

Große Befehlsliste

SYSTEM TR 440

H. Schüle

Änderungsstand

6															

In diese Felder sind fortlaufend die Nummern der jeweiligen Änderung einzutragen. Sie können dadurch feststellen, ob Sie von uns alle Änderungen erhalten haben.

Beispiel:

1	2	3	4	5			

Sie haben die fortlaufenden Nummern 1 bis 5 unserer Änderungen eingetragen. Bekommen Sie jetzt die Änderung mit der fortlaufenden Nummer 7, so wissen Sie, daß Ihnen die Änderung Nummer 6 fehlt. Anforderungen bezüglich des Änderungsdienstes richten Sie bitte an die untenstehende Adresse, Abt. TC/VS 13.

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	27

Bedeutung der Spalten

Bezeichnung	Code	adr	Wirkung	mod2	R	veränderliche Spalten	(M)	BU	TK	Werk	lakte	Int.	Bemerkungen
1	2	3	4	5	6	7	8	9	10	11	12	13	14
Erläuterungen des Mnemocodes	EMC	n	Hauptwirkung	+	x		x	E ₂	#1	B R	22	AB	Spezielles

* (15)

**

*** Fußnoten

- 1 Bezeichnung: Hier steht die Befehlsbezeichnung.
Die Buchstaben die zum Mnemocode führen, sind unterstrichen.
- 2 Code: Externcode (mnemotechnische Abkürzung der Befehlsbezeichnung)
- 3 adr: Adressenteil des Befehls
- 4 Wirkung: Hier wird die hauptsächlichste Wirkung des Befehls angegeben. In den veränderlichen Spalten 7 werden auch evtl. auftretende Nebenwirkungen angegeben.
- 5 mod2: Vom vorhergehenden Befehl kann ein Modifikator 2. Art (mod2) vorhanden sein. Es ist hier angegeben wie dieser Modifikator auf den beschriebenen Befehl einwirkt.
+ = mod2 wird während der Abrufphase zum Adressenteil des Befehls addiert. Anschließend wird mod2 gelöscht.
adr := adr + mod2
mod2 := 0
sp = Der Befehl wird während der Ausführungsphase speziell modifiziert.
Die Art der Modifizierung ist bei diesen Befehlen angegeben.
- 6 R: Ein x in dieser Spalte zeigt an, daß der Befehl als Zweitcode beim Registerbefehl R zugelassen ist.
- 7 veränderliche Spalten: Diese Spalten sind je nach Wirkung des Befehls unterschiedlich benannt. Im allgemeinen definieren diese Spalten den Inhalt der angegebenen Register nach Ausführung des Befehls. Bei Spalten ohne Eintragung und Registern, die nicht aufgeführt sind, erfolgt keine Änderung des betreffenden Inhaltes.
Folgende Schreibweisen werden verwendet:
Bei allen Registern mit Typenkennung und bei den Speicherzellen wird vor einem Semikolon die Typenkennung angegeben (t₁; = Typenkennung der Speicherzelle n).
Steht nach dem Semikolon und vor einem Komma 0 bzw. v, so wird damit ausgesagt, daß das Register links mit Null oder vorzeichengleich aufgefüllt wird (t₁;v;(m) = In dem Register, das in dieser Spalte angegeben ist, steht der Inhalt der Speicherzelle m, links mit Vorzeichen aufgefüllt; beim Transport wird die TK aus der Speicherzelle m mit in dieses Register gebracht).
Ist eine Klammer — in zwei Registerspalten eingetragen, so steht in beiden Registern der gleiche Wert.
- 8 (M): Für das Markenregister ist durch x angegeben, daß bei diesem Befehl das Markenbit berücksichtigt wird. Als Wirkung gehört dazu, daß beim Transport eines Zahlwortes in ein Register das erste Bit dem zweiten angeglichen wird. Ist das Zahlwort markiert, so wird außerdem das Markenregister M gesetzt.
(M) := (M)∨(n)
a = beliebiges Register
(a)₁ := (a)₂
- 9 BU-Alarm: Hier wird angegeben, unter welchen Bedingungen ein Bereichsüberschreitungs-Alarm gegeben wird (nur bei TK 0 und 1).
> = übergelaufen ≥ = über- oder untergelaufen
- 10 TK-Alarm: Diese Spalte gibt an, bei welcher Bedingung ein Typenkennungs-Alarm auftritt.
- 11 Werk: Es wird angegeben, ob der Befehl das Befehlswerk (B), das Rechenwerk (R) oder beide belegt.
- 12 Takte: Hier wird die Ausführungszeit in Takten angegeben. Es handelt sich zum Teil um Mittelwerte. Sind unter "Werk" Befehls- und Rechenwerk aufgeführt, so gilt die Zeit für beide Werke. Für das Rechenwerk kann in vielen Fällen eine kürzere Zeit benötigt werden, die aber dann ohne Vorteil ist. Ist unter Werk nur das Rechenwerk aufgeführt, dann können während der angegebenen Zeit parallel dazu alle Befehle ablaufen, die nur das Befehlswerk ansprechen.
- 13 Int.: Diese Spalte nennt den Internode des Befehls in zwei Sedezimalen.
- 14 Bemerkungen: Hier sind spezielle Vermerke und Erläuterungen zum Befehl aufgeführt.
- 15 Fußnoten: Sternchen in den Spalten werden in den Fußnoten erläutert.

Befehlsabruf	8	Takte (Mittelwert)
belegt Rechenwerk	1	
Modifizierung 1. Art	8	
Modifizierung 2. Art (wenn bei "mod2: +")	8	
bei Operand aus dem Speicher	8	
bei erfüllter Sprungbed.	5	
Für genauere Zeitberechnungen s. TR 440 Befehls-Lexikon.		

Zu den angegebenen Zeiten kommen für einfache Zeitberechnungen nachfolgend diese Zeiten hinzu, die in der Abrufphase liegen und das Befehlswerk belegen:

Erklärung der Zeichen

Bezeichnung der Register im Rechenwerk:

A	Akkumulator	} 48 Bits Information 2 Bits Typenkenennung
Q	Quotientenregister	
D	Multiplikandenregister	
H	Hilfsregister	
Y	Schifftzähler	8 Bits
M	Markenregister	1 Bit
A, Q	doppeltlanges Register A und Q	
H, Q	doppeltlanges Register H und Q	

Bezeichnung der Register im Befehlswerk:

B	Bereitadressenregister	} 24 Bits
F	Befehlsfolgerregister	
X	Indexbasenregister	22 Bits
K	Merklichterregister	} 8 Bits
U	Unterprogrammregister	

Variable:

mod1	Modifikator erster Art	} 24-Bit-Größe im Register B oder in einem nicht adressierbaren Register
mod2	Modifikator zweiter Art	
op	Operationscode eines Befehls	8 Bits
adr	Inhalt des Adressenteils eines Befehls	(16 Bits auf 24 Bits erweitert)
n	Speicheradresse eines Ganzwortes	} 16 Bits
	nur geradzählige Adressen, ungeradzählige werden um 1 vermindert.	
	Speicheradresse eines Halbwortes	
m	Zahl oder Operand	
z	Indexadresse	} Indizes: L für links Parameter R für rechts
i	Parameter	
p	Spezifikation (s evtl. unterteilt in s ₁ , s ₂ , ...)	
s	Zweitcode (Code für den Zweitbefehl)	

Zeichen und ihre Bedeutung:

:=	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.
::	Links- und rechtsstehende Größen werden miteinander vertauscht. Beispiel: $\langle A \rangle :: \langle H \rangle$ Die Inhalte von A und H werden vertauscht.

t _x	Typenkenennung im Register x oder in der Speicherzelle x
1, 1	Typenkenennung in beiden Registern eines doppeltlangen Registers

0,	der linke Teil des Registers ist mit Null aufgefüllt
----	--

v,	der linke Teil des Registers ist vorzeichen- gleich aufgefüllt
----	--

< >	Inhalt eines Registers, einer Speicher- oder Indexzelle Beispiel: $\langle A \rangle$ Inhalt des Registers A (einschließlich Typenkenennung)
-----	--

< >, < >	Inhalt von zwei getrennten Registern Beispiel: $\langle A \rangle, \langle Q \rangle$ Inhalt von A und Q
----------	--

< , >	Inhalt zweier Register, die zusammengefaßt sind Beispiel: $\langle A, Q \rangle$ Inhalt der zu einem doppelt langen Register vereinigten Register A und Q
-------	---

	Betrag einer Größe, Beispiel: $ \langle n \rangle $ Betrag vom Inhalt der Speicherzelle n
--	---

Indizes für Teile eines Wortes:

< >*	Vorzeichen vom Inhalt
< >†	Typenkenennung vom Inhalt
< >*	Marke vom Inhalt
< A > ₁	Bit 1 im Register A
< A > ₄₁₋₄₈	Bits 41 bis 48 im Register A
< n > ₉₋₂₄	Bits 9 bis 24 in der Speicherzelle (Drittelwort)
< n > _{1,2}	Bits 1 und 2 in der Speicherzelle

zur Zählung der Bits siehe Seite Wortstruktur

Logische Verknüpfungen:

UND-Verknüpfung (Konjunktion)

a := b ^ c			
0	0	0	0
0	0	L	L
0	L	0	L
L	L	L	L

ODER-Verknüpfung (Disjunktion)

a := b v c			
0	0	0	0
L	0	L	0
L	L	L	L
L			

Antivalenz-Verknüpfung (exklusives ODER)

