

Betr.: DER LEERE RECHNER - Druckfehlerberichtigungen -

Wir bitten im LEEREN RECHNER die folgenden Korrekturen vorzunehmen.

Außerdem sind die beiliegenden Blätter auszutauschen bzw. neu einzuordnen.

CNZ Blatt 2: Zeile 6 von unten muß heißen:
W7.und W8. wie Fall A oder B

DA,DSB: Ausführungszeit: Fall A: 5,30 μ s
Fall B: 1,15 μ s

E, EMB, EMU, ENZ, EZ, MAB, MABI, MU, RLR, T:
Die Ausführungszeiten sind um 0,20 μ s zu verringern.

HALT, SW: Die Voraussetzung muß lauten:
BEBY=L oder BEBT1=L oder BEBT=BEWA=L oder
BEBT=BEWH=L

LC: Ausführungszeit: 0,65 μ s

LMC, LMT, ZMC:
Ausführungszeit: Fall A: 0,65 μ s

PDP: Im Kästchen BW rechts oben ist ein F einzusetzen.

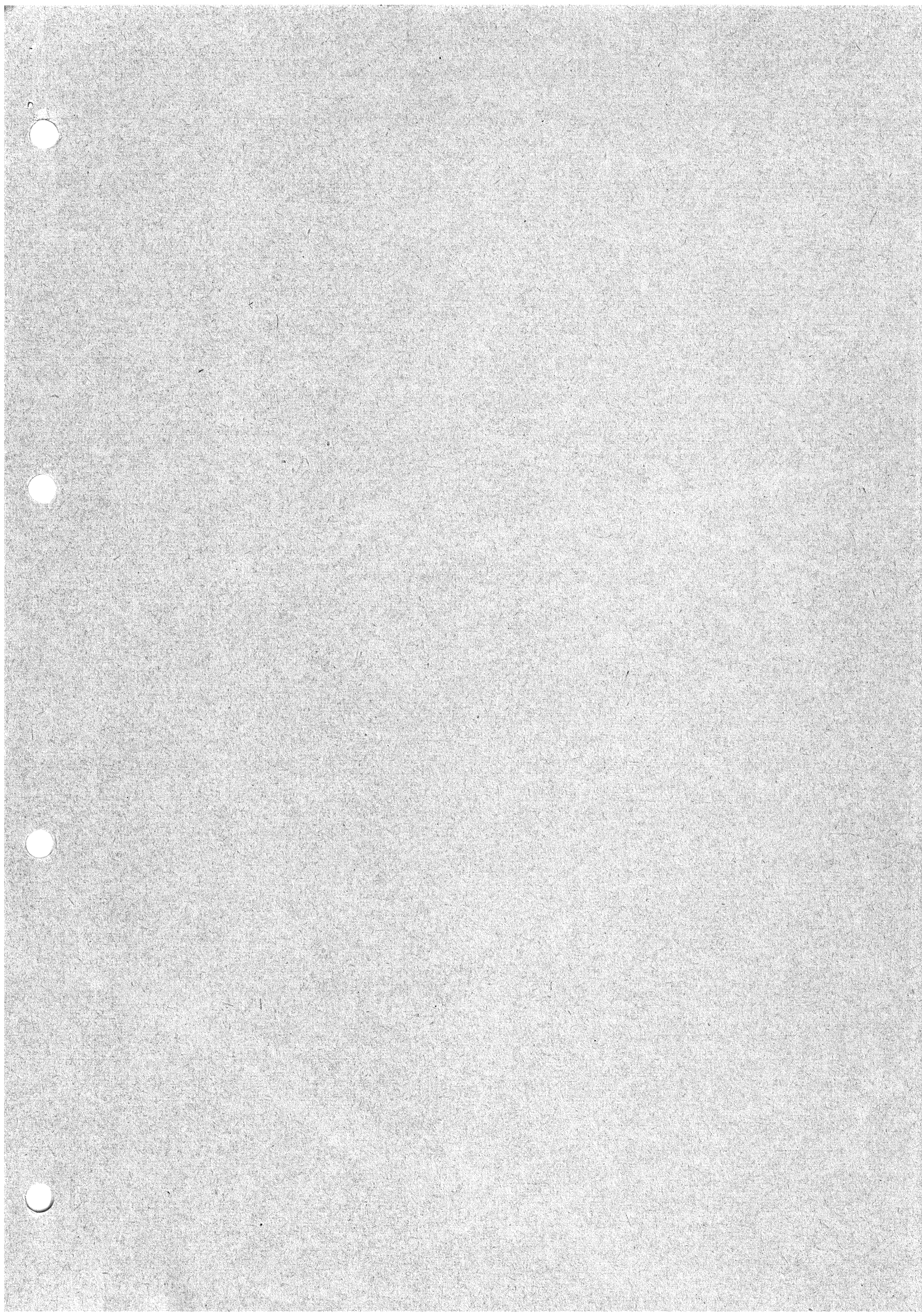
R Blatt 3: Es ist noch MLD hinzuzufügen

SFBE: Ausführungszeit: 0,75 μ s

STN: Bei Sprungbedingung SB muß es heißen: $s_2 \neq 0$
anstelle von $\langle s \rangle_2 \neq 0$

VMO: Auf Blatt 1 muß (nur bei 1. und 2. Auflage) als
vorletzte Zeile eingefügt werden:
Die Indexregister werden jedoch nicht zurückge-
speichert.

VMO: Auf Blatt 2 muß die letzte Zeile lauten:
0,50 μ s, falls $a_{21}=L$



Interne Mitteilung für E 44 Nr. 12/66

Typ: Vorläufige Beschreibung

Thema: Handbuch: Wirkung der TR 440-Befehle
DER LEERE RECHNER

Datum: Juli 1968 (3. Auflage)

Autor(en): Buneß, Dr.Höhenwarter, Pfrang, Sievers

Erläuterungen:

GR/E87
I/12/66
080768

29. Juli 1968

<u>Inhaltsverzeichnis</u>	<u>Seite</u>
1. Erklärung der benutzten Zeichen	1
1.1. Rechenwerksregister	1
1.2. Befehlswerksregister	1
1.3. Variable	1
1.4. Zuweisungen	2
1.5. Inhalt, Bits	3
1.6. Logische Verknüpfungen	3
1.7. Darstellung der Spezifikationen	4
2. Informationsdarstellung	4
2.1. Grundstruktur	4
2.2. Zahlworte	5
2.3. Befehls- und Alphaworte	6
3. Wirkungsweise des Befehlswerks	7
3.1. Allgemeines	7
3.2. Phase 1	7
3.3. Phase 2	7
4. Bemerkungen zu Sprungbefehlen (und Tabellenbefehlen)	8
4.1. Vergleiche zwischen zwei Größen	8
4.2. Sprung in andere Großseite	10
4.3. Ausführungszeit bei erfüllter Bedingung	10
5. Bemerkungen zu den arithmetischen Befehlen	10
5.1. Invertierung	10
5.2. Betrag	10
5.3. Rechenoperationen	11
6. Einzelbeschreibungen der Befehle	11 a
6.1. Erklärung des Formulars	11 a
6.2. Ausführungszeiten	12
7. Internspezifikationen	13

1. Erklärung der benutzten Zeichen

1.1. Rechenwerksregister

A (für RA)	<u>A</u> kkumulator
Q (für RQ)	Multiplikator - <u>Q</u> uotientenregister
D (für RD)	Multiplikandenregister
H (für RH)	<u>H</u> ilfsregister
M (für RM)	<u>M</u> arkenregister
Y (für RY)	Schifftzähler

1.2. Befehlswerksregister

B (für BB)	<u>B</u> ereitadreßregister
F (für BF)	Befehls <u>f</u> olgezähler
K (für BK)	Merk <u>l</u> ichterregister
U (für BU)	<u>U</u> nterprogrammordnungszähler

1.3. Variable

mod1	<u>M</u> odifikationsgröße in Phase <u>1</u> (in keinem Register lokalisierbar)
mod2	<u>M</u> odifikationsgröße in Phase <u>2</u> (in keinem Register lokalisierbar)
ro	<u>R</u> egister <u>o</u> perand (48 Bits, zusätzlich 2 Bits Typenennung)
op	<u>O</u> perationsteil eines Befehls
adr16	<u>A</u> dreßteil eines Befehls im Speicher (<u>16</u> -Bit-Adreßteil)
adr	<u>A</u> dreßteil eines Befehls während der Ausführung (24-Bit-Adreßteil)
adr ist entweder	
n	Adresse einer Speicherzelle (Ganzwort)
n ₂	Adresse einer Speicherzelle (<u>H</u> albwort)
n*	Wahladresse (Ganz- oder Halbwortadresse oder Zahl nach freier Wahl)

oder setzt sich zusammen aus

i	<u>I</u> ndexadresse
p	<u>P</u> arameter
s	<u>S</u> pezifikation
c	Befehls <u>c</u> ode

Bemerkung: i, p, s können zur Unterscheidung mit den Indizes L oder R versehen werden (links, rechts)
s kann außerdem in $s_1 s_2$ aufgeteilt sein.

1.4. Zuweisungen

a) Bitanzahl der Zielgröße \leq Bitanzahl der Quellengröße

:= Die rechtsstehende Quellengröße wird der linksstehenden Zielgröße zugewiesen. Falls die Quellengröße aus mehr Bits besteht als die Zielgröße, werden die linken überzähligen Bits der Quellengröße ignoriert.

b) Bitanzahl beider Größen ist gleich

== Rechtsstehende und linksstehende Größe werden vertauscht.

c) Bitanzahl der Zielgröße $>$ Bitanzahl der Quellengröße

:o= Die rechtsstehende Quellengröße wird der linksstehenden Zielgröße zugewiesen. Dabei wird die Zielgröße links mit 0-Bits aufgefüllt.

:v= Die rechtsstehende Quellengröße wird der linksstehenden Zielgröße zugewiesen. Dabei wird die Zielgröße links mit Bits gleich dem linken Bit (Vorzeichen) der Quellengröße aufgefüllt.

1.5. Inhalt, Bits

$\langle \rangle$	Inhalt eines Registers, einer Speicher- oder Indexzelle
$\langle \rangle_1$	<u>erstes</u> (linkes) Bit
$\langle \rangle_{2,40}$	<u>zweites</u> bis <u>vierzigstes</u> Bit
$\langle n_3 \rangle$	abgekürzt für $\langle n_2 \rangle_{9,24}$ (<u>Drittelwort</u>)
$\langle \rangle_t$	<u>Typen</u> kennung (bei Halbwörtern die TK des Ganzwortes, das das Halbwort enthält)
$\langle n \rangle_m$	<u>Marken</u> bit bei Typenkennung 0 und 1 (gleichbedeutend mit $\langle n \rangle_1$)

Bemerkung: Bit-Angaben erfolgen nicht nur bei Inhalten, sondern auch bei Variablen, ausgenommen n.

1.6. Logische Verknüpfungen

\wedge	logisches Und $0 = 0 \wedge 0$ $0 = 0 \wedge L$ $0 = L \wedge 0$ $L = L \wedge L$
\vee	logisches (einschließliches) Oder $0 = 0 \vee 0$ $L = 0 \vee L$ $L = L \vee 0$ $L = L \vee L$
\neq	Antivalenz (ausschließliches Oder) $0 = 0 \neq 0$ $L = 0 \neq L$ $L = L \neq 0$ $0 = L \neq L$

$\langle \overline{} \rangle$

Negation (Invertierung)

$$L = \overline{O}$$

$$O = \overline{L}$$

1.7. Darstellung der Spezifikationen

()

Die runden Klammern haben nur die Bedeutung des Zusammenfassens.

[]

Die Größen in eckigen Klammern können wahlweise weggelassen werden.

|

trennt Größen, genau eine von ihnen muß ausgewählt werden.

v

trennt Größen, mindestens eine von ihnen muß ausgewählt werden (oder beliebig viel mehr).

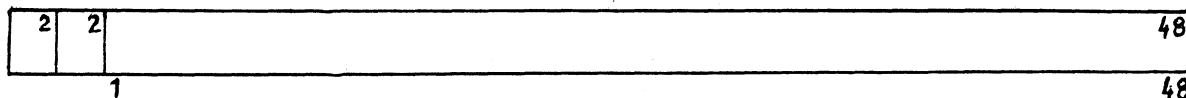
Bemerkung:

Das Zeichen v hat eine verwandte, aber nicht identische Bedeutung mit dem logischen Oder. Verwechslungen können nicht eintreten.

2. Informationsdarstellung

2.1. Grundstruktur

Einheit der Verarbeitung ist ein Ganzwort. Die vier großen Rechenwerksregister A, Q, D, H stellen je ein solches Ganzwort dar, ebenso die Kernspeicherzellen.



Die Gesamtwortlänge von 52 Bits setzt sich zusammen aus:

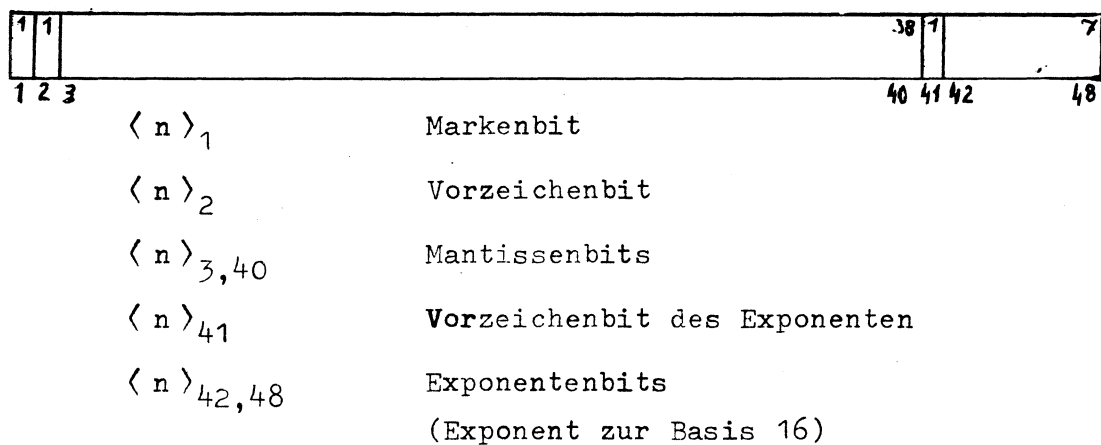
2	Bits	Dreierprobe
2	Bits	Typenkennung
48	Bits	eigentliche Information, je nach Typenkennung

$\langle \rangle_t = 0$	Gleitkommazahl
$\langle \rangle_t = 1$	Festkommazahl
$\langle \rangle_t = 2$	Befehlswort
$\langle \rangle_t = 3$	Alphawort

2.2. Zahlworte

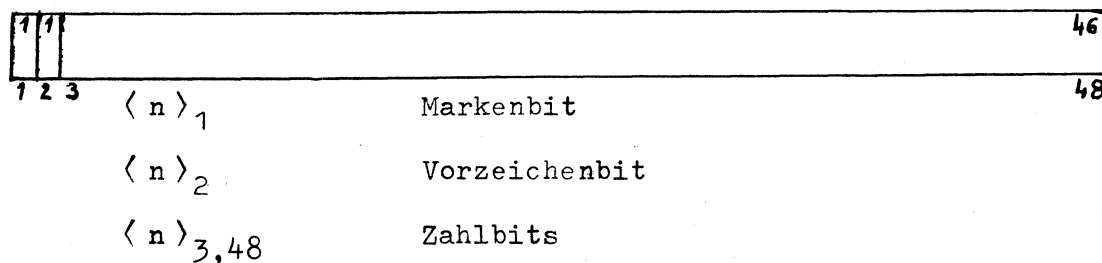
a) Im Speicher:

1. Gleitkommazahl:



Das Komma ist zwischen $\langle n \rangle_2$ und $\langle n \rangle_3$ anzunehmen.

2. Festkommazahl:



linksbündige Festkommazahl:

das Komma ist zwischen

$\langle n \rangle_2$ und $\langle n \rangle_3$ anzunehmen.

rechtsbündige Festkommazahl:

das Komma ist nach

$\langle n \rangle_{48}$ anzunehmen.

b) Im Register:

Bei Typenkennung 0 und 1 gilt bei Transport aus dem Speicher in ein Register:

Die Marke wird abgetrennt und

$$\langle \rangle_1 := \langle \rangle_2$$

c) Doppelt genaue Gleitkommazahl:

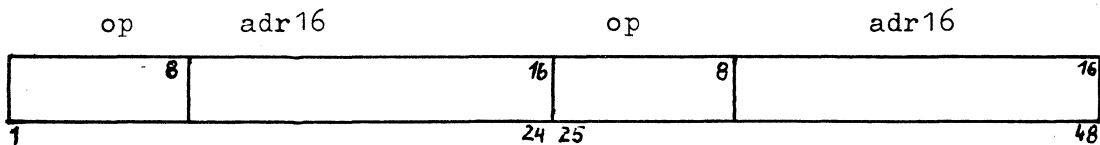
Sie besteht aus einer Gleitkommazahl und einer Festkommazahl, die die Fortsetzung der Mantisse enthält. Beide haben das gleiche Vorzeichen.

d) Doppelt genaue Festkommazahl:

Sie besteht aus zwei Festkommazahlen, jede mit Vorzeichen und TK1.

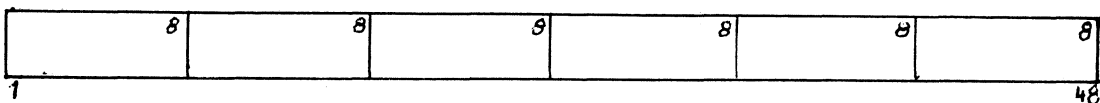
2.3. Befehls- und Alphaworte

1. Ein Befehlsganzwort im Speicher besteht aus zwei Befehlshalbworten zu je 8 + 16 Bit:



Es wird immer ein Ganzwort aus dem Speicher ausgelesen und in das Befehlsregister eingeschrieben. Steht ein Befehl in einer Halbzelle mit einer geraden Adresse n, so ist es daher unmöglich, durch die Ausführung dieses Befehls den Befehl in n+1 in seiner Ausführung zu ändern.

2. Ein Alphawort kann aus sechs Oktaden bestehen.



Ebenso kann es als positive 48-stellige Dualzahl interpretiert werden.

3. Wirkungsweise des Befehlswerks

3.1. Allgemeines

Als Übergangsgrößen von einem Befehl zum nächsten dienen die Variablen mod1 und mod2, beide mit 24 Bit, außerdem op und adr, sowie ro (beim Register - Befehl).

Die Ausführung jedes Befehls läßt sich in zwei Teile gliedern, Phase 1 und Phase 2. Phase 1 läuft bei jedem Befehl gleich ab, oder wird übersprungen, falls der unmittelbar vorher ausgeführte Befehl den Ausgang 2 hat (siehe unten). Daher genügt es, sich bei den Einzelbeschreibungen auf Phase 2 zu beschränken.

3.2. Phase 1

Wirkung: 1. $adr := adr_{16} + mod1$
2. $mod1 := 0$

Damit ist aus dem 16-Bit-Adreßteil im Speicher ein 24-Bit-Adreßteil der Befehlsausführung geworden.

3.3. Phase 2

1. Nach Phase 1 wird der Befehl interpretiert als:

op adr

Die Wirkung ist bei den Einzelbeschreibungen nachzulesen.

2. Definition: $\langle F \rangle$ = Adresse der Halbzelle, aus der der gerade bearbeitete Befehl ausgelesen wurde.

3. Der Übergang von einem Befehl zum nächsten wird durch den jeweiligen Ausgang bestimmt.

Wenn kein Ausgang angegeben ist, hat der Befehl Ausgang 1.

Ausgang 1 (normaler Ausgang):

mod2 := 0
(falls nicht neu definiert)
 $\langle F \rangle := \langle F \rangle + 1$
(falls nicht neu gesetzt)
Sprung auf Phase 1 des nächsten Befehls.

Ausgang 2:

$\langle F \rangle := \langle F \rangle$
Sprung auf Phase 2 des definierten Befehls.

4. Bemerkungen zu Sprungbefehlen (und Tabellenbefehlen)

4.1. Vergleiche zwischen zwei Größen

Verglichen werden die Inhalte von A und H, (bei Tabellenbefehlen (ohne Maske) die von A und D). Die Art des Vergleichs wird von der höheren der beiden Typenkennungen bestimmt.

Fall A: $\langle A \rangle_t = \langle H \rangle_t = 0$

$\langle A \rangle_1 = \langle A \rangle_2, \langle H \rangle_1 = \langle H \rangle_2$

Gleichheit, falls:

$\langle A \rangle_i = \langle H \rangle_i$

$i = 1, 2, \dots, 40$

und falls

$\langle A \rangle_i = \langle H \rangle_i$

$i = 41, \dots, 48$

Gleitkommazahlen werden
in normalisierter Form
verglichen.
(Siehe Befehl NRM G)

Mantisse

Exponent

oder ein Exponent = + 0
und der andere = - 0

Bemerkung:

falls $\langle A \rangle_1 \neq \langle A \rangle_2$
oder $\langle H \rangle_1 \neq \langle H \rangle_2$

werden Vergleiche im
allgemeinen fehlerhaft
ausgeführt.

Fall B: $\text{Max } (\langle A \rangle_t, \langle H \rangle_t) = 1$

Gleichheit, falls:

$$\langle A \rangle_i = \langle H \rangle_i \\ i = 1, 2, \dots, 48$$

oder eine Größe = + 0
und die andere = - 0

Bemerkung:

falls $\langle A \rangle_1 \neq \langle A \rangle_2$
oder $\langle H \rangle_1 \neq \langle H \rangle_2$
gelten $\langle A \rangle_1$ und $\langle H \rangle_1$
als Vorzeichenbits.

Fall C: $\text{Max } (\langle A \rangle_t, \langle H \rangle_t) \geq 2$

Gleichheit genau dann,
wenn: $\langle A \rangle_i = \langle H \rangle_i$
 $i = 1, 2, \dots, 48$

über- oder untergelaufene
Gleitkommazahl

Beide Größen werden als
Festkommazahl interpretiert.

über- oder untergelaufene
Zahl

Beide Größen werden als
positive Dualzahlen von
48 Bit interpretiert.

4.2. Sprung in andere Großseite

Wird ein Sprungbefehl durch die Befehle MABI oder MU definiert, so folgt:

$$\langle F \rangle := n_2$$

Der ganze 24-Bit-Adreßteil wird in den Befehlsfolgezähler übertragen.

Wird ein Sprungbefehl nicht durch MABI oder MU definiert, so gilt das in den Einzelbeschreibungen Gesagte. (Es wird nicht aus der Großseite herausgesprungen)

4.3. Ausführungszeit bei erfüllter Bedingung

Bei einem Sprungbefehl mit erfüllter Sprungbedingung und unbedingten Sprungbefehlen sind 0,3 μ s zur Ausführungszeit der Phase 1 des nächsten Befehls hinzuzurechnen.

5. Bemerkungen zu den arithmetischen Befehlen

5.1. Invertierung

Definition: $\langle \rangle_i := \overline{\langle \rangle_i}$

wobei $1 \leq i \leq 48$

5.2. Betrag

Fall A: $\langle \rangle_t = 0$

$|\langle \rangle_{1,40}| := \langle \rangle_{1,40}$ falls $\langle \rangle_1 = \langle \rangle_2 = 0$

$|\langle \rangle_{1,40}| := \overline{\langle \rangle_{1,40}}$ falls $\langle \rangle_1 = \langle \rangle_2 = 1$

Gleitkomma

Zahl ist positiv

Zahl war negativ

Fall B: $\langle \rangle_t = 1$

$|\langle \rangle_{1,48}| := \langle \rangle_{1,48}$ falls $\langle \rangle_1 = \langle \rangle_2 = 0$

$|\langle \rangle_{1,48}| := \overline{\langle \rangle}_{1,48}$ falls $\langle \rangle_1 = \langle \rangle_2 = L$

Fall C: $\langle \rangle_t \geq 2$

$|\langle \rangle_{1,48}| := \langle \rangle_{1,48}$

Festkomma

Zahl ist positiv

Zahl war negativ

Befehls- und

α -Worte

gelten als

positiv

5.3. Rechenoperationen

Negative Zahlen werden im (B-1)-Komplement dargestellt. Bei Addition und Subtraktion wird der links eventuell entstehende Übertrag auf das rechte Bit des Ergebnisses addiert (Einerrücklauf).

Da die Subtraktion im (B-1)-Komplement erfolgt, ist $-\langle \rangle$ gleichbedeutend mit $\overline{\langle \rangle}$. Bei Gleitkommaoperationen bezieht sich das -Zeichen nur auf die Mantisse.

Entsteht bei der Verknüpfung zweier m-stelliger Zahlen ein Ergebnis mit mehr als m Stellen, so wird es betragsmäßig gerundet, falls ein nur m-stelliges Ergebnis verlangt wird. Dies geschieht, indem man auf die (m+1)te Stelle des Betrags ein L addiert und dann nach der m-ten Stelle abschneidet. Speziell bei Addition und Subtraktion zweier normalisierter Gleitkommazahlen geschieht folgendes:

Sind die Exponenten der beiden Operanden ungleich, so wird der Operand mit dem kleineren Exponenten entsprechend der Exponentendifferenz denormalisiert, wobei man sich vorstellen soll, daß keine Stellen verloren gehen. Die Mantisse des Operanden mit dem größeren Exponenten wird mit vorzeichengleichen Bits auf die selbe Anzahl von Bits rechts verlängert. Nun wird addiert mit eventuellem Einerrücklauf.

a) Ist das Ergebnis nicht übergelaufen und bereits normalisiert, und eine betragsmäßige Rundung in der 40. Stelle (durch Addition eines L auf die 41. Stelle des Betrages) würde zu keinem Überlauf führen, so wird das Ergebnis in der 40. Stelle betragsmäßig gerundet.

- b) Ist das Ergebnis übergelaufen oder würde die betragsmäßige Rundung zu Überlauf führen, so wird es um eine Tetrade denormalisiert (Exponentenüberlauf möglich) und dann in der 40. Stelle betragsmäßig gerundet.
- c) Ist das Ergebnis normalisierbar (folglich auch nicht übergelaufen, Rundung kann keinen Überlauf ergeben), so wird es normalisiert und dann in der 40. Stelle betragsmäßig gerundet. Die Normalisierung wird jedoch nur soweit ausgeführt, als die anschließende Rundung nicht zu Überlauf führt. Bei der Normalisierung werden, falls vorhanden, berechnete Stellen, sonst vorzeichen-gleiche, nachgezogen.

Nach der 40. Stelle wird die Mantisse abgeschnitten.

Bei unnormalisierten Operanden kann die Rundung entfallen, außerdem können Mantissenstellen verloren gehen.

6. Einzelbeschreibungen der Befehle

6.1. Erklärung des Formulars

Auf den Beschreibungsformularen sind die Wirkungen jedes einzelnen Befehls des TR 440 in Phase 2 angegeben.

Dazu muß noch folgendes erklärt werden: Es bedeuten die Kästchen

R : Zulassung als Zweitcode beim Befehl R.

Falls +, dann erlaubt.

BW: Belegung des Befehlswerks durch den Befehl.

Sinngemäß wie bei RW.

RW: Belegung des Rechenwerks durch den Befehl. Ein + bedeutet, daß das Rechenwerk überhaupt angesprochen wird. Falls leicht möglich, werden genau die Register angegeben, die verändert werden.

Zur Verdeutlichung dient bei vielen Befehlen eine Kommentierung auf der rechten Blattseite. Sie ist von der Beschreibung der Wirkungen durch einen senkrechten Strich getrennt.

6.2. Ausführungszeiten

Phase 1 dauert im Mittel $0,4 \mu\text{s}$. Falls einer der Befehle MF, MCF vorangeht, sind $0,4 \mu\text{s}$ hinzuzurechnen, falls einer der Befehle MFU, MCFU, MD vorangeht, sind $0,5 \mu\text{s}$ hinzuzurechnen. Die angegebenen Ausführungszeiten beziehen sich auf Phase 2.

Falls ein Modifizierbefehl 2. Art vorangeht und als erste Wirkung die Addition von mod2 angegeben ist, sind der Ausführungszeit des nächsten oder des neu definierten Befehls $0,4 \mu\text{s}$ hinzuzurechnen.

Falls der Befehl zu Beginn einen Operanden aus dem Speicher ausliest, sind i.a. noch $0,4 \mu\text{s}$ zur Ausführungszeit hinzuzurechnen.

Belegt ein Befehl in Phase 2 nur das Rechenwerk, ist das Befehlswerk frei für die Operationen die ausschließlich im Befehlswerk ablaufen. Das sind: Phase 1 des nächsten Befehls und auch Phase 2, falls dabei nur das Befehlswerk belegt wird.

7. Internspezifikationen

m_1, \dots, m_8 bedeuten die Bits des Spezifikationsteils. Steht unter einem der m_i ein - , so ist m_i ohne Bedeutung.

IR

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
A	Q	D	H	B	-	-	-

Es sind alle 2^8 Bitanordnungen erlaubt.

Falls $m_1=m_2=m_3=m_4=0$, Wirkung wie Nullbefehl.

LA

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
F	2	E	3	H	T	V	M

Erlaubt ist nur $s = [V \vee 2 \vee F \vee 3 \vee E \vee M] \parallel H[M] \parallel T[M]$

Nicht erlaubte Spezifikationen ergeben undefinierte Befehlsausführung.

LR, ST, STN

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
s_1	A	Q	D	H	-	-	-

s_1	m_1	m_2
0	C	C
1	C	L
2	L	C
3	L	L

Bei LR sind alle 2^8 möglichen Bitanordnungen erlaubt.

Bei ST, STN darf von den Bits m_3, \dots, m_6 höchstens eines = L sein, sonst undefinierte Befehlsausführung; außerdem muß gelten $m_7=0$, andernfalls wird immer gesprungen.

LZL, NL, SL, SLL, SLN, SNL, SW

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
1	2	3	4	5	6	7	8

Es sind alle 2^8 möglichen Bitanordnungen erlaubt.

Bei LZL hat sowohl s_L als auch s_R diese Gestalt.

MRX, RX

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
A	Q	D	H	B	N	*	C

* wenn $m_7 = 0$: Befehl RX

wenn $m_7 = 1$: Befehl MRX

Von den Bits m_1, \dots, m_5 darf höchstens eines = L sein, sonst undefinierte Befehlsausführung.

NRM

	8	4	2	1	8	4	2	1
s	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
N	O	O	O	-	-	-	-	-
L	O	O	L	-	-	-	-	-
F	O	L	O	-	-	-	-	-
F4	O	L	L	-	-	-	-	-
G	L	O	O	-	-	-	-	-
*	L	O	L	-	-	-	-	-
FG	L	L	O	-	-	-	-	-
*	L	L	L	-	-	-	-	-

* Diese Bitanordnungen führen zu undefinierter Befehlsausführung.

R, RLR

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
A	Q	D	H	\overline{m}_5	-	-	\overline{L}
F	U	B	Y	m_5	-	-	\overline{L}

Genau eines der Bits m_1, \dots, m_4 muß = L sein, sonst undefinierte Befehlsausführung.

(Ausnahme: ist der definierte Befehl c der Nullbefehl, ist $m_1=m_2=m_3=m_4 = 0$ erlaubt.)

