\langle A, Q \rangle := \langle A, Q \rangle - \langle n \rangle$	+	x	$1, 1; \langle A, Q \rangle - \langle n \rangle$		$t_n; \langle n \rangle$			x	$\geq$	$\neq 1$	R	17	7F	
Multipliziere	ML	n	$\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle$	+	x	$1, 1; \langle A \rangle \cdot \langle n \rangle$		$t_n; \langle n \rangle$		$+0$	x	$\neq 1$		R	55	54	Ergebnis: $\langle A \rangle_v = \langle Q \rangle_v$
Multipliziere negativ	MLN	n	$\langle A, Q \rangle := -\langle A \rangle \cdot \langle n \rangle$	+	x	$1, 1; -\langle A \rangle \cdot \langle n \rangle$		$t_n; \langle n \rangle$		$+0$	x	$\neq 1$		R	55	58	
Multipliziere akkumulierend	MLA	n	$\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$	+	x	$1, 1; \langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$		$t_n; \langle n \rangle$		$+0$	x	$\geq$	$\neq 1$	R	66	56	$\langle H \rangle_v$ und $\langle Q \rangle_v$ ohne Bedeutung für die Ausführung des Befehls Operand: $\langle H \rangle_v$ kann $\neq \langle Q \rangle_v$ Ergeb.: $\langle A \rangle_v$ kann $\neq \langle Q \rangle_v$
Multipliziere akkumulierend negativ	MAN	n	$\langle A, Q \rangle := -\langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$	+	x	$1, 1; -\langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$		$t_n; \langle n \rangle$		$+0$	x	$\geq$	$\neq 1$	R	66	5A	

A																	
Addiere Adressenteil	AA	z	$\langle A \rangle := \langle A \rangle + z$	+		$TK = 0: 0; \langle A \rangle \cdot 16^{\pm z}$	$1; Q, z$					$\geq$		R	13	98	nach Modifiz. $TK = 0:  z  \leq 2^7$ $TK = 1:  z  \leq 2^{23}$ $TK = 2: z \leq 2^{16}$ $TK = 3:  z  \leq 2^{23}$
Subtrahiere Adressenteil	SBA	z	$\langle A \rangle := \langle A \rangle - z$	+		$TK = 1: 1; \langle A \rangle \pm v, z$	$1; y, z$					$\geq$		R	13	99	
						$TK = 2: 2; \langle A \rangle_{1-22} \pm \langle A \rangle_{23-48} \pm z$	$1; 0, z$					$\geq$					
						$TK = 3: 3; \langle A \rangle \pm v, z$	$1; v, z$					$\geq$					

\* gerundet \*\* Voraussetzung: bei DV, DVD:  $|\langle A \rangle| < |\langle n \rangle|$   
bei DVI :  $|\langle n \rangle| < |\langle A \rangle|$  im anderen Fall: Gleitkommaergebnis, BU-Alarm

# Gleitkomma-Arithmetik

Bezeichnung	Code	adr	Wirkung	mod2	R	Alarm										Bü	TK	Werk	Takte	Int.	Bemerkungen
						<A>	<n>	<D>	<Q>	<Y>	<M>	x	>	≠0							
Gleitkomma <u>addiere</u>	GA	n	<A> := <A> + <n>	+	x	0; <A> + <n>*		$t_n; \langle n \rangle$	0; +0	**	x	>	≠0			R	28	4B			
Gleitkomma <u>addiere Betrag</u>	GAB	n	<A> := <A> +  <n>	+	x	0; <A> +  <n> *		$t_n; \langle n \rangle$	0; +0	**	x	>	≠0			R	28	52			
Gleitkomma <u>addiere im Speicher</u>	GAC	n	<n> := <A> + <n>	+			0; <A> + <n>*	0; <A> + <n>*	0; +0	**	x	>	≠0	B	R	37	4A				
Gleitkomma <u>subtrahiere</u>	GSB	n	<A> := <A> - <n>	+	x	0; <A> - <n>*		$t_n; \langle n \rangle$	0; +0	**	x	>	≠0			R	28	4F			
Gleitkomma <u>subtrahiere Betrag</u>	GSBB	n	<A> := <A> -  <n>	+	x	0; <A> -  <n> *		$t_n; \langle n \rangle$	0; +0	**	x	>	≠0			R	28	53			
Gleitkomma <u>subtrahiere im Speicher</u>	GSBC	n	<n> := <n> - <A>	+			0; <n> - <A>*	0; <n> - <A>*	0; +0	**	x	>	≠0	B	R	37	4E				
Gleitkomma <u>subtrahiere invers</u>	GSBI	n	<A> := <n> - <A>	+	x	0; <n> - <A>*		$t_n; \langle n \rangle$	0; +0	**	x	>	≠0			R	28	48			
Gleitkomma <u>subtrahiere von D</u>	GSBD	n	<A> := <D> - <n>	+	x	0; <D> - <n>*			0; +0	**	x	>	≠0			R	30	4C			
Gleitkomma <u>multipliziere</u>	GML	n	<A> := <A> · <n>	+	x	0; <A> · <n>*		$t_n; \langle n \rangle$	0; +0	**	x	>	≠0			R	54	5E			
Gleitkomma <u>multipliziere negativ</u>	GMLN	n	<A> := -<A> · <n>	+	x	0; -<A> · <n>*		$t_n; \langle n \rangle$	0; +0	**	x	>	≠0			R	54	5C			
Gleitkomma <u>multipliziere akkumulierend</u>	GMLA	n	<A> := <A> · <n> + <H>	+	x	0; <A> · <n> + <H>*		$t_n; \langle n \rangle$	0; +0	**	x	>	≠0			R	97	5F			
Gleitkomma <u>multipliziere akkum. negativ</u>	GMAN	n	<A> := -<A> · <n> + <H>	+	x	0; -<A> · <n> + <H>*		$t_n; \langle n \rangle$	0; +0	**	x	>	≠0			R	97	5D			
Gleitkomma <u>dividiere</u>	GDV	n	<A> := <A> : <n>	+	x	0; <A> : <n>*		0; +0	0; +0	**	x	>	≠0			R	213	64			
Gleitkomma <u>dividiere invers</u>	GDVI	n	<A> := <n> : <A>	+	x	0; <n> : <A>*		0; +0	0; +0	**	x	>	≠0			R	216	66			
<u>Addiere unnormalisiert</u>	AU	n	<A> := <A> + <n>	+	x	0; <A> + <n>		$t_n; \langle n \rangle$	0; +0	+0	x	>	≠0			R	26	49			
<u>Subtrahiere unnormalisiert</u>	SBU	n	<A> := <A> - <n>	+	x	0; <A> - <n>		$t_n; \langle n \rangle$	0; +0	+0	x	>	≠0			R	26	4D			
Bilde <u>reziproken Wert</u>	REZ	n	<A> := 1 : <n>	+	x	0; 1 : <n>*		0; +0	0; +0	**	x	>	≠0			R	213	65			

						<A, Q>	H									
<u>Doppelte Genauigkeit: Addiere</u>	DA	n	<A, Q> := <A, Q> + <n, n+2>			0, 1; <A, Q> + <n, n+2>*	1; +0	1; +0	**	x	>	2, 3	BR	106	F0	$\langle A \rangle_t = \langle n \rangle_t = 0$ $\langle Q \rangle_t = \langle n+2 \rangle_t = 1$
<u>Doppelte Genauigkeit: Subtrahiere</u>	DSB	n	<A, Q> := <A, Q> - <n, n+2>			0, 1; <A, Q> - <n, n+2>*	1; +0	1; +0	**	x	>	2, 3	BR	106	F1	
<u>Doppelte Genauigkeit: Multipliziere</u>	DML	n	<A, Q> := <A, Q> · <n, n+2>			0, 1; <A, Q> · <n, n+2>*	1; +0	1; +0	**	x	>	2, 3	BR	243	F2	
Gleitkomma <u>multipliz. auf doppelte Genauigk.</u>	MLD	n	<A, Q> := <A> · <n>	x		0, 1; <A> · <n>*	1; +0	1; +0	**	x	>	≠0	BR	64	F3	

						<A, z>												
<u>Addiere Adressenteil (TK = 0)</u>	AA	z	<A> := <A> · 16 <sup>z</sup>	+		0; <A> · 16 <sup>z</sup>	1; 0, z				>				R	13	98	nur bei <A> <sub>t</sub> = 0; bei TK ≠ 0 siehe Festkomma-Arithmetik.  z  < 2 <sup>7</sup> nach Modifizierung
<u>Subtrahiere Adressenteil (TK = 0)</u>	SBA	z	<A> := <A> · 16 <sup>-z</sup>	+		0; <A> · 16 <sup>-z</sup>	1; 0, z				>				R	13	99	

\* normalisiert und gerundet

\*\* Anzahl der Binärstellen um die das Ergebnis normalisiert wurde.

Falls Ergebnis = ±0 oder Exponentenunterlauf: <Y> := +0

# Boolesche Operationen

Bezeichnung	Code	adr	Wirkung	mod2	R	$\langle A \rangle$	$\langle D \rangle$	Alarm			Bemerkungen		
								$\langle M \rangle$	BÜ	Werk			
<u>VEL</u>	VEL	n	$\langle A \rangle := \langle A \rangle \vee \langle n \rangle$	+	x	$t_{n,x}; \langle A \rangle \vee \langle n \rangle$	$t_n; \langle n \rangle$	x		R	5	68	
<u>AUT</u>	AUT	n	$\langle A \rangle := \langle A \rangle \# \langle n \rangle$	+	x	$t_{n,x}; \langle A \rangle \# \langle n \rangle$	$t_n; \langle n \rangle$	x		R	5	69	
<u>ET</u>	ET	n	$\langle A \rangle := \langle A \rangle \wedge \langle n \rangle$	+	x	$t_{n,x}; \langle A \rangle \wedge \langle n \rangle$	$t_n; \langle n \rangle$	x		R	5	6A	
Setze <u>zusammen</u>	ZUS	n	$\langle A \rangle_x := \langle A \rangle_x$ : für $\langle H \rangle_x = 0$ $\langle A \rangle_x := \langle n \rangle_x$ : für $\langle H \rangle_x = L$	+	x	$t_{n,x}(\langle A, n \rangle) \dots$ ; (s. Wirkung)	$t_n; \langle n \rangle$	x		R	5	6B	
<u>VEL</u> Adressenteil	VLA	z	$\langle A \rangle := \langle H \rangle \vee z$	+		$t_H; \langle H \rangle \vee (0, z)$	$1; 0, z$			R	8	88	} z: 0...65535
<u>AUT</u> Adressenteil	ATA	z	$\langle A \rangle := \langle H \rangle \# z$	+		$t_H; \langle H \rangle \# (0, z)$	$1; 0, z$			R	8	89	
<u>ET</u> Adressenteil	ETA	z	$\langle A \rangle := \langle H \rangle \wedge z$	+		$t_H; \langle H \rangle \wedge (0, z)$	$1; 0, z$			R	8	8A	