a := b # c	
0	0
L	0
L	L
0	L

Negation

a := ¬ b	
0	L
L	0

Transportbefehle

Bezeichnung	Code	adr	Wirkung	mod2 R	(A)	(Q)	(H)	(D)	(Y)	(M)	Werk	Int.	Bemerkungen
Bringe (nach A)	B	n	$\langle A \rangle := \langle n \rangle$	+	$t_n; \langle n \rangle$					x	R 2	70	
Bringe unverändert	BU	n	$\langle A \rangle := \langle n \rangle$	+	$t_n; \langle n \rangle$						R 2	D3	unverändert bei jeder TK
Bringe und Reserviere	BR	n	$\langle H \rangle := \langle A \rangle$ $\langle A \rangle := \langle n \rangle$	+	$t_n; \langle n \rangle$		$t_n; \langle A \rangle$			x	R 5	76	
Bringe negativ	BN	n	$\langle A \rangle := -\langle n \rangle$	+	$t_n; -\langle n \rangle$					x	R 5	75	
Bringe negativ und Reserviere	BNR	n	$\langle H \rangle := \langle A \rangle$ $\langle A \rangle := -\langle n \rangle$	+	$t_n; -\langle n \rangle$		$t_n; \langle A \rangle$			x	R 5	77	
Bringe Betrag	BB	n	$\langle A \rangle := \langle n \rangle $	+	$t_n; \langle n \rangle $					x	R 5	74	
Bringe nach Q	BQ	n	$\langle Q \rangle := \langle n \rangle$	+	x	$t_n; \langle n \rangle$				x	R 3	72	
Bringe nach D	BD	n	$\langle D \rangle := \langle n \rangle$	+	x			$t_n; \langle n \rangle$		x	R 3	71	
Bringe nach H	BH	n	$\langle H \rangle := \langle n \rangle$	+	x		$t_n; \langle n \rangle$			x	R 3	73	
Bringe nach Q und bringe (nach A)	BQB	n	$\langle A \rangle := \langle n \rangle$ $\langle Q \rangle := \langle n \rangle$	+	$t_n; \langle n \rangle$	$t_n; \langle n \rangle$				x	R 2	DA	
Bringe zwei Wörter	BZ	n	$\langle A \rangle := \langle n \rangle$ $\langle Q \rangle := \langle n+2 \rangle$	+	$t_n; \langle n \rangle$	$t_{n+2}; \langle n+2 \rangle$				x	B R 15	D9	
Bringe zwei Wörter negativ	BZN	n	$\langle A \rangle := -\langle n \rangle$ $\langle Q \rangle := -\langle n+2 \rangle$	+	$t_n; -\langle n \rangle$	$t_{n+2}; -\langle n+2 \rangle$				x	B R 15	D1	
Bringe Teilwort	BT	n	$\langle A \rangle_x := \langle n \rangle_x$ falls $\langle Q \rangle_x = 0$ $\langle A \rangle_x := 0$ falls $\langle Q \rangle_x = 1$	+	$t_n; \dots$ (S. Wirk.) x $\langle A \rangle$ wird um p Stellen rechts geschiftet, 0 nachgezogen						R 7 +p	F6	p: Anzahl der 1, die rechtsbündig in Q stehen. x: 1,2,...,48
Bringe nächstes Zeichen	BNZ	il i_R	$\langle A \rangle := 0$, Zeichen gemäß a und b rechtsb. $\langle A \rangle_x := \langle a \rangle_x$ b := b + 1 wenn b < d b := +0 wenn b = d $\langle i_R \rangle := \langle i_R \rangle + 2$	sp	$t_n; 0, \dots$ (S. Wirk.)	undefiniert $\langle i_L \rangle_{alt}$					B R 41	E6	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 von $\langle i_L \rangle$ bedeutungslos
Bringe zwei Halbwörter	BZ2	m	$\langle A \rangle := \langle m, m+1 \rangle$	+	$t_m; \langle m, m+1 \rangle$						B R 10	D8	
Bringe Halbwort	B2	m	$\langle A \rangle := \langle m \rangle$	+	$t_m; 0, \langle m \rangle$						R 9	6E	
Bringe Halbwort mit Vorzeichen	B2V	m	$\langle A \rangle := \langle m \rangle$	+	1; v, $\langle m \rangle$						R 10	6F	
Bringe Halbwort mit Vorzeichen negativ	B2VN	m	$\langle A \rangle := -\langle m \rangle$	+	1; -v, $\langle m \rangle$						R 12	67	
Bringe Drittelwort	B3	m	$\langle A \rangle := \langle m \rangle_{9-24}$	+	1; 0, $\langle m \rangle_{9-24}$						R 9	6C	
Bringe Drittelwort mit Vorzeichen	B3V	m	$\langle A \rangle := \langle m \rangle_{9-24}$	+	1; v, $\langle m \rangle_{9-24}$						R 10	6D	
Bringe aus Leitblock	BLEI	p	$\langle A \rangle := \langle \langle BL \rangle \cdot 2^p + p \rangle$	+	$t; \langle \langle BL \rangle \cdot 2^p + p \rangle$						B R 13	EE	p: +0...255 BL = Leitadressenregister
Bringe (und setze) Steuerbits (und Sperren)	BSS	s_L s_R	$\langle A \rangle :=$ Steuerbits falls $a_0=L$		2; Steuerbits falls $a_0=L$						B R 3	FB	s_L : Bits a_9 bis a_6 s_R : Bits a_1 bis a_4 s_R ist ohne Bedeutung, muß aber angegeben werden.

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

Bringe und speichere Indexbasis		BCI	n	$\langle X \rangle := \langle n \rangle_{3-24}$ $\langle U \rangle := \langle n \rangle_{4,1-48}$			$3; 0, \langle X \rangle, 0, \langle U \rangle$			$\langle X \rangle$	$\langle U \rangle$		B	17x +66	B6	x = Anzahl der zurückge- speicherten Indexreg.
Speichere		C	n	$\langle n \rangle := \langle A \rangle$	+	0	$t_n; \langle A \rangle$					$\geq$	B R 8	80		
Speichere unverändert		CU	n	$\langle n \rangle := \langle A \rangle$	+	$\langle A \rangle_1$	$t_n; \langle A \rangle$						B R 8	DO		unverändert bei jeder TK
Speichere und bringe Reserve		CR	n	$\langle n \rangle := \langle A \rangle$ $\langle A \rangle := \langle H \rangle$	+	0	$t_n; \langle A \rangle$	$t_n; \langle H \rangle$			$\geq$	B R 9	81			
Speichere negativ		CN	n	$\langle n \rangle := -\langle A \rangle$	+	0	$t_n; -\langle A \rangle$				$\geq$	B R 9	84			
Speichere Betrag		CB	n	$\langle n \rangle := \langle A \rangle $	+	0	$t_n; \langle A \rangle $				$\geq$	B R 9	85			
Speichere markiert		CMT	n	$\langle n \rangle := \langle A \rangle$	+	L	$t_n; \langle A \rangle$				$\geq$	2,3 B R 8	82			
Speichere mit Marke aus Register		CMR	n	$\langle n \rangle := \langle A \rangle$	+	$\langle M \rangle$	$t_n; \langle A \rangle$				$\geq$	2,3 B R 8	83			
Speichere mit Marke aus Speicher		CMC	n	$\langle n \rangle := \langle A \rangle$	+	$\langle n \rangle_m$	$t_n; \langle A \rangle$				$\geq$	2,3 B R 13	A2			
Speichere aus Q		CQ	n	$\langle n \rangle := \langle Q \rangle$	+	0	$t_n; \langle Q \rangle$				$\geq$	B R 9	87			
Speichere aus D		CD	n	$\langle n \rangle := \langle D \rangle$	+	0	$t_n; \langle D \rangle$				$\geq$	B R 9	86			
Speichere aus H		CH	n	$\langle n \rangle := \langle H \rangle$	+	0	$t_n; \langle H \rangle$				$\geq$	B R 9	8F			
Speichere zwei Wörter		CZ	n	$\langle n \rangle := \langle A \rangle$ $\langle n+2 \rangle := \langle Q \rangle$	+	0	$t_n; \langle A \rangle$	$t_n; \langle Q \rangle$			$\geq$	B R 16	DB			

Speichere Teilwort		CT	n	$\langle n \rangle_x := \langle A \rangle_x$	+		$\langle A \rangle$ um p Stellen links im Kreis geschifft $\langle n \rangle_x := \langle A \rangle_x$ 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 zurückgeschifft	$\langle D \rangle$				B R 22	F7	+2p	p: Anzahl der L, die rechts- bündig im Reg. Q stehen x: 1,2,...,48
--------------------	--	----	---	--	---	--	---	---------------------	--	--	--	--------	----	-----	--

Speichere nächstes Zeichen		CNZ	il i _R	Voraussetzung: $\langle A \rangle = 0, f$ Bits rechte f Bits von $\langle A \rangle$ in a gemäß b eingesetzt $\langle A \rangle := \langle a \rangle$ $\langle A \rangle_x := \langle a \rangle_x$ b := b + 1 wenn b < d b := +0 $\langle i_R \rangle := \langle i_R \rangle + 2$ wenn b = d	sp		$\langle A \rangle$	$\langle n \rangle$	$\langle B \rangle$	$\langle Y \rangle$	$\langle Q \rangle$	$\langle D \rangle$				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
							$t_n; \langle a \rangle$		$\langle i_L \rangle_{ait}$	undef. geschif. tet; andere Bits=0	abzusp. Zeichen aus A, gem. b geschif. undef.		B R 55	E7		

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

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

Transportbefehle (Fortsetzung)

Bezeichnung	Code	adr	Wirkung	mod2 R	(B)	(i)	(i _r)	(s ₁)	(K)	Werk	Takte	Int.	Bemerkungen
Registertausch	RT	s	(s ₁) := (s ₂)	+	(s ₁) und (s ₂) werden vertauscht einschl. TK					R	4	97	s ₁ , s ₂ : A, Q, D oder H; s ₁ ≠ s ₂
Index: <u>Bringe</u>	XB	i	(B) := (i)		(i)					B	1	08	Bits 9 - 11 = OLO
Index: <u>Speichere</u>	XC	i	(i) := (B)			(B)				B	3	18	Bit 9 = 0
Index: <u>Speichere negativ</u>	XCN	i	(i) := -(B)			-(B)				B	5		Bit 9 = L
<u>Tausch-Transport</u> in Indexzellen	TTX	i _r i _r	(B) := (i _r) (i _r) := (i _r)		(i _r)		(i _r)			B	11	0D	
<u>Transport</u> aus Indexzelle nach Indexzelle	TXX	i _r i _r	(i _r) := (i _r) (B) := (i _r)		(i _r)		(i _r)			B	4	0C	
<u>Transport</u> aus Indexzelle nach <u>Rechenwerk</u>	TXR	s i	(s ₁) := (i) (B) := (i)		(i)			1; v, x(i)		B R	5	8C	s ₁ : A, Q, D und H s ₂ : leer: + statt ± N: -
<u>Transport</u> aus <u>Rechenwerk</u> nach Indexzelle	TRX	s i	(i) := (s ₁) (B) := (s ₁)							B R	6	9C	s ₁ : A, Q, D, H oder leer s ₂ : leer: + statt ± N: -
<u>Transport</u> aus <u>Speicher</u> nach <u>B</u>	TCB	m	(B) := (m)	+	(m)					B	2	39	
<u>Transport</u> aus <u>B</u> nach <u>Speicher</u>	TBC	m	(m) := (B)						(B)	B	7	07	
<u>Bringe</u> und <u>speichere</u> <u>Merklichter</u>	BCL	m	(m) ₁₇₋₂₄ := (K)						(K)	B	18	06	(m) ₁₋₁₆ bleiben erhalten

Voraussetzung

<u>Wortgruppen</u> <u>transport</u> <u>vorwärts</u>	WTV	i _r i _r	(i _r) := (i _r) + 2k (B) := (i _r) + 2((B) - 1)							B	21a + 10	22	k: 0, 1, 2, 3, ..., (B-1) a: Anzahl der Ganzwörter
<u>Wortgruppen</u> <u>transport</u> <u>rückwärts</u>	WTR	i _r i _r	(i _r) := (i _r) - 2k (B) := (i _r) - 2((B) - 1)							B	21a + 10	23	

Wirkung

Bequemes <u>Bringen</u> aller <u>Register</u>	QBR	n	Die Register werden unverändert gebracht							B R	84	FE	T: Prüfregister (nicht zugreifbares Register)
Bequemes <u>Speichern</u> aller <u>Register</u>	QCR	n	Die Register werden unverändert abgespeichert							B R	55	FF	T: Prüfregister (nicht zugreifbares Register)

Spezifikationen

<u>Zeichenketten</u> <u>verarbeitung</u>	Version <u>Bringen</u> Version <u>Speichern</u>	2K	s i	(A) bzw. (A, Q) := p Oktaden Gem. a; ggf. verkürzen zu Tetraden (a) := (A) bzw. (A, Q) p Okt. Ggf. verlängern aus Tetraden								min. 122	FC	a = (i) wenn s ₄ = leer a = (D) ₂₅₋₄₈ wenn i = D a = (H) ₂₅₋₄₈ wenn i = H a = (B) wenn i = B a: Oktadenadresse
--	--	----	-----	---	--	--	--	--	--	--	--	----------	----	---

Transportiere Oktaden	z	min.	191	z: 1...65535
wenn $\langle H \rangle_4 = 3$	z	min.	209	a: $\langle H \rangle_4 = 4 = \text{Okt.adr.Zielgebiet}$
wenn $\langle H \rangle_4 = 2$	z	min.	209	q: $\langle H \rangle_4 = 4 = \text{Okt.adr.Quellgebiet}$
	z	min.	209	x: 0, 1, 2, ..., z-1
	z	min.	209	y: 0, 1, 2, 3, 4, 5, 0, 1, 2, ...