RT

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
A	Q	D	H	-	-	-	-

Genau zwei der Bits $m_1 \dots m_4$ müssen = L sein, sonst undefinierte Befehlsausführung.

SBIT

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
s_2				s_1			

$$s_1 = 32m_3 + 16m_4 + 8m_5 + 4m_6 + 2m_7 + m_8$$

s_2	m_1	m_2
A	0	0
Q	0	L
D	L	0
H	L	L

SH

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
A	Q	L	K	Z	R	U	B

Es sind alle 2^8 möglichen Bitanordnungen erlaubt.

Wenn $m_5 = L$, sind m_1 und m_2 bedeutungslos.

Wenn $m_4 = L$, ist m_6 bedeutungslos.

Wenn $m_3 = L$ und $m_5 = 0$, ist m_6 bedeutungslos.

Wenn $m_1 = m_2 = m_5 = 0$, sind m_3, m_4, m_6, m_7 bedeutungslos.

SHB

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
L	-	-	-	-	-	-	-

Es sind alle 2^8 Bitanordnungen erlaubt.

Wenn $m_1 = 0$, dann Spezifikation R

TRX, TXR

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
A	Q	D	H	N	-	-	-

TRX: Von den Bits $m_1 \dots m_4$ darf höchstens eines = L sein, sonst undefinierte Befehlsausführung.

TXR: Es sind alle 2^8 Bitanordnungen erlaubt.

US

8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
6	8	C	G	-	-	-	-

Von den Bits m_1, m_2, m_3 muß genau eines = L sein, sonst undefinierte Befehlsausführung.

Wenn $m_4 = 0$, dann Teilspezifikation E.

ZTR

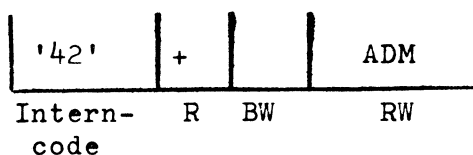
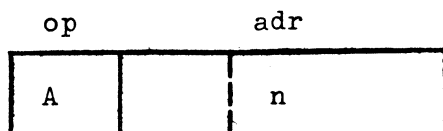
8	4	2	1	8	4	2	1
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
s_1	A	Q	D	H	M	-	-

s_1	m_1	m_2
0	0	0
1	0	L
2	L	0
3	L	L

Von den Bits m_3, \dots, m_6 darf höchstens eines = L sein, sonst undefinierte Befehlsausführung.

A

ADDIERE



Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W3. $\langle A \rangle := \langle A \rangle + \langle n \rangle$

$\langle A \rangle_t := \max(\langle A \rangle_t, \langle n \rangle_t)$

W4. falls $\langle n \rangle_t \leq 1$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

falls $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$:

A-Alarm

Marke bei Zahl-
wort

Ergebnis überge-
laufen

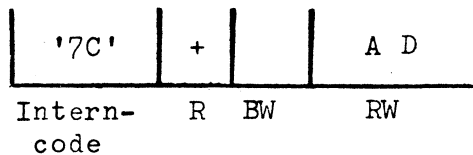
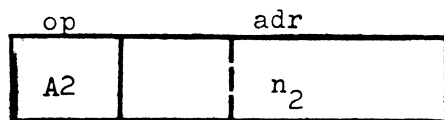
Bemerkung: Die Addition geschieht wie bei Tk 1 (modulo $2^{48}-1$).
Die Vorzeichenstellen der Operanden können ungleich
sein.

$\langle A \rangle := +0$ nur wenn beide
Operanden = +0

Ausführungszeit: 0,40 μs

A2

ADDIERE HALBWORT



Wirkung:

W1. $n_2 := n_2 + \text{mod}2$

W2. $\langle A \rangle_{25,48} := \langle A \rangle_{25,48} + \langle n_2 \rangle$

W3. $\langle A \rangle_{1,24} := \langle A \rangle_{25}$

W4. $\langle D \rangle := +0$

$\langle D \rangle_t := \langle A \rangle_t$

Falls $\langle A \rangle_{25} \neq \langle A \rangle_{26}$:

W5. A-Alarm

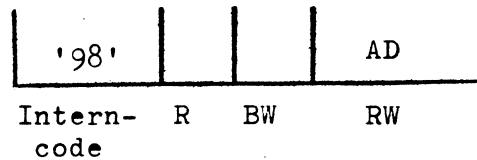
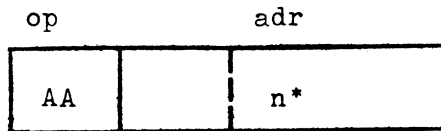
Die Addition geschieht
modulo $2^{24}-1$

$\langle A \rangle_{25}$ und $\langle A \rangle_{26}$ werden
als Vorzeichenstellen
aufgefaßt

Ausführungszeit: 0,70 μ s

ADDIERE ADRESSTEIL

AA



Wirkung: W1. $n^* := n^* + \text{mod}2$

Variante 1: $\langle A \rangle_t = 0$

W2. $\langle D \rangle := \text{rechte 8 Bits von } n^*$
 $\langle D \rangle_t := 1$

W3. Die rechten 8 Bits von n^* werden zu $\langle A \rangle_{41,48}$ addiert. Der Einerrücklauf erfolgt von $\langle A \rangle_{41}$ zu $\langle A \rangle_{48}$

Fall A: $-127 \leq \text{Ergebnis} \leq +127$

W4. entfällt

Fall B: $\text{Ergebnis} \geq 128$

W4. A-Alarm

Gleitkomma

andere Bits von n^* ohne Bedeutung
Exponent in A

Exponent über-
gelaufen

Exponent um
255 zu klein

Fall C: Ergebnis ≤ -128

W4. $\langle A \rangle_{1,40} := -0$
 $\langle A \rangle_{41,48} := -127$

Exponent unter-
gelaufen

normalisierte
Gleitkommanull

Variante 2: $\langle A \rangle_t = 1 \text{ oder } 3$

Festkomma,
Oktaden

W2. $\langle D \rangle :v= n^*$
 $\langle D \rangle_t := 1$

W3. Die 24 Bits von n^* werden durch
Vorzeichenangleich zu einer 48-
Bit-Größe ergänzt und zu den
48 Bit von $\langle A \rangle$ addiert. Der Einer-
rücklauf erfolgt von $\langle A \rangle_1$ zu $\langle A \rangle_{48}$.
 $\langle A \rangle := \text{Ergebnis}$
 Falls $\langle A \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$: A-Alarm

Variante 3: $\langle A \rangle_t = 2$

Befehle

W2. $\langle D \rangle := \text{rechte 16 Bits von } n^*$
 $\langle D \rangle_t := 1$

W3. Die rechten 16 Bits von n^* werden
zu den rechten 16 Bits von $\langle A \rangle$
addiert. Der Einerrücklauf erfolgt
von $\langle A \rangle_{33}$ zu $\langle A \rangle_{48}$. $\langle A \rangle_{1,32}$ blei-
ben unverändert.

linke 8 Bits von
 n^* ohne Bedeutung

Bemerkung: +0 als Ergebnis in allen Varianten nur
dann, wenn beide Operanden +0 sind.

Ausführungszeit: 0,65 μs

AB

ADDIERE BETRAG

op	adr
AB	n

'40'	+		ADM
Intern- code	R	BW	RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. $\langle A \rangle := \langle A \rangle + |\langle n \rangle|$
 $\langle A \rangle_t := \max(\langle A \rangle_t, \langle n \rangle_t)$

W4. falls $\langle n \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

falls $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$:
 A-Alarm

Marke bei Zahl-
worten

Ergebnis überge-
laufen

Bemerkung: Die Addition geschieht wie bei Tk 1 (modulo $2^{48}-1$).
 Die Vorzeichenstellen der Operanden können ungleich sein.

Falls $\langle n \rangle_t \geq 2$, gilt $|\langle n \rangle| = \langle n \rangle$

Falls $\langle n \rangle_t = 0$, geschieht die Betragsbildung wie bei Tk 1.

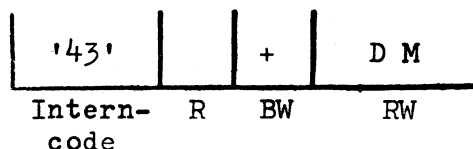
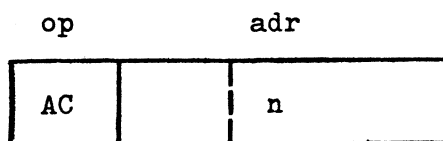
$\langle A \rangle := +0$ nur wenn vor der Addition

$\langle A \rangle = +0$ war, $|\langle n \rangle| = +0$

Ausführungszeit: 0,40 μs

AC

ADDIERE IM SPEICHER



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle A \rangle + \langle n \rangle$
 $\langle D \rangle_t := \max(\langle A \rangle_t, \langle n \rangle_t)$
 falls $n_t \leq 1$ zusätzlich:
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
worten

Fall A: $\langle D \rangle_t \geq 2$

W3. $\langle n \rangle := \langle D \rangle$
 $\langle n \rangle_t := \langle D \rangle_t$

Fall B: $\langle D \rangle_t \leq 1$ und $\langle D \rangle_1 = \langle D \rangle_2$

W3. $\langle n \rangle_1 := \langle n \rangle_1$
 $\langle n \rangle_{2,48} := \langle D \rangle_{2,48}$
 $\langle n \rangle_t := \langle D \rangle_t$

Marke im Speicher
bleibt

Fall C: $\langle D \rangle_t \leq 1$ und $\langle D \rangle_1 \neq \langle D \rangle_2$

W3. A-Alarm

Ergebnis über-
gelaufen

Bemerkung: Die Addition geschieht wie bei Tk 1 (modulo $2^{48}-1$).
 Die Vorzeichenstellen der Operanden können ungleich
 sein.

In den Fällen A und B gilt:

$\langle n \rangle := +0$ nur wenn beide Operanden = +0

Ausführungszeit: 0,85 μ s

GR/EP1
 I/12/66
 201267

29. Juli 1968

ADDIERE IN A,Q

op	adr
AQ	n

'7E'	+		AQDM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$:

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Fall A: $\langle A \rangle_t = \langle Q \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$
und $\langle Q \rangle_1 = \langle Q \rangle_2$

W4. $\langle A, Q \rangle := \langle A, Q \rangle + \langle n \rangle$

W5. falls $\langle A \rangle_1 \neq \langle A \rangle_2$:

A-Alarm

Fall B: $\langle A \rangle_t = \langle Q \rangle_t = \langle n \rangle_t = 1$ und $(\langle A \rangle_1 \neq \langle A \rangle_2$
oder $\langle Q \rangle_1 \neq \langle Q \rangle_2$)

W4. $\langle A, Q \rangle := \text{undefiniert}$

W5. falls $\langle A \rangle_1 \neq \langle A \rangle_2$:

A-Alarm

Marke bei Zahlwörtern

Normalfall

Der Einerrücklauf erfolgt
von $\langle A \rangle_1$ zu $\langle Q \rangle_{48}$

Ergebnis übergelaufen

$\langle A \rangle$ oder $\langle Q \rangle$ übergelaufen

Fall C: $\langle A \rangle_t$ oder $\langle Q \rangle_t$ oder $\langle n \rangle_t \neq 1$

falsche Tk

W4. Tk-Alarm

Bemerkung: $\langle A \rangle$ und $\langle Q \rangle$ können verschiedene Vorzeichen haben.

Ausführungszeit: Fall A,B: 0,85 μ s
Fall C: 0,30 μ s

AT

ADDIERE TEILWORT

op	adr
AT	n

'F4'	+		A D
Intern- code	R	BW	RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle Q \rangle$

$\langle D \rangle_t := \langle Q \rangle_t$

W3. Es wird eine Hilfsgröße qr gebildet:

qr := um p Stellen im Kreis nach rechts geschiftete $\langle Q \rangle$, wobei p = Anzahl der in Q rechts anstehenden L-Bits. Falls jedoch $\langle Q \rangle = \text{LLLL..L}$, dann p = 0

W4. Es wird eine Hilfsgröße nr gebildet:

nr := um p Stellen nach rechts geschiftete $\langle n \rangle$

W5. $nr_i := 0$ falls $qr_i = L$

$\langle A \rangle_i := L$ falls $qr_i = L$

W6. Die Operanden $\langle A \rangle$ und nr werden als positive, vorzeichenlose ganze Festkommazahlen aufgefaßt.

Q	LLLL0000	LLLLLLLL	0000000000	LLLLLL
n	0000	n ₁ n ₂	00000000	n ₂₋₁ n ₂ n ₂₊₁
qr	LLLLLL	LLLL	0000	LLLLLLLL
nr	0000000000	n ₁ n ₂	00000000	n ₂₋₁ n ₂ n ₂₊₁
A	LLLLLLLLLL	AAAA	LLLLLLLL	AAAAAAAAAA
A	0000000000	EEEE	00000000	EEEEEEEEEE

Falls $\langle A \rangle + nr < 2^{48}$:

$\langle A \rangle := \langle A \rangle + nr$

Falls $\langle A \rangle + nr \geq 2^{48}$:

$\langle A \rangle := \langle A \rangle + nr - 2^{48}$

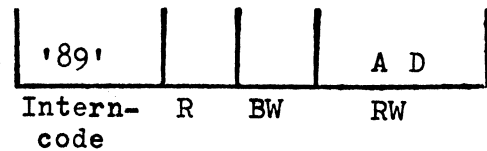
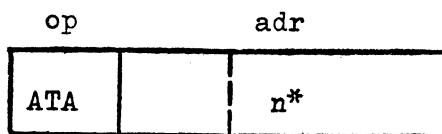
A-Alarm

W7. $\langle A \rangle_1 = 0$ falls $qr_1 = L$

Ausführungszeit: (1,05 + 0,10 p) μs

ATA

ATA ADRESSTEIL



Wirkung: W1. $n^* := n^* + \text{mod}2$

W2. $\langle A \rangle := \langle H \rangle \div n^*$
 $\langle A \rangle_t := \langle H \rangle_t$

W3. $\langle D \rangle := n^*$
 $\langle D \rangle_t := 1$

Ausführungszeit: 0,40 μ s

ADDIERE UNNORMALISIERT

op	adr
AU	n

'49'	+		AQDYM
Intern- code	R	BW	RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A:

$\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := \langle A \rangle + \langle n \rangle$; Ergebnis nicht
gerundet, nicht
normalisiert

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle := +0$

W7. falls Exponent des Ergebnisses
übergelaufen:

A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unterge-
laufener Operand

W4. $\langle A \rangle := \text{undefiniert}$

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle := \text{undefiniert}$
 evtl. A-Alarm

Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

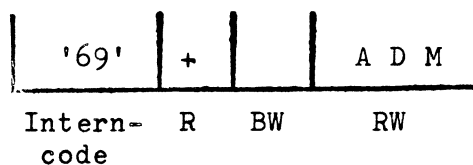
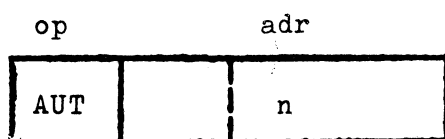
falsche Tk

W4. Tk-Alarm

Ausführungszeit: Fall A: $1,3 \mu s$
 Fall B: $1,3 \mu s$
 Fall C: $0,1 \mu s$

AUT

AUT



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle n \rangle_t \geq 2$

W2. $\langle A \rangle := \langle A \rangle \neq \langle n \rangle$
 $\langle A \rangle_t := \text{Max} (\langle A \rangle_t, \langle n \rangle_t)$

W3. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

Fall B: $\langle n \rangle_t \leq 1$

W2. $\langle A \rangle_1 := \langle A \rangle_1 \neq \langle n \rangle_2$
 $\langle A \rangle_k := \langle A \rangle_k \neq \langle n \rangle_k$
 $2 \leq k \leq 48$
 $\langle A \rangle_t := \text{Max} (\langle A \rangle_t, \langle n \rangle_t)$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W3. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

Zahlwort

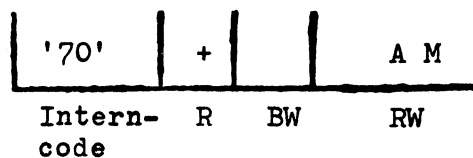
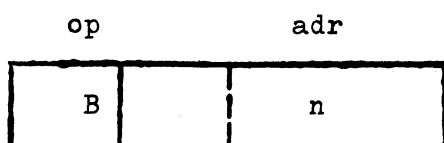
Markenstelle

Marke

Ausführungszeit: 0,25 μs

B

BRINGE (NACH A)



Wirkung:

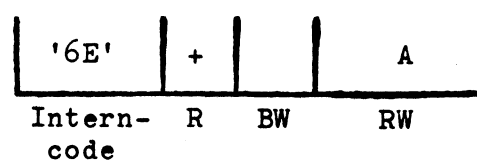
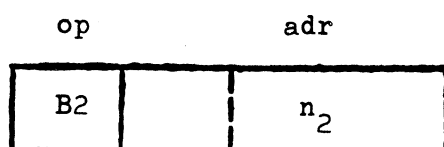
- W1. $n := n + \text{mod} 2$
- W2. $\langle A \rangle := \langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle_t$
- W3. falls $\langle n \rangle_t \leq 1$
 $\langle A \rangle_1 := \langle A \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Zahlwort
Vorzeichenangleich
Marke

Ausführungszeit: 0,10 μ s

B2

BRINGE HALBWORT

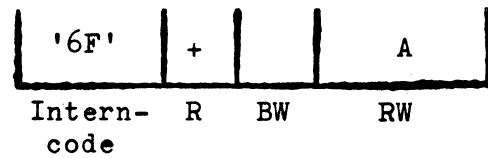
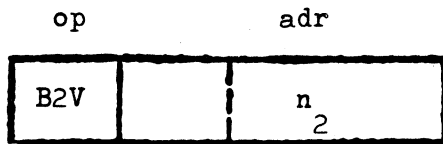


Wirkung: W1. $n_2 := n_2 + \text{mod}2$
W2. $\langle A \rangle := \langle n_2 \rangle$
 $\langle A \rangle_t := \langle n_2 \rangle_t$

Ausführungszeit: 0,45 μs

B2V

BRINGE HALBWORT MIT VORZEICHEN

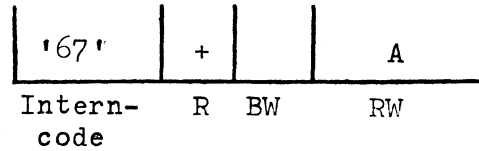
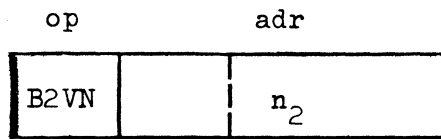


Wirkung: W1. $n_2 := n_2 + \text{mod}2$
W2. $\langle A \rangle := \langle n_2 \rangle$
 $\langle A \rangle_t := 1$

Ausführungszeit: 0,50 μs

B2VN

BRINGE HALBWORT MIT VORZEICHEN NEGATIV



Wirkung:

W1. $n_2 := n_2 + \text{mod}2$

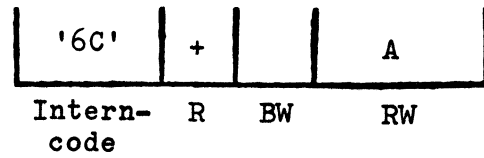
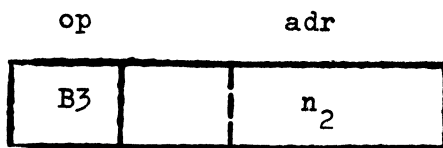
W2. $\langle A \rangle : v = \overline{\langle n_2 \rangle}$

$\langle A \rangle_t := 1$

Ausführungszeit: 0,60 μ s

B3

BRINGE DRITTELWORT

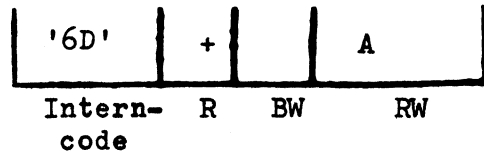
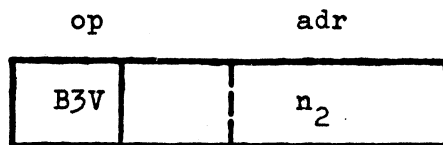


Wirkung: W1. $n_2 := n_2 + \text{mod}2$
W2. $\langle A \rangle := \langle n_3 \rangle$
 $\langle A \rangle_t := 1$

Ausführungszeit: 0,45 μs

B3V

BRINGE DRITTELWORT MIT VORZEICHEN

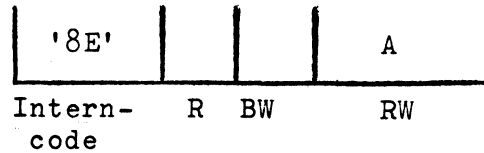
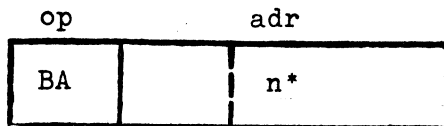


Wirkung: W1. $n_2 := n_2 + \text{mod}2$
W2. $\langle A \rangle := \langle n_3 \rangle$
 $\langle A \rangle_t := 1$

Ausführungszeit: 0,50 μ s

BA

BRINGE ADRESSTEIL



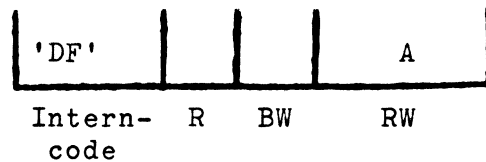
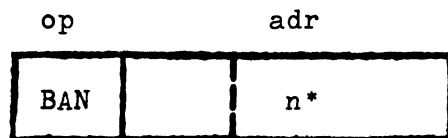
Wirkung: W1. $n^* := n^* + \text{mod}2$

W2. $\langle A \rangle : v = n^*$
 $\langle A \rangle_t := 1$

Ausführungszeit: 0,10 μ s

BAN

BRINGE ADRESSTEIL NEGATIV



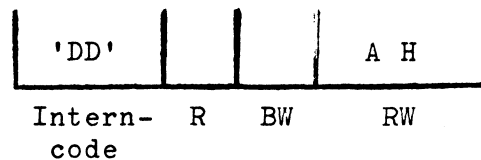
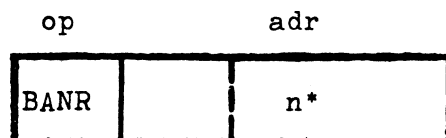
Wirkung: W1. $n^* := n^* + \text{mod}2$

W2. $\langle A \rangle := -n^*$
 $\langle A \rangle_t := 1$

Ausführungszeit: 0,25 μ s

BANR

BRINGE ADRESSTEIL NEGATIV UND RESERVIERE



Wirkung: W1. $n^* := n^* + \text{mod}2$

W2. $\langle H \rangle := \langle A \rangle$
 $\langle H \rangle_t := \langle A \rangle_t$

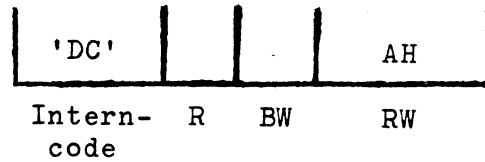
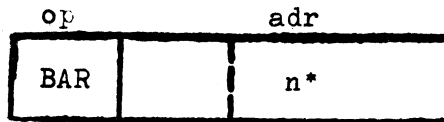
W3. $\langle A \rangle := -n^*$
 $\langle A \rangle_t := 1$

Reservieren des $\langle A \rangle$

Ausführungszeit: 0,25 μ s

BAR

BRINGE ADRESSTEIL UND RESERVIERE



Wirkung: W1. $n^* := n^* + \text{mod}2$

W2. $\langle H \rangle := \langle A \rangle$
 $\langle H \rangle_t := \langle A \rangle_t$

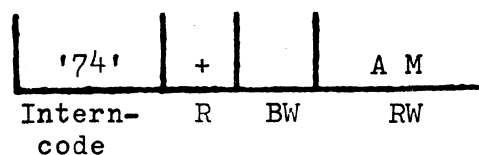
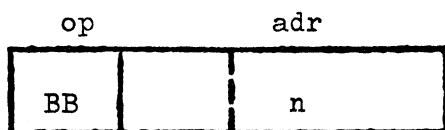
W3. $\langle A \rangle := n^*$
 $\langle A \rangle_t := 1$

reservieren des $\langle A \rangle$

Ausführungszeit: 0,15 μ s

BB

BRINGE BETRAG



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle n \rangle_t \geq 2$

W2. $\langle A \rangle := \langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle_t$

Fall B: $\langle n \rangle_t \leq 1$ und $\langle n \rangle_2 = 0$

W2. $\langle A \rangle := \langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

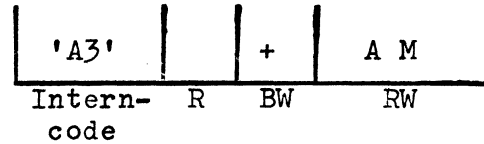
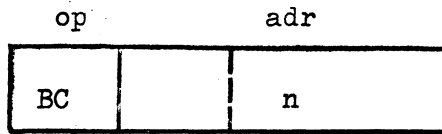
Fall C: $\langle n \rangle_t \leq 1$ und $\langle n \rangle_2 = L$

W2. $\langle A \rangle := -\langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle_t$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

$\langle n \rangle$ positiv

$\langle n \rangle$ negativ

Ausführungszeit: 0,25 μ s

BRINGE UND SPEICHERE

Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle A \rangle_t \geq 2$ oder $\langle n \rangle_t \geq 2$

W2. Ausführung B-Befehl und
gleichzeitig C-Befehl ab W2.

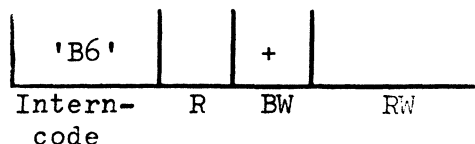
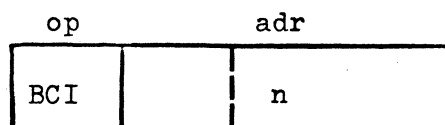
Fall B: $\langle A \rangle_t \leq 1$ und $\langle n \rangle_t \leq 1$

W2. Ausführung B-Befehl und
gleichzeitig CMC-Befehl ab W2.

Bemerkung: Im Alarmfall unterbleibt das Abspeichern

BC ist eingriffssperrender Code
(BEEN := L)

Ausführungszeit: 0,65 μ s

BRINGE UND SPEICHERE INDEXBASIS UND U

Wirkung: W1. $\langle n \rangle_t := 3$

Fall A. Der Rechner läuft im Systemmodus

W2. $\langle 4_2 \rangle := \langle n \rangle_{1,24}$
 $\langle U \rangle := \langle n \rangle_{41,48}$
 $\langle n \rangle_{25,40} := +0$

W3. $\langle 4_2 \rangle_t := 3$ falls Abwickler-
 flip-flop = 0
 $\langle 4_2 \rangle_t := 2$ falls Abwickler-
 flip-flop = L

W4. $\langle XB \rangle := \langle 4_2 \rangle$

Austausch der Index-
 basen und von U

Indexbasis neu setzen

Fall B. Der Rechner läuft nicht im Systemmodus

W2. $\langle (\langle BL \rangle \cdot 2^8 + 4)_2 \rangle := \langle n \rangle_{1,24}$
 $\langle U \rangle := \langle n \rangle_{41,48}$
 $\langle n \rangle_{25,40} := +0$

W3. $\langle (\langle BL \rangle \cdot 2^8 + 4)_2 \rangle_t := 3$ falls Ab-
 wicklerflip-
 flop = 0
 $\langle (\langle BL \rangle \cdot 2^8 + 4)_2 \rangle_t := 2$ falls Ab-
 wicklerflip-
 flop = L

W4. $\langle XB \rangle := \langle (\langle BL \rangle \cdot 2^8 + 4)_2 \rangle$

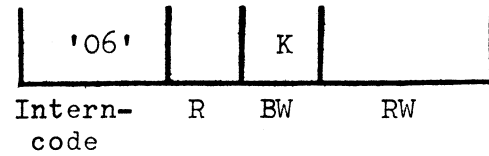
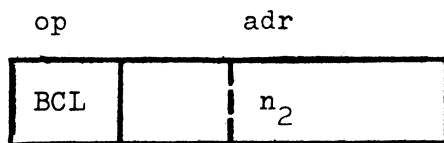
Austauschen der rel.
 Indexbasen und von U

Indexbasis neu setzen

Ausführungszeit: $(0,85 p + 3,3) \mu s$
 wobei p = Anzahl der zurückzuspeichernden Indexregister

BCL

BRINGE UND SPEICHERE MERKLICHTER



Wirkung: W1. $\langle K \rangle ::= \langle n_2 \rangle_{17,24}$

Der aktuelle Merklichter-
stand wird ausgetauscht
mit dem in $\langle n_2 \rangle$ durch
die letzten 8-Bit defi-
nierten Merklichterstand.

Die anderen Bits von
 $\langle n_2 \rangle$ bleiben erhalten.