# Halbwort-Arithmetik

	Code	adr	Wirkung	mod2	R	$\langle A \rangle$	$\langle Q \rangle$	$\langle D \rangle$	$\langle Y \rangle$	Alarm			Bemerkungen	
										$\langle M \rangle$	BÜ	Werk		
<u>Addiere Halbwort</u>	A2	m	$\langle A \rangle := \langle A \rangle_{25-48} + \langle m \rangle$	+	x	$t_A; v, (\langle A \rangle_{25-48} + \langle m \rangle)$		$t_A; +0$			R	14	7C	
<u>Subtrahiere Halbwort</u>	SB2	m	$\langle A \rangle := \langle A \rangle_{25-48} - \langle m \rangle$	+	x	$t_A; v, (\langle A \rangle_{25-48} - \langle m \rangle)$		$t_A; +0$			R	14	7D	
<u>Multipliziere Halbwort</u>	M2	m	$\langle A \rangle := \langle A \rangle \cdot \langle m \rangle$	+	x	$t_A; \langle A \rangle \cdot \langle m \rangle$	$t_A; +0$	$t_A; +0$	+0		R	49,5	7A	beim Ergebnis entfallen die linken 24 Bits
<u>Multipliziere Halbwort negativ</u>	M2N	m	$\langle A \rangle := -\langle A \rangle \cdot \langle m \rangle$	+	x	$t_A; -\langle A \rangle \cdot \langle m \rangle$	$t_A; +0$	$t_A; +0$	+0		R	49,5	78	
<u>Multipliziere Halbwort mit Rundung</u>	M2R	m	$\langle A \rangle := \langle A \rangle \cdot \langle m \rangle$ gerundet	+	x	$t_A; (\langle A \rangle \cdot \langle m \rangle)$ gerundet	$t_A; +0$	$t_A; +0$	+0		R	36,5	7B	Beim Ergebnis entfallen die rechten 24 Bits und es wird gerundet
<u>Multipliziere Halbwort negativ mit Rundung</u>	M2NR	m	$\langle A \rangle := -\langle A \rangle \cdot \langle m \rangle$ gerundet	+	x	$t_A; (-\langle A \rangle \cdot \langle m \rangle)$ gerundet	$t_A; +0$	$t_A; +0$	+0		R	36,5	79	

# Teilwort-Arithmetik

	Code	adr	Wirkung	mod2	R	$\langle A \rangle$	$\langle D \rangle$	Alarm			Bemerkungen		
								$\langle M \rangle$	BÜ	Werk			
<u>Addiere Teilwort</u>	AT	n	$\langle A \rangle := \langle A \rangle_x + \langle n \rangle_x$	+	x	Hilfsgröße qr := um p Stellen rechts im Kreis geschifteter $\langle Q \rangle$ Hilfsgröße nr := um p Stellen rechts geschifteter $\langle n \rangle$	$t_q; \langle Q \rangle$	>		R	21+2p	F4	p = Anzahl der rechts in Q anstehenden l-Bits Falls $\langle Q \rangle = LL\dots L$ dann p = 0
<u>Subtrahiere Teilwort</u>	SBT	n	$\langle A \rangle := \langle A \rangle_x - \langle n \rangle_x$	+	x	Das Nullfeld von qr schneidet in nr und $\langle A \rangle$ Teilwörter aus; diese werden addiert bzw. subtrahiert. Die anderen Stellen werden auf 0 gelöscht. $t_A$ ; Ergebnis	$t_q; \langle Q \rangle$	<		R	21+2p	F5	Die Zahlen werden als positive, vorzeichenlose, ganze Festkommazahlen aufgefaßt.

# Index-Arithmetik

Bezeichnung	adr	Wirkung	mod2	R	$\langle B \rangle$	$\langle i \rangle$	$\langle i_s \rangle$	Werk	Takte	Int.	Bemerkungen
Erhöhe <u>B</u> um Speicher	HBC	m		x	$\langle B \rangle + \langle m \rangle$			B	6,5	3C	
Vermindere <u>B</u> um Speicher	VBC	m		x	$\langle B \rangle - \langle m \rangle$			B	8,5	15	
Erhöhe <u>B</u> um Adressenteil	HBA	z			$\langle B \rangle + z$			B	7,5	11	z: 0...65 535
Vermindere <u>B</u> um Adressenteil	VBA	z			$\langle B \rangle - z$			B	8,5	13	
Erhöhe <u>B</u> um Parameter mal Indexzelle	HBPX	p i			$\langle B \rangle := \langle B \rangle + p \cdot \langle i \rangle$			B	5,5   p   +3	OF	p: ±1...±15
Erhöhe Indexzelle um Parameter	HXP	p i			$\langle B \rangle := \langle i \rangle + p$ $\langle i \rangle := \langle i \rangle + p$	$\langle i \rangle + p$	$\langle i \rangle + p$	B	10,5	2C	p: ±0...±127
Erhöhe Indexzelle um Indexzelle	HXX	i <sub>L</sub> i <sub>R</sub>			$\langle B \rangle := \langle i_R \rangle + \langle i_L \rangle$ $\langle i_R \rangle := \langle i_R \rangle + \langle i_L \rangle$	$\langle i_R \rangle + \langle i_L \rangle$	$\langle i_R \rangle + \langle i_L \rangle$	B	14,5	2E	
Vermindere Indexzelle um Indexzelle	VXX	i <sub>L</sub> i <sub>R</sub>			$\langle B \rangle := \langle i_R \rangle - \langle i_L \rangle$ $\langle i_R \rangle := \langle i_R \rangle - \langle i_L \rangle$	$\langle i_R \rangle - \langle i_L \rangle$	$\langle i_R \rangle - \langle i_L \rangle$	B	14,5	2F	
Register und Indexzelle	RX	s i			$\langle B \rangle := \langle i \rangle \pm \langle s_1 \rangle$ Falls $s_3 = C$ : $\langle i \rangle := \langle i \rangle \pm \langle s_1 \rangle$	$\langle i \rangle \pm \langle s_1 \rangle$ $\langle i \rangle \pm \langle s_1 \rangle$	nur falls $s_3 = C$ $\langle i \rangle \pm \langle s_1 \rangle$	B R	12	* 8D	$s_1$ : A,Q,D,H (rechtes Halbwort) oder B $s_2$ : leer = positiv + N = negativ - } statt ± $s_3$ : leer = nicht zurückspeichern C = zurückspeichern

\*siehe Internspezifikation auf Seite 23

# Setzen und Löschen

					$\langle A \rangle$	$\langle H \rangle$	$\langle s_2 \rangle$				
Bringe Adressenteil	BA	z		+	1;v,z				R	2	8E
Bringe Adressenteil und reserviere	BAR	z		+	1;v,z	$t_A; \langle A \rangle$			R	3	DC
Bringe Adressenteil negativ	BAN	z		+	1;-(v,z)				R	5	DF
Bringe Adressenteil negativ und reserviere	BANR	z		+	1;-(v,z)	$t_A; \langle A \rangle$			R	5	DD
Lösche Register	LR	s		+	$\langle s_2 \rangle := +0$ $\langle s_2 \rangle_t := s_1$		$s_1; +0$		R	3	9A
Lösche in <u>A</u>	LA	s		+	$\langle A \rangle_s := 0$ $\langle M \rangle := 0$ nur bei $s = M$			Wirkung: s = F: $\langle A \rangle_{1-40} := 0 \hat{=}$ Mantissenteil s = 2: $\langle A \rangle_{1-24} := 0 \hat{=}$ linkes Halbwort s = E: $\langle A \rangle_{41-48} := 0 \hat{=}$ Exponententeil s = 3: $\langle A \rangle_{33-48} := 0 \hat{=}$ rechtes Drittel s = V: $\langle A \rangle_{1-2} := 0 \hat{=}$ Vorzeichenstellen s = M: $\langle M \rangle := 0 \hat{=}$ Markenregister s = H: $\langle A \rangle_{1-42} := 0 \hat{=}$ ohne rechte Hexade s = T: $\langle A \rangle_{1-44} := 0 \hat{=}$ ohne rechte Tetrade	R	4	8F
								Die Spezifikationen F bis M können kombiniert verwendet werden H und T nur einzeln od. mit M erlaubt			

z: 0...65535

$s_1$ : 0,1,2 oder 3 (TK)  
 $s_2$ : A,Q,D und H

Bezeichnung	Code	adr	Wirkung	mod2	R	<B>	<i>	<U>	Werk	Takte	Int.	Bemerkungen
Index: <u>Bringe Adressenteil</u>	XBA	z	<B> := z			z			B	1	01	z: 0...65535
Index: <u>Bringe Adressenteil negativ</u>	XBAN	z	<B> := -z			-z			B	2	19	
Setze <u>Index</u>	ZX	p i	<i> := p <B> := p			v,p	v,p		B	6	1A	p: ±0...±127
Setze <u>Unterprogrammregister</u>	ZU	i	<U> := i	+				i	B	1	3E	

Bezeichnung	Code	adr	Wirkung	mod2	R	<B>	<i>	<U>	Werk	Takte	Int.	Bemerkungen
Setze <u>Typenkennung im Register</u>	ZTR	s	<s <sub>2</sub> > <sub>t</sub> := s <sub>1</sub> <M> := L, falls s <sub>3</sub> = M	+		s <sub>1</sub>	L, falls s <sub>3</sub> = M		R	5	92	s <sub>1</sub> : 0,1,2,3 oder leer s <sub>2</sub> : A,Q,D,H oder leer s <sub>3</sub> : leer oder M
Setze <u>Typenkennung 0</u>	ZT0	n	<n> <sub>t</sub> := 0	+		0			B	13	C8	
Setze <u>Typenkennung 1</u>	ZT1	n	<n> <sub>t</sub> := 1	+		1			B	13	C9	
Setze <u>Typenkennung 2</u>	ZT2	n	<n> <sub>t</sub> := 2	+		2			B	13	CA	
Setze <u>Typenkennung 3</u>	ZT3	n	<n> <sub>t</sub> := 3	+		3			B	13	CB	

Bezeichnung	Code	adr	Wirkung	mod2	R	<B>	<i>	<U>	Werk	Takte	Int.	Bemerkungen
<u>Lösche Speicher</u>	LC	n	<n> := +0 <n> <sub>n</sub> := <n> <sub>n</sub>	+		t <sub>n</sub> ; +0	<n> <sub>n</sub>		B	13	33	
<u>Lösche markiert</u>	LMT	n	<n> := +0 <n> <sub>n</sub> := L	+		t <sub>n</sub> ; +0	L		B	13	32	
<u>Lösche Marke im Speicher</u>	LMC	n	<n> <sub>n</sub> := 0	+			0		B	13	31	
Setze <u>Marke im Speicher</u>	ZMC	n	<n> <sub>n</sub> := L	+			L		B	13	30	
<u>Lösche und setze Merklichter</u>	LZL	s <sub>L</sub> s <sub>R</sub>	<K> <sub>s<sub>R</sub></sub> := 0 <K> <sub>s<sub>L</sub></sub> := L				L	0	B	3	10	s: Merklichter 0,1,2,3,4,5,6,7 und 8 0 bedeutet kein Merklicht (0 muß angegeben werden)
<u>Negiere Merklichter</u>	NL	s	<K> <sub>s</sub> := <K> <sub>s</sub> invertiert				invertiert		B	3	12	
Setze <u>Indexbasis</u>	ZI	m	<X> := <m> <sub>3-24</sub>					<m> <sub>3-24</sub>	B	17 x + 58	B7	x: Anzahl der Indexregister, die zurückgespeichert werden