Festkomma-Arithmetik

Bezeichnung	Code	adr	Wirkung	mod2 R	(A)	(Q)	(D)	(H)	(Q)	(B)	(M)	Alarm	Int.	Bemerkungen
Addiere	A	n	$\langle A \rangle := \langle A \rangle + \langle n \rangle$	+	$t_{aax}; \langle A \rangle + \langle n \rangle$		$t_a; \langle n \rangle$			x		R	8	42
Addiere Betrag	AB	n	$\langle A \rangle := \langle A \rangle + \langle n \rangle $	+	$t_{aax}; \langle A \rangle + \langle n \rangle $		$t_a; \langle n \rangle$			x		R	8	40
Addiere im Speicher	AC	n	$\langle n \rangle := \langle n \rangle + \langle A \rangle$	+			$t_{aax}; \langle n \rangle + \langle A \rangle$			x		R	17	43
Subtrahiere	SB	n	$\langle A \rangle := \langle A \rangle - \langle n \rangle$	+	$t_{aax}; \langle A \rangle - \langle n \rangle$		$t_a; \langle n \rangle$			x		R	8	46
Subtrahiere Betrag	SB	n	$\langle A \rangle := \langle A \rangle - \langle n \rangle $	+	$t_{aax}; \langle A \rangle - \langle n \rangle $		$t_a; \langle n \rangle$			x		R	8	41
Subtrahiere im Speicher	SBC	n	$\langle n \rangle := \langle n \rangle - \langle A \rangle$	+			$t_{aax}; \langle n \rangle - \langle A \rangle$			x		R	17	47
Subtrahiere invers	SBI	n	$\langle A \rangle := \langle n \rangle - \langle A \rangle$	+	$t_{aax}; \langle n \rangle - \langle A \rangle$		$t_a; \langle n \rangle$			x		R	8	44
Subtrahiere von D	SBD	n	$\langle A \rangle := \langle D \rangle - \langle n \rangle$	+	$t_{aax}; \langle D \rangle - \langle n \rangle$		$t_a; \langle n \rangle$			x		R	11	45
Multipliziere mit Rundung	MLR	n	$\langle A \rangle := \langle A \rangle \cdot \langle n \rangle$	+	$1; (\langle A \rangle \cdot \langle n \rangle) *$		$t_a; \langle n \rangle$			x		R	57	55
Multipliziere negativ mit Rundung	MNR	n	$\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle$	+	$1; (-\langle A \rangle \cdot \langle n \rangle) *$		$t_a; \langle n \rangle$			x		R	57	59
Multipliziere akkumulierend mit Rundung	MAR	n	$\langle A \rangle := \langle A \rangle \cdot \langle n \rangle + \langle H \rangle$	+	$1; (\langle A \rangle \cdot \langle n \rangle) * + \langle H \rangle$		$t_a; \langle n \rangle$			x		R	67	57
Multipliziere akkumulierend negativ mit Rundung	MANR	n	$\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle + \langle H \rangle$	+	$1; (-\langle A \rangle \cdot \langle n \rangle) * + \langle H \rangle$		$t_a; \langle n \rangle$			x		R	67	58
Dividiere	DV	n	$\langle A \rangle := \langle A \rangle : \langle n \rangle$	+	$1; \langle A \rangle : \langle n \rangle$		$t_a; \langle n \rangle$			x		R	220	60
Dividiere invers	DVI	n	$\langle A \rangle := \langle n \rangle : \langle A \rangle$	+	$1; \langle n \rangle : \langle A \rangle$		$t_a; \langle n \rangle$			x		R	222	62
Dividiere doppelt lang	DVD	n	$\langle A \rangle := \langle A, Q \rangle : \langle n \rangle$	+	$1; \langle A, Q \rangle : \langle n \rangle$		$t_a; \langle n \rangle$			x		R	228	61

Addiere in AQ	AQ	n	$\langle A, Q \rangle := \langle A, Q \rangle + \langle n \rangle$	+	$1, 1; \langle A, Q \rangle + \langle n \rangle$		$t_a; \langle n \rangle$			x		R	17	7E
Subtrahiere in AQ	SBQ	n	$\langle A, Q \rangle := \langle A, Q \rangle - \langle n \rangle$	+	$1, 1; \langle A, Q \rangle - \langle n \rangle$		$t_a; \langle n \rangle$			x		R	17	7F
Multipliziere	ML	n	$\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle$	+	$1, 1; \langle A \rangle \cdot \langle n \rangle$		$t_a; \langle n \rangle$			x		R	55	54
Multipliziere negativ	MLN	n	$\langle A, Q \rangle := -\langle A \rangle \cdot \langle n \rangle$	+	$1, 1; -\langle A \rangle \cdot \langle n \rangle$		$t_a; \langle n \rangle$			x		R	55	58
Multipliziere akkumulierend	MLA	n	$\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$	+	$1, 1; \langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$		$t_a; \langle n \rangle$			x		R	66	56
Multipliziere akkumulierend negativ	MAN	n	$\langle A, Q \rangle := -\langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$	+	$1, 1; -\langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$		$t_a; \langle n \rangle$			x		R	66	5A

Addiere Adressenteil	AA	z	$\langle A \rangle := \langle A \rangle + z$	+			$1; Q, z$					R	13	98
Subtrahiere Adressenteil	SBA	z	$\langle A \rangle := \langle A \rangle - z$	+			$1; V, z$					R	13	99

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

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

$\langle A \rangle$															
Addiere <u>Adressenteil</u> (TK = 0)	AA	z	$\langle A \rangle := \langle A \rangle \cdot 16^z$	+	$0; \langle A \rangle \cdot 16^{tz}$		$1; 0; z$				>	R	13	98	nur bei $\langle A \rangle_t = 0$; bei TK $\neq 0$ siehe Festkomma-Arithmetik.
Subtrahiere <u>Adressenteil</u> (TK = 0)	SBA	z	$\langle A \rangle := \langle A \rangle \cdot 16^{-z}$	+	$0; \langle A \rangle \cdot 16^{-z}$		$1; 0; z$				>	R	13	99	$ z < 2^7$ nach Modifizierung

* normalisiert und gerundet
 ** Anzahl der Binärstellen um die das Ergebnis normalisiert wurde.
 Falls Ergebnis = ± 0 oder Exponentenunterlauf: $\langle Y \rangle := +0$

Boolesche Operationen

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

Halbwort-Arithmetik

			$\langle A \rangle$	$\langle Q \rangle$	$\langle D \rangle$	$\langle Y \rangle$
<u>Addiere Halbwort</u>	A2	m	$\langle A \rangle := \langle A \rangle_{2s-4s} + \langle m \rangle$		$t_A; +0$	$t_A; +0$
<u>Subtrahiere Halbwort</u>	SB2	m	$\langle A \rangle := \langle A \rangle_{2s-4s} - \langle m \rangle$		$t_A; +0$	$t_A; +0$
<u>Multipliziere Halbwort</u>	M2	m	$\langle A \rangle := \langle A \rangle \cdot \langle m \rangle$	$t_A; +0$	$t_A; +0$	$t_A; +0$
<u>Multipliziere Halbwort negativ</u>	M2N	m	$\langle A \rangle := -\langle A \rangle \cdot \langle m \rangle$	$t_A; +0$	$t_A; +0$	$t_A; +0$
<u>Multipliziere Halbwort mit Rundung</u>	M2R	m	$\langle A \rangle := \langle A \rangle \cdot \langle m \rangle$ gerundet	$t_A; +0$	$t_A; +0$	$t_A; +0$
<u>Multipliziere Halbwort negativ mit Rundung</u>	M2NR	m	$\langle A \rangle := -\langle A \rangle \cdot \langle m \rangle$ gerundet	$t_A; +0$	$t_A; +0$	$t_A; +0$

Teilwort-Arithmetik

			$\langle A \rangle$	$\langle D \rangle$	$\langle Y \rangle$	$\langle D \rangle$
<u>Addiere Teilwort</u>	AT	n	$\langle A \rangle := \langle A \rangle_x + \langle n \rangle_x$	Hilfsgröße qr := um p Stellen rechts im Kreis geschifteter $\langle Q \rangle$	$t_q; \langle Q \rangle$	P = Anzahl der rechts in Q anstehenden L-Bits Falls $\langle Q \rangle = 11...1$ dann p = 0
<u>Subtrahiere Teilwort</u>	SBT	n	$\langle A \rangle := \langle A \rangle_x - \langle n \rangle_x$	Hilfsgröße nr := um p Stellen rechts geschifteter $\langle n \rangle$	$t_q; \langle Q \rangle$	Die Zahlen werden als positive, vorzeichenlose, ganze Festkommazahlen aufgefaßt.
				Das Nullfeld von qr schneidet in nr und $\langle A \rangle$ Teilwörter aus; diese werden addiert bzw. subtrahiert. Die anderen Stellen werden auf 0 gelöst.		
				$t_A; \text{Ergebnis}$		