Ausführungszeit: 0,90 μ s

BD

BRINGE NACH D

op	adr
BD	n

'71'	+		D M
Intern- code	R	BW	RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle D \rangle_1 := \langle D \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Zahlwort
 Vorzeichenangleich
 Marke

Ausführungszeit: 0,15 μ s

BH

BRINGE NACH H

op	adr
BH	n

'73'	+		H M
Intern- code	R	BW	RW

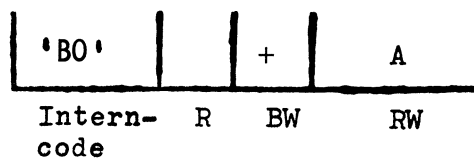
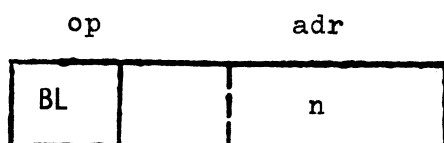
Wirkung: W1. $n := n + \text{mod}2$
 W2. $\langle H \rangle := \langle n \rangle$
 $\langle H \rangle_t := \langle n \rangle_t$
 W3. falls $\langle n \rangle \leq 1$
 $\langle H \rangle_1 := \langle H \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Vorzeichenangleich u.
 Marke bei Zahlworten

Ausführungszeit: 0,15 μ s

BL

BRINGE UND LÖSCHE



Wirkung:

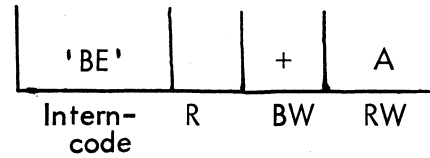
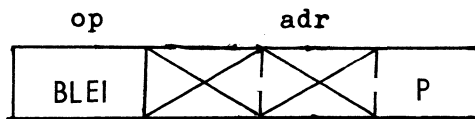
- W1. $n := n + \text{mod}2$
- W2. $\langle A \rangle_{1,48} := \langle n \rangle_{1,48}$
 $\langle A \rangle_t := \langle n \rangle_t$
- W3. $\langle n \rangle := +0$
 $\langle n \rangle_t := 0$

ein Ganzwort gelangt
unverändert nach A

Ausführungszeit: 0,85 μ s

BLEI

BRINGE AUS LEITBLOCK



Wirkung: W1. $\text{adr} := \text{adr} + \text{mod}2$

Adreßteil: $+ 0 \leq p \leq 255$

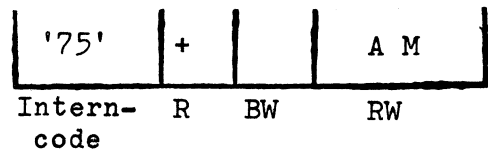
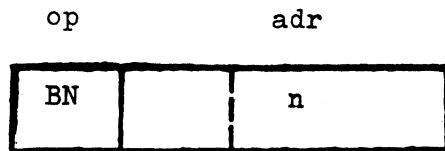
$$\begin{aligned} \text{W2. } \langle A \rangle_{1,48} &:= \langle \langle \text{BL} \rangle 2^8 + p \rangle_{1,48} \\ \langle A \rangle_t &:= \langle \langle \text{BL} \rangle 2^8 + p \rangle_t \end{aligned}$$

ein Ganzwort gelangt
unverändert aus dem
Leitblock nach A.
BL = Leitadreßregister

Ausführungszeit: $0,65 \mu\text{s}$

BN

BRINGE NEGATIV



Wirkung:

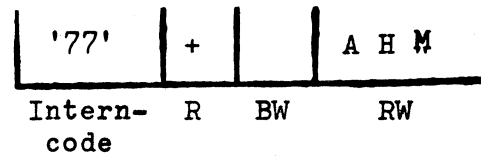
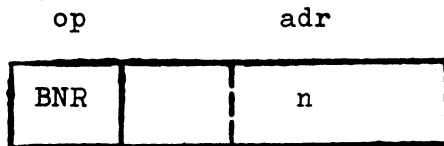
- W1. $n := n + \text{mod}2$
- W2. $\langle A \rangle := -\langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle_t$
- W3. falls $\langle n \rangle_t \leq 1$
 $\langle A \rangle_1 := \langle A \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Zahlwort
Vorzeichenangleich
Marke

Ausführungszeit: 0,25 μ s

BNR

BRINGE NEGATIV U. RESERVIERE



Wirkung:

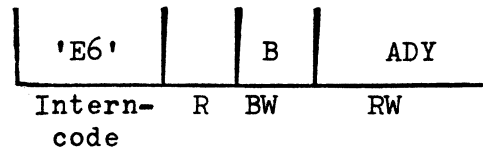
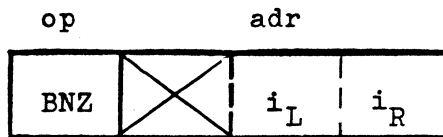
- W1. $n := n + \text{mod}2$
- W2. $\langle H \rangle := \langle A \rangle$
 $\langle H \rangle_t := \langle A \rangle_t$
- W3. $\langle A \rangle := -\langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle_t$
- W4. falls $\langle n \rangle_t \leq 1$
 $\langle A \rangle_1 := \langle A \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Zahlwort
 Vorzeichenangleich
 Marke

Ausführungszeit: 0,25 μ s

BNZ

BRINGE NÄCHSTES ZEICHEN



Eingangsinformation:

$\langle i_L \rangle_{9,12}$ = Anzahl der Bits der im Speicherwort mit der Adresse $\langle i_R \rangle + \text{mod}2$ stehenden Zeichen. Es ist erlaubt
 $\langle i_L \rangle_{9,12} = 4|6|8|12$

$\langle i_L \rangle_{21,24}$ = Nummer des Zeichens, das aus dem Speicherwort ausgelesen werden soll.
Linksbündig im Speicherwort steht das Zeichen mit der Nummer 0, die folgenden Zeichen mit nächsthöherer Nummer schließen sich lückenlos an.

Die übrigen Bits von $\langle i_L \rangle$ sind bedeutungslos.

Wirkung: W1. $\langle B \rangle := \langle i_L \rangle$

W2. $\langle D \rangle := \text{undefiniert}$

$\langle D \rangle_t := \text{undefiniert}$

W3. $\langle A \rangle$:= durch Zeichennummer
adressiertes Zeichen
von $\langle \langle i_R \rangle + \text{mod} 2 \rangle$

$\langle A \rangle_t := \langle \langle i_R \rangle + \text{mod} 2 \rangle_t$

W4. $\langle Y \rangle :=$ undefiniert

Fall A: $\langle i_L \rangle_{21,24} <$ maximale Zeichennummer

W5. $\langle i_L \rangle_{21,24} := \langle i_L \rangle_{21,24} + 1$

W6. entfällt.

Fall B: $\langle i_L \rangle_{21,24} =$ maximale Zeichennummer

W5. $\langle i_L \rangle_{17,24} := +0$

W6. $\langle i_R \rangle := \langle i_R \rangle + 2$

Fall C: $\langle i_L \rangle_{21,24} >$ maximale Zeichennummer

W5. und W6. wie Fall A oder B.

ausgelesenes Zeichen

falls Zeichennr. <
max. Zeichennr.:

Zeichennr. um 1 er-
höht ($\langle i_R \rangle$ bleibt er-
halten)

falls Zeichennr. =
max. Zeichennr.:

Zeichennr. := +0 und

Adresse um 2 erhöht

fehlerhafte Programmierung

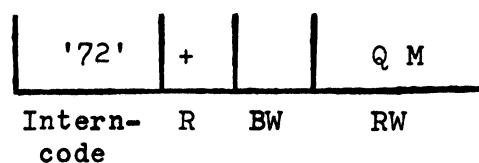
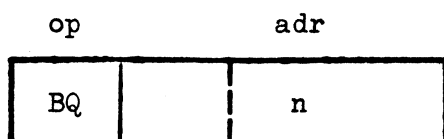
Bemerkung: Maximale Zeichennummer= 11 bei 4-Bit-Zeichen

7	"	6	"	"
5	"	8	"	"
3	"	12	"	"

Ausführungszeit: 2,05 μ s

BQ

BRINGE NACH Q



Wirkung:

- W1. $n := n + \text{mod}2$
- W2. $\langle Q \rangle := \langle n \rangle$
 $\langle Q \rangle_t := \langle n \rangle_t$
- W3. falls $\langle n \rangle_t \leq 1$
 $\langle Q \rangle_1 := \langle Q \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Zahlwort
Vorzeichenangleich
Marke

Ausführungszeit: 0,15 μ s

BQB

BRINGE NACH Q U. BRINGE (NACH A)

op	adr
BQB	n

'DA'	+		A Q M
Intern- code	R	BW	RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle A \rangle := \langle n \rangle$
 $\langle Q \rangle := \langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle_t$
 $\langle Q \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle A \rangle_1 := \langle A \rangle_2$
 $\langle Q \rangle_1 := \langle Q \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Zahlwort

Vorzeichenangleich in A
Q

Marke

Ausführungszeit: 0,10 μ s

BR

BRINGE U. RESERVIERE

op	adr
BR	n

'76'	+		A H M
Intern- code	R	BW	RW

Wirkung:

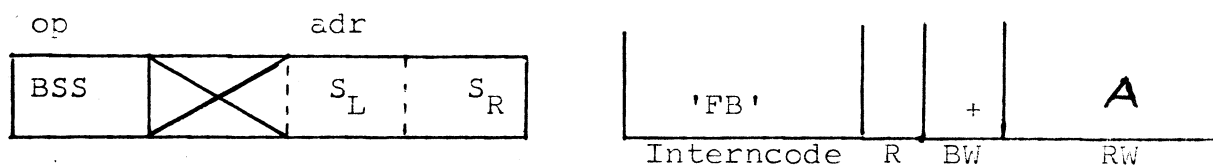
- W1. $n := n + \text{mod}2$
- W2. $\langle H \rangle := \langle A \rangle$
 $\langle H \rangle_t := \langle A \rangle_t$
- W3. $\langle A \rangle := \langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle_t$
- W4. falls $\langle n \rangle_t \leq 1$
 $\langle A \rangle_1 := \langle A \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Zahlwort
Vorzeichenangleich
Marke

Ausführungszeit: 0,25 μ s

BSS

Bringe und setze Steuerbits und Sperren



Adreßteil:

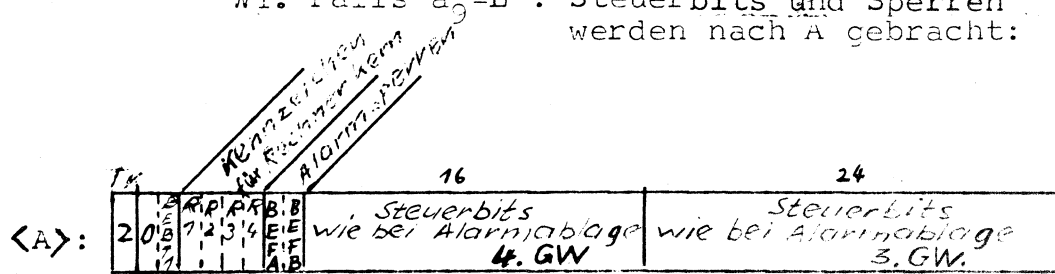
adr = $a_1 a_2 \dots a_{24}$

a_1 bis a_8 und a_{13} bis a_{18} ohne Bedeutung

Spezifikation nur intern angebbbar

Wirkung:

W1. Falls $a_9 = L$: Steuerbits und Sperren werden nach A gebracht:



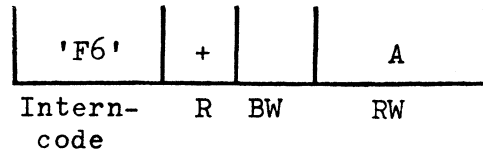
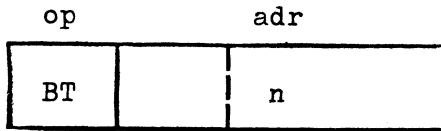
Die Steuerflip-flop und die Sperren bleiben unverändert

W2. Falls $a_{10} = L$: Voraussetzung: nur im System- oder Spezialmodus zulässig oder im Normalmodus in Wartungsvariante, sonst keine Wirkung.

Falls $a_{19}=L$: BEFE:=0	Ändern der Sperren
Falls $a_{20}=L$: BEFA:=0 BEFB:=0	Eingriffssperre löschen
Falls $a_{21}=L$: BEFB:=0	Alarmsperre 1 und 2 löschen
Falls $a_{22}=L$: BEFE:=L	Alarmsperre 2 löschen
Falls $a_{23}=L$: BEFA:=L	Eingriffssperre
Falls $a_{24}=L$: BEFA:=L BEFB:=L	Alarmsperre 1
Sich widersprechende Angaben führen zu undefinierter Befehlsausführung	Alarmsperre 1 und 2
W3. Falls $a_{11} = L$: Voraussetzungen wie bei W2.	Normieren der Seitenassoziativregister
$\langle BP1 \rangle_{1,2}, \langle BP2 \rangle_{1,2}, \langle BP3 \rangle_{1,2}, \langle BP4 \rangle_{1,2} := 0$	Gültigkeits- und Abspeicherbits löschen
Alterungsflipflops: BP1 wird ältestes Register : : : BP4 wird jüngstes Register	
W4. Falls $a_{12} = L$: Voraussetzungen wie bei W2.	Normieren der Indexassoziativregister
$\langle BI1 \rangle_{1,2}, \langle BI2 \rangle_{1,2}, \langle BI3 \rangle_{1,2}, \langle BI4 \rangle_{1,2} := 0$	Gültigkeits- und Rückspeicherbits löschen
Alterungsflipflops: BI1 wird ältestes Register : : : BI4 wird jüngstes Register	

Ausführungszeit: 0,15 μs

BRINGE TEILWORT



Wirkung:

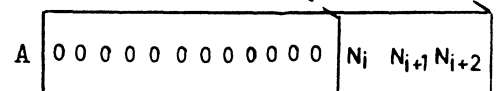
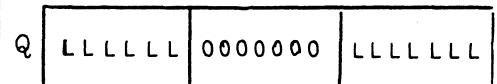
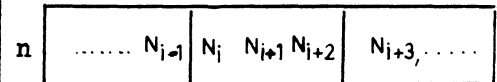
W1. $n := n + \text{mod}2$

W2. $\langle A \rangle_i := \langle n \rangle_i$ falls $\langle Q \rangle_i = 0$

$\langle A \rangle_i := 0$, falls $\langle Q \rangle_i = L$
für $i = 1, 2, \dots, 48$

$\langle A \rangle_t := \langle n \rangle_t$

W3. $\langle A \rangle$ wird nach rechts um p Stellen
geschiftet, wobei $p = \text{Anzahl der}$
in Q rechts anstehenden L-Bits.
Von links werden 0 nachgezogen.



Ausführungszeit: $(0,35 + 0,10p) \mu s$

BRINGE UNVERÄNDERT

op	adr
BU	n

'D3'	+		A
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle A \rangle_{1,48} := \langle n \rangle_{1,48}$

$\langle A \rangle_t := \langle n \rangle_t$

Ein Ganzwort gelangt
unverändert nach A

Ausführungszeit: 0,10 μs

BRINGE ZWEI WORTE

BZ

op	adr
BZ	n

'D9'		+	AQM
Intern- code	R	BW	RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle A \rangle := \langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle A \rangle_1 := \langle A \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W4. $\langle Q \rangle := \langle n+2 \rangle$
 $\langle Q \rangle_t := \langle n+2 \rangle_t$

W5. falls $\langle n+2 \rangle_t \leq 1$
 $\langle Q \rangle_1 := \langle Q \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n+2 \rangle_m$

Wie Befehl B n

Zahlwort

Vorzeichenangleich

Marke

Wie Befehl BQ (n+2)

Zahlwort

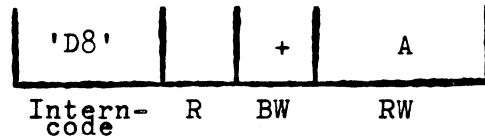
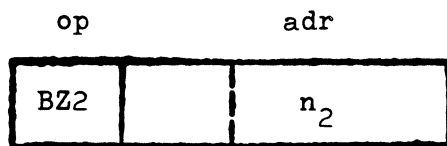
Vorzeichenangleich

Marke

Ausführungszeit: 0,75 μ s

BZ2

BRINGE ZWEI HALBWORTE



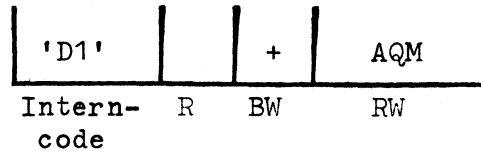
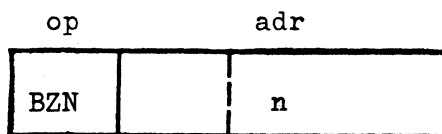
Wirkung: W1. $n_2 := n_2 + \text{mod}2$
W2. $\langle A \rangle_{1,24} := \langle n_2 \rangle$
 $\langle A \rangle_{25,48} := \langle (n+1)_2 \rangle$
 $\langle A \rangle_t := \langle n_2 \rangle_t$

Zwei hintereinanderliegende
Halbworte gelangen unverändert
nach A

Ausführungszeit: 0,50 μs

BZN

BRINGE ZWEI WORTE NEGATIV



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle A \rangle := -\langle n \rangle$
 $\langle A \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle A \rangle_1 := \langle A \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W4. $\langle Q \rangle := -\langle n+2 \rangle$
 $\langle Q \rangle_t := \langle n+2 \rangle_t$

W5. falls $\langle n+2 \rangle_t \leq 1$
 $\langle Q \rangle_1 := \langle Q \rangle_2$
 $\langle M \rangle := \langle M \rangle \vee \langle n+2 \rangle_m$

Zahlwort

Vorzeichenangleich

Marke

Zahlwort

Vorzeichenangleich

Marke

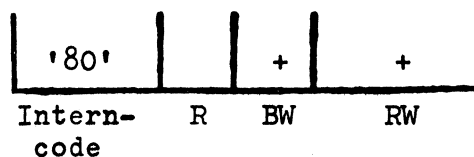
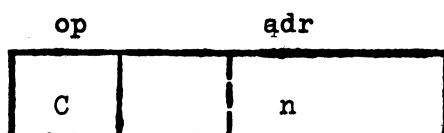
Ausführungszeit: 0,75 μ s

GR/EP1
 T/12/66
 201267

29. Juli 1968

C

SPEICHERE



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle A \rangle_t \geq 2$

W2. $\langle n \rangle := \langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

Fall B: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W2. $\langle n \rangle := \langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

W3. $\langle n \rangle_1 := 0$

Fall C: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W2. A-Alarm

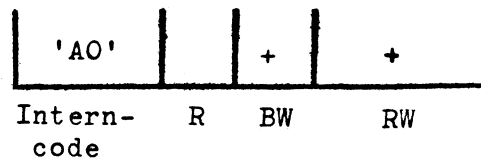
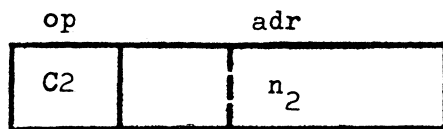
nicht übergelaufenes
Zahlwort

übergelaufenes
Zahlwort

Ausführungszeit: Fälle A,B: 0,40 μs
 Fall C: 0,20 μs

C2

SPEICHERE HALBWORT



Wirkung: W1. $n_2 := n_2 + \text{mod}2$

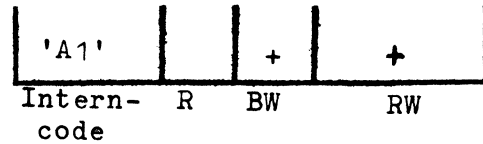
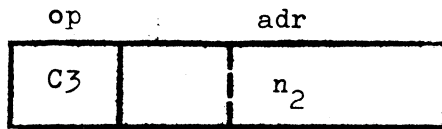
W2. $\langle n_2 \rangle := \langle A \rangle_{25,48}$

Das Halbwort wird
unverändert ein-
gesetzt

Ausführungszeit: 0,75 μ s

C3

SPEICHERE DRITTELWORT



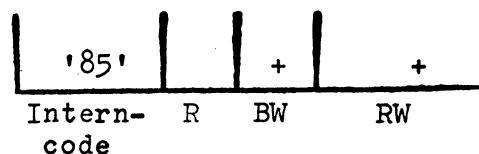
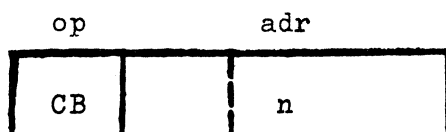
Wirkung: W1. $n_2 := n_2 + \text{mod}2$

W2. $\langle n_3 \rangle := \langle A \rangle_{33,48}$

Das Drittelwort wird
unverändert eingesetzt

Ausführungszeit: 0,75 μ s

SPEICHERE BETRAG



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle A \rangle_t \geq 2$

W2. $\langle n \rangle := \langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

Fall B: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 = \langle A \rangle_2 = 0$

W2. $\langle n \rangle := \langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

Fall C: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 = \langle A \rangle_2 = L$

W2. $\langle n \rangle := -\langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

W3. $\langle n \rangle_1 := 0$

Fall D: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W2. A-Alarm

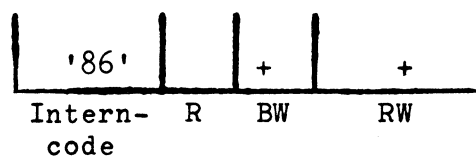
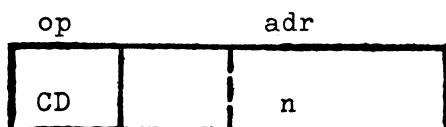
positives Zahlwort

negatives Zahlwort

bei $\langle A \rangle_t = 0$ wird nur
die Mantisse invertiert

übergelaufenes Zahlwort

Ausführungszeit: Fälle A,B,C: 0,45 μs
 Fall D: 0,20 μs

SPEICHERE AUS D

Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle D \rangle_t \geq 2$

W2. $\langle n \rangle := \langle D \rangle$
 $\langle n \rangle_t := \langle D \rangle_t$

Fall B: $\langle D \rangle_t \leq 1$ und $\langle D \rangle_1 = \langle D \rangle_2$

W2. $\langle n \rangle := \langle D \rangle$
 $\langle n \rangle_t := \langle D \rangle_t$
 $\langle n \rangle_1 := 0$

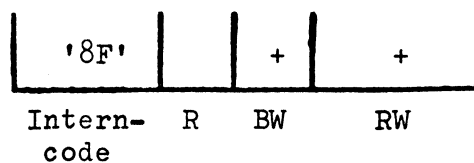
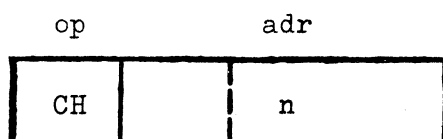
Fall C: $\langle D \rangle_t \leq 1$ und $\langle D \rangle_1 \neq \langle D \rangle_2$

W2. A-Alarm

nicht übergelaufenes
Zahlwort

übergelaufenes
Zahlwort

Ausführungszeit: Fälle A,B: 0,45 μs
 Fall C: 0,20 μs

SPEICHERE AUS H

Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle H \rangle_t \geq 2$

W2. $\langle n \rangle := \langle H \rangle$
 $\langle n \rangle_t := \langle H \rangle_t$

Fall B: $\langle H \rangle_t \leq 1$ und $\langle H \rangle_1 = \langle H \rangle_2$

W2. $\langle n \rangle := \langle H \rangle$
 $\langle n \rangle_t := \langle H \rangle_t$
 $\langle n \rangle_1 := 0$

Fall C: $\langle H \rangle_t \leq 1$ und $\langle H \rangle_1 \neq \langle H \rangle_2$

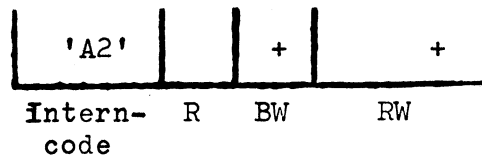
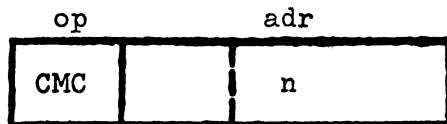
W2. A-Alarm

nicht übergelaufenes
Zahlwort

übergelaufenes
Zahlwort

Ausführungszeit: Fälle A,B: 0,45 μ s
 Fall C: 0,20 μ s

SPEICHERE MIT MARKE AUS SPEICHER



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle A \rangle_t \leq 1$ und $\langle n \rangle_t \leq 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W2. $\langle n \rangle_{2,48} := \langle A \rangle_{2,48}$
 $\langle n \rangle_t := \langle A \rangle_t$

W3. $\langle n \rangle_1 := \langle n \rangle_1$

Markenstelle in n
bleibt erhalten

Fall B: $\langle A \rangle_t \leq 1$ und $\langle n \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

Übergelaufenes
Zahlwort

W2. A-Alarm

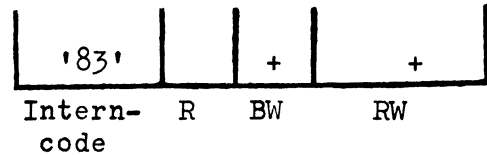
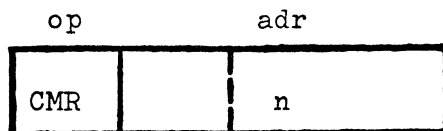
Fall C: $\langle A \rangle_t \geq 2$ oder $\langle n \rangle_t \geq 2$

kein Zahlwort

W2. Tk-Alarm

Ausführungszeit: Fall A: $0,65 \mu s$
 Fälle B,C: $0,20 \mu s$

SPEICHERE MIT MARKE AUS REGISTER



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W2. $\langle n \rangle := \langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

W3. $\langle n \rangle_1 := \langle M \rangle$

Normalfall

Fall B: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W2. A-Alarm

Übergelaufenes
Zahlwort

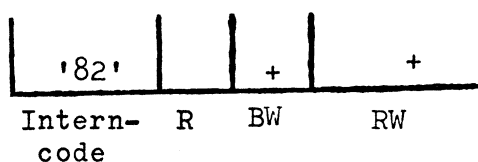
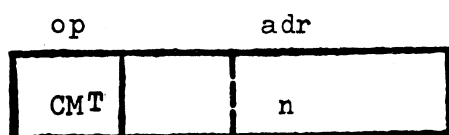
Fall C: $\langle A \rangle_t \geq 2$

W2. Tk-Alarm

kein Zahlwort

Ausführungszeit: Fall A: $0,40 \mu s$
 Fälle B,C: $0,20 \mu s$

SPEICHERE MARKIERT



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W2. $\langle n \rangle := \langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

W3. $\langle n \rangle_1 := L$

Normalfall

Fall B: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W2. A-Alarm

Übergelaufenes
Zahlwort

Fall C: $A_t \geq 2$

W2. Tk-Alarm

kein Zahlwort

Ausführungszeit: Fall A: 0,40 μs
 Fälle B,C: 0,20 μs

SPEICHERE NEGATIV

op	adr
CN	n

'84'		+	+
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle A \rangle_t \geq 2$

W2. $\langle n \rangle := -\langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

Fall B: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W2. $\langle n \rangle := -\langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

W3. $\langle n \rangle_1 := 0$

Fall C: $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W2. A-Alarm

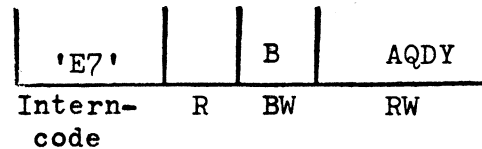
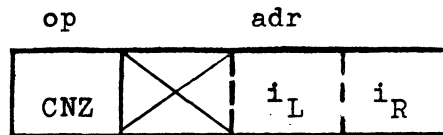
Falls $\langle A \rangle_t = 0$
 wird nur die
 Mantisse invertiert

Markenstelle bei
 Zahlworten

übergelaufenes Zahlwort

Ausführungszeit: Fälle A,B: 0,45 μs
 Fall C: 0,20 μs

SPEICHERE NÄCHSTES ZEICHEN



Eingangsinformation:

$\langle i_L \rangle_{9,12}$ = Anzahl der Bits der im Speicherwort mit der Adresse $\langle i_R \rangle + \text{mod} 2$ stehenden Zeichen. Es ist erlaubt $\langle i_L \rangle_{9,12} = 4|6|8|12$

$\langle i_L \rangle_{21,24}$ = Nummer des Zeichens, das im Speicherwort ersetzt wird. Linksbündig im Speicherwort steht das Zeichen mit der Nummer 0. Diesem Zeichen schließen sich lückenlos die Zeichen mit den nächst höheren Nummern an.

Die übrigen Bits von $\langle i_L \rangle$ sind bedeutungslos.

In A steht rechtsbündig das Zeichen, das abgespeichert werden soll. Die übrigen Stellen von A müssen 0 sein. (Andernfalls werden andere Zeichen im Speicher und Q zerstört.)

Wirkung: W1. $\langle B \rangle := \langle i_L \rangle$

W2. $\langle D \rangle := \text{undefiniert}$

$\langle D \rangle_t := \text{undefiniert}$

W3. $\langle Q \rangle :=$ abzuspeicherndes Zeichen aus A,
gemäß der Zeichennummer geschif-
tet. Die anderen Bits in Q werden
gelöscht.

$$\langle Q \rangle_t := \langle A \rangle_t$$

W4. In $\langle i_R \rangle + \text{mod}2$ wird das durch die Zeichen-
nummer adressierte Zeichen durch das in A
stehende Zeichen überschrieben.

$$W5. \langle A \rangle := \langle \langle i_R \rangle + \text{mod}2 \rangle$$

$$\langle A \rangle_t := \langle \langle i_R \rangle + \text{mod}2 \rangle_t$$

$$W6. \langle Y \rangle := \text{undefiniert}$$

Fall A: $\langle i_L \rangle_{21,24} < \text{maximale Zeichennummer}$

falls Zeichennr. $< \text{max.}$
Zeichennr.:

$$W7. \langle i_L \rangle_{21,24} := \langle i_L \rangle_{21,24} + 1$$

Zeichennr. um 1 erhöht
($\langle i_R \rangle$ bleibt erhalten)

W8. entfällt.

Fall B: $\langle i_L \rangle_{21,24} = \text{maximale Zeichennummer}$

falls Zeichennr. = max.
Zeichennr.:

$$W7. \langle i_L \rangle_{17,24} := +0$$

Zeichennr. := +0 und

$$W8. \langle i_R \rangle := \langle i_R \rangle + 2$$

Adresse um 2 erhöht

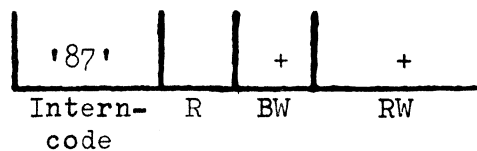
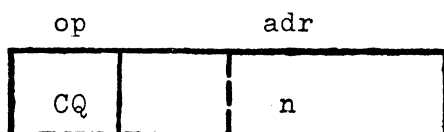
Fall C: $\langle i_L \rangle_{21,24} > \text{maximale Zeichennummer}$

fehlerhafte Programmierung

W5. und W6. wie Fall A oder B.