# Sprünge

Bezeichnung	Code	adr	Wirkung	mod2	R	$\langle F \rangle_{9-24}$	$\langle F \rangle_{1-24}$	$\langle H \rangle$	$\langle B \rangle$	$\langle U \rangle$	U-Alarm	Werk	Takte	Int.	Bemerkungen
<u>Null</u> befehl	NULL	z	keine Wirkung		x							B	1	00	Nur übliche Erhöhung des Befehlsfolgeregisters. z: ohne Bedeutung, muß angegeben werden.
<u>Warte</u> befehl	WB	z	z Uhrimpulse warten (Uhrimpulse alle 10µs)									B R	s.Be-merkung.	F8	z: 0...65535 Takte: bei z = 0: 3 Takte bei z ≠ 0: (10z-5)µs
<u>Sprunge</u>	S	m	$\langle F \rangle_{9-24} := m$	+		m						B	1	36	
<u>Sprunge</u> nach <u>E</u> rsatzung	SE	m	$\langle F \rangle := \langle m \rangle + \text{mod}2$	sp	x		$\langle m \rangle + \text{mod}2$					B	2	BC	Sprung in and. Großseite mögl.
<u>Sprunge</u> und bringe $\langle F \rangle + 1$ nach <u>B</u>	SFB	m	$\langle B \rangle := \langle F \rangle + 1$ $\langle F \rangle_{9-24} := m$	+		m			$\langle F \rangle + 1$			B	1	3A	
<u>Sprunge</u> nach <u>E</u> rsatzung und bringe $\langle F \rangle + 1$ nach <u>B</u>	SFBE	m	$\langle B \rangle := \langle F \rangle + 1$ $\langle F \rangle_{1-24} := \langle m \rangle + \text{mod}2$	sp			$\langle m \rangle + \text{mod}2$		$\langle F \rangle + 1$			B R	15	FA	Sprung in andere Großseite mögl.
<u>Sprunge</u> in <u>U</u> nterprogramm	SU	m	$\langle U \rangle := \langle U \rangle + 1$ $\langle \langle U \rangle \rangle := \langle F \rangle + 1$ $\langle F \rangle_{9-24} := m$	+		m			$\langle U \rangle + 1$	$\langle U \rangle_{alt} = 254$		B	8	38	$\langle U \rangle$ : 0...255 ( $\langle U \rangle$ : 255 + 1 = 0)
<u>Sprunge</u> in <u>U</u> nterprogramm nach <u>E</u> rsatzung	SUE	m	$\langle U \rangle := \langle U \rangle + 1$ $\langle \langle U \rangle \rangle := \langle F \rangle + 1$ $\langle F \rangle := \langle m \rangle + \text{mod}2$	sp	x		$\langle m \rangle + \text{mod}2$		$\langle U \rangle + 1$	$\langle U \rangle_{alt} = 254$		B	8	BD	$\langle U \rangle$ : 0...255 ( $\langle U \rangle$ : 255 + 1 = 0) Sprung in and. Großseite mögl.
<u>Sprunge</u> ins <u>S</u> ystem und <u>r</u> eserviere	SSR	Pl Pr	$\langle F \rangle := \langle a + 6 \rangle$ $\langle B \rangle := \text{adr}$	+			$\langle a + 6 \rangle$				adr	B R	48	BB	$\left. \begin{matrix} Pl \\ Pr \end{matrix} \right\} : 0...255$ a = $\langle BL \rangle$ , $2^8$ = Anfangsadresse des Leitblocks

			Sprungbedingung				$\langle A \rangle$								
<u>Sprunge</u> wenn <u>i</u> dentisch <u>0</u>	SIO	m	$\langle A \rangle = 0$	+		m *						B R	1	A4	
<u>Sprunge</u> wenn <u>n</u> icht <u>0</u>	SNO	m	$\langle A \rangle \neq 0$	+		m *						B R	1	A7	
<u>Sprunge</u> wenn <u>g</u> rößer <u>g</u> leich <u>0</u>	SGGO	m	$\langle A \rangle \geq 0$	+		m *						B R	1	A6	$\langle A \rangle_t = 2$ oder 3: Sprung immer!
<u>Sprunge</u> wenn <u>g</u> rößer <u>0</u>	SGO	m	$\langle A \rangle > 0$	+		m *						B R	1	D4	
<u>Sprunge</u> wenn <u>k</u> leiner <u>g</u> leich <u>0</u>	SKGO	m	$\langle A \rangle \leq 0$	+		m *						B R	1	A5	
<u>Sprunge</u> wenn <u>k</u> leiner <u>0</u>	SKO	m	$\langle A \rangle < 0$	+		m *						B R	1	D5	$\langle A \rangle_t = 2$ oder 3: Sprung nie!
<u>Sprunge</u> wenn <u>r</u> echtes Bit in A gesetzt	SR	m	$\langle A \rangle_{4e} = 1$	+		m *						B R	1	B8	
<u>Sprunge</u> wenn <u>r</u> echtes Bit in A <u>n</u> icht gesetzt	SRN	m	$\langle A \rangle_{4e} = 0$	+		m *						B R	1	BA	
<u>Sprunge</u> wenn <u>i</u> dentisch	SI	m	$\langle A \rangle = \langle H \rangle$	+		m *	norm**	norm**				B R	**	6	AC
<u>Sprunge</u> wenn <u>n</u> icht identisch	SN	m	$\langle A \rangle \neq \langle H \rangle$	+		m *	norm**	norm**				B R	**	6	AD
<u>Sprunge</u> wenn <u>g</u> rößer <u>g</u> leich	SGG	m	$\langle A \rangle \geq \langle H \rangle$	+		m *	norm**	norm**				B R	**	6	AF
<u>Sprunge</u> wenn <u>g</u> rößer	SG	m	$\langle A \rangle > \langle H \rangle$	+		m *	norm**	norm**				B R	**	6	AB
<u>Sprunge</u> wenn <u>k</u> leiner <u>g</u> leich	SKG	m	$\langle A \rangle \leq \langle H \rangle$	+		m *	norm**	norm**				B R	**	6	AE
<u>Sprunge</u> wenn <u>k</u> leiner	SK	m	$\langle A \rangle < \langle H \rangle$	+		m *	norm**	norm**				B R	**	6	AA

Bezeichnung	Code	adr	Sprungbedingung	mod? R	$\langle F \rangle_{e-24}$	$\langle i \rangle$	$\langle M \rangle \langle K \rangle_s$	$\langle D \rangle$	$\langle U \rangle$	Alarm	Werk	lakte	Int.	Bemerkungen																													
Springe wenn Index <u>identisch</u> 0	SXI	m	$\langle B \rangle = \pm 0$								B	1	1	24	$\langle B \rangle_1 = \langle B \rangle_v$																												
Springe wenn Index <u>nicht</u> identisch 0	SXN	m	$\langle B \rangle \neq \pm 0$								B	1	1	27																													
Springe wenn Index <u>größer</u> gleich 0	SXGG	m	$\langle B \rangle \geq \pm 0$								B	1	1	25																													
Springe wenn Index <u>größer</u> 0	SXG	m	$\langle B \rangle > \pm 0$								B	1	1	CE																													
Springe wenn Index <u>kleiner</u> gleich 0	SXXG	m	$\langle B \rangle \leq \pm 0$								B	1	1	26																													
Springe wenn Index <u>kleiner</u> 0	SXK	m	$\langle B \rangle < \pm 0$								B	1	1	CF																													
Springe wenn Indexgröße <u>rechtes</u> Bit = L	SXR	m	$\langle B \rangle_{24} = L$								B	1	1	B2																													
Springe wenn Indexgröße <u>rechtes</u> Bit <u>nicht</u> L	SXRN	m	$\langle B \rangle_{24} = 0$								B	1	1	B3																													
Springe und <u>zähle</u> wenn Index kleiner 0	SZX	p i	$\langle i \rangle < \pm 0$			$\langle F \rangle + p *$	$\langle i \rangle + 1 *$				B	11	2	0A	$\langle i \rangle_1 = \langle i \rangle_v$ p: $\pm 0 \dots \pm 127$																												
Springe wenn <u>Marke</u>	SM	m	$\langle M \rangle = L$	+				0			B	1	1	34																													
Springe wenn <u>Marke</u> <u>nicht</u>	SMN	m	$\langle M \rangle = 0$	+				0			B	1	1	35																													
Springe wenn <u>arithmetischer</u> Alarm (BÜ-Alarm)	SAA	m	BÜ-Alarm	+						BÜ-Alarm wird gelöscht	B R	1	1	A9																													
Springe wenn <u>Alarm</u> (Typenkennung)	SAT	m	TK-Alarm	+						TK-Alarm wird gelöscht	B R	1	1	A8																													
Springe wenn <u>Typenkennung</u>	ST	p s	$\langle s_2 \rangle_t = s_1$	+		$\langle F \rangle + p *$					B R	10	3	90	p: $\pm 0 \dots \pm 127$ $s_1$ : 0,1,2 oder 3 (TK) $s_2$ : A,Q,D oder H																												
Springe wenn <u>Typenkennung</u> <u>nicht</u>	STN	p s	$\langle s_2 \rangle_t \neq s_1$	+		$\langle F \rangle + p *$					B R	10	3	91																													
Springe wenn <u>Bit</u> gesetzt	SBIT	p s	$\langle s_2 \rangle_{s_1} = L$			$\langle F \rangle + p *$	Tabelle für Zeitberechnung: <table border="1" style="font-size: small;"> <tr> <td><math>s_1</math></td> <td>1</td> <td>2</td> <td>...</td> <td>14</td> <td>15</td> <td>16</td> <td>17</td> <td>...</td> <td>31</td> <td>32</td> <td>33</td> <td>...</td> <td>47</td> <td>48</td> </tr> <tr> <td>q</td> <td>16</td> <td>15</td> <td>...</td> <td>3</td> <td>2</td> <td>1</td> <td>16</td> <td>...</td> <td>2</td> <td>1</td> <td>16</td> <td>...</td> <td>2</td> <td>1</td> </tr> </table>			$s_1$	1	2	...	14	15	16	17	...	31	32	33	...	47	48	q	16	15	...	3	2	1	16	...	2	1	16	...	2	1	B R	7+2q	F9	$s_1$ : Bitnummer 1...48 $s_2$ : Register A, Q, D oder H p: Sprungweite $\pm 0 \dots \pm 127$ q: s. Tab. für Zeitberechn.
$s_1$	1	2	...	14	15	16	17	...	31	32	33	...	47	48																													
q	16	15	...	3	2	1	16	...	2	1	16	...	2	1																													
Springe wenn <u>Exponent</u> <u>größer</u> gleich	SEGG	pl pr	$\langle A \rangle_{41-48} \geq p_r$	+		$\langle F \rangle + p_l *$			$t_A; \langle A \rangle$	falls $\langle A \rangle_t \neq 0$ : TK-Alarm	B R	10	4	93	$p_l$ : Sprungweite p: $\pm 0 \dots \pm 127$ $p_r$ : Vergl.-Exponent ****																												
Springe wenn <u>Merklicht</u>	SL	p s	eines der $\langle K \rangle_s = L$			$\langle F \rangle + p *$					B	9	1	1E	p: $\pm 0 \dots \pm 127$ s: Merkleichter 1,2,... und 8																												
Springe wenn <u>Merklicht</u> und <u>lösche</u>	SLL	p s	eines der $\langle K \rangle_s = L$			$\langle F \rangle + p *$		0			B	9	1	1F																													
Springe wenn <u>Merklicht</u> <u>nicht</u>	SLN	p s	alle $\langle K \rangle_s = 0$			$\langle F \rangle + p *$					B	9	1	1C																													
Springe wenn <u>Merklicht</u> <u>nicht</u> sonst <u>lösche</u>	SNL	p s	alle $\langle K \rangle_s = 0$			$\langle F \rangle + p *$		0			B	9	1	1D																													

						$\langle A \rangle$	$\langle Y \rangle$							
<u>Prüfe</u> <u>Dreierprobe</u> und <u>springe</u> wenn richtig	PDP	n	DP = richtig	+		$\langle F \rangle + 2 *$	$t_n; \langle n \rangle$	$\langle n \rangle_{s-6} OL, DP, \langle n \rangle_t$	R	21	21	B9	DP: Dreierprobenbits Transp. unverän. bei jeder TK DP <u>nicht</u> richt.: kein DP-Alarm	

\* bei erfüllter Sprungbedingung  
\*\* wenn  $\langle A \rangle_t = \langle H \rangle_t = 0$ :  $\langle A \rangle$  :=  $\langle A \rangle$  normalisiert  
 $\langle H \rangle$  :=  $\langle H \rangle$  normalisiert  
ist der Exponent +0 wird er zu -0, zur  
Ausführungszeit kommt ein Takt hinzu

\*\*\* bei nicht erfüllter Sprungbedingung  
\*\*\*\* 0...127 (positiv), NO...N127 (negativ)