Index-Arithmetik

Bezeichnung

Code adr Wirkung

mod2 R

<i>

<i_R>

Werk Takte Int. Bemerkungen

Erhöhe <u>B</u> um Speicher	HBC	m	$\langle B \rangle := \langle B \rangle + \langle m \rangle$	x	$\langle B \rangle + \langle m \rangle$				B	6,5	3C	
Vermindere <u>B</u> um Speicher	VBC	m	$\langle B \rangle := \langle B \rangle - \langle m \rangle$	x	$\langle B \rangle - \langle m \rangle$				B	8,5	15	
Erhöhe <u>B</u> um Adressenteil	HBA	z	$\langle B \rangle := \langle B \rangle + z$		$\langle B \rangle + z$				B	7,5	11	
Vermindere <u>B</u> um Adressenteil	VBA	z	$\langle B \rangle := \langle B \rangle - z$		$\langle B \rangle - z$				B	8,5	13	z: 0...65 535
Erhöhe <u>B</u> um Parameter mal Indexzelle	HBPX	p i	$\langle B \rangle := \langle B \rangle + p \cdot \langle i \rangle$		$\langle B \rangle + p \cdot \langle i \rangle$				B	5,5 p +3	OF	p: ±1...±15
Erhöhe Indexzelle um Parameter	HXP	p i	$\langle i \rangle := \langle i \rangle + p$		$\langle i \rangle + p$				B	10,5	2C	p: ±0...±127
Erhöhe Indexzelle um Indexzelle	HXX	i _L i _R	$\langle B \rangle := \langle i_R \rangle + \langle i_L \rangle$ $\langle i_R \rangle := \langle i_R \rangle + \langle i_L \rangle$		$\langle i_R \rangle + \langle i_L \rangle$			$\langle i_R \rangle + \langle i_L \rangle$	B	14,5	2E	
Vermindere Indexzelle um Indexzelle	VXX	i _L i _R	$\langle B \rangle := \langle i_R \rangle - \langle i_L \rangle$ $\langle i_R \rangle := \langle i_R \rangle - \langle i_L \rangle$		$\langle i_R \rangle - \langle i_L \rangle$			$\langle i_R \rangle - \langle i_L \rangle$	B	14,5	2F	
Register und Indexzelle	RX	s i	$\langle B \rangle := \langle i \rangle \pm \langle s_1 \rangle$ Falls $s_3 = C$: $\langle i \rangle := \langle i \rangle \pm \langle s_1 \rangle$		$\langle i \rangle \pm \langle s_1 \rangle$ 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 + } statt ± N = negativ - } s ₃ : leer = nicht zurückspeichern C = zurückspeichern

*siehe Internspezifikation auf Seite 23

Setzen und Löschen

Bringe Adressenteil	BA	z	$\langle A \rangle := z$	+		1; v, z				R 2	8E	z: 0....65535	
Bringe Adressenteil und reserviere	BAR	z	$\langle H \rangle := \langle A \rangle$ $\langle A \rangle := z$	+		1; v, z	$t_A; \langle A \rangle$			R 3	DC		
Bringe Adressenteil negativ	BAN	z	$\langle A \rangle := -z$	+		1; -(v, z)				R 5	DF		
Bringe Adressenteil negativ und reserviere	BANR	z	$\langle H \rangle := \langle A \rangle$ $\langle A \rangle := -z$	+		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	s_1 : 0, 1, 2 oder 3 (TK) s_2 : A, Q, D und H	
Lösche in A	LA	s	$\langle A \rangle_s := 0$ $\langle M \rangle := 0$ nur bei $s = M$	+		Wirkung $s = F: \langle A \rangle_{1-40} := 0 \hat{=} \text{ Mantissesteil}$ $s = 2: \langle A \rangle_{1-24} := 0 \hat{=} \text{ linkes Halbwort}$ $s = E: \langle A \rangle_{1-48} := 0 \hat{=} \text{ Exponententeil}$ $s = 3: \langle A \rangle_{33-48} := 0 \hat{=} \text{ rechtes Drittel}$ $s = V: \langle A \rangle_{1-2} := 0 \hat{=} \text{ Vorzeichenstellen}$ $s = M: \langle M \rangle := 0 \hat{=} \text{ Markenregister}$ $s = H: \langle A \rangle_{1-42} := 0 \hat{=} \text{ ohne rechte Hexade}$ $s = T: \langle A \rangle_{1-44} := 0 \hat{=} \text{ ohne rechte Tetra-de}$			Die Spezifikationen F bis M können kombiniert verwendet werden H und T nur einzeln od. mit M erlaubt			R 4	8B

Index: <u>Bringe Adressenteil</u>	XBA	z	 := z			z			B	1	01	
Index: <u>Bringe Adressenteil negativ</u>	XBAN	z	 := -z			-z			B	2	19	z: 0...65535
Setze <u>Index</u>	ZX	p i	<i> := p := p			v, p	v, p		B	6	1A	p: ±0...±127
Setze <u>Unterprogrammregister</u>	ZU	i	<U> := i	+			i		B	1	3E	

			$\langle s_2 \rangle_t$	$\langle M \rangle$								
Setze <u>Typenkennung im Register</u>	ZTR	s	$\langle s_2 \rangle_t := s_1$ $\langle M \rangle := L$, falls $s_3 = M$	+								
Setze <u>Typenkennung 0</u>	ZTO	n	$\langle n \rangle_t := 0$	+		0						
Setze <u>Typenkennung 1</u>	ZT1	n	$\langle n \rangle_t := 1$	+		1						
Setze <u>Typenkennung 2</u>	ZT2	n	$\langle n \rangle_t := 2$	+		2						
Setze <u>Typenkennung 3</u>	ZT3	n	$\langle n \rangle_t := 3$	+		3						

			$\langle n \rangle$	$\langle n \rangle_a$	$\langle K \rangle_{aL}$	$\langle K \rangle_{aR}$	$\langle X \rangle$	Alarm TK				
Lösche <u>Speicher</u>	LC	n	$\langle n \rangle := +0$ $\langle n \rangle_a := \langle n \rangle_a$	+								
Lösche <u>markiert</u>	LMT	n	$\langle n \rangle := +0$ $\langle n \rangle_a := L$	+								
Lösche <u>Marke im Speicher</u>	LMC	n	$\langle n \rangle_a := 0$	+								
Setze <u>Marke im Speicher</u>	ZMC	n	$\langle n \rangle_a := L$	+								
Lösche und setze <u>Merklichter</u>	LZL	sL sR	$\langle K \rangle_{aR} := 0$ $\langle K \rangle_{aL} := L$			L	0					
Negiere <u>Merklichter</u>	NL	s	$\langle K \rangle_a := \langle K \rangle_a$ invertiert			invertiert						
Setze <u>Indexbasis</u>	ZI	m	$\langle X \rangle := \langle m \rangle_{a-24}$				$\langle m \rangle_{a-24}$					

s: Merklichter
0, 1, 2, 3, 4, 5, 6, 7 und 8
0 bedeutet kein Merklicht
(0 muß angegeben werden)

x: Anzahl der Indexregister,
die zurückgespeichert werden

12

Bezeichnung	Code	adr	Wirkung	mod2 R	$\langle F \rangle_{9-24}$	$\langle H \rangle$	$\langle B \rangle$	$\langle U \rangle$	U-Alarm	B	1	00	Bemerkungen
<u>Nullbefehl</u>	NULL	z	keine Wirkung										Nur übliche Erhöhung des Befehlsfolgeregisters. z: ohne Bedeutung, muß angegeben werden.
<u>Wartebefehl</u>	WB	z	z Uhrimpulse warten (Uhrimpulse alle 10µs)										z: 0...65535 Takte: bei z = 0: 3 Takte bei z ≠ 0: (10z-5)µs
<u>Springe</u>	S	m	$\langle F \rangle_{9-24} := m$	+									
<u>Springe nach Ersetzung</u>	SE	m	$\langle F \rangle := \langle m \rangle + \text{mod}2$	sp x									
<u>Springe und bringe $\langle F \rangle + 1$ nach $\underline{B}$</u>	SFB	m	$\langle B \rangle := \langle F \rangle + 1$ $\langle F \rangle_{9-24} := m$	+			$\langle F \rangle + 1$						
<u>Springe nach Ersetzung und bringe $\langle F \rangle + 1$ nach $\underline{B}$</u>	SFBE	m	$\langle B \rangle := \langle F \rangle + 1$ $\langle F \rangle_{1-24} := \langle m \rangle + \text{mod}2$	sp			$\langle F \rangle + 1$						
<u>Springe in Unterprogramm</u>	SU	m	$\langle U \rangle := \langle U \rangle + 1$ $\langle \langle U \rangle \rangle := \langle F \rangle + 1$ $\langle F \rangle_{9-24} := m$	+				$\langle U \rangle + 1$	$\langle U \rangle_{alt} = 254$				
<u>Springe in Unterprogramm nach Ersetzung</u>	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				$\langle U \rangle + 1$	$\langle U \rangle_{alt} = 254$				
<u>Springe ins System und reserviere</u>	SSR	Pl P _a	$\langle F \rangle := \langle a + 6 \rangle$ $\langle B \rangle := \text{adr}$	+									P _l } : 0...255 P _a } : Anfangsadresse des Leitblocks

$\langle A \rangle$

<u>Springe</u> wenn <u>identisch</u> <u>0</u>	SIO	m	$\langle A \rangle = 0$	+	m *						B R	1 A4
<u>Springe</u> wenn <u>nicht</u> <u>0</u>	SNO	m	$\langle A \rangle \neq 0$	+	m *						B R	1 A7
<u>Springe</u> wenn <u>größer</u> <u>gleich</u> <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>Springe</u> wenn <u>größer</u> <u>0</u>	SGO	m	$\langle A \rangle > 0$	+	m *						B R	1 D4
<u>Springe</u> wenn <u>kleiner</u> <u>gleich</u> <u>0</u>	SKGO	m	$\langle A \rangle \leq 0$	+	m *						B R	1 A5
<u>Springe</u> wenn <u>kleiner</u> <u>0</u>	SKO	m	$\langle A \rangle < 0$	+	m *						B R	1 D5 $\langle A \rangle_t = 2$ oder 3: Sprung nie!
<u>Springe</u> wenn <u>rechtes</u> Bit in A <u>gesetzt</u>	SR	m	$\langle A \rangle_{*s} = L$	+	m *						B R	1 B8
<u>Springe</u> wenn <u>rechtes</u> Bit in A <u>nicht</u> <u>gesetzt</u>	SRN	m	$\langle A \rangle_{*s} = 0$	+	m *						B R	1 BA
<u>Springe</u> wenn <u>identisch</u>	SI	m	$\langle A \rangle = \langle H \rangle$	+	m *	norm**	norm**				B R	** 6 AC
<u>Springe</u> wenn <u>nicht</u> <u>identisch</u>	SN	m	$\langle A \rangle \neq \langle H \rangle$	+	m *	norm**	norm**				B R	** 6 AD
<u>Springe</u> wenn <u>größer</u> <u>gleich</u>	SGG	m	$\langle A \rangle \geq \langle H \rangle$	+	m *	norm**	norm**				B R	** 6 AF
<u>Springe</u> wenn <u>größer</u>	SG	m	$\langle A \rangle > \langle H \rangle$	+	m *	norm**	norm**				B R	** 6 AB
<u>Springe</u> wenn <u>kleiner</u> <u>gleich</u>	SKG	m	$\langle A \rangle \leq \langle H \rangle$	+	m *	norm**	norm**				B R	** 6 AE
<u>Springe</u> wenn <u>kleiner</u>	SK	m	$\langle A \rangle < \langle H \rangle$	+	m *	norm**	norm**				B R	** 6 AA