<u>Bemerkung:</u> Maximale Zeichennummer =	11	bei	4-Bit-Zeichen
	7	"	6 " "
	5	"	8 " "
	3	"	12 " "

Ausführungszeit: 2,75 μs

SPEICHERE AUS Q

Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle Q \rangle_t \geq 2$

W2. $\langle n \rangle := \langle Q \rangle$
 $\langle n \rangle_t := \langle Q \rangle_t$

Fall B: $\langle Q \rangle_t \leq 1$ und $\langle Q \rangle_1 = \langle Q \rangle_2$

W2. $\langle n \rangle := \langle Q \rangle$
 $\langle n \rangle_t := \langle Q \rangle_t$
 $\langle n \rangle_1 := 0$

Fall C: $\langle Q \rangle_t \leq 1$ und $\langle Q \rangle_1 \neq \langle Q \rangle_2$

W2. A-Alarm

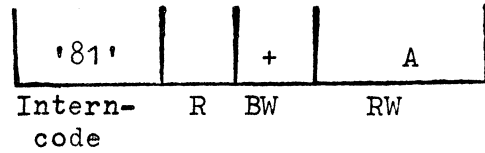
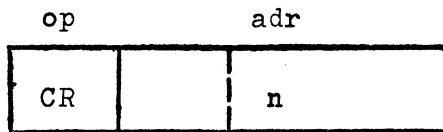
nicht übergelaufenes
Zahlwort

übergelaufenes
Zahlwort

Ausführungszeit: Fall A,B: 0,45 μs
 Fall C: 0,20 μs

CR

SPEICHERE UND BRINGE RESERVE



Wirkung:

- W1. $n := n + \text{mod}2$
- W2. Ausführung des C-Befehls ab W2
- W3. $\langle A \rangle := \langle H \rangle$
 $\langle A \rangle_t := \langle H \rangle_t$
- W4. Ausgang 1

W3 auch wenn
A-Alarm bei C-Befehl

Ausführungszeit: 0,45 μ s

CT

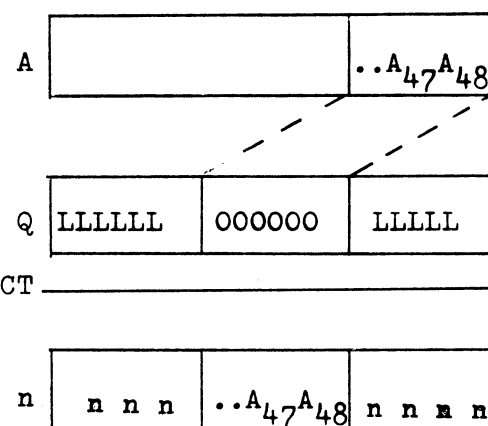
SPEICHERE TEILWORT

op	adr
CT	n

'F7'		+	AD
Intern- code	R	BW	RW

Wirkung:

- W1. $n := n + \text{mod}2$
- W2. $\langle A \rangle$ wird nach links um p Stellen im Kreis geschiftet, wobei p = Anzahl der in Q rechts anstehenden L-Bits.
- W3. $\langle n \rangle_i := \langle n \rangle_i$ falls $\langle Q \rangle_i = L$
 $\langle n \rangle_i := \langle A \rangle_i$ falls $\langle Q \rangle_i = 0$
für $i = 1, 2, \dots, 48$
- W4. $\langle A \rangle$ wird im Kreis in die ursprüngliche Lage zurückgeschiftet.
- W5. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

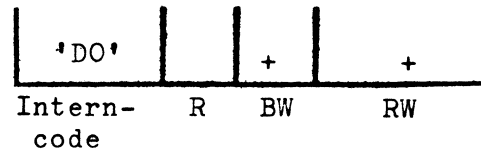
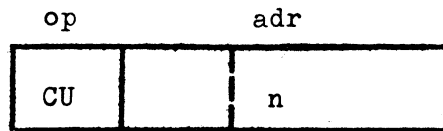


Ergebnis steht in D

Ausführungszeit: (1,10 + 0,10p) μ s

CU

SPEICHERE UNVERÄNDERT



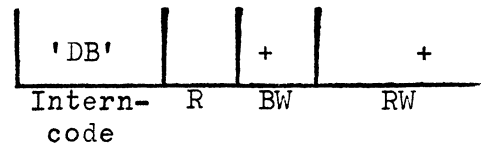
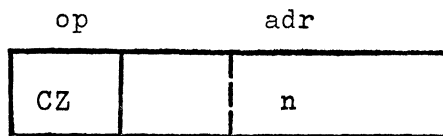
Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle n \rangle := \langle A \rangle$
 $\langle n \rangle_t := \langle A \rangle_t$

Ausführungszeit: 0,40 μs

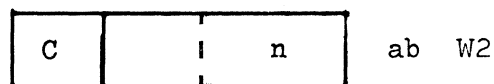
CZ

SPEICHERE ZWEI WÖRTE

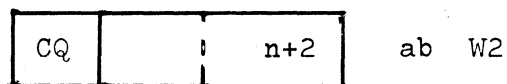


Wirkung: W1. $n := n + \text{mod}2$

W2. Ausführung des Befehls



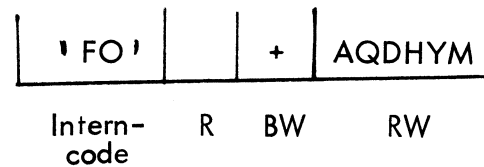
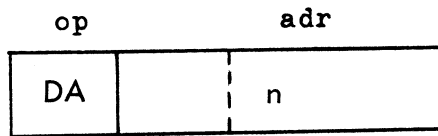
W3. Ausführung des Befehls



W4. Ausgang 1

Ausführungszeit: 0,80 μ s

DOPPELTE GENAUIGKEIT: ADDIERE

Wirkung:W1. Falls $\langle n \rangle_t \leq 1$:

$$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$$

W2. Falls $\langle n+2 \rangle_t \leq 1$:

$$\langle M \rangle := \langle M \rangle \vee \langle n+2 \rangle_m$$

W3. $\langle D \rangle, \langle H \rangle := + 0$

$$\langle D \rangle_t, \langle H \rangle_t := 1$$

Fall A:(a) $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle Q \rangle_t = \langle n+2 \rangle_t = 1$ und(b) $\langle A \rangle_1 = \langle A \rangle_2 = \langle Q \rangle_1 = \langle Q \rangle_2$ und $\langle n \rangle_2 = \langle n+2 \rangle_2$ W4. $\langle A, Q \rangle := \langle A, Q \rangle + \langle n, n+2 \rangle$

Das Ergebnis ist normalisiert und bei normalisierten Operanden auch gerundet

W5. $\langle Y \rangle :=$ Anzahl der Binärstellen, um die das Ergebnis normalisiert wurde,

falls Ergebnis = 0 oder Exponentenunterlauf:

$$\langle Y \rangle := + 0$$

W6. Falls Exponent des Ergebnisses übergelaufen:

A-Alarm

Fall B:

Bedingung (a) oder (b) nicht erfüllt

W4. entfällt

$$W5. \langle Y \rangle := + 0$$

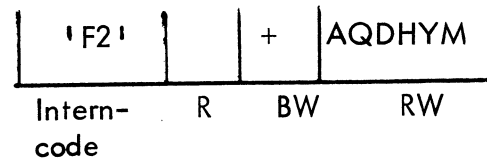
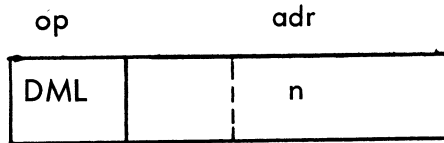
W6. Falls (b) nicht erfüllt:

A-Alarm

Falls (a) nicht erfüllt:

Tk-Alarm

Ausführungszeit:

DOPPELTE GENAUIGKEIT: MULTIPLIZIEREWirkung:

W1. Falls $\langle n \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W2. Falls $\langle n+2 \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n+2 \rangle_m$

W3. $\langle D \rangle, \langle H \rangle := +0$
 $\langle D \rangle_t, \langle H \rangle_t := 1$

Fall A: (a) $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle Q \rangle_t = \langle n+2 \rangle_t = 1$ und
 (b) $\langle A \rangle_1 = \langle A \rangle_2 = \langle Q \rangle_1 = \langle Q \rangle_2$ und $\langle n \rangle_2 = \langle n+2 \rangle_2$

W4. $\langle A, Q \rangle := \langle A, Q \rangle \cdot \langle n, n+2 \rangle$

Das Ergebnis ist gerundet und normalisiert

W5. $\langle Y \rangle :=$ Anzahl der Binärstellen, um die das Ergebnis normalisiert wurde.

Falls Mantisse eines Faktors $\neq \pm 0$ oder Exponentenunterlauf:

$\langle Y \rangle := +0$

W6. Falls Exponentenüberlauf und Mantissen beider Faktoren $\neq \pm 0$:
 A-Alarm

Fall B: Bedingung (a) oder (b) nicht erfüllt

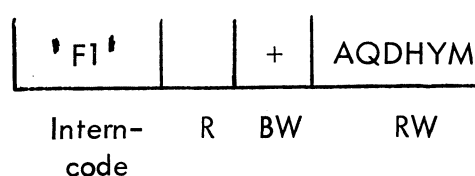
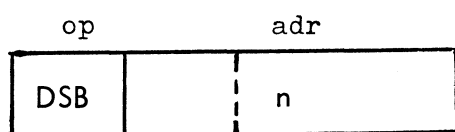
W4. entfällt

W5. $\langle Y \rangle := +0$

W6. Falls (b) nicht erfüllt: A-Alarm
 Falls (a) nicht erfüllt: Tk-Alarm

Ausführungszeit: Fall A: 12,15 μ s
 Fall B: 1,1 μ s

DOPPELTE GENAUIGKEIT: SUBTRAHIERE

Wirkung:

W1. Falls $\langle n \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W2. Falls $\langle n+2 \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n+2 \rangle_m$

W3. $\langle D \rangle, \langle H \rangle := + 0$
 $\langle D \rangle_t, \langle H \rangle_t := 1$

Fall A:

(a) $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle Q \rangle_t = \langle n+2 \rangle_t = 1$ und
 (b) $\langle A \rangle_1 = \langle A \rangle_2 = \langle Q \rangle_1 = \langle Q \rangle_2$ und $\langle n \rangle_2 = \langle n+2 \rangle_2$

W4. $\langle A, Q \rangle := \langle A, Q \rangle - \langle n, n+2 \rangle$

Das Ergebnis ist normalisiert und bei normalisierten Operanden auch gerundet

W5. $\langle Y \rangle :=$ Anzahl der Binärstellen, um die das Ergebnis normalisiert wurde,

falls Ergebnis = 0 oder Exponentenunterlauf:

$\langle Y \rangle := + 0$

W6. Falls Exponent des Ergebnisses übergelaufen:

A-Alarm

Fall B:

Bedingung (a) oder (b) nicht erfüllt

W4. entfällt

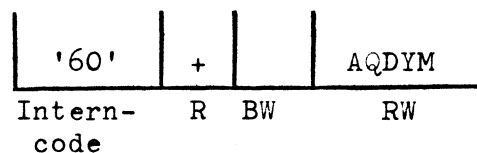
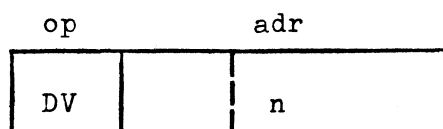
W5. $\langle Y \rangle := + 0$

W6. Falls (b) nicht erfüllt: A-Alarm

Falls (a) nicht erfüllt: Tk-Alarm

Ausführungszeit:

DIVIDIERE



Wirkung: W1. $n := n + \text{mod}2$

W2. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahlworten

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle n \rangle \neq \pm 0$
und $\langle A \rangle_1 = \langle A \rangle_2$ und $|\langle A \rangle| < |\langle n \rangle|$

Normalfall

W3. $\langle A \rangle := \langle A \rangle : \langle n \rangle$
 $\langle Q \rangle := \text{Rest, wobei } \text{sgn}(\text{Rest}) = \text{sgn}(\text{Dividend})$
 $\langle Q \rangle_t := 1$

Die Zahlen werden als
echte Brüche aufgefaßt

W4. $\langle D \rangle := +0$
 $\langle D \rangle_t := 1$

W5: $\langle Y \rangle := +0$

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle n \rangle \neq \pm 0$ und $\langle A \rangle_1 = \langle A \rangle_2$
und $|\langle A \rangle| \geq |\langle n \rangle|$

Gleitkommafall

W3. $\langle A \rangle := \langle A \rangle : \langle n \rangle$, wobei Ergebnis
in Gleitkommadarstellung,
normalisiert, nicht ge-
rundet

$$\langle A \rangle_t := 0$$

$$\langle Q \rangle := +0$$

$$\langle Q \rangle_t := 1$$

W4. $\langle D \rangle := +0$

$$\langle D \rangle_t := 1$$

W5. $\langle Y \rangle := +0$

A-Alarm

Fall C:

$\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle n \rangle \neq \pm 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Dividend

W3. $\langle A \rangle := \text{undefiniert}$

$$\langle A \rangle_t := 0 \text{ oder } 1$$

$$\langle Q \rangle := \text{undefiniert}$$

$$\langle Q \rangle_t := 1$$

W4. $\langle D \rangle := +0$

$$\langle D \rangle_t := 1$$

W5. $\langle Y \rangle := +0$

evtl. A-Alarm

Fall D:

$\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle n \rangle = \pm 0$

Divisor = 0

W3. $\langle D \rangle := \langle A \rangle$

$$\langle D \rangle_t := \langle A \rangle_t$$

$$\langle Q \rangle := +0$$

$$\langle Q \rangle_t := 1$$

W4. $\langle A \rangle := +0$

W5. $\langle Y \rangle := +0$

A-Alarm

Fall E: $\langle A \rangle_t \neq 1$ oder $\langle n \rangle_t \neq 1$

falsche Tk

W3. $\langle A \rangle$ bleibt erhalten

$\langle Q \rangle := +0$

$\langle Q \rangle_t := 1$

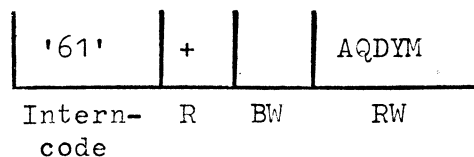
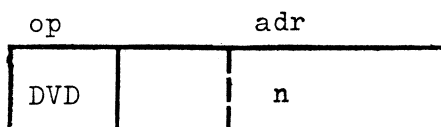
W4. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

Tk-Alarm

Ausführungszeit: Fall A: 11,0 μs
 Fall B: 12,9 μs
 Fall C: etwa 12 μs
 Fall D: 0,15 μs
 Fall E: 0,10 μs

DIVIDIERE DOPPELT LANG



Wirkung:

W1. $n := n + \text{mod}2$

W2. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahlwörtern

Fall A:

$\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle n \rangle \neq 0$ und
 $\langle A \rangle_1 = \langle A \rangle_2$ und $\langle Q \rangle_1 = \langle Q \rangle_2$ und
 $|\langle A \rangle|$ (nach Vorzeichenangleich) $< |\langle n \rangle|$

Normalfall

(siehe Befehl VAQ)

Die Zahlen werden als
echte Brüche aufgefaßt

W3. $\langle A \rangle := \langle A, Q \rangle : \langle n \rangle$
 $\langle Q \rangle := \text{Rest}, \text{ wobei } \text{sgn}(\text{Rest}) =$
 $\text{sgn}(\text{Dividend})$

$\langle Q \rangle_t := 1$

W4. $\langle D \rangle := +0$
 $\langle D \rangle_t := 1$

W5. $\langle Y \rangle := +0$

Fall B:

$\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle n \rangle \neq 0$ und
 $\langle A \rangle_1 = \langle A \rangle_2$ und $\langle Q \rangle_1 = \langle Q \rangle_2$ und
 $|\langle A \rangle|$ (nach Vorzeichenangleich) $\geq |\langle n \rangle|$

Gleitkommafall

(siehe Befehl VAQ)

W3. $\langle A \rangle := \langle A, Q \rangle : \langle n \rangle$, Ergebnis in
Gleitkommadarstellung,
normalisiert, nicht gerundet

$\langle A \rangle_t := 0$

$\langle Q \rangle := +0$

$\langle Q \rangle_t := 1$

W4. $\langle D \rangle := +0$

$\langle D \rangle_t := 1$

W5. $\langle Y \rangle := +0$

A-Alarm

Fall C: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle n \rangle \neq \pm 0$ und
 $(\langle A \rangle_1 \neq \langle A \rangle_2$ oder $\langle Q \rangle_1 \neq \langle Q \rangle_2)$

über-oder unterge-
laufener Dividend

W3. $\langle A \rangle := \text{undefiniert}$

$\langle A \rangle_t := 0$ oder 1

$\langle Q \rangle := \text{undefiniert}$

$\langle Q \rangle_t := 1$

W4. $\langle D \rangle := +0$

$\langle D \rangle_t := 1$

W5. $\langle Y \rangle := +0$

eventuell A-Alarm

Fall D: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle n \rangle = \pm 0$ und $\langle Q \rangle_1 = \langle Q \rangle_2$

Divisor = 0

W3. $\langle Q \rangle :=$ Dividend 2. Teil nach Vorzeichenangleich zwischen A und Q

(siehe Befehl VAQ)

$\langle Q \rangle_t := 1$

$\langle D \rangle :=$ Dividend 1. Teil nach Vorzeichenangleich

$\langle D \rangle_t := 1$

W4. $\langle A \rangle := +0$

A-Alarm

Fall E: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle n \rangle = \pm 0$ und $\langle Q \rangle_1 \neq \langle Q \rangle_2$

Divisor = 0

Dividend Teil 2 über- oder untergelaufen.

W3. $\langle A \rangle := +0$

$\langle Q \rangle :=$ undefiniert

$\langle Q \rangle_t := 1$

W4. $\langle D \rangle :=$ undefiniert

$\langle D \rangle_t := 1$

W5. $\langle Y \rangle := +0$

A-Alarm

Fall F: $\langle A \rangle_t = 1$ und $\langle n \rangle_t \neq 1$ und $\langle Q \rangle_1 = \langle Q \rangle_2$

falsche TK des Divisors

W3. $\langle A, Q \rangle := \langle A, Q \rangle$ mit Vorzeichenangleich zwischen A und Q

(siehe Befehl VAQ)

W4. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

Tk-Alarm

Fall G: $\langle A \rangle_t = 1$ und $\langle n \rangle_t \neq 1$ und $\langle Q \rangle_1 \neq \langle Q \rangle_2$

W3. $\langle A, Q \rangle := \text{undefiniert}$

W4. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

Tk-Alarm

falsche Tk des Divi-
sors und Dividend
2. Teil über- oder
untergelaufen

Fall H: $\langle A \rangle_t \neq 1$

W3. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

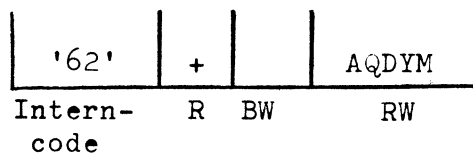
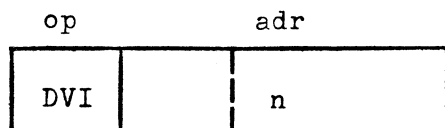
Tk-Alarm

falsche Tk des
Dividenden

Ausführungszeit: Fall A: 11,4 μs
 Fall B: 13,3 μs
 Fall C: etwa 12 μs
 Fall D: 0,55 μs
 Fall E: 0,55 μs
 Fall F: 0,50 μs
 Fall G: 0,50 μs
 Fall H: 0,10 μs

DVI

DIVIDIERE INVERS



Wirkung: W1. $n := n + \text{mod}2$

W2. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle \neq \pm 0$ und $\langle A \rangle_1 = \langle A \rangle_2$
und $|\langle n \rangle| < |\langle A \rangle|$

W3. $\langle A \rangle := \langle n \rangle : \langle A \rangle$
 $\langle Q \rangle := \text{Rest, wobei } \text{sgn}(\text{Rest}) = \text{sgn}(\text{Dividend})$
 $\langle Q \rangle_t := 1$

W4. $\langle D \rangle := +0$
 $\langle D \rangle_t := 1$

W5. $\langle Y \rangle := +0$

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle \neq \pm 0$ und $\langle A \rangle_1 = \langle A \rangle_2$
und $|\langle n \rangle| \geq |\langle A \rangle|$

W3. $\langle A \rangle := \langle n \rangle : \langle A \rangle$, Ergebnis in
 Gleitkommadarstellung,
 normalisiert, nicht ge-
 rundet

Marke bei Zahl-
wörtern

Normalfall

Die Zahlen werden als
echte Brüche aufgefaßt

Dividend aus Speicher

Gleitkommafall

$$\langle A \rangle_t := 0$$

$$\langle Q \rangle := +0$$

$$\langle Q \rangle_t := 1$$

W4. $\langle D \rangle := +0$

$$\langle D \rangle_t := 1$$

W5. $\langle Y \rangle := +0$

A-Alarm

Fall C: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle \neq \pm 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Divisor

W3. $\langle A \rangle := \text{undefiniert}$

$$\langle A \rangle_t := 0 \text{ oder } 1$$

$$\langle Q \rangle := \text{undefiniert}$$

$$\langle Q \rangle_t := 1$$

W4. $\langle D \rangle := +0$

$$\langle D \rangle_t := 1$$

W5. $\langle Y \rangle := +0$

evtl. A-Alarm

Fall D: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle = \pm 0$

Divisor = 0

W3. $\langle A \rangle := +0$

$$\langle Q \rangle := +0$$

$$\langle Q \rangle_t := 1$$

W4. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W5. $\langle Y \rangle := +0$
 A-Alarm

Fall E: $\langle A \rangle_t \neq 1$ oder $\langle n \rangle_t \neq 1$

falsche Tk

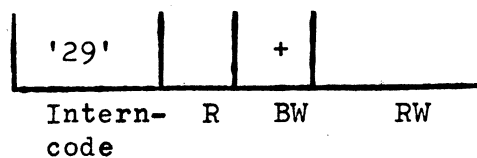
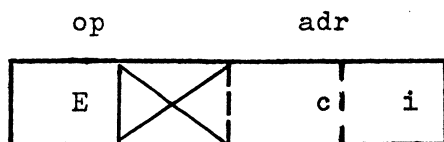
W3. $\langle A \rangle$ bleibt erhalten
 $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 1$

W4. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$
 Tk-Alarm

Ausführungszeit: Fall A: 11,1 μ s
 Fall B: 13,0 μ s
 Fall C: etwa 12 μ s
 Fall D: 0,15 μ s
 Fall E: 0,10 μ s

E

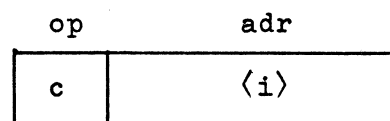
ERSETZE



Wirkung: W1. op:= c
adr:= <i>

W2. Ausgang 2

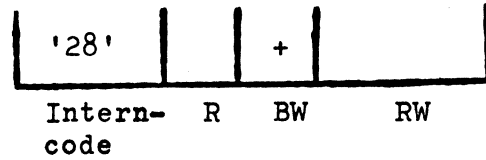
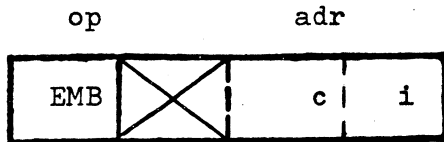
Definition des Befehls



Ausführungszeit: 0,50µs

EMB

ERSETZE UND MODIFIZIERE MIT B

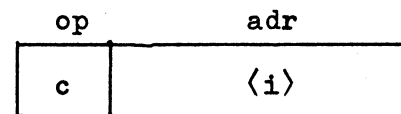


Wirkung: W1. op := c
adr := <i>

W2. mod2 :=

W3. Ausgang 2.

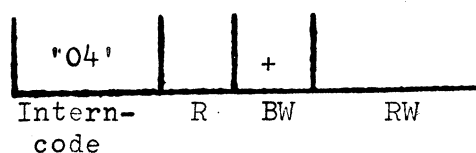
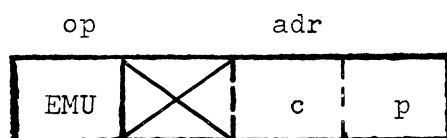
Definition des Befehls



Ausführungszeit: 0,50 μ s

EMU

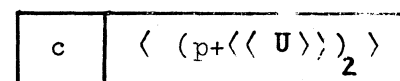
ERSETZE NACH MODIFIZIERUNG ÜBER U



Wirkung: W1. op := c
 adr := $\langle (p + \langle \langle U \rangle \rangle)_2 \rangle$
 $-127 \leq p \leq 127$

W2. Ausgang 2

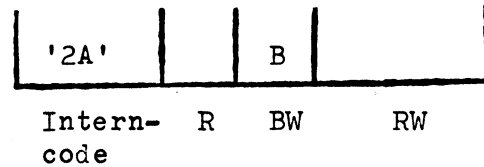
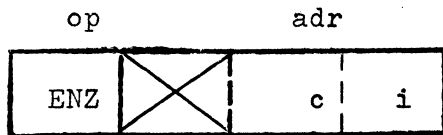
Definition des Befehls



Ausführungszeit: 1,775 μ s

ENZ

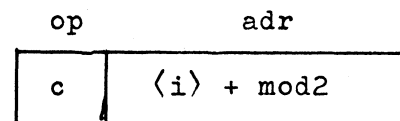
ERSETZE NEGATIV ZÄHLEND



Wirkung:

- W1. $op := c$
 $adr := \langle i \rangle + mod2$
- W2. $mod2 := 0$
- W3. $\langle i \rangle := \langle i \rangle - 2$
 $\langle B \rangle := \langle i \rangle - 2$
- W4. Ausgang 2

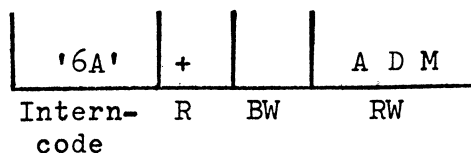
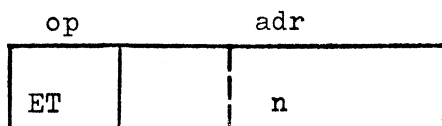
Definition des Befehls



Ausführungszeit: 0,70 μs

ET

ET



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle n \rangle_t \geq 2$

W2. $\langle A \rangle := \langle A \rangle \wedge \langle n \rangle$
 $\langle A \rangle_t := \text{Max} (\langle A \rangle_t, \langle n \rangle_t)$

W3. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

Fall B: $\langle n \rangle_t \leq 1$

W2. $\langle A \rangle_1 := \langle A \rangle_1 \wedge \langle n \rangle_2$
 $\langle A \rangle_k := \langle A \rangle_k \wedge \langle n \rangle_k$
 $2 \leq k \leq 48$
 $\langle A \rangle_t := \text{Max} (\langle A \rangle_t, \langle n \rangle_t)$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W3. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

Zahlwort

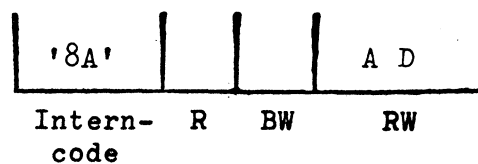
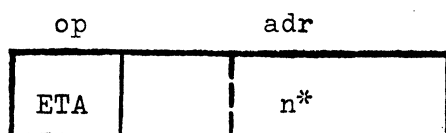
Markenstelle

Marke

Ausführungszeit: 0,25 μs

ETA

ET ADRESSTEIL



Wirkung: W1. $n^* = n^* + \text{mod}2$

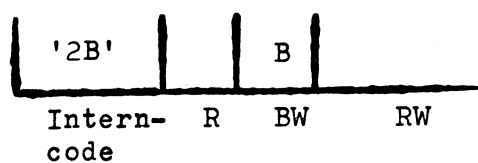
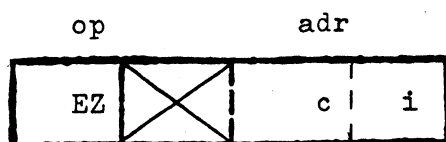
W2. $\langle A \rangle := \langle H \rangle \wedge n^*$
 $\langle A \rangle_t := \langle H \rangle_t$

W3. $\langle D \rangle :o= n^*$
 $\langle D \rangle_t := 1$

Ausführungszeit: 0,40 μ s

EZ

ERSETZE ZÄHLEND



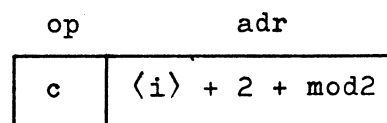
Wirkung: W1. $op := c$
 $adr := \langle i \rangle + 2 + mod2$

W2. $mod2 := 0$

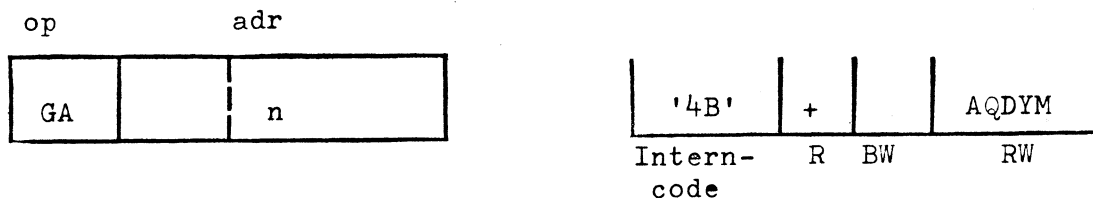
W3. $\langle B \rangle := \langle i \rangle + 2$
 $\langle i \rangle := \langle i \rangle + 2$

W4. Ausgang 2

Definition des Befehls



Ausführungszeit: 0,55 μs

GLEITKOMMA ADDIERE

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei
Zahlwörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := \langle A \rangle + \langle n \rangle$, das Ergebnis ist normalisiert

Falls beide Operanden normalisiert waren, ist das Ergebnis gerundet. Bei nicht normalisierten Operanden braucht dies nicht zu gelten, auch kann dann das Ergebnis von einer Stelle an (bis zu der es korrekt ist) nur noch vorzeichengleiche (statt berechneter) Stellen enthalten.

W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ Anzahl der Binärstellen, um die das Ergebnis normalisiert wurde,

falls Ergebnis = 0 oder Exponentenunterlauf:

$\langle Y \rangle := +0$

W7. falls Exponent des Ergebnisses
übergelaufen:

A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Operand

W4. $\langle A \rangle :=$ undefiniert

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ undefiniert

evtl. A-Alarm

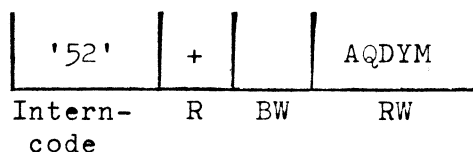
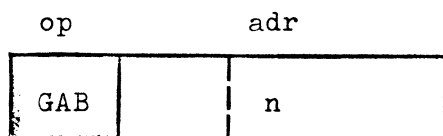
Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

W4. Tk-Alarm

Ausführungszeit: Fall A: 1,4 μ s
Fall B: 1,4 μ s
Fall C: 0,1 μ s

GLEITKOMMA ADDIERE BETRAG



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahlwörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := \langle A \rangle + |\langle n \rangle|$, das Ergebnis ist normalisiert

Falls beide Operanden normalisiert waren, ist das Ergebnis gerundet. Bei nicht normalisierten Operanden braucht dies nicht zu gelten, auch kann dann das Ergebnis von einer Stelle an (bis zu der es korrekt ist) nur noch vorzeichengleiche (statt berechneter) Stellen enthalten.