# Modifizieren



Bezeichnung	Code	adr	Wirkung	mod2	R	mod1	mod2...	<B>	<i>	Werk	Takte	Int.	Bemerkungen
Modifiziere in jedem Fall	MF	i	mod1 := <i> + mod2 <B> := <i> + mod2	sp		<i> *	0	<i> *		B	1	0B	
Modifiziere in jedem Fall mit unveränd. B	MFU	i	mod1 := <i> + mod2	sp		<i> *	0			B	1	08	Bits 9 - 11 = 00L
Modifiziere aus Speicher in jedem Fall	MCF	m	mod1 := <m> + mod2 <B> := <m> + mod2	sp	x	<m> *	0	<m> *		B	2	16	
Modifiziere aus Speicher in jedem Fall mit unverändertem B	MCFU	m	mod1 := <m> + mod2	sp	x	<m> *	0			B	1	3D	
Modifiziere doppelt	MD	i <sub>L</sub> i <sub>R</sub>	mod1 := <i <sub>Rmod2 := &lt;i<sub>L&lt;B&gt; := &lt;i<sub>L <td>sp</td> <td></td> <td>&lt;i<sub>R <td>&lt;i<sub>L <td>&lt;i<sub>L <td></td> <td>B</td> <td>5</td> <td>09</td> <td></td> </sub></td></sub></td></sub></td></sub></sub></sub>	sp		<i <sub>R <td>&lt;i<sub>L <td>&lt;i<sub>L <td></td> <td>B</td> <td>5</td> <td>09</td> <td></td> </sub></td></sub></td></sub>	<i <sub>L <td>&lt;i<sub>L <td></td> <td>B</td> <td>5</td> <td>09</td> <td></td> </sub></td></sub>	<i <sub>L <td></td> <td>B</td> <td>5</td> <td>09</td> <td></td> </sub>		B	5	09	
Modifiziere	M	i	<B> := <i> + mod2 mod2 := <i> + mod2	sp			<i> *	<i> *		B	1	08	Bits 9 - 11 = L00
Modifiziere nach Erhöhung	MH	p i	mod2 := <i> + p <B> := <i> + p <i> := <i> + p				<i> + p	<i> + p	<i> + p	B	10,5	2D	p: ±0...±127
Modifiziere nach Erhöhung um Indexzelle	MHX	i <sub>L</sub> i <sub>R</sub>	mod2 := <i <sub>LR&lt;B&gt; := &lt;i<sub>LR&lt;i<sub>RLR <td></td> <td></td> <td></td> <td>&lt;i<sub>LR <td>&lt;i<sub>LR <td>&lt;i<sub>RLR <td>B</td> <td>14,5</td> <td>0E</td> <td></td> </sub></td></sub></td></sub></td></sub></sub></sub>				<i <sub>LR <td>&lt;i<sub>LR <td>&lt;i<sub>RLR <td>B</td> <td>14,5</td> <td>0E</td> <td></td> </sub></td></sub></td></sub>	<i <sub>LR <td>&lt;i<sub>RLR <td>B</td> <td>14,5</td> <td>0E</td> <td></td> </sub></td></sub>	<i <sub>RLR <td>B</td> <td>14,5</td> <td>0E</td> <td></td> </sub>	B	14,5	0E	
Modifiziere mit Register und Indexzelle	MRX	s i	mod2 := <i> ± <s <sub>1&lt;B&gt; := &lt;i&gt; ± &lt;s<sub>1falls s<sub>3</sub> = C: &lt;i&gt; := &lt;i&gt; ± &lt;s<sub>1 <td></td> <td></td> <td></td> <td>&lt;i&gt; ± &lt;s<sub>1 <td>&lt;i&gt; ± &lt;s<sub>1 <td>falls s<sub>3</sub> = C: &lt;i&gt; ± &lt;s<sub>1 <td>B</td> <td>12</td> <td>8D **</td> <td>s<sub>1</sub>: A, Q, D, H (rechtes Halbwort) oder B s<sub>2</sub>: leer: + } statt ± : N : - s<sub>3</sub>: leer: nicht speichern : C : zurückspeichern</td> </sub></td></sub></td></sub></td></sub></sub></sub>				<i> ± <s <sub>1 <td>&lt;i&gt; ± &lt;s<sub>1 <td>falls s<sub>3</sub> = C: &lt;i&gt; ± &lt;s<sub>1 <td>B</td> <td>12</td> <td>8D **</td> <td>s<sub>1</sub>: A, Q, D, H (rechtes Halbwort) oder B s<sub>2</sub>: leer: + } statt ± : N : - s<sub>3</sub>: leer: nicht speichern : C : zurückspeichern</td> </sub></td></sub></td></sub>	<i> ± <s <sub>1 <td>falls s<sub>3</sub> = C: &lt;i&gt; ± &lt;s<sub>1 <td>B</td> <td>12</td> <td>8D **</td> <td>s<sub>1</sub>: A, Q, D, H (rechtes Halbwort) oder B s<sub>2</sub>: leer: + } statt ± : N : - s<sub>3</sub>: leer: nicht speichern : C : zurückspeichern</td> </sub></td></sub>	falls s <sub>3</sub> = C: <i> ± <s <sub>1 <td>B</td> <td>12</td> <td>8D **</td> <td>s<sub>1</sub>: A, Q, D, H (rechtes Halbwort) oder B s<sub>2</sub>: leer: + } statt ± : N : - s<sub>3</sub>: leer: nicht speichern : C : zurückspeichern</td> </sub>	B	12	8D **	s <sub>1</sub> : A, Q, D, H (rechtes Halbwort) oder B s <sub>2</sub> : leer: + } statt ± : N : - s <sub>3</sub> : leer: nicht speichern : C : zurückspeichern
Modifiziere aus Speicher	MC	m	<B> := <m> + mod2 mod2 := <m> + mod2	sp	x		<m> *	<m> *		B	2	14	
Modifiziere aus Speicher nach Ersetzungen	MCE	m	<B> := <<...<m>...>> + mod2 mod2 := <<...<m>...>> + mod2	sp	x		<<...<m>...>> *	<<...<m>...>> *		B	13x -9	17	x: Anzahl der Halbwörter in der Ersetzkette Abbruch wenn: <<...<m>...>> <sub>1</sub> = L
Modifiziere mit Adressenteil	MA	z	<B> := z + mod2 mod2 := z + mod2	sp			z *	z *		B	1	03	z: 0...65 535
Modifiziere mit negativem Adressenteil	MNA	z	<B> := -z + mod2 mod2 := -z + mod2	sp			-z *	-z *		B	2	02	

\* Ist vom vorhergehenden Befehl ein Modifikator 2. Art vorhanden, so wird er addiert

\*\* siehe Internspezifikation auf Seite 23

# Ersetzen ( und modifizieren )



Bezeichnung	Code	adr	Wirkung	mod2	R	op	adr	(B)	(i)	mod2 <sub>1,2</sub>	Werk	lakte	Int.	Bemerkungen																	
<u>E</u> rsetze	E	c i	adr := <i> mod2 := mod2	sp		c	<i>			mod2	B	4	29																		
<u>E</u> rsetze <u>z</u> ählend	EZ	c i	adr := <i> + 2 + mod2 <B> := <i> + 2 <i> := <i> + 2	sp		c	<i> + 2 * <i> + 2	<i> + 2		0	B	5	2B																		
<u>E</u> rsetze <u>n</u> egativ <u>z</u> ählend	ENZ	c i	adr := <i> + mod2 <B> := <i> - 2 <i> := <i> - 2	sp		c	<i> * <i> - 2	<i> - 2		0	B	8	2A																		
<u>E</u> rsetze und <u>m</u> odifiziere mit <u>B</u>	EMB	c i	adr := <i> mod2 := <B>			c	<i>			<B>	B	4	28																		
<u>M</u> odifiziere <u>A</u> dressenteil mit <u>B</u>	MAB	c p	adr := <B> + p <B> := <B> + p			c	<B> + p	<B> + p		0	B	9,5	20	p: ±0...±127																	
<u>M</u> odifiziere <u>A</u> dressenteil mit <u>B</u> bei <u>I</u> nvarianz der Sprungadresse	MABI	c p	adr := <B> + p <B> := <B> + p			c	<B> + p	<B> + p		0	B	9,5	3F	p: ±0...±127 **																	
<u>M</u> odifiziere über <u>U</u>	MU	c p	adr := <<U>> + p			c	<<U>> + p			0	B	14,5	05	<U> := <U>-1 wenn c Sprungbef. u. Bedingung erfüllt ist** p: ±0...±127																	
<u>E</u> rsetze nach <u>M</u> odifizierung über <u>U</u>	EMU	c p	adr := <<<U>> + p mod2 := mod2	sp		c	<<<U>> + p			mod2	B	29,5	04	p: ±0...±127																	
<u>R</u> elativ- <u>A</u> dressierung mit <u>R</u> egisterinhalt	RLR	c s	adr := <s> + <F> mod2 := mod2	sp		c	<s> + <F>	falls s = F: adr := 2·<F> + 1		mod2	B R	10	EO	s: A, Q, D, H = rechte Hälfte AL, QL, DL, HL = linke Hälfte F, B, U, Y																	
<u>R</u> egisteradressierung	R	c s	operand := <s <sub>1</sub> >			c	<s <sub>1</sub> > = Operand:	<table border="0"> <tr> <td>s<sub>1</sub> = A, Q, D oder H:</td> <td>ro := t<sub>s1</sub>; &lt;s<sub>1</sub>&gt;***</td> </tr> <tr> <td>s<sub>1</sub> = Y, U</td> <td>: ro := 1; 0, &lt;s<sub>1</sub>&gt;</td> </tr> <tr> <td>s<sub>1</sub> = B</td> <td>: ro := 1; v, &lt;B&gt;</td> </tr> <tr> <td>s<sub>1</sub> = F</td> <td>: ro := 1; v, &lt;F&gt;+1</td> </tr> </table>	s <sub>1</sub> = A, Q, D oder H:	ro := t <sub>s1</sub> ; <s <sub>1</sub> >***	s <sub>1</sub> = Y, U	: ro := 1; 0, <s <sub>1</sub> >	s <sub>1</sub> = B	: ro := 1; v, <B>	s <sub>1</sub> = F	: ro := 1; v, <F>+1							<table border="0"> <tr> <td>s<sub>2</sub>: L</td> <td>: linke 24 Bits</td> </tr> <tr> <td>:</td> <td>leer : rechte 24 Bits</td> </tr> <tr> <td>c</td> <td>: erlaubter Code</td> </tr> <tr> <td>ro</td> <td>: Registeroperand</td> </tr> </table>	s <sub>2</sub> : L	: linke 24 Bits	:	leer : rechte 24 Bits	c	: erlaubter Code	ro	: Registeroperand
s <sub>1</sub> = A, Q, D oder H:	ro := t <sub>s1</sub> ; <s <sub>1</sub> >***																														
s <sub>1</sub> = Y, U	: ro := 1; 0, <s <sub>1</sub> >																														
s <sub>1</sub> = B	: ro := 1; v, <B>																														
s <sub>1</sub> = F	: ro := 1; v, <F>+1																														
s <sub>2</sub> : L	: linke 24 Bits																														
:	leer : rechte 24 Bits																														
c	: erlaubter Code																														
ro	: Registeroperand																														
<u>T</u> ue	T	m	op, adr := <m> mod2 := mod2	sp	x	<m> <sub>1-8</sub>	0, <m> <sub>9-24</sub>			mod2	B	5	CC	<m> <sub>t</sub> : beliebig																	

\* Ist vom vorhergehenden Befehl ein Modifikator 2. Art vorhanden, so wird er addiert

\*\* Sprung in eine andere Großseite ist möglich

\*\*\* bei s<sub>1</sub> = A, Q, D oder H und <s<sub>1</sub>><sub>t</sub> = 0 oder 1:  
<s<sub>1</sub>><sub>1</sub> := <s<sub>1</sub>><sub>a</sub>  
<s<sub>1</sub>><sub>2</sub> := <s<sub>1</sub>><sub>v</sub>

Als Zweitcodes (c) sind alle Befehls-codes zugelassen.  
Bei dem Befehl R jedoch nur:

- A A2 AB AQ AT AU AUT
- B B2 B2V B2VN B3 B3V BB BD BH BN
- BNR BQ BQB BR BT BU
- DV DVD DVI
- ET
- GA GAB GDV GDVI GMAN GML GMLA
- GMLN GSB GSBB GSBD GSBI
- HBC