Bezeichnung	Code	adr	Sprungbedingung	mod2 R	$\langle F \rangle_{s=24}$	$\langle i \rangle$	$\langle M \rangle \langle K \rangle_s$	$\langle D \rangle$	$\langle U \rangle$	Alarm	Werk	akt. Int.	Bemerkungen
<u>Springe</u> wenn <u>Index</u> <u>identisch</u> 0	SXI	m	$\langle B \rangle = \pm 0$		m *						B	1 1 24	
<u>Springe</u> wenn <u>Index</u> <u>nicht identisch</u> 0	SXN	m	$\langle B \rangle \neq \pm 0$		m *						B	1 1 27	
<u>Springe</u> wenn <u>Index</u> <u>größer</u> <u>gleich</u> 0	SXGG	m	$\langle B \rangle \geq \pm 0$		m *						B	1 1 25	$\langle B \rangle_1 = \langle B \rangle_v$
<u>Springe</u> wenn <u>Index</u> <u>größer</u> 0	SXG	m	$\langle B \rangle > \pm 0$		m *						B	1 1 OE	
<u>Springe</u> wenn <u>Index</u> <u>kleiner</u> <u>gleich</u> 0	SXKG	m	$\langle B \rangle \leq \pm 0$		m *						B	1 1 26	
<u>Springe</u> wenn <u>Index</u> <u>kleiner</u> 0	SXK	m	$\langle B \rangle < \pm 0$		m *						B	1 1 CF	
<u>Springe</u> wenn <u>Index</u> <u>gröÙe</u> <u>rechtes</u> <u>Bit</u> = L	SXR	m	$\langle B \rangle_{24} = L$		m *						B	1 1 B2	
<u>Springe</u> wenn <u>Index</u> <u>gröÙe</u> <u>rechtes</u> <u>Bit</u> <u>nicht</u> L	SXRN	m	$\langle B \rangle_{24} = 0$		m *						B	1 1 B3	
<u>Springe</u> und <u>zähle</u> wenn <u>Index</u> <u>kleiner</u> 0	SZX	p i	$\langle i \rangle < \pm 0$		$\langle F \rangle + p *$	$\langle i \rangle + 1 *$					B	1 1 2 OA	$\langle i \rangle_1 = \langle i \rangle_v$ p: $\pm 0 \dots \pm 127$
<u>Springe</u> wenn <u>Marke</u>	SM	m	$\langle M \rangle = L$	+	m *		0				B	1 1 34	
<u>Springe</u> wenn <u>Marke</u> <u>nicht</u>	SMN	m	$\langle M \rangle = 0$	+	m *		0				B	1 1 35	
<u>Springe</u> wenn <u>arithmetischer Alarm</u> (BU-Alarm)	SAA	m	BU-Alarm	+	m *					BU-Alarm wird gelöscht	B R	1 1 A9	
<u>Springe</u> wenn <u>Alarm</u> (<u>Typenennung</u>)	SAT	m	TK-Alarm	+	m *					TK-Alarm wird gelöscht	B R	1 1 A8	
<u>Springe</u> wenn <u>Typenennung</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
<u>Springe</u> wenn <u>Typenennung</u> <u>nicht</u>	STN	p s	$\langle s_2 \rangle_t \neq s_1$	+	$\langle F \rangle + p *$						B R	10 3 91	
<u>Springe</u> wenn <u>Bit</u> <u>gesetzt</u>	SBIT	p s	$\langle s_2 \rangle_{s_1} = L$		$\langle F \rangle + p *$	Tabelle für Zeitberechnung: 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.
<u>Springe</u> wenn <u>Exponent</u> <u>größer</u> <u>gleich</u>	SEGG	p ₁ p ₂	$\langle A \rangle_{s_1-s_2} \geq p_2$	+	$\langle F \rangle + p *$					falls $\langle A \rangle_t \neq 0$: TK-Alarm	B R	10 4 93	p ₁ : Sprungweite $\pm 0 \dots \pm 127$ p ₂ : Vergl.-Exponent ****
<u>Springe</u> wenn <u>Merklicht</u>	SL	p s	eines der $\langle K \rangle_s = L$		$\langle F \rangle + p *$						B	9 1 1E	
<u>Springe</u> 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	
<u>Springe</u> wenn <u>Merklicht</u> <u>nicht</u>	SLN	p s	alle $\langle K \rangle_s = 0$		$\langle F \rangle + p *$						B	9 1 1C	p: $\pm 0 \dots \pm 127$ s: Merklichter 1, 2, ..., und 8
<u>Springe</u> wenn <u>Merklicht</u> <u>nicht</u> <u>sonst 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 <u>richtig</u>	PDP	n	DP = richtig	+	$\langle F \rangle + 2 *$	$t_n: \langle n \rangle$	$\langle n \rangle_{s=OL, DP, (n)}$	R 21 21 B	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
**** $0 \dots 127$ (positiv), $NO \dots N127$ (negativ)

Modifizieren

Bezeichnung	Code	adr	Wirkung	mod2 R	mod1	mod2 neu		<i>	Werk	Info	Bemerkungen
Modifiziere in jedem Fall	MF	i	mod1 := <i> + mod2 := <i> + mod2	sp	<i> *	0	<i> *		B	1	OB
Modifiziere in jedem Fall mit unveränd. B	MFU	i	mod1 := <i> + mod2	sp	<i> *	0			B	1	OB Bits 9 - 11 = OOL
Modifiziere aus Speicher in jedem Fall	MCF	m	mod1 := <m> + mod2 := <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	il ir	mod1 := <ir> + mod2 mod2 := <il> := <il>	sp	<ir> *	<il>	<il>		B	5	09
Modifiziere	M	i	 := <i> + mod2 mod2 := <i> + mod2	sp		<i> *	<i> *		B	1	OB Bits 9 - 11 = L00
Modifiziere nach Erhöhung	MH	p i	mod2 := <i> + p := <i> + p <i> := <i> + p			<i> + p	<i> + p		B	10,5	2D p: ±0...±127
Modifiziere nach Erhöhung um Indexzelle	MHX	il ir	mod2 := <il> + <ir> := <il> + <ir> <ir> := <il> + <ir>			<il> + <ir>	<il> + <ir>		B	14,5	OE
Modifiziere mit Register und Indexzelle	MRX	s i	mod2 := <i> ± <s1> := <i> ± <s1> falls s3 = C: <i> := <i> ± <s1>			<i> ± <s1>	<i> ± <s1>	falls s3 = C: <i> ± <s1>	B	12	8D **
Modifiziere aus Speicher	MC	m	 := <m> + mod2 mod2 := <m> + mod2	sp x		<m> *	<m> *		B	2	14
Modifiziere aus Speicher nach Ersetzungen	MCE	m	 := <...<m>...> + mod2 mod2 := <...<m>...> + mod2	sp x		<...<m>...> * das 1. Bit wird dem 2. angeglichen	<...<m>...> * das 1. Bit wird dem 2. angeglichen		B	13x -9	17
Modifiziere mit Adressenteil	MA	z	 := z + mod2 mod2 := z + mod2	sp		z *	z *		B	1	03
Modifiziere mit negativem Adressenteil	MNA	z	 := -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

s1: A, Q, D, H (rechtes Halbwort)
oder B
s2: leer: + } statt ±
N: - }
s3: leer: nicht speichern
C: zurückspeichern

x: Anzahl der Halbwörter
in der Ersetzkette
Abbruch wenn:
<...<m>...>1 = L

z: 0...65 535

Ersetzen (und modifizieren)

Bezeichnung	Code	adr	Wirkung	mod2	R	op	odr	(B)	(i)	mod2 _{neu}	Werk	Ikarte	Int.	Bemerkungen
<u>Ersetze</u>	E	c i	adr := <i> mod2 := mod2		sp	c	<i>			mod2	B	4	29	
<u>Ersetze zählend</u>	EZ	c i	adr := <i> + 2 + mod2 := <i> + 2 <i> := <i> + 2		sp	c	<i> + 2 * <i> + 2	<i> + 2	<i> + 2	0	B	5	2B	
<u>Ersetze negativ zählend</u>	ENZ	c i	adr := <i> + mod2 := <i> - 2 <i> := <i> - 2		sp	c	<i> * <i> - 2	<i> - 2	<i> - 2	0	B	8	2A	
<u>Ersetze und modifiziere mit B</u>	EMB	c i	adr := <i> mod2 := 			c	<i>				B	4	28	
<u>Modifiziere Adressenteil mit B</u>	MAB	c p	adr := + p := + p			c	 + p + p	 + p		0	B	9,5	20	p: ±0...±127
<u>Modifiziere Adressenteil mit B bei Invarianz der Sprungadresse</u>	MABI	c p	adr := + p := + p			c	 + p + p	 + p		0	B	9,5	3F	p: ±0...±127 *****
<u>Modifiziere über U</u>	MU	c p	adr := <<U>> + p			c	<<U>> + p				B	14,5	05	<U> := <U> - 1 wenn c Sprungbef. u. Bedingung erfüllt ist** p: ±0...±127
<u>Ersetze nach Modifizierung über U</u>	EMU	c p	adr := <<U>> + p mod2 := mod2		sp	c	<<U>> + p			mod2	B	29,5	04	p: ±0...±127
<u>Relativ-Adressierung mit Registerinhalt</u>	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>Registeradressierung</u>	R	c s	operand := <s1>		+	c	Reg.- <s1> = Operand: s1 = A, Q, D oder H: ro := t, s1, i, <s1> *** s1 = Y, U : ro := 1; 0, <s1> s1 = B : ro := 1; v, s1 = F : ro := 1; v, <F> + 1				B R	4	96	s2: L : linke 24 Bits : leer : rechte 24 Bits c : erlaubter Code ro: Registeroperand
<u>Tue</u>	T	m	op, adr := <m> mod2 := mod2		sp x	<m> 1-s	0, <m> s-24			mod2	B	5	CC	<m>: beliebig

* Ist vom vorhergehenden Befehl ein Modifikator 2. Art vorhanden, so wird er addiert
 ** Sprung in eine andere Großseite ist möglich
 *** bei s1 = A, Q, D oder H und <s1>4 = 0 oder 1:
 <s1>1 := <s1>
 <s1>2 := <s1>
 **** für c die Sprungbefehle von Seite 12 u. 13 (mit Ausnahme der Befehle NULL, PDP, SSR und WB)
 bei erfüllter Sprungbedingung ist ein Sprung in eine andere Großseite möglich.