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ Anzahl der Binärstellen, um die das Ergebnis normalisiert wurde,

falls Ergebnis = 0 oder Exponentenunterlauf:

$\langle Y \rangle := +0$

W7. falls Exponent des Ergebnisses
übergelaufen:

A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Operand

W4. $\langle A \rangle :=$ undefiniert

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ undefiniert

evtl. A-Alarm

Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

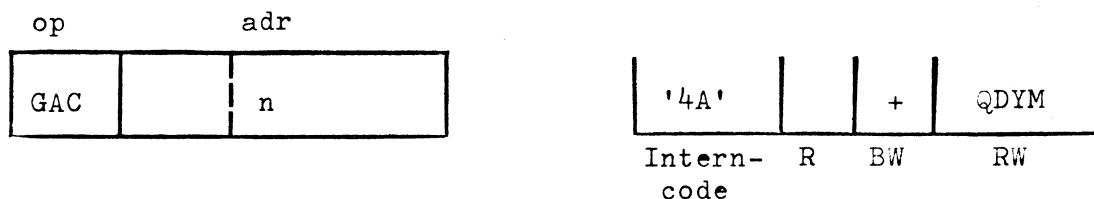
falsche Tk

W4. Tk-Alarm

Ausführungszeit: Fall A: 1,4 μ s
Fall B: 1,4 μ s
Fall C: 0,1 μ s

GAC

GLEITKOMMA ADDIERE IM SPEICHER



Wirkung: W1. $n := n + \text{mod}2$

W2. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W3. $\langle D \rangle := \langle n \rangle + \langle A \rangle$, das Ergeb-
 nis ist normalisiert

Falls beide Operanden normalisiert
waren, ist das Ergebnis gerundet.
Bei nicht normalisierten Operanden
braucht dies nicht zu gelten, auch
kann dann das Ergebnis von einer
Stelle an (bis zu der es korrekt
ist) nur noch vorzeichengleiche
(statt berechneter) Stellen enthalten

W4. falls Exponent des Ergebnisses
 nicht übergelaufen:

$\langle n \rangle_1 := \langle n \rangle_1$

$\langle n \rangle_{2,48} := \langle D \rangle_{2,48}$

falls Exponent des Ergebnisses über-
gelaufen:

A-Alarm

W5. $\langle q \rangle := +0$

$\langle q \rangle_t := 0$

W6. $\langle Y \rangle :=$ Anzahl der Binärstellen, um
die das Ergebnis normalisiert
wurde,
falls Ergebnis = 0 oder Exponentenun-
terlauf:
 $\langle Y \rangle := 0$

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Operand

W3. $\langle D \rangle :=$ undefiniert

$\langle D \rangle_t := 0$

W4. $\langle n \rangle :=$ undefiniert, $\langle n \rangle_m$ bleibt
erhalten

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ undefiniert

evtl. A-Alarm

Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

W3. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

Tk-Alarm

Ausführungszeit: Fall A: 1,85 μ s

Fall B: 1,85 μ s

Fall C: 0,10 μ s

GLEITKOMMA DIVIDIERE

op		adr
GDV		n

'64'	+		AQDYM
Intern- code	R	BW	RW

Wirkung:

- W1. $n := n + \text{mod}2$
- W2. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$ und $\langle n \rangle_{1,40} \neq \pm 0$ Normalfall

- W3. $\langle D \rangle := +0$
 $\langle D \rangle_t := 0$
- W4. $\langle A \rangle := \langle A \rangle : \langle n \rangle$, das Ergebnis ist
normalisiert und gerundet
- W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$
- W6. $\langle Y \rangle :=$ Anzahl der Binärstellen um
die das Ergebnis normali-
siert wurde
falls Dividend $\neq \pm 0$ oder Exponenten-
unterlauf:
 $\langle Y \rangle := +0$
- W7. falls Dividend $\neq \pm 0$ und Exponenten-
überlauf:
A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$ und $\langle n \rangle_{1,40} \neq \pm 0$

über- oder unter-
gelaufener Dividend

- W3. $\langle D \rangle := +0$
 $\langle D \rangle_t := 0$
- W4. $\langle A \rangle := \text{undefiniert}$
- W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$
- W6. $\langle Y \rangle := \text{undefiniert}$
 evtl. A-Alarm

Fall C: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle n \rangle_{1,40} = \pm 0$

Divisor = 0

- W3. $\langle D \rangle := \langle A \rangle$
 $\langle D \rangle_t := 0$
- W4. $\langle A \rangle := +0$
- W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$
- W6. $\langle Y \rangle := +0$
 A-Alarm

Fall D: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

- W3. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$
- W4. $\langle Q \rangle := \langle A \rangle$
 $\langle Q \rangle_t := \langle A \rangle_t$
 Tk-Alarm

Ausführungszeit: Fall A: 10,65 μs
 Fall B: 10,65 μs
 Fall C: 0,15 μs
 Fall D: 0,10 μs

GLEITKOMMA DIVIDIERE INVERS

op	adr
GDVI	n

'66'	+		AQDYM
Intern- code	R	BW	RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$ und

$\langle A \rangle_{1,40} \neq \pm 0$

Normalfall

W3. $\langle D \rangle := +0$
 $\langle D \rangle_t := 0$

W4. $\langle A \rangle := \langle n \rangle : \langle A \rangle$, das Ergebnis
ist normalisiert und ge-
rundet

W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ Anzahl der Binärstellen
um die das Ergebnis nor-
malisiert wurde
falls Dividend $(\langle n \rangle) = \pm 0$ oder
Exponentenunterlauf:
 $\langle Y \rangle := +0$

W7. falls Dividend $(\langle n \rangle) \neq \pm 0$ und
Exponentenüberlauf:
A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$ und

$$\frac{\langle A \rangle_{1,40} \neq \pm 0}{}$$

- W3. $\langle D \rangle := +0$
 $\langle D \rangle_t := 0$
- W4. $\langle A \rangle := \text{undefiniert}$
- W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$
- W6. $\langle Y \rangle := \text{undefiniert}$
 evtl. A-Alarm

über- oder unter-
gelaufener Divisor

Fall C: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_{1,40} = \pm 0$

- W3. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$
- W4. $\langle A \rangle := +0$
- W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$
- W6. $\langle Y \rangle := +0$
 A-Alarm

Divisor = 0

Fall D: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

- W3. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$
- W4. $\langle Q \rangle := \langle A \rangle$
 $\langle Q \rangle_t := \langle A \rangle_t$
 Tk-Alarm

falsche Tk

Ausführungszeit: Fall A: 10,8 μs

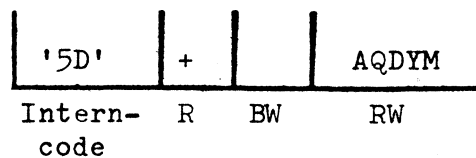
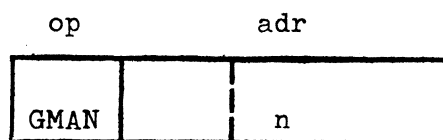
Fall B: 10,8 μs

Fall C: 0,15 μs

Fall D: 0,10 μs

GMAN

GLEITKOMMA MULTIPLIZIERE AKKUMULIEREND NEGATIV



Wirkung: wie GMLA, nur ersetze man bei Variante 1:

W4. $\langle A \rangle := \langle A \rangle \cdot \langle n \rangle$

durch

W4. $\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle$

Ausführungszeit: wie bei GMLA

GLEITKOMMA MULTIPLIZIERE

op		adr
GML		n

'5E'	+		AQDYM
Intern- code	R	BW	RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := \langle A \rangle \cdot \langle n \rangle$, das Ergebnis ist gerundet; bei normalisierten Faktoren auch normalisiert.

W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$

W6. falls Ergebnis nicht normalisiert wurde oder Mantisse eines Faktors $= \pm 0$ oder Exponentenunterlauf:
 $\langle Y \rangle := +0$
 sonst
 $\langle Y \rangle := 4$

W7. falls Exponentenüberlauf und Mantissen beider Faktoren $\neq \pm 0$:
 A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Operand

W4. $\langle A \rangle :=$ undefiniert

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ undefiniert

evtl. A-Alarm

Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

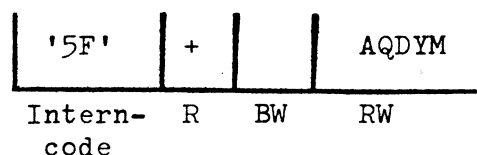
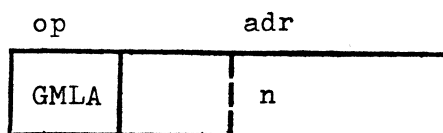
W4. Tk-Alarm

Ausführungszeit: Fall A: 2,7 μ s

Fall B: 2,7 μ s

Fall C: 0,1 μ s

GLEITKOMMA MULTIPLIZIERE AKKUMULIEREND



Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Variante 1: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

W4. $\langle A \rangle := \langle A \rangle \cdot \langle n \rangle$, das Ergebnis ist gerundet, bei normalisierten Faktoren auch normalisiert

W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$

Fall A: (Exponent des Produkts nicht übergelaufen oder Mantisse eines Faktors = ± 0) und $\langle H \rangle_t = 0$ und $\langle H \rangle_1 = \langle H \rangle_2$

Normalfall

W6. $\langle A \rangle := \langle A \rangle + \langle H \rangle$, das Ergebnis ist normalisiert, bei normalisierten Operanden auch gerundet

W7. $\langle Y \rangle :=$ Anzahl der Binärstellen, um die
das Ergebnis der Addition nor-
malisiert wurde,
falls Ergebnis = 0 oder Expo-
nentenunterlauf:
 $\langle Y \rangle := +0$

W8. falls Exponent des Ergebnisses der
Addition übergelaufen:
A-Alarm

Fall B: (Exponent des Produkts nicht übergelaufen oder
Mantisse eines Faktors = ± 0) und $\langle H \rangle_t = 0$ und
 $\langle H \rangle_1 \neq \langle H \rangle_2$

H übergelaufen

W6. $\langle A \rangle :=$ undefiniert
W7. $\langle Y \rangle :=$ undefiniert
W8. evtl. A-Alarm

Fall C: (Exponent des Produkts nicht übergelaufen oder
Mantisse eines Faktors = ± 0) und $\langle H \rangle_t \neq 0$

Falsche Tk in H

W6. falls Produkt nicht normalisiert wurde
oder Mantisse eines Faktors = ± 0 oder
Exponentenunterlauf:
 $\langle Y \rangle := +0$
sonst
 $\langle Y \rangle := 4$
W7. Tk-Alarm

Fall D: Exponent des Produkts übergelaufen und
Mantissen beider Faktoren $\neq \pm 0$

W6. wie bei Fall C
W7. A-Alarm

Variante 2: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Operand

W4. $\langle A \rangle := \text{undefiniert}$

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle := \text{undefiniert}$

W7. evtl. A-Alarm oder Tk-Alarm

Variante 3: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

W4. Tk-Alarm

Ausführungszeit: Variante 1 und 2: Falls akkumuliert wurde: 4,85 μs
Falls nicht akkumuliert wurde: 2,70 μs
Variante 3: 0,10 μs

GLEITKOMMA MULTIPLIZIERE NEGATIV

op	adr
GMLN	n

'5C'	+		AQDYM
Intern- code	R	BW	RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle$, das Ergebnis
ist gerundet, bei normalisierten
Faktoren auch normalisiert

W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$

W6. falls Ergebnis nicht normalisiert
wurde oder Mantisse eines Faktors
 $= \pm 0$ oder Exponentenunterlauf:

$\langle Y \rangle := 0$

sonst

$\langle Y \rangle := 4$

W7. falls Exponentenüberlauf und Man-
tissen beider Faktoren $\neq 0$:

A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W4. $\langle A \rangle :=$ undefiniert

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ undefiniert
evtl. A-Alarm

über- oder unter-
gelaufener Operand

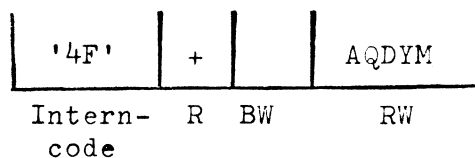
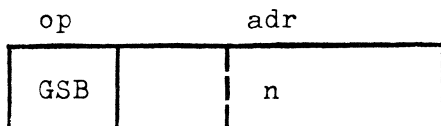
Fall C: $A_t \neq 0$ oder $n_t \neq 0$

W4. Tk-Alarm

falsche Tk

Ausführungszeit: Fall A: 2,7 μ s
Fall B: 2,7 μ s
Fall C: 0,1 μ s

GLEITKOMMA SUBTRAHIERE



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahlwörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := \langle A \rangle - \langle n \rangle$, das Ergebnis ist normalisiert

Falls beide Operanden normalisiert waren, ist das Ergebnis gerundet. Bei nicht normalisierten Operanden braucht dies nicht zu gelten, auch kann dann das Ergebnis von einer Stelle an (bis zu der es korrekt ist) nur noch vorzeichengleiche (statt berechneter) Stellen enthalten.

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ Anzahl der Binärstellen, um die das Ergebnis normalisiert wurde,

falls Ergebnis = 0 oder Exponentenunterlauf:

$\langle Y \rangle := +0$

W7. falls Exponent des Ergebnisses
übergelaufen:

A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Operand

W4. $\langle A \rangle :=$ undefiniert

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ undefiniert

evtl. A-Alarm

Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

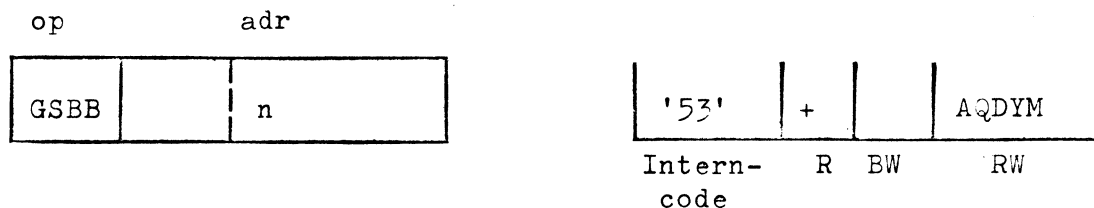
W4. Tk-Alarm

Ausführungszeit: Fall A: 1,4 μ s

Fall B: 1,4 μ s

Fall C: 0,1 μ s

GLEITKOMMA SUBTRAHIERE BETRAG



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := \langle A \rangle - |\langle n \rangle|$, das Ergeb-
nis ist normalisiert

Falls beide Operanden normalisiert
waren, ist das Ergebnis gerundet.
Bei nicht normalisierten Operanden
braucht dies nicht zu gelten, auch
kann dann das Ergebnis von einer
Stelle an (bis zu der es korrekt
ist) nur noch vorzeichengleiche
(statt berechneter) Stellen enthalten.

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ Anzahl der Binärstellen, um
die das Ergebnis normalisiert
wurde,

falls Ergebnis = 0 oder Exponentenun-
terlauf:

$\langle Y \rangle := +0$

W7. falls Exponent des Ergebnisses über-
gelaufen: A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Operand

W4. $\langle A \rangle :=$ undefiniert

W5, $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ undefiniert

evtl. A-Alarm

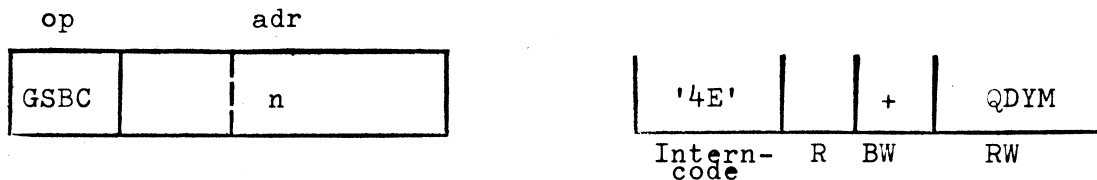
Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

W4. Tk-Alarm

Ausführungszeit: Fall A: 1,4 μ s
Fall B: 1,4 μ s
Fall C: 0,1 μ s

GLEITKOMMA SUBTRAHIERE IM SPEICHER



Wirkung: W1. $n := n + \text{mod}2$

W2. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W3. $\langle D \rangle := \langle n \rangle - \langle A \rangle$, das Ergeb-
nis ist normalisiert

Falls beide Operanden normalisiert
waren, ist das Ergebnis gerundet.
Bei nicht normalisierten Operanden
braucht dies nicht zu gelten, auch
kann dann das Ergebnis von einer
Stelle an (bis zu der es korrekt
ist) nur noch vorzeichengleiche
(statt berechneter) Stellen enthalten.

W4. falls Exponent des Ergebnisses nicht
übergelaufen:

$\langle n \rangle_1 := \langle n \rangle_1$
 $\langle n \rangle_{2,48} := \langle D \rangle_{2,48}$

falls Exponent des Ergebnisses über-
gelaufen:

A-Alarm

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ Anzahl der Binärstellen, um
die das Ergebnis normalisiert
wurde,

falls Ergebnis = 0 oder Exponentenun-
terlauf:

$\langle Y \rangle := +0$

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener Ope-
rand

W3. $\langle D \rangle :=$ undefiniert

$\langle D \rangle_t := 0$

W4. $\langle n \rangle :=$ undefiniert, $\langle n \rangle_m$ bleibt
erhalten.

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ undefiniert

evtl. A-Alarm

Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

W3. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

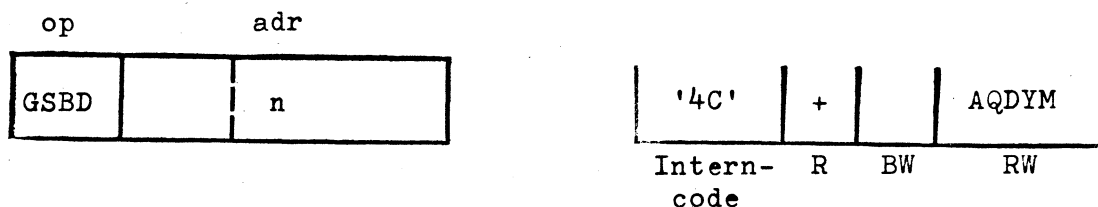
Tk-Alarm

Ausführungszeit: Fall A: 1,85 μ s

Fall B: 1,85 μ s

Fall C: 0,10 μ s

GLEITKOMMA SUBTRAHIERE VON D



Wirkung: W1. $n := n + \text{mod}2$

W2. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A: $\langle D \rangle_t = \langle n \rangle_t = 0$ und $\langle D \rangle_1 = \langle D \rangle_2$

Normalfall

W3. $\langle A \rangle := \langle D \rangle - \langle n \rangle$, das Ergeb-
 nis ist normalisiert

Falls beide Operanden normalisiert
 waren, ist das Ergebnis gerundet.
 Bei nicht normalisierten Operanden
 braucht dies nicht zu gelten, auch
 kann dann das Ergebnis von einer
 Stelle an (bis zu der es korrekt
 ist) nur noch vorzeichengleiche
 (statt berechneter) Stellen enthalten.

W4. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$

W5. $\langle Y \rangle :=$ Anzahl der Binärstellen,
 um die das Ergebnis nor-
 malisiert wurde,
 falls Ergebnis = 0 oder Exponenten-
 unterlauf:
 $\langle Y \rangle := +0$

W6. falls Exponent des Ergebnisses
übergelaufen:

A-Alarm

Fall B: $\langle D \rangle_t = \langle n \rangle_t = 0$ und $\langle D \rangle_1 \neq \langle D \rangle_2$

über- oder unterge-
laufener Operand

W3. $\langle A \rangle :=$ undefiniert

W4. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W5. $\langle Y \rangle :=$ undefiniert

evtl. A-Alarm

Fall C: $\langle D \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

W3. $\langle A \rangle := \langle D \rangle$

W4. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

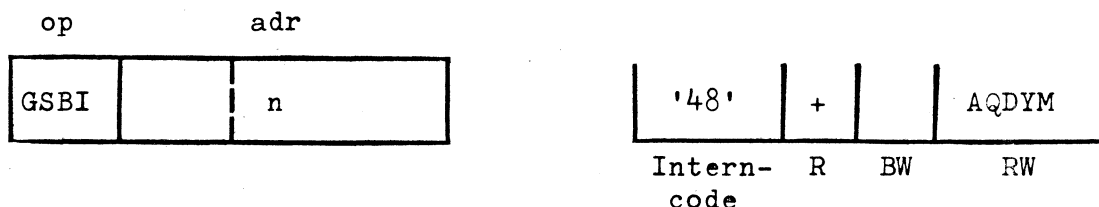
Tk-Alarm

Ausführungszeit: Fall A: 1,5 μ s

Fall B: 1,5 μ s

Fall C: 0,15 μ s

GLEITKOMMA SUBTRAHIERE INVERS



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei
Zahlwörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := \langle n \rangle - \langle A \rangle$, das Ergebnis ist normalisiert

Falls beide Operanden normalisiert waren, ist das Ergebnis gerundet. Bei nicht normalisierten Operanden braucht dies nicht zu gelten, auch kann dann das Ergebnis von einer Stelle an (bis zu der es korrekt ist) nur noch vorzeichengleiche (statt berechneter) Stellen enthalten.

W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ Anzahl der Binärstellen, um die das Ergebnis normalisiert wurde,

falls Ergebnis = 0 oder Exponentenunterlauf:

$\langle Y \rangle := +0$

W7. falls Exponent des Ergebnisses über-
gelaufen:

A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unter-
gelaufener
Operand

W4. $\langle A \rangle :=$ undefiniert

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ undefiniert

evtl. A-Alarm

Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

falsche Tk

W4. $\langle A \rangle_{1,40} := \overline{\langle A \rangle_{1,40}}$

Tk-Alarm

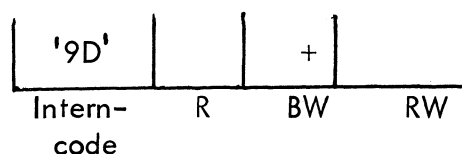
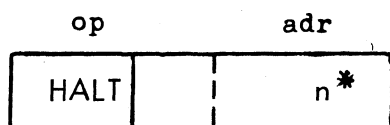
Ausführungszeit: Fall A: 1,4 μ s

Fall B: 1,4 μ s

Fall C: 0,1 μ s

HALT

HALT



Voraussetzung: BEBY = L oder BEBT = BEWA = L , sonst Makro

Wirkung: W1. BEKH := L

W2. Das bewirkt, daß so lange eine Schleife der Abrufphase des nächsten Befehls durchlaufen wird, bis BEKH gelöscht wird. (BEKH kann u. a. durch die WEITER-Taste gelöscht werden.)

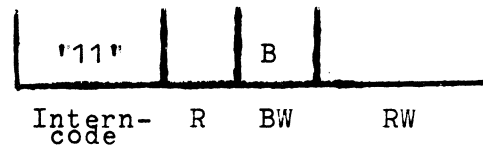
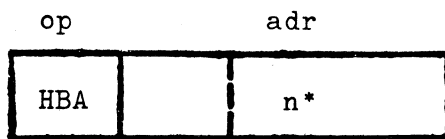
Während die Schleife durchlaufen wird (während der Rechner hält), steht der nächste Befehl in BC und BA

W3. Wird die WEITER-Taste gedrückt, dann wird der Befehl ausgeführt, der auf HALT folgt; das Programm wird also normal fortgesetzt.

Ausführungszeit: 0,10 μ s

HBA

ERHÖHE B UM ADRESSTEIL

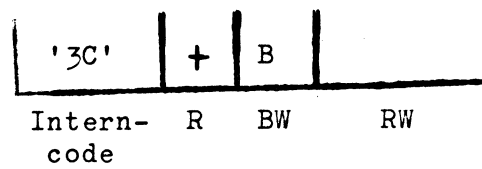
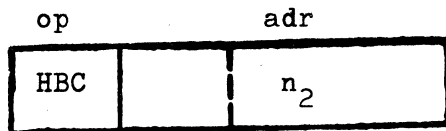


Wirkung: W1. $\langle B \rangle := \langle B \rangle + n^*$

Ausführungszeit: 0,375 μ s

HBC

ERHÖHE B UM SPEICHER

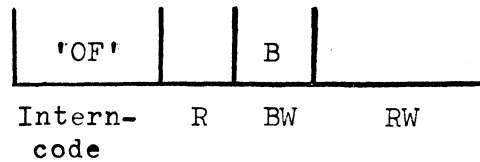
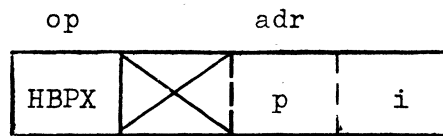


Wirkung: W1. $\langle B \rangle := \langle B \rangle + \langle n_2 \rangle$

Ausführungszeit: 0,325 μ s

HBPX

ERHÖHE B UM PARAMETER MAL INDEXZELLE



Wirkung:

Fall A: $-15 \leq p \leq 15$

W1. $\langle B \rangle := \langle B \rangle + p \cdot \langle i \rangle$

Die Verknüpfung geschieht modulo $2^{24}-1$

Fall B: $|p| > 15$

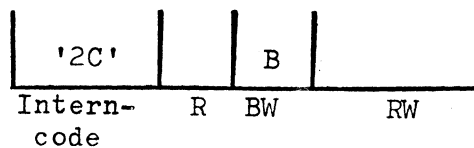
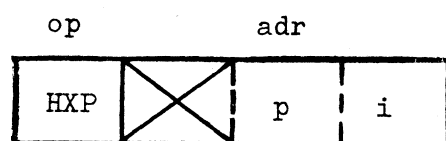
W1. entfällt

Ausführungszeit: Fall A: $(0,275 |p| + 0,15) \mu s$

Fall B: $0,10 \mu s$

HXP

ERHÖHE INDEXZELLE UM PARAMETER



Wirkung:

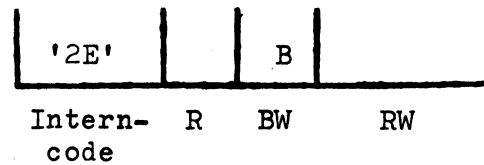
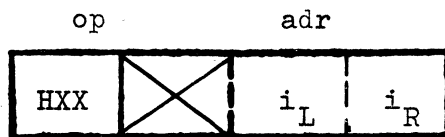
W1. $\langle i \rangle := \langle i \rangle + p$
 $-127 \leq p \leq 127$

W2. $\langle B \rangle := \langle i \rangle$

Ausführungszeit: 0,525 μ s

HXX

ERHÖHE INDEXZELLE UM INDEXZELLE



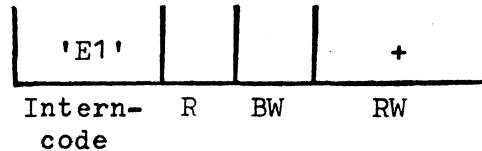
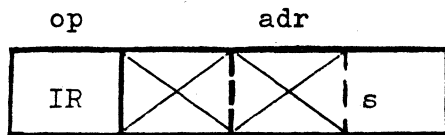
Wirkung:

W1. $\langle i_R \rangle := \langle i_R \rangle + \langle i_L \rangle$

W2. $\langle B \rangle := \langle i_R \rangle$

Ausführungszeit: 0,725 μ s

INVERTIERE REGISTER



Adreßteil: $s = (AvQvDvH)[B]$

B: Betrag

Wirkung:

Fall A: $s = [AvQvDvH]$

- W1. Der Inhalt der adressierten Register wird invertiert. Bei Tk 0 werden nur die linken 40 Bits (Mantisse) invertiert. Sonst alle 48 Bits.

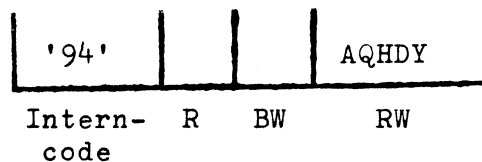
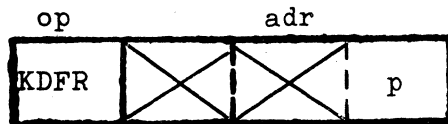
Fall B: $s = [AvQvDvH] B$

- W1. Der Inhalt der adressierten Register wird betragsmäßig hergestellt. Bei Tk 0 und Tk 1 gilt nur das erste (linke) Bit als Vorzeichen. Bei Tk 0 bleiben die rechten 8 Bits (Exponent) erhalten. Bei Tk 2 und Tk 3 keine Wirkung.

Ausführungszeit: $(0,20p + 0,05) \mu s$
wobei p=Anzahl der adressierten Register, jedoch mindestens 1

KDFR

KONVERTIERE DEZIMALZAHL IN FESTKOMMAZAHL RECHTSBÜNDIG



Eingangsgrößen: Im doppelt langen Register A,Q steht rechtsbündig die zu konvertierende ganze positive Dezimalzahl d in p Tetraden. Die linken Tetraden sind beliebig und ohne Bedeutung. Die Tk der beiden Register ist beliebig.

$$\langle A, Q \rangle = t_{23} t_{22} \dots t_{p-1} \dots t_0$$

$$d = t_{p-1} \cdot 10^{p-1} + \dots + t_0$$

Wirkung: W1. $adr := adr + mod2$

$$p := adr_{17,24}$$

W2. $\langle A \rangle_t, \langle Q \rangle_t, \langle D \rangle_t, \langle H \rangle_t := 1$
 $\langle Q \rangle, \langle D \rangle, \langle H \rangle := +0$
 $\langle Y \rangle := +0$

Fall A: $1 \leq p \leq 13$

W3. $\langle A \rangle := d$ in dualer Darstellg.

Normalfall

Fall B: $p < 1$ oder $p > 13$.

W3. $\langle A \rangle$ ist nicht definiert

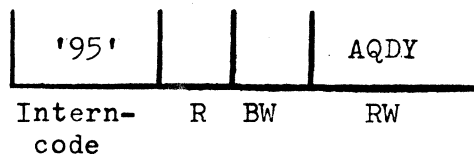
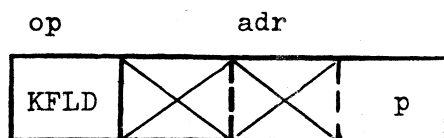
Sonderfall

Bemerkung zu Fall A: Der Befehl wirkt auch wie beschrieben, falls $10 \leq t \leq 15$ ist.

Ausführungszeit: $(0,35p + 1,10) \mu s$

KFLD

KONVERTIERE FESTKOMMAZAH_L LINKSBÜNDIG IN DEZIMALZAH_L



Wirkung: W1. $\text{adr} := \text{adr} + \text{mod}2$

Adreßteil: $p = \text{adr}_{17,24}$

Fall A: $\langle A \rangle_1 = \langle A \rangle_2 = 0$ und $\langle A \rangle > 0$ und $\langle A \rangle_t \leq 1$ und $1 \leq p \leq 13$

Normalfall

W2. $\langle A \rangle$ wird als positiver Dualbruch $b > 0$ aufgefaßt (Komma zwischen $\langle A \rangle_2$ und $\langle A \rangle_3$) und in einen Dezimalbruch d_p von p Dezimalstellen konvertiert, mit dem (gedachten) Komma unmittelbar vor der vordersten der p Dezimalstellen.