- M2 M2N M2NR M2R MAN MANR MAR
- MC MCE MCF MCFU ML MLA MLD MLN
- MLR MNR
- NULL
- REZ
- SB SB2 SBB SBD SBI SBQ SBT SBU SE
- SUE
- T TCB
- VBC VEL
- ZUS

# Aufbereitung

Bezeichnung	Code	adr	Wirkung	mod2	R	Voraussetzung	$\langle A \rangle$	$\langle B \rangle$	$\langle Q \rangle$	$\langle Y \rangle$	Werk	Takte	Int.	Bemerkungen	
<u>Umschlüsseln</u>	US	s i	$\langle A \rangle := \langle A \rangle_{\text{umgeschlüsselt}}$	sp		$\langle A \rangle =$ umzu- schl. Zeich. $\langle i \rangle_{\text{mod}2} =$ Anfangsadr. der Umschl- Tabelle *	$t_n; \langle A \rangle_{\text{umge-}} \text{schl.}$ falls $s_2 = E$ $\langle A \rangle_{1-x} := 0$	$\langle i \rangle_{\text{mod}2}$ Adresse der Um- schl.- Tabelle	falls $s_2 = G$ $t_n; \langle A \rangle_{\text{umge-}} \text{schl.}$	falls $s_2 = G$ Anzahl der umgeschlüs- selt. Zeich.	B R	$s_2 = E: 32$ $s_2 = G: \text{und}$ $s_1 = 6: 176$ $= 8: 136$ $= C: 96$	E3	$s_1$ : Bitlänge der Zeichen 6, 8 oder C (12 Bits) $s_2$ : E = ein Zeichen rechtsbünd. G = alle Zeich. 8, 6 oder 4 $t_n$ : Typenkennung des Wortes, in dem das linke Zeichen steht	
<u>Invertiere Register</u>	IR	s	$\langle s \rangle := \neg \langle s \rangle$ $\langle s \rangle :=  \langle s \rangle $			falls $s_2$ : leer, invertiere falls $s_2$ : B, bilde Betrag im Register					R	$4p + 1$	E1	$s_1$ : Register A, Q, D und H p: Anzahl der adr. Register	
Spezifikation <span style="float: right;">BÜ Alarm</span>															
<u>Schifte</u>	SH	s p	Schift gemäß Spezifika- tion um p Stellen	+		$s_1$ : A = Register A Q = Register Q AQ = Register A und Q getrennt Z = doppelt langes Register A, Q $s_2$ : leer, L = Rechtsschift, Linksschift $s_3$ : leer, K = gestreckter Schift, Kreisschift $s_4$ : leer, R = ohne Rundung, mit Rundung $s_5$ : leer, U = abhängig von TK, unabh. von TK $s_6$ : leer, B = nicht zählen, zählen der aus A geschifteten 'L'-Bits im Reg.Y		$\geq 1$			R	$2(q+r)+5$ $p: 4=q$ Rest r zusätzl. 5 wenn $s_4=R$ u. gerundet wurde	9B	1) BÜ Alarm möglich: nur bei Linksschift in A mit TK=0. oder 1; bei Rechtsschift höchstens um 1 Stelle mit Rundung p= Anzahl der Schiftschritte $\pm 0 \dots +127$ $p: 4 = q$ Rest r	
<u>Schifte in B</u>	SHB	s p	$\langle B \rangle := \langle B \rangle$ geschiftet um p Stellen nach rechts oder links mit Nachziehen von 0-Bits			s: R = Schift nach rechts L = Schift nach links					B	$4 * p + 6$	21	p: Anzahl der Schiftschritte 0...255	
Alarm Y BÜ TK															
<u>Vorzeichenangleich zwischen A und Q</u>	VAQ	z	$\langle A, Q \rangle := \langle A \rangle + \langle Q \rangle \cdot 2^{-4e}$ $\langle Q \rangle_v := \langle A \rangle_1$ falls $\langle A \rangle = \pm 0, \langle Q \rangle \neq \pm 0$ $\langle A \rangle_v := \langle Q \rangle_1$	+		VAQ gleicht die Vorzeichen an und berichtigt das Ergebnis unter Er- haltung des Zahlenwertes		$\langle A \rangle_t \neq 1$			R	2 bei $\langle A \rangle_1 = \langle Q \rangle_v$ 17 bei $\langle A \rangle_1 \neq \langle Q \rangle_v$	63	Adressenteil z ist ohne Bedeutung, muß aber an- gegeben werden	
<u>Normalisiere</u>	NRM	s	bei s = G: $\langle A \rangle := \langle A \rangle_{\text{normalisiert}}$	+		Die Festkommazahl wird als echter Bruch aufgefaßt $\langle A \rangle_t := 0; \langle A, Q \rangle_{\text{gerundet}}, \langle Q \rangle := 1; +0$	4t	$\neq 0$				R	$2t + 3$	9F	s = G, FG, F, F4, N oder L
			bei s = FG: $\langle A \rangle_{\text{cl.eitk.}} := \langle A, Q \rangle_{\text{festk.}}$				+0	$\langle A \rangle_1 \neq 1$	$\langle Q \rangle_1 \neq 1$			$2t + 10$	t: Anzahl der Tetraden- schifte		
			bei s = F: $\langle A, Q \rangle := \langle A, Q \rangle_{\text{normalisiert}}$				4t	$\langle A \rangle_1 \neq 1$	$\langle Q \rangle_1 \neq 1$			$2t + 7$			
			bei s = F4: $\langle A, Q \rangle := \langle A, Q \rangle_{\text{normalisiert}}$				t	$\langle A \rangle_1 \neq 1$	$\langle Q \rangle_1 \neq 1$			$2t + 7$			
			bei s = N falls $\langle A \rangle \neq +0$ $\langle A \rangle := \langle A \rangle$ links geschift. bis $\langle A \rangle_1 = L$				e								beim Schiften werden Nullen nachgezogen
			bei s = L: $\langle A \rangle := \langle A \rangle$ links geschift. bis $\langle A \rangle_1 = 0$				48					$2e + 3$	e: Anzahl der Einerschifte		falls $\langle A \rangle = +0: \langle A \rangle := \langle A \rangle$ beim Schiften werden Nullen nachgezogen

\*Tabelle in Viertelwörtern.  
Die Anfangsadresse ist eine Halbwortadresse

Bezeichnung	Code	adr	Wirkung	mod2	R	Voraussetzung	Y	BÜ	TK	Werk	fakte	Int.	Bemerkungen	
Konvertiere Dezimalzahl in Festkommazahl rechtsbündig	KDFR	p	$\langle A \rangle_{r.estk.} := \langle A, Q \rangle_{bez.}$	+		$\langle A, Q \rangle = \text{pos. Dez.-Zahl}$ Komma rechts	$\langle A \rangle := 1; \langle A, Q \rangle \langle D \rangle := 1; +0$ $\langle Q \rangle := 1; +0 \quad \langle H \rangle := 1; +0$	+0			R	7p + 22	94	p: 1...13 Anzahl der Dezimalstellen, die konvertiert werden
Konvertiere Festkommazahl linksbündig in Dezimalzahl	KFLD	p	$\langle A, Q \rangle_{bez.} := \langle A \rangle_{r.estk.}$	+		$\langle A \rangle = \text{pos. Festk.-Zahl}$ Komma links	Im doppelt langen Register $\langle A, Q \rangle$ steht rechtsbündig die Dezimalzahl jeweils mit Typenkennung = 1. Im Reg. D steht mit TK = 1 der Rest	+0	<	2,3	R	$\langle A \rangle > 0$ 9p + 5	95	p: 1...13 Anzahl der gewünschten Dezimalstellen

\* Tabelle in Viertelwörtern  
Die Anfangsadresse ist eine Halbwortadresse

## Tabelle durchsuchen



	Suchbedingung	Eingangsgrößen	$\langle B \rangle$	$\langle A \rangle$	$\langle Q \rangle$	$\langle D \rangle$	Alarm	BÜ	TK						
Tabelle durchsuchen auf Identität	TLI	n	$\langle n+2k \rangle = \langle D \rangle$ ** Tabellenende: TK $\neq t_0$			$\langle D \rangle = \text{Suchwort}$ $\langle D \rangle_1 = \langle D \rangle_2$ bei $\langle D \rangle_1 = 0$ oder 1	$n^*$	$t_0; \langle D \rangle^{***}$				$\neq \langle D \rangle_t$	B R	15p + 10	EC
Tabelle durchsuchen mit Dehnung	TLD	n	$\langle n+k(B) \rangle \geq \langle D \rangle$ ** Tabellenende: TK $\neq t_0$			$\langle D \rangle = \text{Suchwort}$ $\langle D \rangle_1 = \langle D \rangle_2$ bei $\langle D \rangle_1 = 0$ oder 1 $\langle B \rangle = \text{Dehnungswert}$	$n^*$	$t_0; \langle D \rangle^{***}$				$\neq \langle D \rangle_t$	B R	14,5p + 10,5	EB
Tabelle durchsuchen mit Dehnung und Maske	TDM	n	$\langle n+k(B) \rangle_x = \langle D \rangle_x$ für $\langle H \rangle_x = 0$ ** Tabellenende: TK $\neq t_0$			$\langle D \rangle = \text{Suchwort}$ $\langle H \rangle = \text{Maske}$ $\langle B \rangle = \text{Dehnungswert}$	$n^*$	$t_H; \langle H \rangle^{n^*}$				$\neq \langle D \rangle_t$	B R	14,5p + 2,5	EA
Tabelle durchsuchen auf Maximum	TMAX	n	$\langle n+k(B) \rangle_x = \text{Max.}$ für $\langle H \rangle_x = 0$ Tabellenende: TK $\neq t_n$			$\langle H \rangle = \text{Maske}$ $\langle B \rangle = \text{Dehnungswert}$	$n^*$	$t_n^*; \langle H \rangle^{n^*}$	$t_n^*; \neg \langle H \rangle^{n^*}$				B R	17p + 4	EF
Tabelle durchsuchen auf Minimum	TMIN	n	$\langle n+k(B) \rangle_x = \text{Min.}$ für $\langle H \rangle_x = 0$ Tabellenende: TK $\neq t_n$			$\langle H \rangle = \text{Maske}$ $\langle B \rangle = \text{Dehnungswert}$	$n^*$	$t_n^*; \langle H \rangle^{n^*}$	$t_n^*; \neg \langle H \rangle^{n^*}$				B R	17p + 4	EE
Tabelle durchsuchen logarithmisch	TLOG	n	$\langle n+2k \rangle_x \geq \langle D \rangle_x$ Voraussetzung: } für $\langle H \rangle_x = 0$ $\langle n \rangle_x < \langle D \rangle_x$			$\langle A \rangle = \text{Tabellenlänge in Ganzwört.}$ $2 \leq \langle A \rangle \leq 2^{20}$ $\langle D \rangle = \text{Suchwort}$ $\langle H \rangle = \text{Maske}$	$n^+$	$t_n^*; \langle H \rangle^{n^+}$ falls kein Wort gefund. 3; 0	$t_0; \neg \langle H \rangle^{n^+}$			kein Wort gef.	B R	23q + 46	ED

k: 0,1,2,...  
p: Anzahl der untersuchten Worte in der Tabelle  
x: 1...48  
falls Dehnungswert  $\langle B \rangle = \pm 0$   
Wirkung wie Nullbefehl

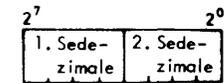
\* Adresse des zuerst gefundenen Wortes, das den Suchbedingungen genügt oder Adresse des ersten Wortes hinter der Tabelle (bei TMAX und TMIN immer Adresse des zuerst gefundenen Wortes)

\*\*\* falls  $\langle D \rangle_t = 0$ :  $\langle D \rangle := \langle D \rangle$  normalisiert, Vergleichsoperand normalisiert (im Speicher unverändert)