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 GS8D GS8I
 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	(A)	(B)	(Q)	(Y)	Werk	Takte	Int.	Bemerkungen
US	s	i	$\langle A \rangle := \langle A \rangle_{\text{umgeschlüsselt}}$	sp	$\langle A \rangle = \text{umzu-}$ schl. Zeich. $\langle i \rangle + \text{mod} 2 =$ Anfangsadr. der Umschl. Tabelle *	$t_1; \langle A \rangle_{\text{umge-}}$ schl. Adresse $\text{falls } s_2 = E$ $\langle A \rangle_1 - x := 0$	$\langle i \rangle + \text{mod} 2$ schl. Adresse der Umschl. Tabelle	$\text{falls } s_2 = G$ $t_1; \langle A \rangle_{\text{umge-}}$ schl. Adresse der Umschl. Tabelle	$\text{falls } s_2 = G$ $t_1; \langle A \rangle_{\text{umge-}}$ schl. Adresse der Umschl. Tabelle	R	$4p + 1$	E1	s_1 : Bitlänge der Zeichen 6, 8 oder C (12 Bits) s_2 : E = ein Zeichen rechtstünd. G = alle Zeich. 8, 6 oder 4 t_1 : Typenkenkung des Wortes, in dem das linke Zeichen steht

IR	s	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													
BÜ Alarm													
SH	s	p	Schift gemäß Spezifikation um p Stellen	+	s_1 : A = Register A 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	≥ 1				R	$2(q+r)+5$ Rest r $p:4=q$ 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 Schifschritte $\neq 0 \dots +127$ $p:4 = q$ Rest r
SHB	s	p	$\langle B \rangle := \langle B \rangle$ geschiftet um p Stellen nach rechts oder links mit Nachziehen von O-Bits		s: R = Schift nach rechts L = Schift nach links					B	$4 \cdot p + 6$	21	p: Anzahl der Schifschritte O...255

Alarm													
Y BÜ TK													
VAQ	z		$\langle A, Q \rangle := \langle A \rangle + \langle Q \rangle \cdot 2^{-4s}$ $\langle Q \rangle_v := \langle A \rangle_1$ 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
NRM	s		bei s = G: $\langle A \rangle := \langle A \rangle_{\text{normalisiert}}$		Die Festkommazahl wird als echter Bruch aufgefaßt $\langle A \rangle_t := 0; \langle A, Q \rangle_{\text{gerundet}}, \langle Q \rangle := 1, +0$	$\neq 0$					$2t + 3$		s = G, FG, F, F4, N oder L
			bei s = FG: $\langle A \rangle_{\text{einstk.}} := \langle A, Q \rangle_{\text{restk.}}$			$\neq 1$					$2t + 10$		t: Anzahl der Tetraden- schifte
			bei s = F: $\langle A, Q \rangle := \langle A, Q \rangle_{\text{normalisiert}}$	+		$\neq 1$					$2t + 7$		
			bei s = F4: $\langle A, Q \rangle := \langle A, Q \rangle_{\text{normalisiert}}$			$\neq 1$					$2t + 7$		
			bei s = N falls $\langle A \rangle \neq +0$ $\langle A \rangle := \langle A \rangle$ links geschift. bis $\langle A \rangle_1 = L$		beim Schiften werden Nullen nachgezogen e	$\neq 1$					$2t + 7$		
			bei s = L: $\langle A \rangle := \langle A \rangle$ links geschift. bis $\langle A \rangle_1 = 0$		falls $\langle A \rangle = +0: \langle A \rangle := \langle A \rangle$ beim Schiften werden Nullen nachgezogen e						$2e + 3$		e: Anzahl der Eierschifte

*Tabelle in Viertelwörtern.
Die Anfangsadresse ist eine Halbwortadresse

Konvertiere Dezimalzahl in Festkommazahl rechtsbündig	KDFR	p	$\langle A \rangle_{\text{rest}} \cdot x := \langle A, Q \rangle_{\text{rest}}$	+	$\langle A, Q \rangle = \text{pos. Dez.-Zahl}$ Komma rechts	$\langle A \rangle := 1; \langle A, Q \rangle$ $\langle Q \rangle := 1; +0$	$\langle D \rangle := 1; +0$ $\langle H \rangle := 1; +0$	+0		R	7p + 22	94	p: 1...13 Anzahl der Dezimalstellen, die konvertiert werden
Konvertiere Festkommazahl linksbündig in Dezimalzahl	KFLD	p	$\langle A, Q \rangle_{\text{rest}} := \langle A \rangle_{\text{rest}} \cdot x$	+	$\langle A \rangle = \text{pos. Festk.-Zahl}$ Komma links $\langle A \rangle_1 = 0$ oder 1 $\langle A \rangle_1 - \langle A \rangle_2 = 0$ Reg. D steht mit TK = 1 der Rest	Im doppelt langen Register die Dezimalzahl jeweils mit Typenkennung = 1. Im Reg. D steht mit TK = 1 der Rest	+0 $\geq$		2,3	R	$\langle A \rangle > 0$ 9p + 5	95	p: 1...13 Anzahl der gewünschten Dezi- malstellen

* Tabelle in Viertelwörtern
Die Anfangsadresse ist eine Halbwortadresse

Tabelle durchsuchen

		Suchbedingung		Eingangsgrößen				Alarm			
								BÜ TK			

Zentralcode

Stand 1972

0000	000L	00LO	00LL	0L00	0L0L	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
NUL	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
		SUB	BEL			"	%	^	+	(	0	A	Q	a	q
1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
		EM	DC1			,	\$	v	-	)	1	B	R	b	r
2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
		CAN	DC2			/	#	~		[	2	C	S	c	s
3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
			DC3			\	\$(x)		/	]	3	D	T	d	t
4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
			DC4				¢				4	E	U	e	u
5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
		NL	FL			^					5	F	V	f	v
6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
SOH	CR					o	@	↑		<	6	G	W	g	w
7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
STX	NF					~	&		=	>	7	H	X	h	x
8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
ETX	VT						*		≠	(MZ)}	8	I	Y	i	y
9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
EOT										.	9	J	Z	j	z
10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
ENQ										,		K	Ä	k	ä
11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
ACK		HT				/			<	:		L	Ö	l	ö
12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
DLE		BS	IS4			,	□		>	;		M	Ü	m	ü
13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
NAK		ESC	IS3			)		10	≡	i		N		n	ß
14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
SYN		SO	IS2			—			≡	?		O		o	
15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255
ETB		SI	IS1			—	π			SP	(PZ)}	P		p	DEL

2 ⁿ	n	2 ⁿ	2 ⁿ	n	2 ⁿ
2	1	0.5	2 251 769 813 685 248	51	0.151 444 069 209 850 062 616 168 462 667 236 328 125
4	2	0.25	4 503 539 627 370 496	52	0. 222 044 604 825 031 306 084 728 333 618 164 062 5
6	3	0.125	9 007 199 254 740 992	53	0. 111 022 302 462 515 654 042 363 168 808 062 031 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	36 028 797 018 983 968	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	0.151 013 877 787 807 814 456 755 265 365 851 135 253 906 25
128	7	0.007 812 5	144 115 188 075 855 872	57	0. 006 938 893 903 907 228 377 647 697 925 567 628 853 125
256	8	0.003 906 25	288 230 376 151 711 744	58	0. 003 469 446 953 814 188 823 848 962 783 813 476 562 5
512	9	0.001 953 125	576 460 752 303 423 488	59	0. 001 734 723 475 876 807 084 411 924 481 391 906 738 281 25
1 024	10	0.000 976 562 5	1 152 921 504 606 946 976	60	0. 000 867 361 737 988 403 205 982 240 685 853 369 140 625
2 048	11	0.000 488 281 25	2 305 843 009 213 693 952	61	0.181 433 690 868 984 201 773 602 981 120 347 978 684 570 312 5
4 096	12	0. 244 140 625	4 611 686 018 427 387 904	62	0. 216 840 434 497 100 886 801 490 590 173 988 342 285 156 25
8 192	13	0. 122 070 312 5	9 223 372 036 854 775 808	63	0. 108 420 217 248 550 440 400 745 280 086 984 171 142 578 125
16 384	14	0. 061 035 156 25	18 446 744 073 709 551 616	64	0. 054 210 108 824 275 221 700 372 640 043 487 085 571 286 082 5
32 768	15	0. 030 517 578 125	36 893 488 147 419 103 232	65	0. 027 105 054 312 137 010 850 186 320 021 748 542 785 644 531 25
65 536	16	0.000 015 258 768 062 5	73 786 976 294 838 206 464	66	0.181 013 552 527 158 068 805 425 083 160 010 874 271 382 822 266 625
131 072	17	0. 007 626 394 531 25	147 573 952 589 676 412 928	67	0. 006 776 263 578 034 402 712 548 590 005 437 135 896 411 132 812 5
262 144	18	0. 003 814 697 265 625	295 147 905 179 352 825 856	68	0. 003 388 131 789 017 201 356 273 290 002 718 567 848 205 596 408 25
524 288	19	0. 001 907 348 632 812 5	590 295 810 358 705 651 712	69	0. 001 694 065 894 508 800 678 136 645 001 359 283 924 102 783 203 125
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	0.000 000 478 837 158 203 125	2 361 183 241 434 822 606 848	71	0.211 423 516 473 627 150 169 534 161 250 339 820 981 025 895 800 781 25
4 194 304	22	0. 238 418 579 101 562 5	4 722 366 482 869 645 213 696	72	0. 211 758 236 813 575 064 767 080 625 169 910 512 847 900 380 625
8 388 608	23	0. 119 208 288 550 781 25	9 444 732 965 738 290 427 392	73	0. 005 879 118 406 787 542 363 540 312 584 955 245 256 423 950 312 5
16 777 216	24	0. 059 604 644 775 390 625	18 889 465 931 478 590 854 784	74	0. 052 939 559 203 383 771 191 770 156 292 477 622 628 211 975 087 656 25
33 554 432	25	0. 029 802 322 387 685 312 5	37 778 931 862 957 161 709 568	75	0. 026 469 779 801 696 885 985 078 146 238 811 314 105 987 548 828 125
67 108 864	26	0.000 000 014 901 161 183 847 656 25	75 557 825 914 323 419 136	76	0.211 013 234 889 800 848 442 797 942 539 073 119 405 657 062 983 774 414 082 5
134 217 728	27	0. 007 450 580 596 923 828 125	151 115 727 451 628 646 838 272	77	0. 006 617 444 900 424 21 398 971 288 536 559 702 828 526 466 887 031 25
268 435 456	28	0. 003 725 290 298 461 914 062 5	302 231 454 903 657 293 678 544	78	0. 003 306 722 450 212 110 699 485 634 768 279 851 414 283 248 443 903 515 625
536 870 912	29	0. 001 862 645 149 230 957 031 25	604 462 908 807 314 587 353 088	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	1 208 925 819 614 629 174 708 176	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	0.000 000 000 465 861 287 307 739 257 812 5	2 417 851 639 229 258 349 412 352	81	0.241 413 590 306 276 513 837 435 704 348 034 981 426 782 906 065 459 433 125
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. 206 795 153 138 256 918 717 852 173 017 490 713 391 453 027 725 219 728 562 5
8 589 934 592	33	0. 116 415 321 828 934 814 453 125	9 671 406 556 917 033 397 649 408	83	0. 103 397 576 569 128 459 358 926 086 508 745 356 895 726 513 882 608 863 281 25
17 179 869 184	34	0. 058 207 680 913 487 407 228 562 5	19 342 813 113 834 066 795 298 816	84	0. 051 698 788 284 564 229 679 463 043 254 372 678 347 963 256 931 304 931 640 625
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. 025 849 384 142 282 114 839 731 521 627 186 339 173 831 628 465 652 820 312 5
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 808 640 625	77 371 252 455 336 267 181 195 264	86	0.241 012 924 897 071 141 057 419 865 780 813 593 169 586 965 814 232 826 232 910 156 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. 006 462 346 535 570 528 709 932 880 406 796 584 793 482 907 116 413 116 455 078 125
274 877 906 944	38	0. 003 637 978 807 091 712 851 680 156 25	309 485 009 821 345 068 724 781 056	88	0. 003 231 174 267 785 264 354 966 440 203 398 292 366 741 453 558 206 558 227 539 062 5
549 753 813 888	39	0. 001 818 989 403 545 856 475 830 078 125	618 970 019 642 680 137 449 562 112	89	0. 001 615 587 133 892 632 177 483 220 101 699 146 198 370 726 779 103 279 113 789 531 25
1 099 511 627 776	40	0. 000 909 494 701 772 928 237 915 039 062 5	1 237 940 039 295 360 274 889 124 224	90	0. 000 807 793 566 946 316 088 741 610 050 849 573 069 185 363 389 551 639 556 884 765 625
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25	2 479 880 078 070 760 549 798 248 448	91	0.271 403 896 783 473 158 044 370 905 025 424 786 549 592 881 694 775 819 778 442 382 812 5
4 398 046 511 104	42	0. 227 373 675 443 232 059 478 759 765 625	4 851 760 157 141 521 099 596 496 896	92	0. 201 948 391 736 579 022 165 402 512 712 363 274 796 340 847 387 909 889 221 191 406 25
8 796 083 022 208	43	0. 113 686 637 721 616 029 739 379 882 812 5	9 693 520 314 293 042 199 192 983 792	93	0. 100 974 195 868 289 511 092 701 256 356 196 637 396 170 423 893 954 944 610 595 703 125
17 592 166 044 416	44	0. 056 843 418 860 808 014 869 689 941 406 25	19 807 040 628 586 084 368 365 987 584	94	0. 050 487 097 834 144 755 546 350 628 178 088 318 698 085 211 846 977 472 305 297 851 562 5
35 184 372 088 832	45	0. 028 421 708 430 404 007 434 844 970 703 125	39 614 081 257 132 168 796 771 975 168	95	0. 025 243 548 967 072 377 773 175 314 069 049 159 349 542 605 923 488 736 152 648 925 781 25
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5	79 228 162 514 264 337 563 533 950 336	96	0.271 012 621 774 483 536 188 886 587 657 044 524 579 674 771 302 961 744 368 076 324 462 890 625
140 737 486 355 328	47	0. 007 105 427 357 601 001 659 711 242 675 781 25	158 456 325 028 528 675 187 087 900 672	97	0. 206 310 887 241 768 094 43 283 828 522 262 239 337 385 651 480 872 184 038 162 231 445 312 5
281 474 976 710 656	48	0. 003 552 713 678 900 500 929 355 621 337 880 625	316 912 650 057 057 360 174 175 801 344	98	0. 003 155 443 620 884 047 221 618 914 281 134 149 892 825 740 436 082 019 881 115 722 656 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. 001 577 721 810 442 023 810 823 412 870 218 046 009 540 557 861 328 125
1 125 899 906 842 624	50	0. 000 888 178 419 700 125 232 338 905 334 472 656 25	1 267 650 600 228 228 401 496 703 205 376	100	0. 000 788 860 905 221 011 805 411 728 565 282 786 228 673 206 435 109 023 004 770 278 880 864 082 5