Es gilt dabei die Formel:

$$d_p = b - 10^{-p} \cdot r_p$$

r_p ist der Konvertierungsrest

$r_p \geq 0$, positiver Dualbruch mit Komma hinter den Vorzeichenstellen.

$$\langle A, Q \rangle := d_p$$

$$\langle A \rangle_t, \langle Q \rangle_t := 1$$

$$\langle D \rangle := r_p$$

$$\langle D \rangle_t := 1$$

$$\langle Y \rangle := 0$$

Fall B: $\langle A \rangle = +0, \langle A \rangle_t \leq 1$

W2. $\langle A \rangle, \langle Q \rangle, \langle D \rangle := +0$
 $\langle A \rangle_t, \langle Q \rangle_t, \langle D \rangle_t := 1$
 $\langle Y \rangle := 0$

Fall C: $\langle A \rangle_t \geq 2$

keine Zahl

W2. Tk-Alarm

Fall D: $\langle A \rangle_t \leq 1$ und ($\langle A \rangle \leq -0$ oder $\langle A \rangle_1 \neq \langle A \rangle_2$)

falsche Zahl

W2. A-Alarm

Fall E: ($p \leq 0$ oder $p \geq 14$) und $\langle A \rangle_1 = \langle A \rangle_2 = 0$ und
 $\langle A \rangle > +0$ und $\langle A \rangle_t \leq 1$

falsches p

W2. $\langle A \rangle_t, \langle Q \rangle_t, \langle D \rangle_t := 1$
 $\langle A \rangle, \langle Q \rangle, \langle D \rangle := \text{undefiniert}$
 $\langle Y \rangle := +0$
A-Alarm

Ausführungszeiten:

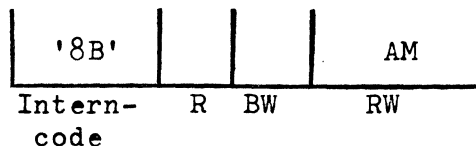
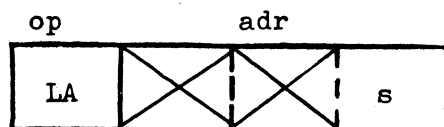
Fall A: $(0,25 + 0,45 p) \mu s$

Fall B,C,D: $0,10 \mu s$

Fall E: unbestimmt

LA

LÖSCHE IN A



Wirkung: W1. $s := \text{adr} + \text{mod}2$

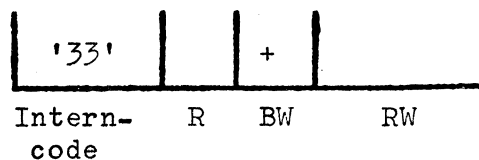
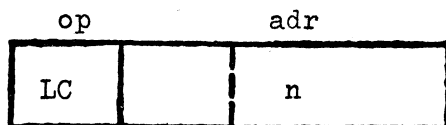
Adreßteil: $s = [Vv2vFv3vEvM] | H[M] | T[M]$

W2.	$\langle A \rangle_{1,2}$	$:= 0$	falls s	V enthält	<u>V</u> orzeichen löschen
	$\langle A \rangle_{1,24}$	$:= 0$	falls s	2 enthält	linkes <u>H</u> albwort löschen
	$\langle A \rangle_{1,40}$	$:= 0$	falls s	F enthält	Mantisse löschen
	$\langle A \rangle_{33,48}$	$:= 0$	falls s	3 enthält	rechtes <u>D</u> rittelwort löschen
	$\langle A \rangle_{41,48}$	$:= 0$	falls s	E enthält	<u>E</u> xponent löschen
	$\langle A \rangle_{1,42}$	$:= 0$	falls s	H enthält	nur rechte <u>H</u> exade bleibt
	$\langle A \rangle_{1,44}$	$:= 0$	falls s	T enthält	nur rechte <u>T</u> etrade bleibt
	$\langle M \rangle$	$:= 0$	falls s	M enthält	<u>M</u> arke löschen
	entfällt		falls s	leer	

Ausführungszeit: 0,20 μ s

LC

LÖSCHE SPEICHER



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle n \rangle_t \geq 2$

W2. $\langle n \rangle_{1,48} := +0$

Fall B: $\langle n \rangle_t \leq 1$

W2. $\langle n \rangle_{2,48} := +0$

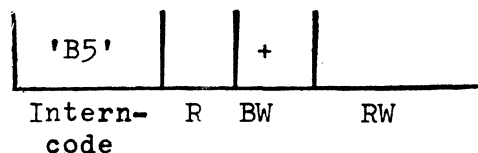
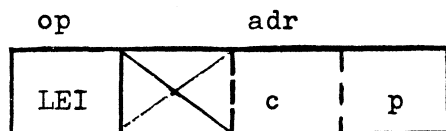
Tk bleibt
erhalten

Zahlwort:
auch Marke
bleibt erhalten

Ausführungszeit: 0,45 μs

LEI

MODIFIZIERE MIT LEITADRESSE



Wirkung:

Fall A: Der Rechner läuft entweder im System- oder Spezialmodus mit $0 \leq p \leq 255$

W1. op := c
 adr := $\langle BL \rangle \cdot 2^8 + p$

W2. Ausgang 2

Bemerkung: Die Adressen werden absolut genommen (Kacheladresse).

Fall B: Der Rechner läuft im Normalmodus oder im Abwicklermodus

W1. Makro

Ausführungszeit: Fall A: 0,50 μ s

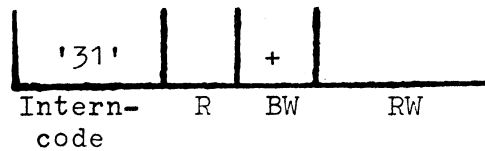
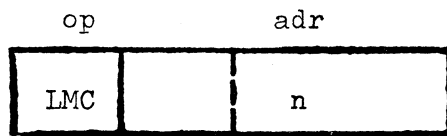
Definition d. Befehls:

c	$\langle BL \rangle \cdot 2^8 + p$
---	------------------------------------

BL = Leitadreßregister

LMC

LÖSCHE MARKE IM SPEICHER



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle n \rangle_t \leq 1$

W2. $\langle n \rangle_1 := 0$

Fall B: $\langle n \rangle_t \geq 2$

W2. Tk-Alarm

Zahlwort

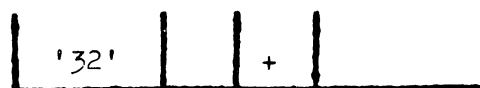
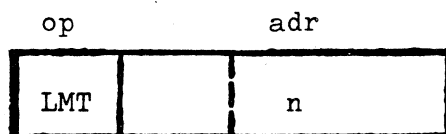
Markenstelle

Ausführungszeit: Fall A = 0,35 μs

Fall B = 0,05 μs

LMT

LÖSCHE MARKIERT



Intern- R BW RW
code

Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle n \rangle_t \leq 1$

W2. $\langle n \rangle_{2,48} := 0$
 $\langle n \rangle_1 := L$

Fall B: $\langle n \rangle_t \geq 2$

W2. Tk-Alarm

Normalfall:
Zahlwort

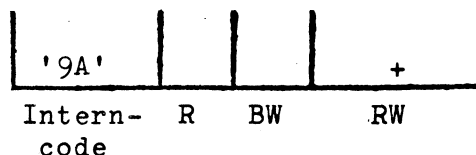
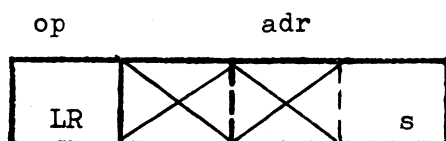
Markenstelle

Ausführungszeit: Fall A = 0,45 μ s

Fall B = 0,05 μ s

LR

LÖSCHE REGISTER



Wirkung: W1. $s := s + \text{mod}2$

Adreßteil: $s := s_1 s_2 ; s_1 = 0|1|2|3$
 $s_2 = [A \ v \ Q \ v \ H \ v \ D]$

Fall A: $s_2 = A \ v \ Q \ v \ H \ v \ D$

W2. $\langle s_2 \rangle_t := s_1$
 $\langle s_2 \rangle := +0$

Fall B: s_2 leer

(kein Register adressiert)

W2. entfällt

Typenkennung

Register

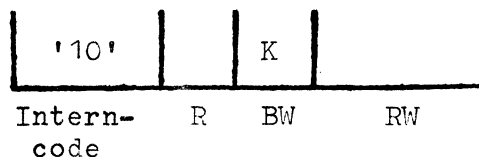
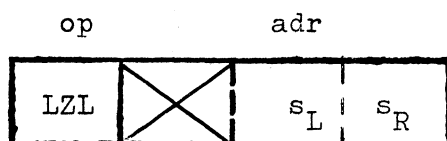
Mindestens
1 Register adressiert

Die angegebenen Register
werden auf +0
gelöscht, Tk : s_1

Ausführungszeit: 0,15 μs

LZL

LÖSCHE UND SETZE MERKLICHTER



Adressteil: s_L : 0v1v2v3v...v8
 s_R : 0v1v2v3v...v8

Wirkung: W1. Die durch s_R bezeichneten
 Merklichter werden gelöscht

 W2. Die durch s_L bezeichneten
 Merklichter werden gesetzt

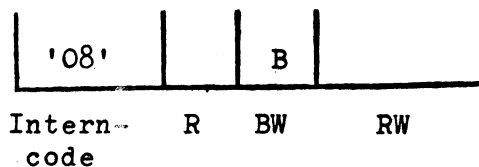
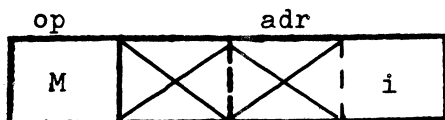
Merklichter 1 bis 8
 in beliebiger Kom-
 bination

0 bedeutet kein
 Merklicht

Ausführungszeit: 0,15 µs

M

MODIFIZIERE

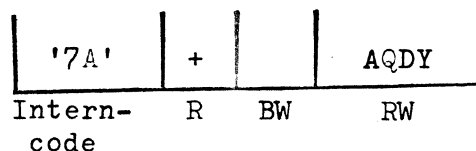
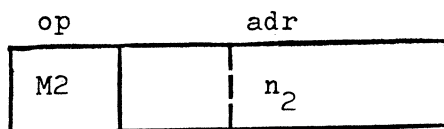


Wirkung: W1. mod2 := <i> + mod2
 W2. := mod2

Bemerkung: Interne Darstellung von M :

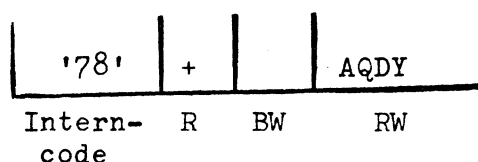
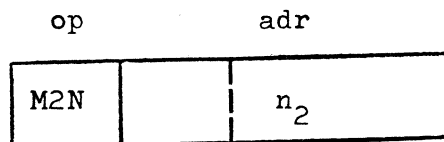
Die Linksadresse muß aus folgenden
 Bits bestehen: LOOXXXXX (X beliebig).
 Andernfalls falsche Befehlsausführung.

Ausführungszeit: 0,05 µs

MULTIPLIZIERE HALBWORT

- Wirkung:
- W1. $n_2 := n_2 + \text{mod}2$
 - W2. Es wird $\langle A \rangle$ mit $\langle n_2 \rangle$ multipliziert wie bei Tk 1 und als wäre $\langle n_2 \rangle_2$ gleich $\langle n_2 \rangle_1$, dh. $\langle n_2 \rangle_2$ ohne Bedeutung; $\langle n_2 \rangle$ und $\langle A \rangle$ werden als ganze Zahlen aufgefaßt, $\langle A \rangle_1$: Vorzeichenstelle, $\langle A \rangle_{2,48}$: Ziffernstellen.
 - W3. $\langle A \rangle :=$ rechte 48 Bits des Ergebnisses; die vorderen 21 gehen verloren.
Falls $|\text{Ergebnis}| \geq 2^{46}$:
A-Alarm
 - W4. $\langle D \rangle := +0$
 $\langle D \rangle_t := \langle A \rangle_t$
 - W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := \langle A \rangle_t$
 - W6. $\langle Y \rangle := +0$

Ausführungszeit: 2,5 μ s

MULTIPLIZIERE HALBWORT NEGATIVWirkung:

- W1. $n_2 := n_2 + \text{mod}2$
- W2. Es wird $-\langle A \rangle$ mit $\langle n_2 \rangle$ multipliziert wie bei Tk 1 und als wäre $\langle n_2 \rangle_2$ gleich $\langle n_2 \rangle_1$, d.h. $\langle n_2 \rangle_2$ ohne Bedeutung; $\langle n_2 \rangle$ und $\langle A \rangle$ werden als ganze Zahlen aufgefaßt, $\langle A \rangle_1$: Vorzeichenstelle, $\langle A \rangle_{2,48}$: Ziffernstellen.
- W3. $\langle A \rangle :=$ rechte 48 Bits des Ergebnisses; die vorderen 21 gehen verloren.
 Falls $|\text{Ergebnis}| \geq 2^{46}$:
 A-Alarm
- W4. $\langle D \rangle := +0$
 $\langle D \rangle_t := \langle A \rangle_t$
- W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := \langle A \rangle_t$
- W6. $\langle Y \rangle := +0$

Ausführungszeit: 2,5 μs

M2NR

MULTIPLIZIERE HALBWORT NEGATIV MIT RUNDUNG

op	adr
M2NR	n_2

'79'	+		AQDY
Intern- code	R	BW	RW

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Alles geschieht wie bei $\langle A \rangle_t = \langle n_2 \rangle_t = 1$ und wie wenn

$\langle n_2 \rangle_2 := \langle n_2 \rangle_1$ ($\langle n_2 \rangle_2$ ist ohne Bedeutung).

$\langle A \rangle$, falls $\langle A \rangle_1 = \langle A \rangle_2$, und $\langle n_2 \rangle$ werden als echte Brüche aufgefaßt.

Falls $\langle A \rangle_1 \neq \langle A \rangle_2$, wird $\langle A \rangle$ als Zahl z mit $1 \leq |z| < 2$ aufgefaßt; $\langle A \rangle_1$ allein gilt dann als Vorzeichen.

W2. $\langle A \rangle := -\langle A \rangle \cdot \langle n_2 \rangle$ gerundet

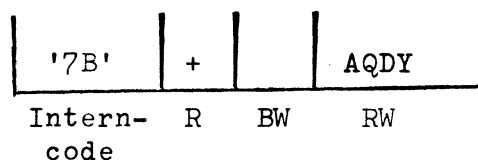
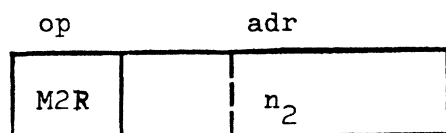
W3. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := \langle A \rangle_t$

W4. $\langle D \rangle := +0$
 $\langle D \rangle_t := \langle A \rangle_t$

W5. $\langle Y \rangle := +0$

Bemerkung: Nur wenn zu Beginn $\langle A \rangle_1 \neq \langle A \rangle_2$ war, kann dies auch für das Ergebnis gelten.

Ausführungszeit: 1,825 μ s

MULTIPLIZIERE HALBWORT MIT RUNDUNGWirkung:

W1. $n_2 := n_2 + \text{mod}2$

Alles geschieht wie bei $\langle A \rangle_t = \langle n_2 \rangle_t = 1$ und wie wenn

$\langle n_2 \rangle_2 := \langle n_2 \rangle_1$ ($\langle n_2 \rangle_2$ ist ohne Bedeutung).

$\langle A \rangle$, falls $\langle A \rangle_1 = \langle A \rangle_2$, und $\langle n_2 \rangle$ werden als echte Brüche aufgefaßt.

Falls $\langle A \rangle_1 \neq \langle A \rangle_2$, wird $\langle A \rangle$ als Zahl z mit $1 < |z| < 2$ aufgefaßt;

$\langle A \rangle_1$ allein gilt dann als Vorzeichen.

W2. $\langle A \rangle := \langle A \rangle \cdot \langle n_2 \rangle$ gerundet

W3. $\langle Q \rangle := +0$

$\langle Q \rangle_t := \langle A \rangle_t$

W4. $\langle D \rangle := +0$

$\langle D \rangle_t := \langle A \rangle_t$

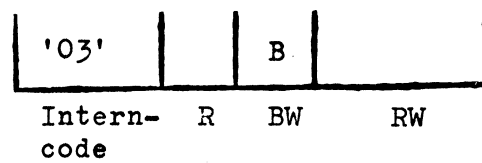
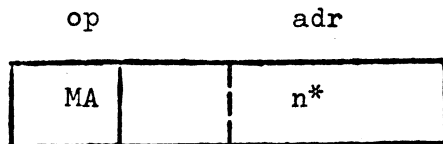
W5. $\langle Y \rangle := +0$

Bemerkung: Nur wenn zu Beginn $\langle A \rangle_1 \neq \langle A \rangle_2$ war, kann dies auch für das Ergebnis gelten.

Ausführungszeit: 1,825 μs

MA

MODIFIZIERE MIT ADRESSTEIL

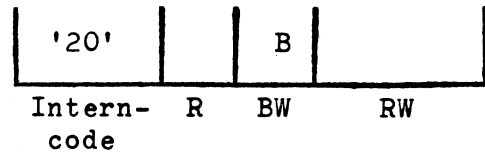
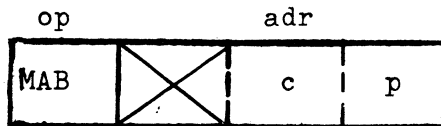


Wirkung: W1. $\text{mod2} := n^* + \text{mod2}$
W2. $\langle B \rangle := \text{mod2}$

Ausführungszeit: 0,05 μ s

MAB

MODIFIZIERE ADRESSTEIL MIT B



Wirkung:

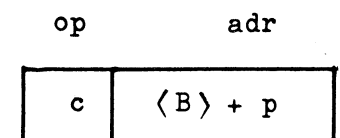
W1. mod2 := 0

W2. op := c
adr := $\langle B \rangle + p$
 $-127 \leq p \leq 127$

W3. $\langle B \rangle := \langle B \rangle + p$

W4. Ausgang 2

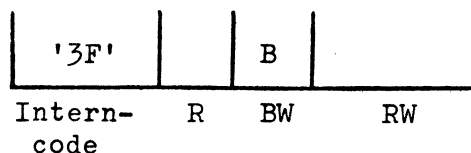
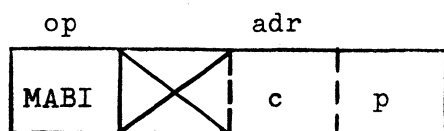
Definition des Befehls



Ausführungszeit: 0,775 μ s

MABI

MODIFIZIERE ADRESSTEIL MIT B BEI INVARIANZ DER SPRUNGADRESSE



Wirkung:

W1. mod2 := 0

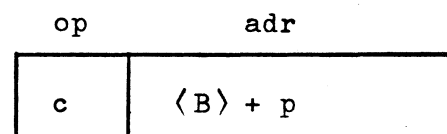
W2. op := c

adr := $\langle B \rangle + p$
 $-127 \leq p \leq 127$

W3. $\langle B \rangle := \langle B \rangle + p$

W4. Ausgang 2

Definition des Befehls:



Bemerkung:

Ist einer der Befehle S, SFB, SU, VSS oder ein bedingter Sprungbefehl mit erfüllter Sprungbedingung, so gilt dort $\langle F \rangle$ anstelle von $\langle F \rangle_{9,24}$.

Sprung in andere Großseite ist also möglich.

PDP und TLOG gelten nicht als Sprungbefehle.

Ausführungszeit: 0,775 μ s

MULTIPLIZIERE AKKUMULIEREND NEGATIV

op	adr
MAN	n

'5A'	+		AQDYM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W4. falls $\langle H \rangle_t \neq 1$ oder $\langle Q \rangle_t \neq 1$
 Tk-Alarm

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W5. $\langle A, Q \rangle := -\langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$
 $\langle Q \rangle_t := 1$
 $\langle Y \rangle := +0$

W7. falls $\langle A \rangle_1 \neq \langle A \rangle_2$
 A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W5. $\langle A \rangle, \langle Q \rangle := \text{undefiniert}$
 $\langle Q \rangle_t := 1$

W6. $\langle Y \rangle := +0$

W7. falls $\langle A \rangle_1 \neq \langle A \rangle_2$
 A-Alarm

Marke bei Zahlwörtern

Normalfall

die Addition geschieht,
 als ob $\langle H \rangle_t = \langle Q \rangle_t = 1$

über- oder untergelaufenes
 Ergebnis

über- oder untergelaufener
 Operand

Fall C: $\langle A \rangle_t \neq 1$ oder $\langle n \rangle_t \neq 1$

falsche Tk

W5. Tk-Alarm

Bemerkung: $\langle Q \rangle_1 \neq \langle Q \rangle_2$ oder $\langle H \rangle_1 \neq \langle H \rangle_2$ löst keinen A-Alarm aus.
Beim Ergebnis können die Vorzeichen in A und Q
verschieden sein.

Ausführungszeit: Fall A und B: 3,30 μ s
Fall C: 0,10 μ s

MULTIPLIZIERE AKKUMULIEREND NEGATIV MIT RUNDUNG

op	adr
MANR	n

'5B'	+		AQDYM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahlwörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle$ gerundet $+\langle H \rangle$

die Addition ist
Tk-unabhängig

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 1$

$\langle Y \rangle := +0$

W6. falls $\langle H \rangle_t \neq 1$

Tk-Alarm

W7. falls $\langle A \rangle_1 \neq \langle A \rangle_2$

A-Alarm

über- oder unterge-
laufenes Ergebnis

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unterge-
laufener Operand

W4. $\langle A \rangle := \text{undefiniert}$

W5.

bis W7. wie Fall A

MAR

MULTIPLIZIERE AKKUMULIEREND MIT RUNDUNG

op	adr
MAR	n

'57'	+		AQDYM
------	---	--	-------

Intern-
code

R

BW

RW

Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahlwörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := \langle A \rangle \cdot \langle n \rangle$ gerundet + $\langle H \rangle$

die Addition ist Tk-unabhängig.

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 1$

$\langle Y \rangle := +0$

W6. falls $\langle H \rangle_t \neq 1$

Tk-Alarm

W7. falls $\langle A \rangle_1 \neq \langle A \rangle_2$

A-Alarm

über- oder untergelaufenes Ergebnis

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder untergelaufener Operand

W4. $\langle A \rangle := \text{undefiniert}$

W5. bis W7. wie Fall A

Fall C: $\langle A \rangle_t \neq 1$ oder $\langle n \rangle_t \neq 1$

Falsche Tk

W4. Tk-Alarm

Bemerkung: $\langle H \rangle_1 \neq \langle H \rangle_2$ löst keinen Alarm aus.

A := +0 nur wenn das gerundete Produkt = +0

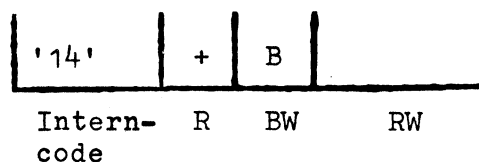
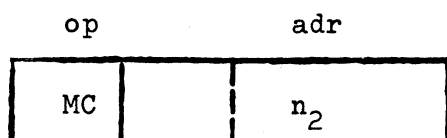
und $\langle H \rangle = +0$

Ausführungszeit: Fall A und B: 3,35 μ s

Fall C: 0,10 μ s

MC

MODIFIZIERE AUS SPEICHER

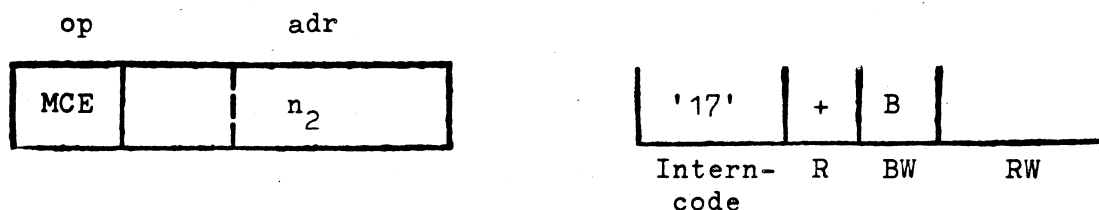


Wirkung: W1. $\text{mod2} := \langle n_2 \rangle + \text{mod2}$
W2. $\langle B \rangle := \text{mod2}$

Ausführungszeit: 0,10 μs

MCE

MODIFIZIERE AUS SPEICHER NACH ERSETZUNGEN



Wirkung: W1. $\text{mod2} := \langle \langle \dots \langle n_2 \rangle_2 \dots \rangle_2 \rangle + \text{mod2}$

Das erste Halbwort in der Ersetzkette ist $\langle n_2 \rangle$, das letzte Halbwort das, bei dem das erste Bit = L ist.

Das erste Bit wird dem 2. angeglichen und das Halbwort als 24Bit-Modifikator genommen. (nach Addition von mod2)

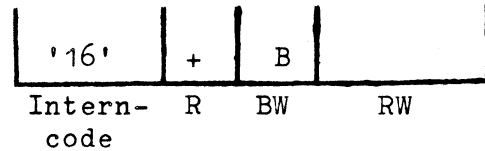
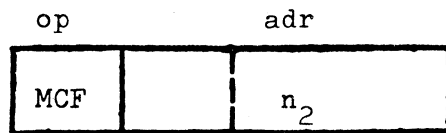
W2. $\langle B \rangle := \text{mod2}$

Ausführungszeit: (0,65k - 0,45) μs

wobei k = Anzahl der Halbwörter in der Ersetzkette

MCF

MODIFIZIERE AUS SPEICHER IN JEDEM FALLE



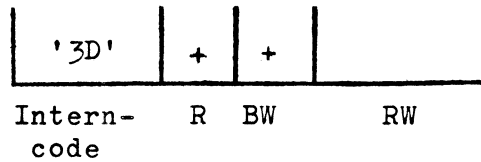
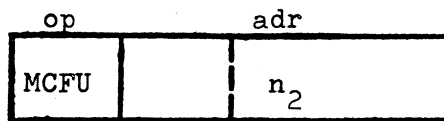
Wirkung: W1. $\text{mod1} := \langle n_2 \rangle + \text{mod2}$

W2. $\langle B \rangle := \text{mod1}$

Ausführungszeit: 0,10 μs

MCFU

MODIFIZIERE AUS SPEICHER IN JEDEM FALLE
MIT UNVERÄNDERTEM B

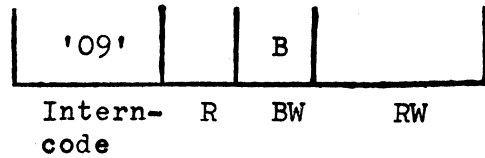
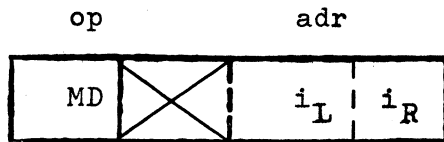


Wirkung: W1. $\text{mod1} := \langle n_2 \rangle + \text{mod2}$

Ausführungszeit: 0,05 μ s

MD

MODIFIZIERE DOPPELT

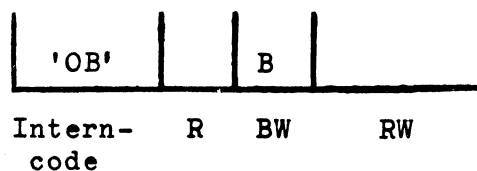
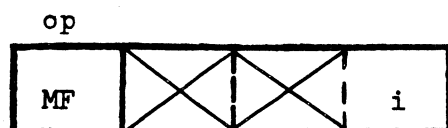


Wirkung: W1. mod1 := $\langle i_R \rangle + \text{mod2}$
 mod2 := $\langle i_L \rangle$
 W2. $\langle B \rangle := \langle i_L \rangle$

Ausführungszeit: 0,25 μ s

MF

MODIFIZIERE IN JEDEM FALLE

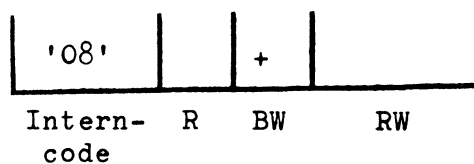
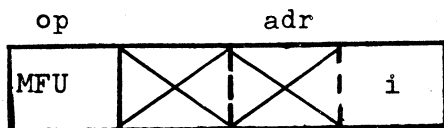


Wirkung: W1. mod1 := <i> + mod2
 W2. := <i> + mod2

Ausführungszeit: 0,05 μ s

MFU

MODIFIZIERE IN JEDEM FALL MIT UNVERÄNDERTEM B



Wirkung: $\text{mod1} := \langle i \rangle + \text{mod2}$

Bemerkung: Interne Darstellung von MFU :

Die Linksadresse muß aus folgenden

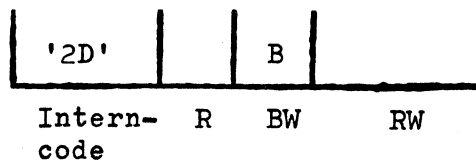
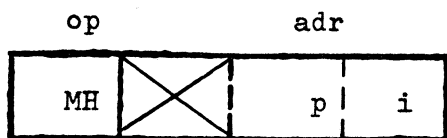
Bits bestehen: OOLXXXXX (X beliebig).

Andernfalls falsche Befehlsausführung.

Ausführungszeit: 0,05 μ s

MH

MODIFIZIERE NACH ERHÖHUNG

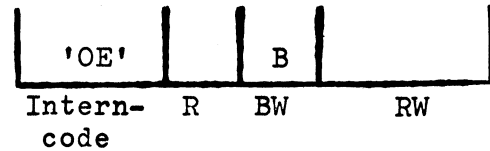
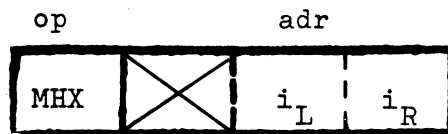


Wirkung: W1. $\text{mod2} := \langle i \rangle + p$
 $-127 \leq p \leq 127$
 W2. $\langle B \rangle := \text{mod2}$
 $\langle i \rangle := \text{mod2}$

Ausführungszeit: 0,525 μs

MHX

MODIFIZIERE NACH ERHÖHUNG UM INDEXZELLE



Wirkung: W1. mod2, $\langle B \rangle, \langle i_R \rangle := \langle i_R \rangle + \langle i_L \rangle$

Ausführungszeit: 0,725 μ s

ML

MULTIPLIZIERE

op	adr
ML	n

'54'	+		AQDYM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W4. $\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle$
 hierbei gilt $\langle A \rangle_{1,2} = \langle Q \rangle_{1,2}$
 $\langle Q \rangle_t := 1$

W5. $\langle Y \rangle := +0$

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W4. $\langle A \rangle, \langle Q \rangle := \text{undefiniert}$
 $\langle Q \rangle_t := 1$

W5. $\langle Y \rangle := +0$

Fall C: $\langle A \rangle_t \neq 1$ oder $\langle n \rangle_t \neq 1$

W4. Tk - Alarm

Marke bei Zahlworten

Normalfall; Tk 1

gleiche Vorzeichen
in A und Q

über - oder unterge-
laufene Zahl

Falsche Tk

MULTIPLIZIERE AKKUMULIEREND



Wirkung:

W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W4. falls $\langle H \rangle_t \neq 1$ oder $\langle Q \rangle_t \neq 1$
 Tk-Alarm

Marke bei Zahlwörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W5. $\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$
 $\langle Q \rangle_t := 1$

W6. $\langle Y \rangle := +0$

W7. falls $\langle A \rangle_1 \neq \langle A \rangle_2$
 A-Alarm

Normalfall

die Addition geschieht,
 als ob $\langle H \rangle_t = \langle Q \rangle_t = 1$

vorderer Teil des Ergebnisses über- oder untergelaufen

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W5. $\langle A \rangle, \langle Q \rangle := \text{undefiniert}$
 $\langle Q \rangle_t := 1$

W6. $\langle Y \rangle := +0$

W7. falls $\langle A \rangle_1 \neq \langle A \rangle_2$
 A-Alarm

über- oder untergelaufene Zahl

Falsche Tk

Bemerkung: $\langle Q \rangle_1 \neq \langle Q \rangle_2$ oder $\langle H \rangle_1 \neq \langle H \rangle_2$ löst keinen A-Alarm aus.

Beim Ergebnis können die Vorzeichen in A und in Q verschieden sein.