\*\* Wird kein Wort gefunden:  
Abbruch mit TK-Alarm

$n^+$  niedrigste Adresse der nach den Suchbedingungen gefundenen Wörter oder erstes Wort hinter der Tabelle

# Zentralcode



0000	000L	00LO	00LL	0LOO	0LOL	0LLO	0LLL	L000	L00L	L0LO	L0LL	LL00	LL0L	LLLO	LLLL		
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
<sup>0</sup> NUL	<sup>16</sup> NL6	<sup>32</sup> SUB	<sup>48</sup>	<sup>64</sup>	<sup>80</sup>	<sup>96</sup> "	<sup>112</sup> %	<sup>128</sup> ^	<sup>144</sup> +	<sup>160</sup> (	<sup>176</sup> 0	<sup>192</sup> A	<sup>208</sup> Q	<sup>224</sup> a	<sup>240</sup> q	0	0000
<sup>1</sup>	<sup>17</sup> NL5	<sup>33</sup> EM	<sup>49</sup>	<sup>65</sup>	<sup>81</sup>	<sup>97</sup> ,	<sup>113</sup> §	<sup>129</sup> v	<sup>145</sup> -	<sup>161</sup> )	<sup>177</sup> 1	<sup>193</sup> B	<sup>209</sup> R	<sup>225</sup> b	<sup>241</sup> r	1	000L
<sup>2</sup>	<sup>18</sup> NL4	<sup>34</sup>	<sup>50</sup>	<sup>66</sup>	<sup>82</sup>	<sup>98</sup> /	<sup>114</sup> #	<sup>130</sup> ¬	<sup>146</sup>	<sup>162</sup> [	<sup>178</sup> 2	<sup>194</sup> C	<sup>210</sup> S	<sup>226</sup> c	<sup>242</sup> s	2	00LO
<sup>3</sup>	<sup>19</sup> NL3	<sup>35</sup> T1	<sup>51</sup>	<sup>67</sup>	<sup>83</sup>	<sup>99</sup> \	<sup>115</sup> \$	<sup>131</sup>	<sup>147</sup> /	<sup>163</sup> ]	<sup>179</sup> 3	<sup>195</sup> D	<sup>211</sup> T	<sup>227</sup> d	<sup>243</sup> t	3	00LL
<sup>4</sup>	<sup>20</sup> NL2	<sup>36</sup> T2	<sup>52</sup>	<sup>68</sup>	<sup>84</sup>	<sup>100</sup>	<sup>116</sup> ¢	<sup>132</sup>	<sup>148</sup>	<sup>164</sup> {	<sup>180</sup> 4	<sup>196</sup> E	<sup>212</sup> U	<sup>228</sup> e	<sup>244</sup> u	4	0LOO
<sup>5</sup>	<sup>21</sup> NL	<sup>37</sup> TE	<sup>53</sup> FL	<sup>69</sup>	<sup>85</sup>	<sup>101</sup> ^	<sup>117</sup>	<sup>133</sup>	<sup>149</sup>	<sup>165</sup> }	<sup>181</sup> 5	<sup>197</sup> F	<sup>213</sup> V	<sup>229</sup> f	<sup>245</sup> v	5	0LOL
<sup>6</sup>	<sup>22</sup> CR	<sup>38</sup>	<sup>54</sup>	<sup>70</sup>	<sup>86</sup>	<sup>102</sup> °	<sup>118</sup> @	<sup>134</sup> ↑	<sup>150</sup>	<sup>166</sup> <	<sup>182</sup> 6	<sup>198</sup> G	<sup>214</sup> W	<sup>230</sup> g	<sup>246</sup> w	6	0LLO
<sup>7</sup>	<sup>23</sup> NF	<sup>39</sup>	<sup>55</sup>	<sup>71</sup>	<sup>87</sup>	<sup>103</sup> ~	<sup>119</sup> &	<sup>135</sup>	<sup>151</sup> =	<sup>167</sup> >	<sup>183</sup> 7	<sup>199</sup> H	<sup>215</sup> X	<sup>231</sup> h	<sup>247</sup> x	7	0LLL
<sup>8</sup>	<sup>24</sup> VT	<sup>40</sup>	<sup>56</sup>	<sup>72</sup>	<sup>88</sup>	<sup>104</sup>	<sup>120</sup> *	<sup>136</sup>	<sup>152</sup> ≠	<sup>168</sup> MZ	<sup>184</sup> 8	<sup>200</sup> I	<sup>216</sup> Y	<sup>232</sup> i	<sup>248</sup> y	8	L000
<sup>9</sup>	<sup>25</sup> VT3	<sup>41</sup>	<sup>57</sup>	<sup>73</sup>	<sup>89</sup>	<sup>105</sup>	<sup>121</sup>	<sup>137</sup>	<sup>153</sup>	<sup>169</sup> .	<sup>185</sup> 9	<sup>201</sup> J	<sup>217</sup> Z	<sup>233</sup> j	<sup>249</sup> z	9	L00L
<sup>10</sup>	<sup>26</sup> VT4	<sup>42</sup>	<sup>58</sup>	<sup>74</sup>	<sup>90</sup>	<sup>106</sup>	<sup>122</sup>	<sup>138</sup>	<sup>154</sup>	<sup>170</sup> ,	<sup>186</sup>	<sup>202</sup> K	<sup>218</sup> Ä	<sup>234</sup> k	<sup>250</sup> ä	A	L0LO
<sup>11</sup>	<sup>27</sup> VT5	<sup>43</sup>	<sup>59</sup>	<sup>75</sup>	<sup>91</sup>	<sup>107</sup>	<sup>123</sup>	<sup>139</sup>	<sup>155</sup> <	<sup>171</sup> :	<sup>187</sup>	<sup>203</sup> L	<sup>219</sup> Ö	<sup>235</sup> l	<sup>251</sup> ö	B	L0LL
<sup>12</sup>	<sup>28</sup> VT6	<sup>44</sup>	<sup>60</sup>	<sup>76</sup>	<sup>92</sup>	<sup>108</sup> ´	<sup>124</sup> ¤	<sup>140</sup>	<sup>156</sup> >	<sup>172</sup> ;	<sup>188</sup>	<sup>204</sup> M	<sup>220</sup> Ü	<sup>236</sup> m	<sup>252</sup> ü	C	LL00
<sup>13</sup>	<sup>29</sup> VT7	<sup>45</sup>	<sup>61</sup>	<sup>77</sup>	<sup>93</sup>	<sup>109</sup> `	<sup>125</sup>	<sup>141</sup> 10	<sup>157</sup> ≡	<sup>173</sup> !	<sup>189</sup>	<sup>205</sup> N	<sup>221</sup>	<sup>237</sup> n	<sup>253</sup> ß	D	LL0L
<sup>14</sup>	<sup>30</sup> VT8	<sup>46</sup>	<sup>62</sup>	<sup>78</sup>	<sup>94</sup>	<sup>110</sup>	<sup>126</sup>	<sup>142</sup>	<sup>158</sup> ≡	<sup>174</sup> ?	<sup>190</sup>	<sup>206</sup> O	<sup>222</sup>	<sup>238</sup> o	<sup>254</sup>	E	LLLO
<sup>15</sup>	<sup>31</sup> NL7	<sup>47</sup>	<sup>63</sup>	<sup>79</sup>	<sup>95</sup>	<sup>111</sup>	<sup>127</sup> π	<sup>143</sup>	<sup>159</sup>	<sup>175</sup> SP	<sup>191</sup> PZ	<sup>207</sup> P	<sup>223</sup>	<sup>239</sup> p	<sup>255</sup>	F	LLLL
Steuerzeichen								Akzente	Sonstige Zeichen	Mathematische Zeichen	Satzzeichen	Ziffern	Großbuchstaben	Kleinbuchstaben			

- |    |                                       |     |                           |     |                                     |     |  |
|----|---------------------------------------|-----|---------------------------|-----|-------------------------------------|-----|--|
| CR | Wagenrücklauf (Carriage Return)       | NF  | Neues Formular (New Form) | PZ  | Plus Null (Plus Zero)               | T2  | Textumschaltung 2                        |
| EM | Ende der Aufzeichnung (End of Medium) | NL  | Neue Zeile (New Line)     | SP  | Zwischenraum (Space)                | TE  | Textende                                 |
| FL | Fluchtsymbol                          | NLx | Vorschub um x Zeilen      | SUB | Substitution (Substitute Character) | VT  | Vertikal-Tabulator (Vertical Tabulation) |
| MZ | Minus Null (Minus Zero)               | NUL | Nil (Null)                | T1  | Textumschaltung 1                   | VTx | Zeilenvorschub gemäß Spur x              |