(:) = Anzahl der Nullen

Konvertierungstafel

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

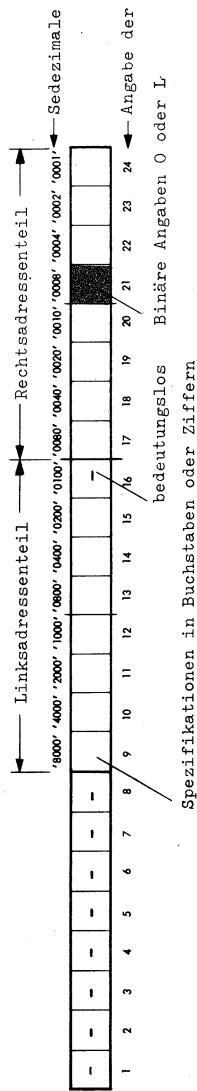
1 000 000	0F4 240	100 000	018 6A0	10 000	002 710	1 000	000 3E8	100	000 064	10	000 00A
2 000 000	1E8 480	200 000	030 D40	20 000	004 E20	2 000	000 700	200	000 0C8	20	000 014
3 000 000	20C 6C0	300 000	049 3E0	30 000	007 530	3 000	000 888	300	000 12C	30	000 01E
4 000 000	300 900	400 000	061 A80	40 000	009 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	001 858	700	000 28C	70	000 046
		800 000	0C3 500	80 000	013 880	8 000	001 F40	800	000 320	80	000 050
		900 000	00B 8A0	90 000	015 F90	9 000	002 328	900	000 384	90	000 05A

2 ⁷	1	Sedezi- male
2 ⁰	2	Sedezi- male

0000	000L	00LO	00LL	0LO0	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
NULL	LZL	MAB	ZMC	AB		DV	B	C	ST	C2	BL		CU	RLR	DA
XBA	HBA	SHB	LMC	SBB		DYD	BD	CR	STN	C3	VPU*		BZN	IR	DSB
MNA	NL	WTV	LMT	A	GAB	DVI	BQ	CMT	ZTR	CMC	SXR		ZDP*		DML
MA	VBA	WTR	LC	AC	GSBB	VAQ	BH	CMR	SEGG	BC	SXRN		BU	US	MLD
EMU	MC	SXI	SM	SBI	ML	GDV	BB	CN	KDFR	SIO	Y*		SGO		AT
MU	VBC	SXGG	SMN	SBD	MLR	REZ	BN	CB	KFLD	SKGO	LEI*		SKO		SBT
BCL	MCF	SXKG	S	SB	MLA	GDVI	BR	CD	R	SGGO	BCI			BNZ	BT
TBC	MCE	SXN	VSS*	SBC	MAR	B2VN	BNR	CQ	RT	SN0	ZI			CNZ	CT
MFU** M,XB	XC** XCN	EMB	SU	GSBI	MLN	VEL	M2N	VLA	AA	SAT	SR	ZT0	BZ2		WB
MD	XBAN	E	TCB	AU	MNR	AUT	M2NR	ATA	SBA	SAA	PDP	ZT1	BZ		SBIT
SZX	ZX	ENZ	SFB	GAC	MAN	ET	M2	ETA	LR	SK	SRN	ZT2	BQB	TDM	SFBE
MF	SW*	EZ		GA	MANR	ZUS	M2R	LA	SH	SG	SSR	ZT3	CZ	TLD	BSS
TXX	SLN	HXP	HBC	GSBD	GMLN	B3	A2	TXR	TRX	SI	SE	T	BAR	TLI	ZK
TTX	SNL	MH	MCFU	SBU	GMAN	B3V	SB2	RX*** MRX	HALT*	SN	SUE		BANR	TLOG	TOK
MHX	SL	HXX	ZU	GSBC	GML	B2	AQ	BA		SKG	BLEI	SXG		TMIN	QBR
HBPX	SLL	VXX	MABI	GSB	GMLA	B2V	SBQ	CH	NRM	SGG	VMO*	SXK	BAN	TMAX	QCR

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

Spezifikationen - Intern



IR S

	9	16	17	18	19	20	21	22	23	24
	-	-	A	Q	D	H	B	-	-	-

Es sind alle 2⁸ möglichen Bitanordnungen erlaubt. Falls die Bits 17 - 20 = 0; Wirkung wie Nullbefehl.

LA S

	9	16	17	18	19	20	21	22	23	24
	-	-	F	2	E	3	H	T	V	M

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

	9	16	17	18	19	20	21	22	23	24
	-	-	S ₁	A	Q	D	H	-	-	-

Es sind alle 2⁸ möglichen Bitanordnungen erlaubt.
s₁: Typenkennung 0,1,2 oder 3

LZL SL SR

	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8

Es sind in jedem Adressenteil jeweils alle 2⁸ möglichen Bitanordnungen erlaubt.

MFU, XB, M i

	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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

MRX, RX si

	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	A	Q	D	H	B	N	C	-	-	-	-	-	-	-	-	-

*MRX: Bit 15 = L
RX: Bit 15 = 0
Von den Bits 9 - 13 darf höchstens eines = L sein, sonst undefinierte Befehlsausführung.

NL S

SL, SLL, SLN, SNL, SW p s

	9	16	17	18	19	20	21	22	23	24
	p	(bei NL: bedeutungslos)	1	2	3	4	5	6	7	8

Es sind alle 2⁸ möglichen Bitanordnungen erlaubt.

NRM S

	9	16	17	18	19	20	21	22	23	24
	-	-	S	-	-	-	-	-	-	-

s besteht nur aus den Bits 17 - 19

** diese Bitanordnungen führen zu undefinierten Befehlsausführungen.

R, RLR CS

	9	16	17	18	19	20	21	22	23	24
	C	-	A	Q	D	H	-	-	-	-

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

	9	16	17	18	19	20	21	22	23	24
	-	-	A	Q	D	H	-	-	-	-

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

SBIT ps

	9	16	17	18	19	20	21	22	23	24
	-	-	S ₂	-	-	-	-	-	-	-

s₂: Register A =
Q =
D =
H =

SH sp

	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	A	Q	L	K	Z	R	U	B	-	-	-	-	-	-	-	-

Es sind alle 2⁸ 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 sp

	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Es sind alle 2⁸ möglichen Bitanordnungen erlaubt
* Bit 9 = L: links
Bit 9 = 0: rechts

ST, STN ps

	9	16	17	18	19	20	21	22	23	24
	-	-	S ₁	A	Q	D	H	-	-	-

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 si

	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	A	Q	D	H	N	-	-	-	-	-	-	-	-	-	-	-

TRX: Von den Bits 9 - 12 darf nur eines = L sein, sonst undefinierte Befehlsausführungen.
TXR: Es sind alle 2⁸ möglichen Bitanordnungen erlaubt.

US si

	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	6	8	C	G	-	-	-	-	-	-	-	-	-	-	-	-

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

XC, XCN i

	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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

ZK si

	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	B	I	V	-	-	-	-	-	-	-	-	-	-	-	-	-

* 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 Indexzeile erwartet

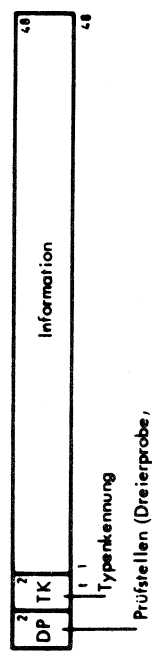
ZTR s

	9	16	17	18	19	20	21	22	23	24
	-	-	S ₁	A	Q	D	H	M	-	-

s₁: Typenkennung 0,1,2 oder 3
Von den Bits 19 - 22 darf höchstens eines = L sein, sonst undefinierte Befehlsausführung.