Ausführungszeit: Fall A und B: 3,30 μ s
Fall C: 0,10 μ s

GLEITKOMMA MULTIPLIZIERE AUF DOPPELTE GENAUIGKEIT

op	adr
MLD	n

'F3'	+	+	AQDHYM
Intern- code	R	B	RW

Wirkung:

W1. Falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W2. $\langle D \rangle, \langle H \rangle := +0$
 $\langle D \rangle_t, \langle H \rangle_t := 1$

Fall A:

(a) $\langle A \rangle_t = \langle n \rangle_t = 0$ und
 (b) $\langle A \rangle_1 = \langle A \rangle_2$

W3. $\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle$
 Das Ergebnis ist normalisiert
 $\langle Q \rangle_t := 1$

W4. $\langle Y \rangle :=$ Anzahl der Binärstellen, um die
 das Ergebnis normalisiert wurde.
 Falls Mantisse eines Faktors $= \pm 0$ oder
 Exponentenunterlauf:
 $\langle Y \rangle := +0$

W5. Falls Exponentenüberlauf und Mantissen bei-
 der Faktoren $\neq \pm 0$:
 A - Alarm

Fall B:

Bedingung (a) oder (b) nicht erfüllt

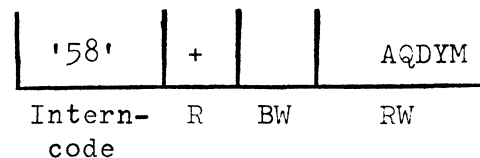
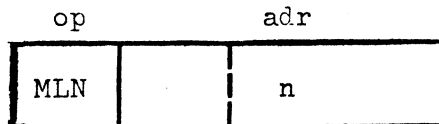
W3. entfällt

W4. $\langle Y \rangle := +0$

W5. Falls (b) nicht erfüllt: A-Alarm
 Falls (a) nicht erfüllt: TK-Alarm

Ausführungszeit:

Fall A: 3,20 μ s
 Fall B: 0,25 μ s

MULTIPLIZIERE NEGATIV

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W4. $\langle A, Q \rangle := -\langle A \rangle \cdot \langle n \rangle$
 hierbei gilt $\langle A \rangle_{1,2} = \langle Q \rangle_{1,2}$
 $\langle Q \rangle_t := \langle 1 \rangle$

W5. $\langle Y \rangle := +0$

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W4. $\langle A \rangle, \langle Q \rangle := \text{undefiniert}$
 $\langle Q \rangle_t := 1$

W5. $\langle Y \rangle := +0$

Fall C: $\langle A \rangle_t \neq 1$ oder $\langle n \rangle_t \neq 1$

W4. Tk - Alarm

Marke bei Zahlwörtern

Normalfall; Tk 1

gleiche Vorzeichen
in A und Q

über - oder unterge-
laufene Zahl

Falsche Tk

Ausführungszeit: Fälle A und B: 2,75 μs

Fall C: 0,10 μs

MLR

MULTIPLIZIERE MIT RUNDUNG

op	adr
MLR	n

'55'	+		AQDYM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W4. $\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle$

W5. $\langle A \rangle := \langle A, Q \rangle$ gerundet

W6. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 1$

W7. $\langle Y \rangle := +0$

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W4. $\langle A \rangle := \text{undefiniert}$

W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 1$

W6. $\langle Y \rangle := +0$

Marke bei Zahlwörtern

Normalfall; Tk 1

über- oder unterge-
laufene Zahl

Fall C: $\langle A \rangle_t \neq 1$ oder $\langle n \rangle_t \neq 1$

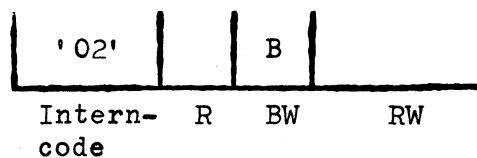
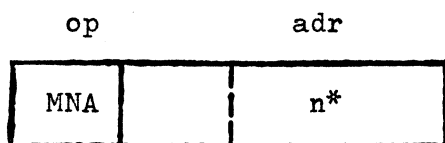
Falsche Tk

W⁴. Tk - Alarm

Ausführungszeit: Fälle A und B: 2,85 μ s
Fall C: 0,10 μ s

MNA

MODIFIZIERE MIT NEGATIVEN ADRESSTEIL



Wirkung: W1. mod2 := -n* + mod2
W2. := mod2

Ausführungszeit: 0,10 μ s

MNR

MULTIPLIZIERE NEGATIV MIT RUNDUNG

op	adr
MNR	n

'59'	+		AQDYM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$

W4. $\langle A, Q \rangle := -\langle A \rangle \cdot \langle n \rangle$

W5. $\langle A \rangle := \langle A, Q \rangle$ gerundet

W6. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 1$

W7. $\langle Y \rangle := +0$

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

W4. $\langle A \rangle := \text{undefiniert}$

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 1$

W6. $\langle Y \rangle := +0$

Marke bei Zahlwörtern

Normalfall; Tk 1

über - oder unterge-
laufene Zahl

Fall C: $\langle A \rangle_t \neq 1$ oder $\langle n \rangle_t \neq 1$

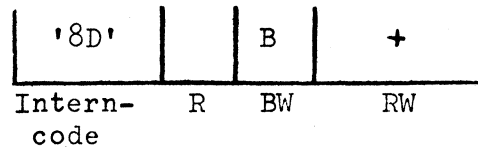
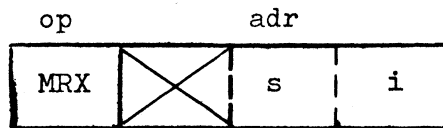
falsche Tk

W4. Tk - Alarm

Ausführungszeit: Fälle A und B: 2,85 μ s
Fall C: 0,10 μ s

MRX

MODIFIZIERE MIT REGISTER UND INDEXZELLE



Spezifikation: $s = s_1 s_2$; $s_1 = A|Q|D|H|B$ | Register
 $s_2 = [N][C]$ | negativ, zurückspeichern

Wirkung: W1. $\text{mod2} := \langle s_1 \rangle + \langle i \rangle$, falls s_2 N nicht enthält
 $\text{mod2} := -\langle s_1 \rangle + \langle i \rangle$, falls s_2 N enthält
 Die Addition läuft über 24 Bits
 W2. $\langle B \rangle := \text{mod2}$
 $\langle i \rangle := \text{mod2}$, nur falls s_2 C enthält

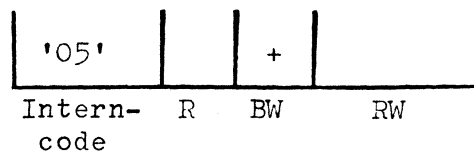
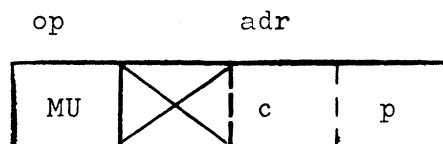
Bemerkung: Falls s_1 leer: Wirkung wie oben, jedoch
 +0 anstelle von $\langle s_1 \rangle$

Interne Darstellung von MRX:

Die Linksadresse muß aus folgenden Bits
 bestehen: XXXX XXIX (X Spezifikationsbits).
 Andernfalls Befehl RX.

Ausführungszeit: 0,60 μ s

MODIFIZIERE MIT U



Wirkung:

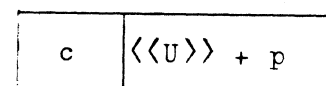
W1. mod2 := 0

W2. op := c

adr := $\langle\langle U \rangle\rangle + p$

$-127 \leq p \leq 127$

Definition des Befehls



Fall A: c ist einer der Befehle S, SE, SFB, SSR, SU, SUE, VSS oder ein bedingter Sprungbefehl, dessen Sprungbedingung erfüllt ist, oder c=R und $\text{adr}_{9,16} = \text{SE oder SUE}$

R SE s

W3. $\langle U \rangle := \langle U \rangle - 1$

W4. Ausgang 2

Fall B: c ist einer der übrigen Befehle

W3. Ausgang 2

Bemerkung zu Fall A:

Beim Befehl c gilt $\langle F \rangle$ anstelle von $\langle F \rangle_{9,24}$

Rücksprung in andere Großseite ist also möglich.

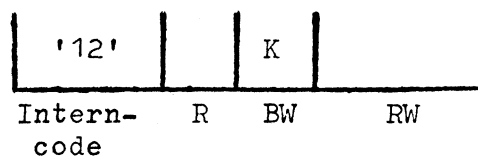
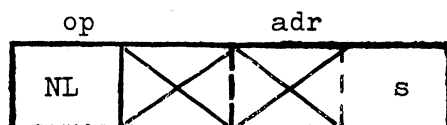
PDP und TLOG gelten nicht als Sprungbefehle.

Ausführungszeit: 1,025 μ s

29. Juli 1968

NL

NEGIERE MERKLICHT



Adressteil: s : 0v1v2v3v...v8

Wirkung: W1. Die durch s bezeichneten
Merklichter werden negiert

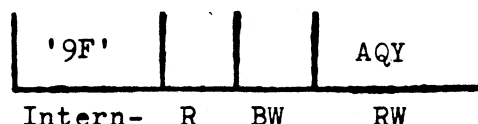
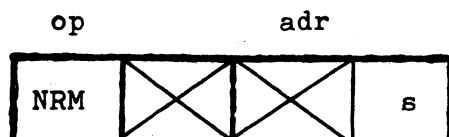
Merklichter 1 bis 8
in beliebiger Kom-
bination

0 bedeutet kein
Merklicht

Ausführungszeit: 0,15 µs

NRM

NORMALISIERE



Wirkung: W1. $\text{adr} := \text{adr} + \text{mod}2$
 $s := \text{adr}_{17,19}$

Adreßteil: $s = G | FG | F | F4 | N | L$

Variante 1: $s = G$

Fall A: $\langle A \rangle_t = 0$ und
 $\langle A \rangle_1 = \langle A \rangle_2 = \dots = \langle A \rangle_5 = \langle A \rangle_6$

W2. Die Mantisse $\langle A \rangle_{1,40}$ wird um Tetraden so lange nach links geschiftet, bis mindestens eines der Bits $\langle A \rangle_3, \langle A \rangle_4, \langle A \rangle_5, \langle A \rangle_6$ ungleich $\langle A \rangle_1$ ist. Beim Schiften werden von rechts Mantissenbits gleich $\langle A \rangle_1$ nachgezogen. Der Exponent $\langle A \rangle_{41,48}$ wird bei jedem Tetradenschift um 1 erniedrigt.

$\langle Y \rangle := 4 m$

wobei $m = \text{Anzahl der Tetradenschifte}$.

Ausnahme: Wird der Exponent < -127 oder ist die Mantisse ± 0 , so ergibt sich:

$\langle A \rangle_{1,40} := -0$
 $\langle A \rangle_{41,48} := -127$
 $\langle Y \rangle := +0$

s besteht nur aus 3 Bits

Gleitkommanormalisierung

$\langle A \rangle$ ist noch nicht normalisiert

$\langle Y \rangle := \text{Anzahl der Einerschifte}$

Mantisse
Exponent

Fall B: $\langle A \rangle_t = 0$ und
 $\langle A \rangle_1, \langle A \rangle_2, \langle A \rangle_3, \dots, \langle A \rangle_6$ nicht alle gleich

W2. $\langle Y \rangle := +0$

Fall C: $\langle A \rangle_t \neq 0$

W2. $\langle Y \rangle := +0$

Tk-Alarm.

$\langle A \rangle$ ist bereits
normalisiert

Falsche Tk

VARIANTE 2 : s = FG

Fall A: $\langle A \rangle_t = \langle Q \rangle_t = 1$ und
 $\langle A \rangle_1 = \langle A \rangle_2 = \langle A \rangle_3 = \langle A \rangle_4 = \langle A \rangle_5 = \langle A \rangle_6 = \langle Q \rangle_1$,
 aber nicht alle Bits von $\langle A \rangle_{3,48}$
 und $\langle Q \rangle_{3,48}$ sind gleich

Umwandlung von
 Fest- in Gleitkomma

$\langle A, Q \rangle$ noch nicht
 normalisiert

W2. Das doppelt lange Register

$\langle A \rangle_1, \dots, \langle A \rangle_{48}, \langle Q \rangle_3, \dots, \langle Q \rangle_{48}$
 wird sooft (m mal) um eine Tetrade
 nach links geschiftet, unter Nach-
 ziehen vorzeichengleicher Stellen,
 bis mindestens eines der Bits
 $\langle A \rangle_3, \langle A \rangle_4, \langle A \rangle_5, \langle A \rangle_6 \neq \langle A \rangle_1$ wird.

m = Anzahl der ge-
 schifteten Tetraden.

$\langle Q \rangle_v$ wird um-
 schiftet.

W3. $\langle Q \rangle := +0$
 $\langle Y \rangle := +0$
 $\langle A \rangle_t := 0$

W4. Falls $\langle A \rangle_1 = \langle A \rangle_{41}$
 $\langle A \rangle_{41,48} := -m$
 Ausgang 1

Keine Rundung nötig
 $\langle A \rangle_{1,40}$ = Mantisse
 $\langle A \rangle_{41,48}$ = Exponent

W4. Falls $\langle A \rangle_1 \neq \langle A \rangle_{41}$
 $\langle A \rangle_{1,40} := \langle A \rangle_{1,40} + L$
 wenn $\langle A \rangle_1 = 0$
 $\langle A \rangle_{1,40} := \langle A \rangle_{1,40} - L$
 wenn $\langle A \rangle_1 = L$

mit Rundung
 Mantisse positiv

Mantisse negativ

W5. Falls $\langle A \rangle_1 = \langle A \rangle_2$
 $\langle A \rangle_{41,48} := -m$
 Ausgang 1

Kein Überlauf bei
 Rundung

W5. Falls $\langle A \rangle_1 \neq \langle A \rangle_2$
 $\langle A \rangle_{1,40}$ wird um eine Tetrade nach
 rechts geschiftet, von links wird
 vorzeichengleich nachgezogen. Die
 bei $\langle A \rangle_{40}$ herausfallende Tetrade
 geht verloren.
 $\langle A \rangle_{41,48} := -m + 1$
 Ausgang 1

Überlauf bei Rundung
 wird beseitigt

Fall B: $\langle A \rangle_t = \langle Q \rangle_t = 1$,
 $\langle A \rangle_1 = \langle A \rangle_2 = \langle Q \rangle_1$,
 $\langle A \rangle_3, \langle A \rangle_4, \langle A \rangle_5, \langle A \rangle_6$ nicht
 alle gleich $\langle A \rangle_1$

W2. $\langle Q \rangle := +0$
 $\langle Y \rangle := +0$
 $\langle A \rangle_t := 0$

W3. Falls $\langle A \rangle_1 = \langle A \rangle_{41}$
 $\langle A \rangle_{41,48} := -0$

Ausgang 1

W3. Falls $\langle A \rangle_1 \neq \langle A \rangle_{41}$
 $\langle A \rangle_{1,40} := \langle A \rangle_{1,40} + L$
 wenn $\langle A \rangle_1 = 0$
 $\langle A \rangle_{1,40} := \langle A \rangle_{1,40} - L$
 wenn $\langle A \rangle_1 = L$

W4. Falls $\langle A \rangle_1 = \langle A \rangle_2$

$\langle A \rangle_{41,48} := -0$

Ausgang 1

W4. Falls $\langle A \rangle_1 \neq \langle A \rangle_2$

$\langle A \rangle_{1,40}$ wird um eine Tetrade nach
 rechts geschiftet, von links wird
 vorzeichengleich nachgezogen. Die
 bei $\langle A \rangle_{40}$ herausfallende Tetrade
 geht verloren.

$\langle A \rangle_{41,48} := +1$

Ausgang 1

$\langle A, Q \rangle$ schon
 normalisiert

keine Rundung nötig

$\langle A \rangle_{1,40}$ = Mantisse

$\langle A \rangle_{41,48}$ = Exponent

Rundung

Mantisse positiv

Mantisse negativ

kein Überlauf
 bei Rundung

Überlauf bei Rundung
 wird beseitigt

Fall C:

$$\langle A \rangle_t = \langle Q \rangle_t = 1, \\ \langle A \rangle_1 = \langle Q \rangle_1, \langle A \rangle_1 \neq \langle A \rangle_2$$

Festkommazahl über-oder
untergelaufen

$$W2. \quad \text{Falls } \langle A \rangle_1 = \langle A \rangle_{41}$$

$$\langle Q \rangle := +0$$

$$\langle Y \rangle := +0$$

$$\langle A \rangle_t := 0$$

$$W2. \quad \text{Falls } \langle A \rangle_1 \neq \langle A \rangle_{41}$$

$$\langle Q \rangle := +0$$

$$\langle Y \rangle := +0$$

$$\langle A \rangle_t := 0$$

$$\langle A \rangle_{1,40} := \langle A \rangle_{1,40} + L \\ \text{wenn } \langle A \rangle_1 = 0$$

$$\langle A \rangle_{1,40} := \langle A \rangle_{1,40} - L \\ \text{wenn } \langle A \rangle_1 = L$$

Rundung

Mantisse positiv

Mantisse negativ

Bemerkung: Falls durch die Rundung sich
 $\langle A \rangle_1$ ändert, erhält man ein
 falsches Ergebnis

W3. $\langle A \rangle_{1,40}$ wird um eine Tetrade nach
 rechts geschiftet, von links wird
 vorzeichengleich nachgezogen. Die
 bei $\langle A \rangle_{40}$ herausfallende Tetrade
 geht verloren.

$$\langle A \rangle_{41,48} := +1$$

$$\langle A \rangle_{1,40} = \text{Mantisse}$$

$$\langle A \rangle_{41,48} = \text{Exponent}$$

Fall D: $\langle A \rangle_t = \langle Q \rangle_t = 1$ und
 $\langle A \rangle_1 = \dots = \langle A \rangle_{48} = \langle Q \rangle_1 = \langle Q \rangle_3 = \dots = \langle Q \rangle_{48}$

vorgegebene Zahl
ist Null

W2. $\langle Q \rangle := +0$
 $\langle Y \rangle := +0$
 $\langle A \rangle_t := 0$
 $\langle A \rangle_{1,40} := -0$
 $\langle A \rangle_{41,48} := -127$

Mantisse
Exponent

Fall E: $\langle A \rangle_1 = \langle Q \rangle_1$ und
 $(\langle A \rangle_t \neq 1 \text{ oder } \langle Q \rangle_t \neq 1)$

falsche TK

W2. $\langle Y \rangle := +0$
 TK-Alarm

Fall F: $\langle A \rangle_1 \neq \langle Q \rangle_1$

Vorzeichen nicht
angeglichen

W2. $\langle Y \rangle := +0$
 A-Alarm

VARIANTE 3: $s = F$ oder F^4

Festkommanormalisierung

Fall A: $\langle A \rangle_t = \langle Q \rangle_t = 1$ und

$$\langle A \rangle_1 = \langle A \rangle_2 = \langle A \rangle_3 = \langle A \rangle_4 = \langle A \rangle_5 = \langle A \rangle_6 = \langle Q \rangle_1,$$

$\langle A, Q \rangle$ normalisierbar

aber nicht alle Bits von $\langle A \rangle$ und $\langle Q \rangle_{3,48}$ gleich

$\langle A, Q \rangle \neq +0$

W2. Das doppelt lange Register

$$\langle A \rangle_1, \langle A \rangle_2, \dots, \langle A \rangle_{48}, \langle Q \rangle_3, \langle Q \rangle_4, \dots, \langle Q \rangle_{48}$$

wird sooft (m mal) um eine Tetrade

nach links geschiftet, unter Nachziehen

vorzeichengleicher Stellen, bis min=

destens eines der Bits

$$\langle A \rangle_3, \langle A \rangle_4, \langle A \rangle_5, \langle A \rangle_6 \neq \langle A \rangle_1 \text{ wird.}$$

$\langle Q \rangle_v$ wird umschiftet

W3. $\langle Y \rangle := 4m$, falls $s = F$

$\langle Y \rangle := m$, falls $s = F^4$

$m = \text{Anzahl der Tetradenschifte}$

Fall B: $\langle A \rangle_t = \langle Q \rangle_t = 1$ und $\langle A \rangle_1 = \langle Q \rangle_1$ und

$(\langle A \rangle_2, \langle A \rangle_3, \langle A \rangle_4, \langle A \rangle_5, \langle A \rangle_6)$ nicht alle
gleich $\langle A \rangle_1$

$\langle A \rangle$ ist bereits
normalisiert oder

$\langle A, Q \rangle = \pm 0$

$$\text{oder } \langle A \rangle_1 = \langle A \rangle_2 = \dots = \langle A \rangle_{48} = \langle Q \rangle_3 = \langle Q \rangle_4 = \dots = \langle Q \rangle_{48}$$

$$\begin{aligned} \text{W2. } \langle Q \rangle_2 &:= \langle Q \rangle_1 \\ \langle Y \rangle &:= +0 \end{aligned}$$

Fall C: $\langle A \rangle_1 \neq \langle Q \rangle_1$

Vorzeichen

nicht angeglichen

W2. $\langle Y \rangle := +0$

A-Alarm

Fall D: $\langle A \rangle_1 = \langle Q \rangle_1$ und $(\langle A \rangle_t \neq 1 \text{ oder } \langle Q \rangle_t \neq 1)$

falsche Tk

W2. $\langle Y \rangle := +0$

Tk-Alarm

VARIANTE 4: $s = N$ Fall A: $\langle A \rangle \neq +0$

W2. $\langle A \rangle$ wird solange (bitweise) nach
links geschiftet, bis $\langle A \rangle_1 = L$.
Von rechts werden 0-Bits nachgezogen.
 $\langle Y \rangle :=$ Anzahl der Schiftschritte

Führende Nullbits
zählen

Normalfall

= Anzahl der führen=
den NullbitsFall B: $\langle A \rangle = +0$ W2. $\langle Y \rangle := 48$ VARIANTE 5: $s = L$

W2. $\langle A \rangle$ wird solange (bitweise) nach
links geschiftet, bis $\langle A \rangle_1 = 0$.
Rechts werden 0-Bits nachgezogen.
 $\langle Y \rangle :=$ Anzahl der Schiftschritte

Führende L-Bits
zählen= Anzahl der
führenden L-BitsAusführungszeiten:

Variante 1: $s = G : (0,10m + 0,15)\mu s$, wobei $m =$ Anzahl der
Tetradenschifte

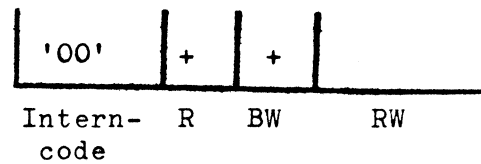
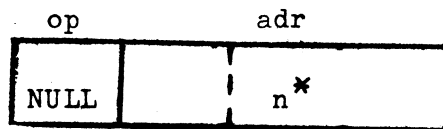
Variante 2: $s = FG : (0,10m + 0,50)\mu s$

Variante 3: $s = F$ oder $F4 : (0,10m + 0,35)\mu s$

Variante 4;5: $s = N$ oder $L : (0,10b + 0,15)\mu s$, wobei $b =$ Anzahl
der Einzelschifte

NULL

NULLBEFEHL



Wirkung: W1. keine Wirkung außer Ausgang 1

Bemerkung: Adreßteil beliebig und ohne Bedeutung
für die Ausführung des Befehls

Ausführungszeit: 0,05 μ s

Prüfe Dreierprobe

op	adr
PDP	n

'B9'			AY
Intern- code	R	BW	RW

- Wirkung:
- W1. $n := n + \text{mod } 2$
 - W2. $\langle A \rangle_{1,48} := \langle n \rangle_{1,48}$
 $\langle A \rangle_t := \langle n \rangle_t$
 - W3. $\langle Y \rangle := \langle n \rangle_{5,6} \cdot 2^6 + 2^4$
 $+ \text{Dreierprobe} \cdot 2^2 + \langle n \rangle_t$
 - W4. Falls Dreierprobe von $\langle n \rangle$ richtig,
 $\langle F \rangle_{9,24} := \langle F \rangle + 2$

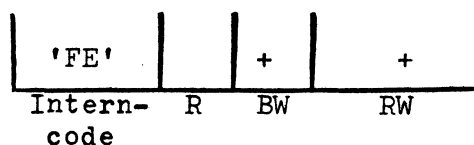
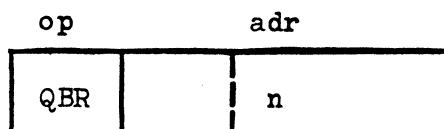
ein Ganzwort ge-
langt unverändert
nach A

$\langle n \rangle_{5,6}$	OL	DP	$\langle n \rangle_t$
---------------------------	----	----	-----------------------

Ausführungszeit: 1,05 μ s

QBR

BEQUEMES BRINGEN ALLER REGISTER



Wirkung:

W1. $\langle H \rangle := \langle n-2 \rangle$
 $\langle H \rangle_t := \langle n-2 \rangle_t$

$\langle D \rangle := \langle n-4 \rangle$
 $\langle D \rangle_t := \langle n-4 \rangle_t$

$\langle Q \rangle := \langle n-6 \rangle$
 $\langle Q \rangle_t := \langle n-6 \rangle_t$

$\langle A \rangle := \langle n-8 \rangle$
 $\langle A \rangle_t := \langle n-8 \rangle_t$

W2. $\langle M \rangle := L, \text{ falls } \langle n-10 \rangle_t = 1 \text{ oder } 3$
 $\langle M \rangle := 0, \text{ falls } \langle n-10 \rangle_t = 0 \text{ oder } 2$

$\langle B \rangle := \langle n-10 \rangle_{1,24}$
 $\langle K \rangle := \langle n-10 \rangle_{25,32}$
 $\langle Y \rangle := \langle n-10 \rangle_{33,40}$
 $\langle U \rangle := \langle n-10 \rangle_{41,48}$

Bemerkung: $\langle n \rangle$ wird auch ausgelesen, ist ohne Bedeutung, muß jedoch in zugewiesener Seite stehen.

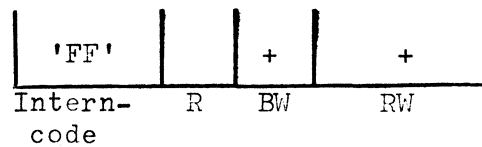
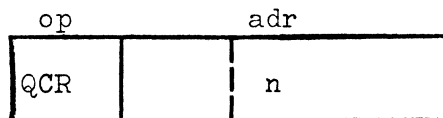
Ausführungszeit: 4,20 μs

Die Register
werden
unverändert
gebracht

B	K	Y	U
---	---	---	---

QCR

BEQUEMES SPEICHERN ALLER REGISTER



Wirkung: W1. $\langle n \rangle_t := 2 + \langle M \rangle$

$\langle n \rangle_{1,24} := \langle B \rangle$
 $\langle n \rangle_{25,32} := \langle K \rangle$
 $\langle n \rangle_{33,40} := \langle Y \rangle$
 $\langle n \rangle_{41,48} := \langle U \rangle$

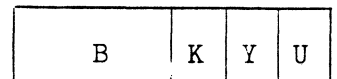
W2. $\langle n+2 \rangle := \langle A \rangle$
 $\langle n+2 \rangle_t := \langle A \rangle_t$

$\langle n+4 \rangle := \langle Q \rangle$
 $\langle n+4 \rangle_t := \langle Q \rangle_t$

$\langle n+6 \rangle := \langle D \rangle$
 $\langle n+6 \rangle_t := \langle D \rangle_t$

$\langle n+8 \rangle := \langle H \rangle$
 $\langle n+8 \rangle_t := \langle H \rangle_t$

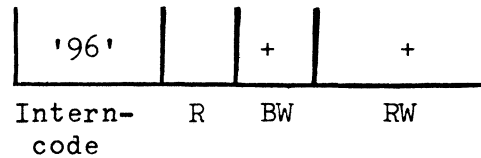
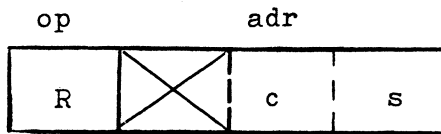
$\langle n+10 \rangle_{1,24} := \langle BT \rangle$
 $\langle n+10 \rangle_{25,48} := +0$
 $\langle n+10 \rangle_t := 2$



Die Register
werden immer
unverändert
abgespeichert

BT = Prüfregister

Ausführungszeit: 2,75 μ s

REGISTERADRESSIERUNG

Wirkung: W1. $\text{adr} := \text{adr} + \text{mod2}$

W2. $\text{mod2} := 0$

Adreßteil: $c = \text{einer der 73 erlaubten Codes,}$
 siehe Bemerkung

$$s = s_1 s_2$$

$$s_1 = A|Q|D|H|B|F|U|Y$$

$$s_2 = [L]$$

Fall A: $s_1 = U|Y$

$$\text{W3.} \quad \text{ro}_{1,40} := +0$$

$$\text{ro}_{41,48} := \langle s_1 \rangle$$

$$\text{ro}_t := 1$$

Fall B: $s_1 = B$

$$\text{W3.} \quad \text{ro} :v= \langle B \rangle$$

$$\text{ro}_t := 1$$

Register

linke 24 Bits

Registeroperand
48 Bits mit Tk

Fall C: $s_1 = F$

W3. $ro := \langle F \rangle + 1$
 $ro_t := 1$

Fall D: $s_1 = A|Q|D|H$

W3. $ro := \langle s_1 \rangle$
 $ro_t := \langle s_1 \rangle_t$

Alle Fälle: W4. $op := c$

Variante 1: c ist einer der erlaubten Codes, jedoch
nicht der NULL-Befehl

Falls c ein Ganzwort adressiert, ersetze
man in der Beschreibung des Befehls c
 $\langle n \rangle$ durch ro und $\langle n \rangle_m$ durch ro_1

Falls c ein Halbwort adressiert und s_2
leer, ersetze man in der Beschreibung²
des Befehls c $\langle n \rangle_2$ durch $ro_{25,48}$

Falls c ein Halbwort adressiert und $s_2=L$,
ersetze man in der Beschreibung des
Befehls c $\langle n \rangle_2$ durch $ro_{1,24}$

W5. Ausgang 2

Variante 2: $c = \text{NULL}$

W5. Ausgang 1

s_2 bedeutungslos
 ro_1 hat analoge Be-
deutung zu $\langle n \rangle_1$

beim NULL-Befehl
keine Wirkung, s
ohne Bedeutung

Bemerkung: Folgende Codes sind erlaubt:

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, MLN, MLR, MNR,
NULL,
REZ,
SB, SB2, SBB, SBD, SBI, SBQ, SBT, SBU, SE, SUE,
T, TCB,
VBC, VEL,
ZUS

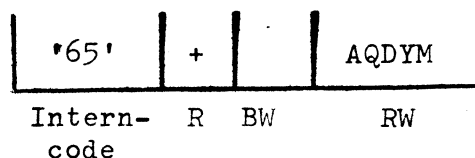
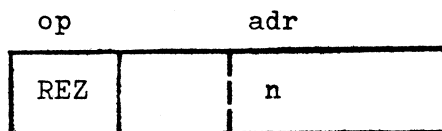
falls c kein erlaubter Code, Wirkung undefiniert

Ausführungszeit: Fälle A, D : 0,20 μ s

Fälle B, C : 0,30 μ s

REZ

BILDE REZIPROKEN WERT



Wirkung:

W1. $n := n + \text{mod}2$

W2. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahl-
wörtern

Fall A: $\langle n \rangle_t = 0$ und $\langle n \rangle_{1,40} \neq 0$

Normalfall

W3. $\langle D \rangle := +0$
 $\langle D \rangle_t := 0$

W4. $\langle A \rangle := 1 : \langle n \rangle$, das Ergebnis ist
normalisiert und
gerundet
 $\langle A \rangle_t := 0$

W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$

W6. $\langle Y \rangle :=$ Anzahl der Binärstellen, um
die das Ergebnis normali-
siert wurde

W7. falls Exponentenüberlauf des Ergeb-
nisses:
A-Alarm

Fall B: $\langle n \rangle_t = 0$ und $\langle n \rangle_{1,40} = \pm 0$

Divisor = 0

W3. $\langle D \rangle := \langle A \rangle$

$\langle D \rangle_t := 0$

W4. $\langle A \rangle := +0$

$\langle A \rangle_t := 0$

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle := +0$

A-Alarm

Fall C: $\langle n \rangle_t \neq 0$

falsche Tk

W3. $\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

W4. $\langle Q \rangle := \langle A \rangle$

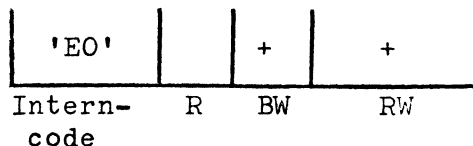
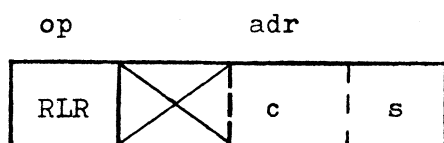
$\langle Q \rangle_t := \langle A \rangle_t$

Tk-Alarm

Ausführungszeit: Fall A: 10,65 μ s
Fall B: 0,15 μ s
Fall C: 0,10 μ s

RLR

RELATIV-ADRESSIERUNG MIT REGISTERINHALT



Spezifikationen:

s : A|Q|D|H|AL|QL|DL|HL|F|U|B|Y

Register; L bedeutet
linke Hälfte

Wirkung:

W1. op := c

W2. $ro_{25,48} := \langle s \rangle$,
falls s = A|Q|D|H|AL|QL|DL|HL|B.