GR2a

# Potenzen von 2

2 <sup>n</sup>	n	2 <sup>-n</sup>	2 <sup>n</sup>	n	2 <sup>-n</sup>
2	1	0.5	2 251 799 813 685 248	51	0.(15) 444 089 209 850 062 616 169 452 667 236 328 125
4	2	0.25	4 503 599 627 370 496	52	0. 222 044 604 925 031 306 084 726 333 618 164 062 5
8	3	0.125	9 007 199 254 740 992	53	0. 111 022 302 462 515 654 042 363 166 809 082 031 25
16	4	0.062 5	18 014 398 509 481 984	54	0. 055 511 151 231 257 827 021 181 583 404 541 015 625
32	5	0.031 25	36 028 797 018 963 968	55	0. 027 755 575 615 628 913 510 590 791 702 270 507 812 5
64	6	0.015 625	72 057 594 037 927 936	56	0.(15) 013 877 787 807 814 456 755 295 395 851 135 253 906 25
128	7	0.007 812 5	144 115 188 075 855 872	57	0. 006 938 893 903 907 228 377 647 697 925 567 626 953 125
256	8	0.003 906 25	288 230 376 151 711 744	58	0. 003 469 446 951 953 614 188 823 848 962 783 813 476 562 5
512	9	0.001 953 125	576 460 752 303 423 488	59	0. 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
1 024	10	0.000 976 562 5	1 152 921 504 606 846 976	60	0. 000 867 361 737 988 403 547 205 962 240 695 953 369 140 625
2 048	11	0.000 488 281 25	2 305 843 009 213 693 952	61	0.(18) 433 680 868 994 201 773 602 981 120 347 976 684 570 312 5
4 096	12	0. 244 140 625	4 611 686 018 427 387 904	62	0. 216 840 434 497 100 886 801 490 560 173 988 342 285 156 25
8 192	13	0. 122 070 312 5	9 223 372 036 854 775 808	63	0. 108 420 217 248 550 443 400 745 280 086 994 171 142 578 125
16 384	14	0. 061 035 156 25	18 446 744 073 709 551 616	64	0. 054 210 108 624 275 221 700 372 640 043 497 085 571 289 062 5
32 768	15	0. 030 517 578 125	36 893 488 147 419 103 232	65	0. 027 105 054 312 137 610 850 186 320 021 748 542 785 644 531 25
65 536	16	0.000 015 258 789 062 5	73 786 976 294 838 206 464	66	0.(18) 013 552 527 156 068 805 425 093 160 010 874 271 392 822 265 625
131 072	17	0. 007 629 394 531 25	147 573 952 589 676 412 928	67	0. 006 776 263 578 034 402 712 546 580 005 437 135 696 411 132 812 5
262 144	18	0. 003 814 697 265 625	295 147 905 179 352 825 856	68	0. 003 388 131 789 017 201 356 273 290 002 718 567 848 205 566 406 25
524 288	19	0. 001 907 348 632 812 5	590 295 810 358 705 651 712	69	0. 001 694 065 894 508 600 678 136 645 001 359 283 924 102 783 203 125
1 048 576	20	0. 000 953 674 316 406 25	1 180 591 620 717 411 303 424	70	0. 000 847 032 947 254 300 339 068 322 500 679 641 962 051 391 601 562 5
2 097 152	21	0.000 000 476 837 158 203 125	2 361 183 241 434 822 606 848	71	0.(21) 423 516 473 627 150 169 534 161 250 339 820 981 025 695 800 781 25
4 194 304	22	0. 238 418 579 101 562 5	4 722 366 482 869 645 213 696	72	0. 211 758 236 813 575 084 767 080 625 169 910 490 512 847 900 390 625
8 388 608	23	0. 119 209 289 550 781 25	9 444 732 965 739 290 427 392	73	0. 105 879 118 406 787 542 383 540 312 584 955 245 256 423 950 195 312 5
16 777 216	24	0. 059 604 644 775 390 625	18 889 465 931 478 580 854 784	74	0. 052 939 559 203 393 771 191 770 156 292 477 622 628 211 975 087 656 25
33 554 432	25	0. 029 802 322 387 695 312 5	37 778 931 862 957 161 709 568	75	0. 026 469 779 601 696 885 595 885 078 146 238 811 314 105 987 548 828 125
67 108 864	26	0.000 000 014 901 161 193 847 656 25	75 557 863 725 914 323 419 136	76	0.(21) 013 234 889 800 848 442 797 942 539 073 119 405 657 052 993 774 414 062 5
134 217 728	27	0. 007 450 580 596 923 828 125	151 115 727 451 828 646 838 272	77	0. 006 617 444 900 424 221 398 971 269 536 559 702 828 526 496 887 207 031 25
268 435 456	28	0. 003 725 290 298 461 914 062 5	302 231 454 903 657 293 676 544	78	0. 003 308 722 450 212 110 699 485 634 768 279 851 414 263 248 443 603 515 625
536 870 912	29	0. 001 862 645 149 230 957 031 25	604 462 909 807 314 587 353 088	79	0. 001 654 361 225 106 055 349 742 817 384 139 925 707 131 624 221 801 757 812 5
1 073 741 824	30	0. 000 931 322 574 615 478 515 625	1 208 925 819 614 629 174 706 176	80	0. 000 827 180 612 553 027 674 871 408 692 069 962 853 565 812 110 900 878 906 25
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5	2 417 851 639 229 258 349 412 352	81	0.(24) 413 590 306 276 513 837 435 704 346 034 981 426 782 906 055 450 439 453 125
4 294 967 296	32	0. 232 830 643 653 869 628 906 25	4 835 703 278 458 516 698 824 704	82	0. 206 795 153 138 256 918 717 852 173 017 490 713 391 453 027 725 219 726 562 5
8 589 934 592	33	0. 116 415 321 826 934 814 453 125	9 671 406 556 917 033 397 649 408	83	0. 103 397 576 569 128 459 358 926 086 508 745 356 695 726 513 862 609 863 281 25
17 179 869 184	34	0. 058 207 660 913 467 407 226 562 5	19 342 813 113 834 066 795 298 816	84	0. 051 698 788 284 564 229 679 463 043 254 372 678 347 863 256 931 304 931 640 625
34 359 738 368	35	0. 029 103 830 456 733 703 613 281 25	38 685 626 227 668 133 590 597 632	85	0. 025 849 394 142 282 114 839 731 521 627 186 339 173 931 628 465 652 465 820 312 5
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625	77 371 252 455 336 267 181 195 264	86	0.(24) 012 924 697 071 141 057 419 865 760 813 593 169 586 965 814 232 826 232 910 156 25
137 438 953 472	37	0. 007 275 957 614 183 425 903 320 312 5	154 742 504 910 672 534 362 390 528	87	0. 006 462 348 535 570 528 709 932 880 406 796 584 793 482 907 116 413 116 455 078 125
274 877 906 944	38	0. 003 637 978 807 091 712 951 660 156 25	309 485 009 821 345 068 724 781 056	88	0. 003 231 174 267 785 264 354 966 440 203 398 292 396 741 453 558 206 558 227 539 062 5
549 755 813 888	39	0. 001 818 989 403 545 856 475 830 078 125	618 970 019 642 690 137 449 562 112	89	0. 001 615 587 133 892 632 177 483 220 101 699 146 726 779 103 279 113 769 531 25
1 099 511 627 776	40	0. 000 909 494 701 772 928 237 915 039 062 5	1 237 940 039 285 380 274 899 124 224	90	0. 000 807 793 566 946 316 088 741 610 050 849 573 099 185 363 389 551 639 556 884 765 625
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25	2 475 880 078 570 760 549 798 248 448	91	0.(27) 403 896 783 473 158 044 370 805 025 424 786 549 592 681 694 775 819 778 442 382 812 5
4 398 046 511 104	42	0. 227 373 675 443 232 059 478 759 765 625	4 951 760 157 141 521 099 596 496 896	92	0. 201 948 391 736 579 022 185 402 512 712 393 274 796 340 847 387 909 889 221 191 406 25
8 796 093 022 208	43	0. 113 686 837 721 616 029 739 379 882 812 5	9 903 520 314 283 042 199 192 993 792	93	0. 100 974 195 868 289 511 092 701 256 356 196 637 398 170 423 693 954 944 610 595 703 125
17 592 186 044 416	44	0. 056 843 418 860 808 014 869 689 941 406 25	19 807 040 628 566 084 398 385 987 584	94	0. 050 487 097 934 144 755 546 350 628 178 098 318 699 085 211 846 977 472 305 297 851 562 5
35 184 372 088 832	45	0. 028 421 709 430 404 007 434 844 970 703 125	39 614 081 257 132 168 796 771 975 168	95	0. 025 243 548 967 072 377 773 175 314 089 049 159 349 542 605 923 488 736 152 648 925 781 25
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5	79 228 162 514 264 337 593 543 950 336	96	0.(27) 012 621 774 483 536 188 886 587 657 044 524 579 674 771 302 961 744 368 076 324 462 890 625
140 737 488 355 328	47	0. 007 105 427 327 601 001 858 711 242 675 781 25	158 456 325 028 528 675 187 087 900 672	97	0. 006 310 887 241 768 094 443 293 828 522 262 289 837 385 651 480 872 184 038 162 231 445 312 5
281 474 976 710 656	48	0. 003 552 713 678 800 500 929 355 621 337 890 625	316 912 650 057 057 350 374 175 801 344	98	0. 003 155 443 620 884 047 221 646 914 261 131 144 918 692 825 740 436 092 019 081 115 722 656 25
562 949 953 421 312	49	0. 001 776 356 839 400 250 464 677 810 668 945 312 5	633 825 300 114 114 700 748 351 602 688	99	0. 001 577 721 810 442 023 610 823 457 130 565 572 459 346 412 870 218 046 009 540 557 861 328 125
1 125 899 906 842 624	50	0. 000 888 178 419 700 125 232 338 905 334 472 656 25	1 267 650 600 228 229 401 496 703 205 376	100	0. 000 788 860 905 221 011 805 411 728 565 282 786 229 673 206 435 219 023 004 770 278 930 664 062 5

# Konvertierungstafel

sedezimal		dezimal		sedezimal		dezimal		sedezimal		dezimal	
100 000	1 048 576	010 000	65 536	001 000	4 096	000 100	256	000 010	16	000 001	1
200 000	2 097 152	020 000	131 072	002 000	8 192	000 200	512	000 020	32	000 002	2
300 000	3 145 728	030 000	196 608	003 000	12 288	000 300	768	000 030	48	000 003	3
400 000	4 194 304	040 000	262 144	004 000	16 384	000 400	1 024	000 040	64	000 004	4
		050 000	327 680	005 000	20 480	000 500	1 280	000 050	80	000 005	5
		060 000	393 216	006 000	24 576	000 600	1 536	000 060	96	000 006	6
		070 000	458 752	007 000	28 672	000 700	1 792	000 070	112	000 007	7
		080 000	524 288	008 000	32 768	000 800	2 048	000 080	128	000 008	8
		090 000	589 824	009 000	36 864	000 900	2 304	000 090	144	000 009	9
		0A0 000	655 360	00A 000	40 960	000 A00	2 560	000 0A0	160	000 00A	10
		0B0 000	720 896	00B 000	45 056	000 B00	2 816	000 0B0	176	000 00B	11
		0C0 000	786 432	00C 000	49 152	000 C00	3 072	000 0C0	192	000 00C	12
		0D0 000	851 968	00D 000	53 248	000 D00	3 328	000 0D0	208	000 00D	13
		0E0 000	917 504	00E 000	57 344	000 E00	3 584	000 0E0	224	000 00E	14
		0F0 000	983 040	00F 000	61 440	000 F00	3 840	000 0F0	240	000 00F	15

dezimal		sedezimal		dezimal		sedezimal		dezimal		sedezimal	
1 000 000	0F4 240	100 000	018 6A0	10 000	002 710	1 000	000 3E8	100	000 064	10	000 00A
2 000 000	1E8 480	200 000	030 D40	20 000	004 E20	2 000	000 7D0	200	000 0C8	20	000 014
3 000 000	2DC 6C0	300 000	049 3E0	30 000	007 530	3 000	000 BB8	300	000 12C	30	000 01E
4 000 000	3D0 900	400 000	061 A80	40 000	009 C40	4 000	000 FA0	400	000 190	40	000 028
		500 000	07A 120	50 000	00C 350	5 000	001 388	500	000 1F4	50	000 032
		600 000	092 7C0	60 000	00E A60	6 000	001 770	600	000 258	60	000 03C
		700 000	0AA E60	70 000	011 170	7 000	001 B58	700	000 2BC	70	000 046
		800 000	0C3 500	80 000	013 880	8 000	001 F40	800	000 320	80	000 050
		900 000	0DB BA0	90 000	015 F90	9 000	002 328	900	000 384	90	000 05A

# Befehlscode Intern - Extern

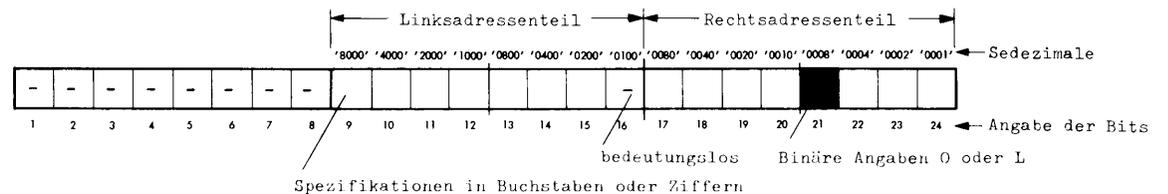


0000	000L	00L0	00LL	0L00	0L0L	0LLO	0LLL	L000	L00L	LOL0	L0LL	LL00	LL0L	LLLO	LLLL		
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
NULL	LZL	MAB	ZMC	AB		DV	B	C	ST	C2	BL		CU	RLR	DA	0	0000
XBA	HBA	SHB	LMC	SBB		DVD	BD	CR	STN	C3	VPU *		BZN	IR	DSB	1	000L
MNA	NL	WTV	LMT	A	GAB	DVI	BQ	CMT	ZTR	CMC	SXR		ZDP *		DML	2	00L0
MA	VBA	WTR	LC	AC	GSBB	VAQ	BH	CMR	SEGG	BC	SXRN		BU	US	MLD	3	00LL
EMU	MC	SXI	SM	SBI	ML	GDV	BB	CN	KDFR	SIO	Y *		SG0		AT	4	0L00
MU	VBC	SXGG	SMN	SBD	MLR	REZ	BN	CB	KFLD	SKG0	LEI *		SK0		SBT	5	0L0L
BCL	MCF	SXKG	S	SB	MLA	GDVI	BR	CD	R	SGG0	BCI			BNZ	BT	6	0LLO
TBC	MCE	SXN	VSS *	SBC	MAR	B2VN	BNR	CQ	RT	SN0	ZI			CNZ	CT	7	0LLL
MFU** M, XB	XC** XCN	EMB	SU	GSBI	MLN	VEL	M2N	VLA	AA	SAT	SR	ZT0	BZ2		WB	8	L000
MD	XBAN	E	TCB	AU	MNR	AUT	M2NR	ATA	SBA	SAA	PDP	ZT1	BZ		SBIT	9	L00L
SZX	ZX	ENZ	SFB	GAC	MAN	ET	M2	ETA	LR	SK	SRN	ZT2	BQB	TDM	SFBE	A	L0L0
MF	SW *	EZ		GA	MANR	ZUS	M2R	LA	SH	SG	SSR	ZT3	CZ	TLD	BSS	B	L0LL
TXX	SLN	HXP	HBC	GSBD	GMLN	B3	A2	TXR	TRX	SI	SE	T	BAR	TLI	ZK	C	LL00
TTX	SNL	MH	MCFU	SBU	GMAN	B3V	SB2	RX** MRX	HALT *	SN	SUE		BANR	TLOG	TOK	D	LL0L
MHX	SL	HXX	ZU	GSBC	GML	B2	AQ	BA		SKG	BLEI	SXG		TMIN	QBR	E	LLLO
HBPX	SLL	VXX	MABI	GSB	GMLA	B2V	SBQ	CH	NRM	SGG	VMO *	SXK	BAN	TMAX	QCR	F	LLLL