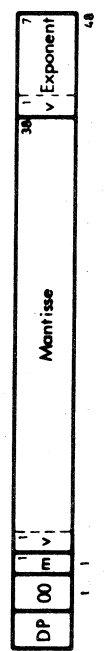
Wortstruktur (im Speicher)

ALLGEMEIN

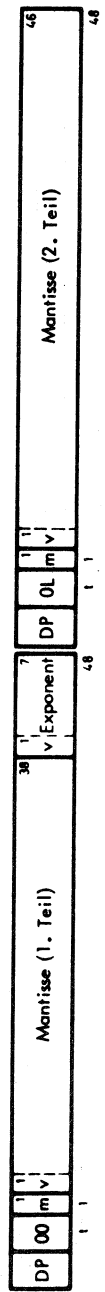


GLEITKOMMAZAHL (Basis 16)

Einfache Länge

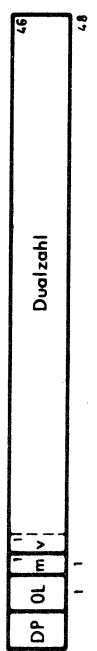


Doppelte Länge

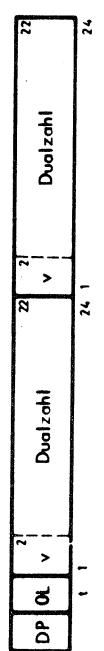


FESTKOMMAZAHL

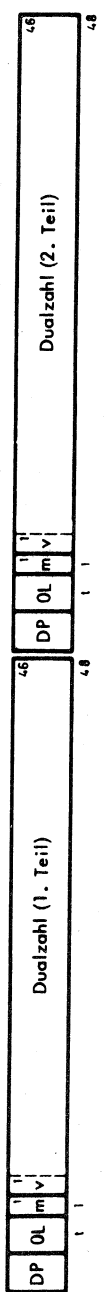
Einfache Länge



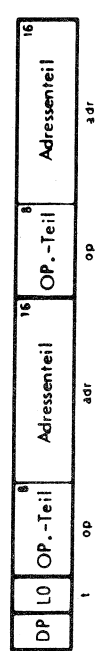
Halbe Länge (zwei Zahlen pro Wort)



Doppelte Länge



BEFEHLE (zwei Befehle pro Wort)



DP	LL	Beliebige Codierung, z.B. 6 Oktaden	48
			48

DP = Bits für Prüfzwecke (Dreierprobe)

TK = Bits für Typenkenennung

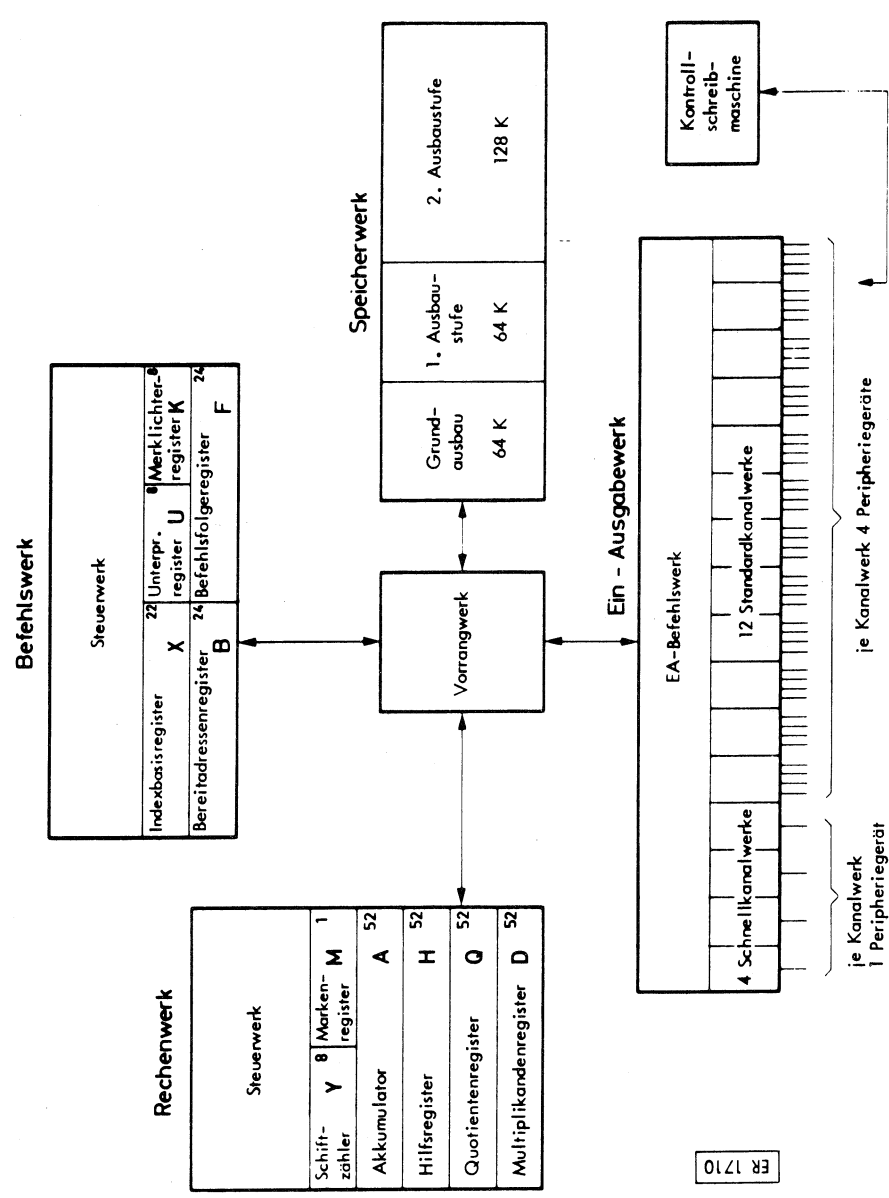
t = Typenkenennung

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

v = Vorzeichen

ER 1711

Blockschaltbild



Alphabetische Liste der Befehle

Code	Int. Seite	mod2	R	Up	Code	Int. Seite	mod2	R	Up	Code	Int. Seite	mod2	R	Up	Code	Int. Seite	mod2	R	Up	Code	Int. Seite	mod2	R	Up	Code	Int. Seite	mod2	R	Up	
A	42	7	+	+	CN	84	5	+		LA	88	10	+		S	36	12	+		SXRN	B3	13								
A2	7C	9	+	+	CNZ	E7	5	sp		LC	33	11	+	+	SAA	A9	13	+		SZX	OA	13								
AA	98	78	+	+	CQ	87	5	+		LMC	31	11	+	+	SAT	A8	13	+												
AB	40	7	+	+	CR	81	5	+		LMT	32	11	+	+	SB	46	7	+	+											
AC	43	7	+	+	CT	F7	5	+	+		9A	10	+		S82	7D	9	+	+											
AQ	7E	7	+	+	CU	D0	5	+		LZL	10	11			SBA	99	78	+	+											
AT	F4	9	+	+	CZ	DB	5	+							SBB	41	7	+	+											
ATA	89	9	+	+						M	*08	14	sp		SBC	47	7	+	+											
AU	69	8	+	+	DA	F0	8		+	M2	7A	9	+	+	SBD	45	7	+	+											
AUT	49	9	+	+	DML	F2	8		+	M2N	78	9	+	+	SBI	44	7	+	+											
B	70	4	+	+	D5B	F1	8		+	M2NR	79	9	+	+	SBIT	F9	13													
B2	6E	4	+	+	DV	60	7	+	+	M2R	78	9	+	+	S8Q	7F	7	+	+											
B2V	6F	4	+	+	DVD	61	7	+	+	MA	03	14	sp		SBT	F5	9	+	+											
B2VN	67	4	+	+	DVI	62	7	+	+	MAB	20	15			SBU	4D	8	+	+											
B3	6C	4	+	+						MABI	3F	15		+	SE	93	13	+	+											
B3V	6D	4	+	+						MAN	5A	7	+	+	SEGG	BC	12	sp	x											
BA	8E	10	+		E	29	15	sp		MANR	58	7	+	+	SFB	3A	12	+												
BAN	DF	10	+		EMB	28	15			MAR	57	7	+	+	SFBE	FA	12	sp												
BANR	DD	10	+		EMU	04	15	sp		MC	14	14	sp	x	SG	AB	12	+												
BAR	DC	10	+		ENZ	2A	15	sp	+	MCE	17	14	sp	x	SG0	D4	12	+												
BB	74	4	+	+	ET	6A	9	+	+	MCF	16	14	sp	x	SGG	AF	12	+												
BC	A3	5	+		ETA	8A	9	+		MCFU	3D	14	sp	x	SGG0	A6	12	+												
BC1	B6	5	+		EZ	2B	15	sp		MD	09	14	sp		SH	9B	16	+												
BCL	06	6	+							MF	*08	14	sp		SHB	21	16	+												
BD	71	4	+	+						MH	2D	14			SI	AC	12	+												
BH	73	4	+	+	GA	48	8	+	+	MHX	0E	14	+	+	SIO	A4	12	+												
BL	80	5	+	+	GAB	52	8	+	+	ML	54	7	+	+	SK	AA	12	+												
BLEI	BE	4	+	+	GAC	4A	8	+	+	MLA	56	7	+	+	SK0	D5	12	+												
BN	75	4	+	+	GDV	64	8	+	+	MLD	56	7	+	+	SKG	AE	12	+												
BNR	77	4	+	+	GMAN	66	8	+	+	MLN	58	7	+	+	SKG0	A5	12	+												
BNZ	E6	4	+	+	GML	5D	8	+	+	MLR	55	7	+	+	SL	1E	13													
BQ	72	4	+	+	GMLA	5E	8	+	+	MNA	02	14	sp	x	SLL	1F	13													
BQB	DA	4	+	+	GMLN	5C	8	+	+	MNR	59	7	+	+	SLN	1C	13	+												
BR	76	4	+	+	GSB	4F	8	+	+	MNR	*8D	14			SM	34	13													
BSS	F8	4	+	+	GSBB	53	8	+	+	MRX	05	15			SMN	35	13	+												
BT	F6	4	+	+	GSBC	4E	8	+	+	MU					SN	AD	12	+												
BU	D3	4	+	+	GSBD	4C	8	+	+	NL	12	11			SN0	A7	12	+												
BZ	D9	4	+	+	GSBI	48	8	+	+	NRM	9F	16	+		SNL	1D	13													
BZ2	D8	4	+	+						NULL	00	12	x		SR	B8	12	+												
BZN	D1	4	+	+	HBA	11	10			PDP	B9	13	+	+	SRN	BA	12	+												
C	80	5	+	+	HBC	3C	10		+						SSR	B8	12	+												
C2	A0	5	+	+	HBPX	0F	10			QBR	FE	6			ST	90	13	+												
C3	A1	5	+	+	HXP	2C	10			OCR	FF	6			STN	91	13	+												
C8	85	5	+	+	HXX	2E	10			R	96	15	+	+	SUE	8D	12	sp	x											
CD	86	5	+	+	IR	E1	16			REZ	65	8	+	+	SXG	CE	13													
CH	8F	5	+	+						RLR	E0	15	sp		SXGG	25	13													
CMC	A2	5	+	+	KDFR	94	17	+	+	RT	97	6	+		SX1	24	13													
CNR	83	5	+	+	KFLD	95	17	+	+	RX	*8D	10			SXK	CF	13													
CMT	82	5	+	+											SXKG	26	13													
															SXN	27	13													
															SXR	82	13													
																									</					

Code: Befehlscode
Int.: Interncode in 2 Sedezimalen

mod2: + = Modifizierung 2. Art

mod2: sp = spezielle Modifizierung 2. Art

R: x = als Zweitcode beim Befehl R zugefassen

Op: Dieser Befehl holt in der Abrufphase einen Operanden aus dem Speicher

* Diese Befehle benutzen gemeinsame Interncodes.
Unterschiede liegen im Adressenteil
siehe Seiten 22/23 "Spezifikationen-Intern".