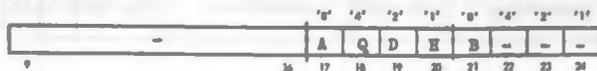
Gr 1

\*nicht für die Programmierung von Operatoren. Der Befehl SSR findet in Abwicklerdiensten Verwendung. \*\*Unterscheidung im Adressenteil (siehe Internspezifikationen)

# Spezifikationen - Intern

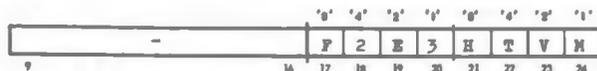


### IR s



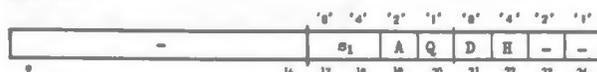
Es sind alle  $2^8$  möglichen Bitanordnungen erlaubt. Falls die Bits 17 - 20 = 0; Wirkung wie Nullbefehl.

### LA s



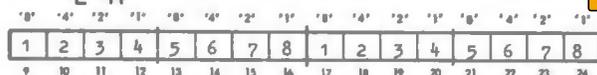
s: F, 2, E, 3, V und M können beliebig kombiniert werden. H oder T dürfen nur einzeln oder mit M verwendet werden. Nicht erlaubte Spezifikationen ergeben undefinierte Befehlsausführungen.

### LR s



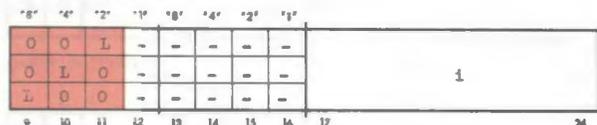
Es sind alle  $2^8$  möglichen Bitanordnungen erlaubt.  
s<sub>1</sub>: Typenkennung 0,1,2 oder 3

### LZL s<sub>L</sub> s<sub>R</sub>



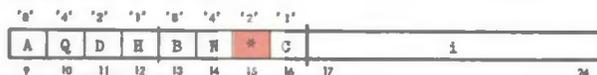
Es sind in jedem Adressteil jeweils alle  $2^8$  möglichen Bitanordnungen erlaubt.

### MFU, XB, M i



MFU: Bits 9 - 11 (Linksadresse) = OOL  
XB : Bits 9 - 11 (Linksadresse) = OLO  
M : Bits 9 - 11 (Linksadresse) = LOO

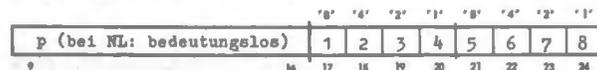
### MRX, RX s i



\*MRX: Bit 15 = L  
RX : Bit 15 = O  
Von den Bits 9 - 13 darf höchstens eines = L sein, sonst undefinierte Befehlsausführung.

### NL s

### SL, SLL, SLN, SNL, SW p s



Es sind alle  $2^8$  möglichen Bitanordnungen erlaubt.

### NRM s

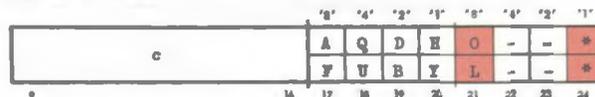


s besteht nur aus den Bits 17 - 19

N	0	0	0
L	0	0	L
F	0	L	0
F4	0	L	L
G	L	0	0
**	L	0	L
FG	L	L	0
**	L	L	L

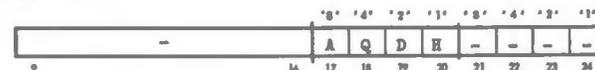
\*\* diese Bitanordnungen führen zu undefinierten Befehlsausführungen.

### R, RLR c s



Genau eines der Bits 17 - 20 muß = L sein, sonst undefinierte Befehlsausführung. (Ausnahme: Falls c = Null, so ist Bit 17 - 20 = 0 erlaubt)  
\* Bit 24 = L rechte 24 Bits } nur bei Halbwortbefehlen  
\* Bit 24 = 0 linke 24 Bits } von Bedeutung

### RT s



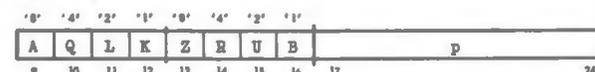
Genau zwei der Bits 17 - 20 müssen = L sein, sonst undefinierte Befehlsausführung.

### SBIT p s



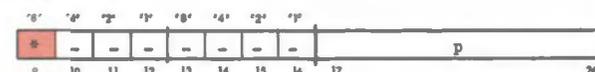
s<sub>2</sub>: Register A = 0 0  
Q = 0 L  
D = L 0  
H = L L  
s<sub>1</sub>: Dualziffern 1-48

### SH s p



Es sind alle  $2^8$  möglichen Bitanordnungen erlaubt.  
Wenn Bit 13 = L, sind die Bits 9 und 10 bedeutungslos.  
Wenn Bit 12 = L, ist Bit 14 bedeutungslos.  
Wenn Bit 11 = L und Bit 13 = 0, ist Bit 14 bedeutungslos.  
Wenn Bit 9,10 und 13 = 0, so sind die Bits 11,12,14 und 15 bedeutungslos.

### SHB s p



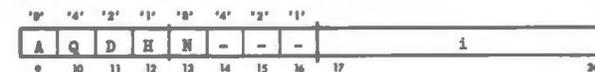
Es sind alle  $2^8$  möglichen Bitanordnungen erlaubt  
\* Bit 9 = L : links  
Bit 9 = 0 : rechts

### ST, STN p s



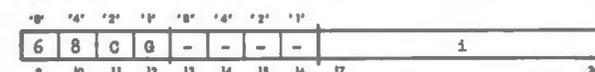
Von den Bits 19 - 22 darf nur eines = L sein, sonst erfolgt eine undefinierte Befehlsausführung. Bit 23 muß = 0 sein, sonst wird in jedem Fall gesprungen.

### TRX, TXR s i



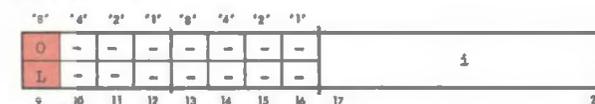
TRX: Von den Bits 9 - 12 darf nur eines = L sein, sonst undefinierte Befehlsausführungen.  
TXR: Es sind alle  $2^8$  möglichen Bitanordnungen erlaubt.

### US s i



Von den Bits 9 - 11 muß genau eines = L sein, sonst undefinierte Befehlsausführung.  
Wenn das Bit 12 = 0, dann Teilspezifikation E.

### XC, XCN i



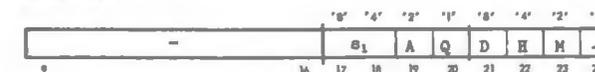
XC : Bit 9 (Linksadresse) = 0  
XCN: Bit 9 (Linksadresse) = L

### ZK s i



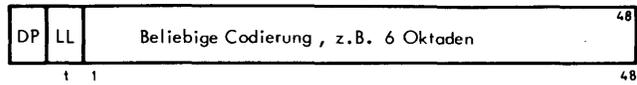
\* Wenn Bit 12 = L(R), muß genau eines der Bits 19-21 = L sein, sonst undefinierte Befehlsausführung  
Wenn Bit 12 = 0(leer), wird in den Bits 17-24 die Adresse einer Indexzelle erwartet

### ZTR s



s<sub>1</sub>: Typenkennung 0,1,2 oder 3  
Von den Bits 19 - 22 darf höchstens eines = L sein, sonst undefinierte Befehlsausführung.





ER 1711

DP = Bits für Prüfzwecke (Dreierprobe)

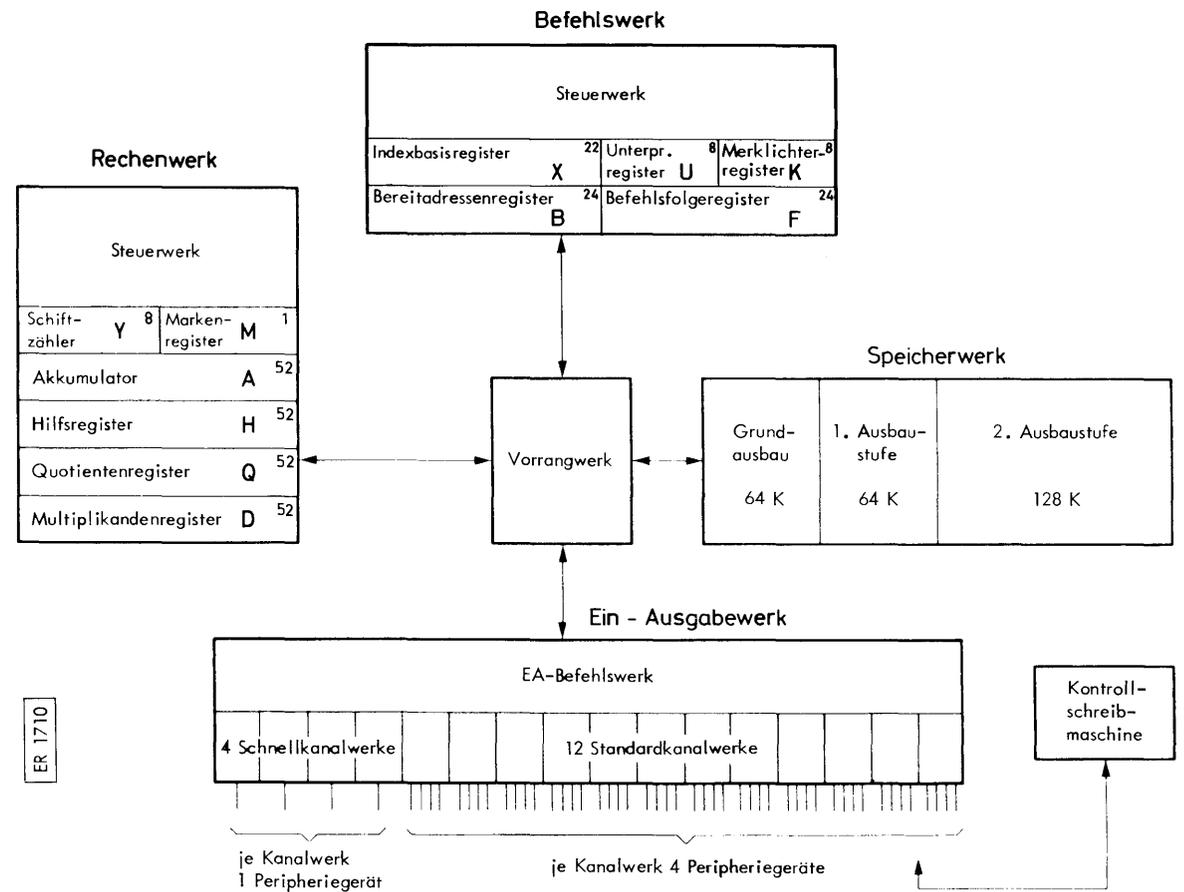
TK = Bits für Typenkennung

t = Typenkennung

m = Marke (nur im Speicher, im Register gleich der v-Stelle)

v = Vorzeichen

## Blockschaltbild



ER 1710