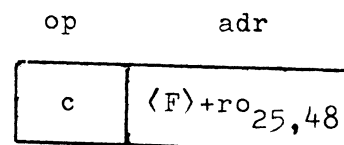
$ro_{25,48} := \langle s \rangle$, falls s = U|Y.

$ro_{25,48} := \langle F \rangle + 1$, falls s = F.

W3. adr := $\langle F \rangle + ro_{25,48}$

W4. Ausgang 2

Definition des Befehls



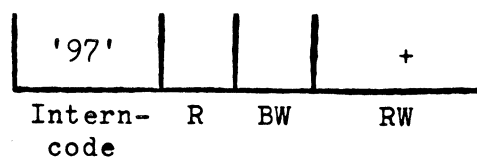
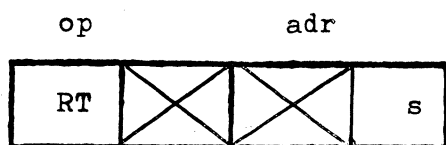
Bemerkung:

Falls s = FL|UL|BL|YL, ist L ohne Bedeutung

Ausführungszeit: 0,80 μ s

RT

REGISTERTAUSCH



Wirkung: W1. $s := s + \text{mod}2$

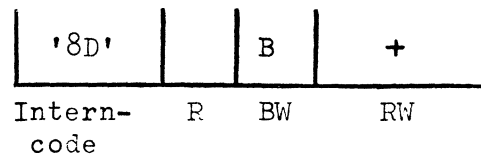
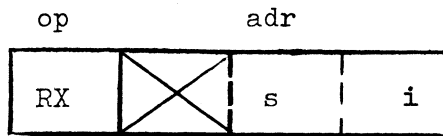
Adreßteil: $s = s_1 s_2$
 $s_1, s_2 = A | H | Q | D$
 $s_1 \neq s_2$

W2. $\langle s_1 \rangle ::= \langle s_2 \rangle$
 $\langle s_1 \rangle_t ::= \langle s_2 \rangle_t$

Ausführungszeit: 0,20 μ s

REGISTER UND INDEXZELLE

RX



Spezifikation: $s = s_1 s_2$; $s_1 = A|Q|D|H|B$ Register
 $s_2 = [N][C]$ negativ, zurückspeichern

Wirkung: W1. $\langle B \rangle := \langle s_1 \rangle + \langle i \rangle$, falls s_2 N nicht enthält
 $\langle B \rangle := -\langle s_1 \rangle + \langle i \rangle$, falls s_2 N enthält
 die Addition läuft über 24 Bits
 W2. $\langle i \rangle := \langle B \rangle$, falls s_2 C enthält
 entfällt , falls s_2 C nicht enthält

Bemerkung: Falls s_1 leer: Wirkung wie oben, jedoch
 +0 anstelle von $\langle s_1 \rangle$

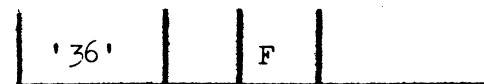
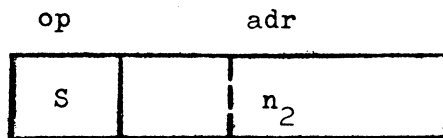
Interne Darstellung von FX:

Die Linksadresse muß aus folgenden Bits
 bestehen: XXXX XXOX (X Spezifikationsbits).
 Andernfalls Befehl MRX

Ausführungszeit: 0,60 μ s

S

Springe



Intern-
code R BW RW

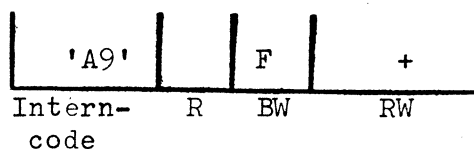
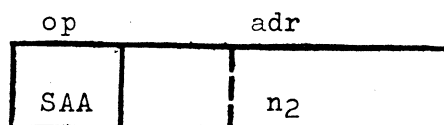
Wirkung: W1. $n_2 := n_2 + \text{mod}2$

W2. $\langle F \rangle_{9,24} := n_2$

Ausführungszeit: 0,05 μ s

SAA

SPRINGE, WENN ALARM (ARITHMETISCH)



Sprungbedingung SB: A-Alarm steht an

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB ist erfüllt

W2. A-Alarm wird gelöscht

$\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

W2. entfällt

Bemerkung:

Bei anstehendem A-Alarm führt das Fehlen dieses Befehls zum Abbruch beim nächsten das Rechenwerk belegenden Befehl, ausgenommen Befehl SAT. Befehle, die nur das Befehlswerk belegen, und der Befehl SAT werden noch ausgeführt.

Ausführungszeit: 0,05 μs

SAT

SPRINGE, WENN ALARM (TYPENKENNUNG)



Sprungbedingung SB: Tk-Alarm steht an

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB ist erfüllt

W2. Tk-Alarm wird gelöscht
 $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

W2. entfällt

Bemerkung:

Bei anstehendem Tk-Alarm führt das Fehlen dieses Befehls zum Abbruch beim nächsten das Rechenwerk belegenden Befehl, ausgenommen Befehl SAA. Befehl, die nur das Befehlswerk belegen, und der Befehl SAA werden noch ausgeführt.

Ausführungszeit: 0,05 μ s

SB

SUBTRAHIERE

op	adr
SB	n

'46'	+		ADM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. $\langle A \rangle := \langle A \rangle - \langle n \rangle$
 $\langle A \rangle_t := \max(\langle A \rangle_t, \langle n \rangle_t)$

W4. falls $\langle n \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$
 falls $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$
 A-Alarm

$\langle A \rangle := \langle A \rangle + \overline{\langle n \rangle}$

Marke bei Zahl-
 worten
 Ergebnis überge-
 laufen

Bemerkung: Die Vorzeichenstellen der Operanden
 können ungleich sein.

Die Subtraktion geschieht wie bei Tk 1 (modulo $2^{48}-1$)

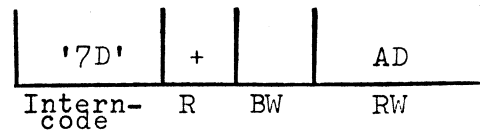
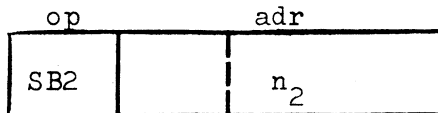
$\langle A \rangle := +0$ nur wenn vor der Subtraktion

$\langle A \rangle = +0$ und $\overline{\langle n \rangle} = +0$

Ausführungszeit: 0,40 μs

SB2

SUBTRAHIERE HALBWORT



Wirkung:

W1. $n_2 := n_2 + \text{mod}2$

W2. $\langle A \rangle_{25,48} := \langle A \rangle_{25,48} - \langle n_2 \rangle$

W3. $\langle A \rangle_{1,24} := \langle A \rangle_{25}$

W4. $\langle D \rangle := +0$

$\langle D \rangle_t := \langle A \rangle_t$

Falls $\langle A \rangle_{25} \neq \langle A \rangle_{26}$:

W5. A-Alarm

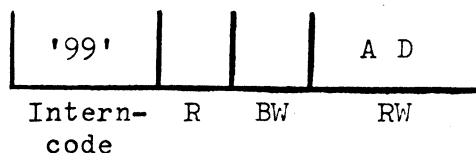
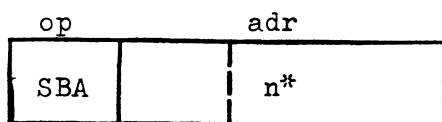
Die Subtraktion geschieht
modulo $2^{24} - 1$

$\langle A \rangle_{25}$ und $\langle A \rangle_{26}$ werden
als Vorzeichenstellen
aufgefaßt

Ausführungszeit: 0,70 μs

SBA

SUBTRAHIERE ADRESSTEIL



Wirkung: W1. $n^* := n^* + \text{mod}2$

Variante 1: $\langle A \rangle_t = 0$

W2. $\langle D \rangle := \text{rechte 8 Bits von } n^*$
 $\langle D \rangle_t := 1$

W3. Die rechten 8 Bits von n^* werden
 von $\langle A \rangle_{41,48}$ subtrahiert.
 Der Einerrücklauf erfolgt von
 $\langle A \rangle_{41}$ zu $\langle A \rangle_{48}$.

Fall A: $-127 \leq \text{Ergebnis} \leq +127$

W4. entfällt

Fall B: $\text{Ergebnis} \geq 128$

W4. $\langle A \rangle_{41,48} := \text{Ergebnis}$
 A-Alarm

Fall C: $\text{Ergebnis} < -128$

W4. $\langle A \rangle_{1,40} := -0$
 $\langle A \rangle_{41,48} := -127$

Gleitkomma

Exponent in A
 andere Bits von n^*
 ohne Bedeutung

Normalfall

Exponent übergelaufen

Exponent um 255 zu klein

Exponent übergelaufen

normalisierte
 Gleitkommanull

Variante 2. $\langle A \rangle_t = 1$ oder 3Festkomma
OktadenW2. $\langle D \rangle :v= n^*$ $\langle D \rangle_t := 1$

W3. Die 24 Bits von n^* werden durch Vorzeichenangleich zu einer 48 Bit-Größe ergänzt und von den 48 Bit des $\langle A \rangle$ subtrahiert. Der Einerrücklauf erfolgt von $\langle A \rangle_1$ zu $\langle A \rangle_{48}$.
 $\langle A \rangle :=$ Ergebnis
 Falls $\langle A \rangle_t = 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$: A-Alarm

Variante 3. $\langle A \rangle_t = 2$

Befehle

W2. $\langle D \rangle :o=$ rechte 16 Bits von n^*
 $\langle D \rangle_t := 1$

W3. Die rechten 16 Bits von n^* werden von den rechten 16 Bits von $\langle A \rangle$ subtrahiert. Der Einerrücklauf erfolgt von $\langle A \rangle_{33}$ zu $\langle A \rangle_{48}$.
 $\langle A \rangle_{1,32}$ bleiben unverändert

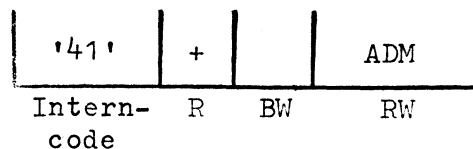
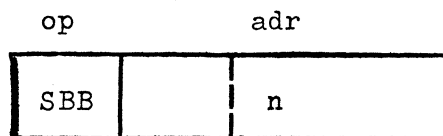
linke 8 Bits
von n^* ohne
Bedeutung

Bemerkung: Variante 1 und 2: +0 als Ergebnis nur dann, wenn beide Operanden (nach Inversion des Subtrahenden) +0, und Variante 3: -0 als Ergebnis genau dann, wenn der Minuend -0 und der Subtrahend +0 ist.

Ausführungszeit: 0,65 μ s

SBB

SUBTRAHIERE BETRAG



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. $\langle A \rangle := \langle A \rangle - |\langle n \rangle|$
 $\langle A \rangle_t := \max(\langle A \rangle_t, \langle n \rangle_t)$

W4. falls $\langle n \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$
 falls $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$:
 A-Alarm

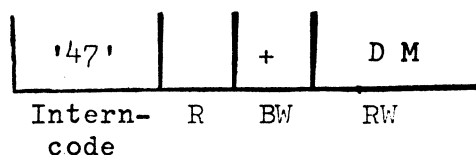
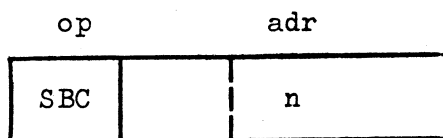
Marke bei Zahl-
 wort
 Ergebnis überge-
 laufen

Bemerkung: Die Vorzeichenstellen der Operanden können ungleich sein.
 Falls $\langle n \rangle_t \geq 2$, gilt $|\langle n \rangle| = \langle n \rangle$
 Falls $\langle n \rangle_t = 0$, geschieht die Betragsbildung wie bei Tk 1.
 Die Subtraktion geschieht wie bei Tk 1 (modulo $2^{48}-1$).
 Es gilt stets $\langle A \rangle \neq +0$

Ausführungszeit: C,40 μs

SBC

SUBTRAHIERE IM SPEICHER



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle - \langle A \rangle$
 $\langle D \rangle_t := \max(\langle A \rangle_t, \langle n \rangle_t)$
 falls $\langle n \rangle_t \leq 1$ zusätzlich
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

$\langle D \rangle := \langle n \rangle + \overline{\langle A \rangle}$

Marke bei Zahl-
worten

Fall A: $\langle D \rangle_t \geq 2$

W3. $\langle n \rangle := \langle D \rangle$
 $\langle n \rangle_t := \langle D \rangle_t$

Fall B: $\langle D \rangle_t \leq 1$ und $\langle D \rangle_1 = \langle D \rangle_2$

W3. $\langle n \rangle_1 := \langle n \rangle_1$
 $\langle n \rangle_{2,48} := \langle D \rangle_{2,48}$
 $\langle n \rangle_t := \langle D \rangle_t$

Marke im Spei-
cher bleibt er-
halten

Fall C: $\langle D \rangle_t \leq 1$ und $\langle D \rangle_1 \neq \langle D \rangle_2$

W3. A-Alarm

Ergebnis über-
gelaufen

Bemerkung: Die Vorzeichenstellen der Operanden
können ungleich sein.

Die Subtraktion geschieht wie bei Tk 1 (modulo $2^{48}-1$).

In den Fällen A und B gilt

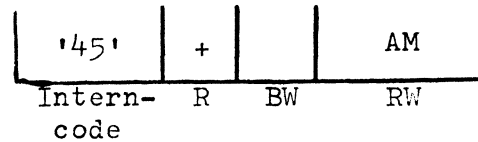
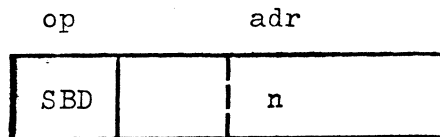
$\langle n \rangle := +0$, nur wenn vor der Subtraktion

$\langle n \rangle = \overline{\langle A \rangle} = +0$

Ausführungszeit: 0,85 μ s

SBD

SUBTRAHIERE VON D



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle A \rangle := \langle D \rangle - \langle n \rangle$
 $\langle A \rangle_t := \max(\langle D \rangle_t, \langle n \rangle_t)$

W3. falls $\langle n \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$
 falls $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$:
 A-Alarm

$\langle A \rangle := \langle D \rangle + \overline{\langle n \rangle}$

Marke bei Zahl-
 wort
 Ergebnis überge-
 laufen

Bemerkung: Die Vorzeichenstellen der Operanden
 können ungleich sein.

Die Subtraktion geschieht wie bei Tk 1 (modulo $2^{48}-1$)

$\langle A \rangle := +0$ nur wenn $\langle D \rangle = \overline{\langle n \rangle} = +0$

Ausführungszeit: 0,55 μs

SBI

SUBTRAHIERE INVERS

op	adr
SBI	n

'44'	+		ADM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. $\langle A \rangle := \langle n \rangle - \langle A \rangle$
 $\langle A \rangle_t := \max(\langle A \rangle_t, \langle n \rangle_t)$

W4. falls $\langle n \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$
 falls $\langle A \rangle_t \leq 1$ und $\langle A \rangle_1 \neq \langle A \rangle_2$
 A-Alarm

$\langle A \rangle := \langle n \rangle + \overline{\langle A \rangle}$

Marke bei Zahlwort

Ergebnis übergelaufen

Bemerkung: Die Vorzeichenstellen der Operanden können ungleich sein.
 Die Subtraktion geschieht wie bei Tk 1 (modulo $2^{48}-1$)
 $\langle A \rangle := +0$ nur wenn vor der Subtraktion
 $\langle n \rangle = \overline{\langle A \rangle} = +0$

Ausführungszeit: 0,40 μs

S B Q

SUBTRAHIERE IN A, Q

op	adr
SBQ	n

'7F'	+		AQDM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$:
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Fall A: $\langle A \rangle_t = \langle Q \rangle_t = \langle n \rangle_t = 1$ und $\langle A \rangle_1 = \langle A \rangle_2$
und $\langle Q \rangle_1 = \langle Q \rangle_2$

W4. $\langle A, Q \rangle := \langle A, Q \rangle - \langle n \rangle$

W5. falls $\langle A \rangle_1 \neq \langle A \rangle_2$:
A-Alarm

Fall B: $\langle A \rangle_t = \langle Q \rangle_t = \langle n \rangle_t = 1$ und $(\langle A \rangle_1 \neq \langle A \rangle_2 \text{ oder } \langle Q \rangle_1 \neq \langle Q \rangle_2)$

W4. $\langle A, Q \rangle := \text{undefiniert}$

W5. falls $\langle A \rangle_1 \neq \langle A \rangle_2$:
A-Alarm

Marke bei Zahlwörtern

Normalfall

Der Einerrücklauf erfolgt
von $\langle A \rangle_1$ zu $\langle A \rangle_{48}$

Ergebnis übergelaufen

$\langle A \rangle$ oder $\langle Q \rangle$ übergelaufen

Fall C: $\langle A \rangle_t$ oder $\langle Q \rangle_t$ oder $\langle n \rangle_t \neq 1$

falsche Tk

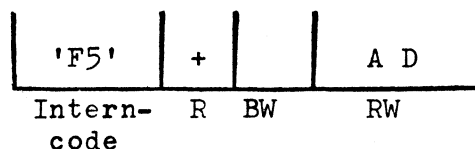
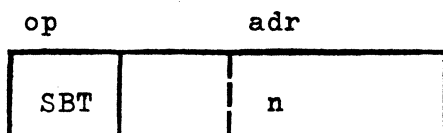
W⁴. Tk-Alarm

Bemerkung: $\langle A \rangle$ und $\langle Q \rangle$ können verschiedene
Vorzeichen haben

Ausführungszeit: Fall A,B: 0,85 μ s
Fall C: 0,30 μ s

SBT

SUBTRAHIERE TEILWORT



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle Q \rangle$

$\langle D \rangle_t := \langle Q \rangle_t$

W3. Es wird eine Hilfsgröße gebildet:

$qr :=$ um p Stellen im Kreis
nach rechts geschiftete $\langle Q \rangle$,
wobei p = Anzahl der in Q rechts
anstehenden L-Bits.

Falls jedoch

$\langle Q \rangle = \text{LLLL}..L$, dann $p = 0$

W4. Es wird eine Hilfsgröße nr gebildet:

$nr :=$ um p Stellen nach rechts
geschiftete $\langle n \rangle$

W5. $nr_i := 0$ falls $qr_i = L$

$\langle A \rangle_i := 0$ falls $qr_i = L$

W6. Die Operanden $\langle A \rangle$ und nr werden als
positive, vorzeichenlose ganze Fest-
kommazahlen aufgefaßt.

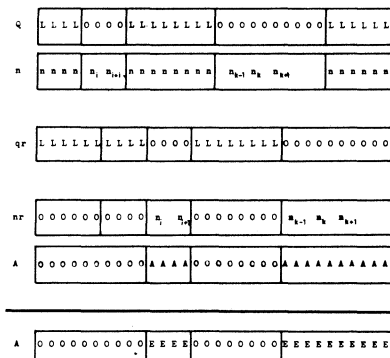
Falls $\langle A \rangle \geq \langle nr \rangle$:

$\langle A \rangle := \langle A \rangle - \langle nr \rangle$

Falls $\langle A \rangle < \langle nr \rangle$:

$\langle A \rangle := \langle A \rangle + 2^{48} - \langle nr \rangle$ und

A-Alarm



B-Komplement

w7. $\langle A \rangle_i := 0$, falls $qr_i = L$

Ausführungszeit: (1,05 + 0,10 p) μ s

SBU

SUBTRAHIERE UNNORMALISIERT

op	adr
SBU	n

'4D'	+		AQDYM
Intern- code	R	BW	RW

Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

W3. falls $\langle n \rangle_t \leq 1$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

Marke bei Zahlwörtern

Fall A: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 = \langle A \rangle_2$

Normalfall

W4. $\langle A \rangle := \langle A \rangle - \langle n \rangle$; Ergebnis nicht
gerundet, nicht normalisiert

W5. $\langle Q \rangle := +0$
 $\langle Q \rangle_t := 0$

W6. $\langle Y \rangle := +0$

W7. falls Exponent des Ergebnisses
übergelaufen:
A-Alarm

Fall B: $\langle A \rangle_t = \langle n \rangle_t = 0$ und $\langle A \rangle_1 \neq \langle A \rangle_2$

über- oder unterge-
laufener Operand

W4. $\langle A \rangle := \text{undefiniert}$

W5. $\langle Q \rangle := +0$

$\langle Q \rangle_t := 0$

W6. $\langle Y \rangle := \text{undefiniert}$
evtl. A-Alarm

Fall C: $\langle A \rangle_t \neq 0$ oder $\langle n \rangle_t \neq 0$

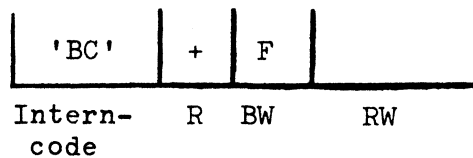
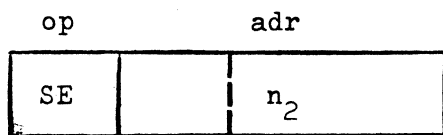
falsche Tk

W4. Tk-Alarm

Ausführungszeit: Fall A: $1,3 \mu s$
Fall B: $1,3 \mu s$
Fall C: $0,1 \mu s$

SE

SPRINGE NACH ERSETZUNG



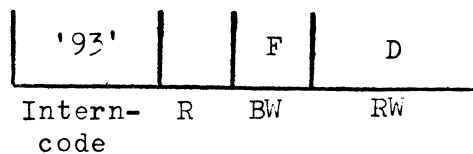
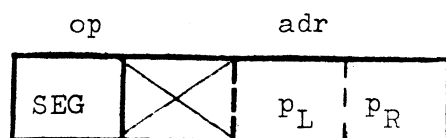
Wirkung: W1. $\langle F \rangle := \langle n_2 \rangle + \text{mod}2$

$\langle n_2 \rangle_t$ ohne Bedeutung

Bemerkung: Adreßteil wie beim Befehl B2. (Falls nicht modifiziert, nur Großseite 0.)

Ausführungszeit: 0,10 μs

SPRUNGE, WENN EXPONENT GRÖßER GLEICH



Wirkung:

W1. $\text{adr} := \text{adr} + \text{mod}2$

W2. $\langle D \rangle := \langle A \rangle$
 $\langle D \rangle_t := \langle A \rangle_t$

Adreßteil:

$-127 \leq p_L \leq 127$
 $-127 \leq p_R \leq 127$

Sprungbedingung SB: $\langle A \rangle_t = 0$ und $\langle A \rangle_{41,48} \geq p_R$

Fall A: SB ist erfüllt

W3. $\langle F \rangle_{9,24} := \langle F \rangle + p_L$

Fall B: SB nicht erfüllt

W3. falls $\langle A \rangle_t = 0$:
entfällt

W3. falls $\langle A \rangle_t \neq 0$:
Tk-Alarm

Sprungparameter
Vergleichs-Exponent

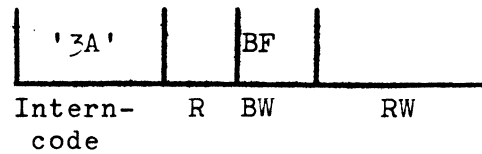
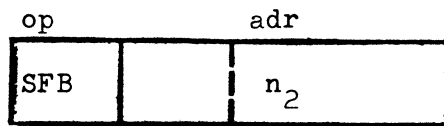
keine Gleitkomma-
zahl

Ausführungszeit:

Fall A: 0,50 μs
 Fall B: 0,20 μs

SFB

SPRINGE UND BRINGE $\langle \underline{F} \rangle + 1$ NACH B



Wirkung: W1. $n_2 := n_2 + \text{mod}2$

W2. $\langle B \rangle := \langle F \rangle + 1$

W3. $\langle F \rangle_{9,24} := n_2$

$\langle B \rangle :=$ Die Adresse,
in der der Folge-
befehl steht

Ausführungszeit: 0,05 μ s

SPRINGE NACH ERSETZUNG UND BRINGE $\langle F \rangle + 1$ NACH B

op	adr
SFBE	n_2

'FA'		BF	+
Intern- code	R	BW	RW

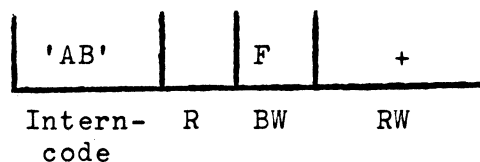
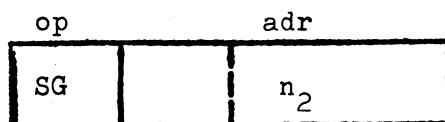
Wirkung: W1. $\langle B \rangle := \langle F \rangle + 1$
 W2. $\langle F \rangle := \langle n_2 \rangle + \text{mod } 2$

$\langle B \rangle :=$ Adresse, wo der Folge-
 befehl steht
 $\langle n_2 \rangle_t$ ohne Bedeutung

Bemerkung: Adressteil wie beim Befehl B2. (~~Falls nicht modifiziert,~~
 nur Großseite 0).

Ausführungszeit:

SPRINGE, WENN GRÖßER



Sprungbedingung SB: $\langle A \rangle > \langle H \rangle$

[1]

Wirkung: W1. $n_2 := n_2 + \text{mod}2$
 falls $\langle A \rangle_t = \langle H \rangle_t = 0$
 zusätzlich
 $\langle A \rangle := \langle A \rangle$ normalisiert
 falls $\langle A \rangle_t = \langle H \rangle_t = 0$ und $\langle H \rangle_1 = \langle H \rangle_2$
 zusätzlich
 $\langle H \rangle := \langle H \rangle$ normalisiert
 Falls in A der Exponent +0 ist,
 wird er in -0 umgeändert.

Gleitkommazahlen

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

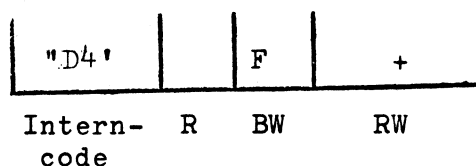
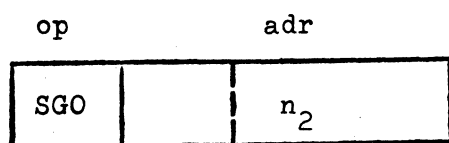
W2. entfällt

Ausführungszeit: 0,35 μ s

[1] siehe Abschnitt 4.1

SGO

SPRINGE, WENN GRÖßER 0



Sprungbedingung SB: $\langle A \rangle > 0$

SB ist erfüllt, wenn bei

$\langle A \rangle_t = 0$: $\langle A \rangle_1 = 0$ und nicht alle Bits $\langle A \rangle_1$ bis $\langle A \rangle_{40}$ gleich Mantisse
 $\langle A \rangle_t = 1$: $\langle A \rangle_1 = 0$ und $\langle A \rangle \neq \pm 0$
 $\langle A \rangle_t \geq 2$: $\langle A \rangle \neq \pm 0$

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB ist erfüllt

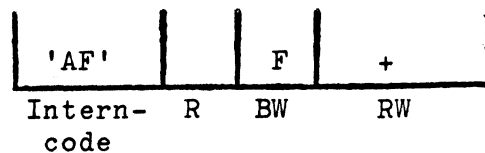
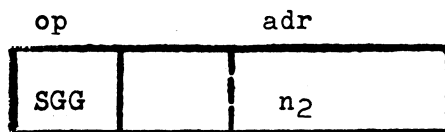
W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

W2. entfällt

Ausführungszeit: 0,05 μs

SPRINGE, WENN GRÖßER GLEICH



Sprungbedingung SB: $\langle A \rangle \geq \langle H \rangle$

[1]

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

falls $\langle A \rangle_t = \langle H \rangle_t = 0$

zusätzlich

$\langle A \rangle := \langle A \rangle$ normalisiert

falls $\langle A \rangle_t = \langle H \rangle_t = 0$ und $\langle H \rangle_1 = \langle H \rangle_2$

zusätzlich

$\langle H \rangle := \langle H \rangle$ normalisiert.

Falls in A der Exponent +0 ist,
wird er in -0 umgeändert.

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

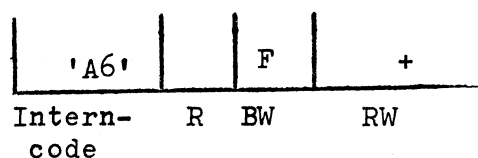
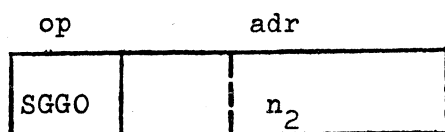
W2. entfällt

Gleitkommazahlen

Ausführungszeit: 0,35 μ s

[1] siehe Abschnitt 4.1

SPRINGE, WENN GRÖßER GLEICH 0



Sprungbedingung SB: $\langle A \rangle \geq 0$

SB ist erfüllt, wenn bei
 $\langle A \rangle_t = 0$: $\langle A \rangle_1 = 0$ oder alle Bits $\langle A \rangle_1$ bis $\langle A \rangle_{40}$ gleich
 $\langle A \rangle_t = 1$: $\langle A \rangle_1 = 0$ oder $\langle A \rangle = \pm 0$
 bei
 $\langle A \rangle_t \geq 2$: immer

Mantisse

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

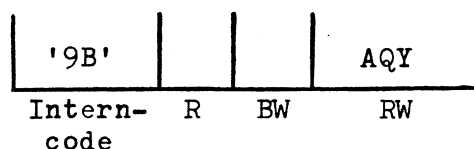
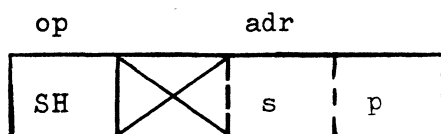
Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

W2. entfällt.

Ausführungszeit: 0,05 μs

SCHIFTE

Wirkung: W1. $\text{adr} := \text{adr} + \text{mod}2$

Adreßteil: Anzahl der Schiftschritte:
 $\pm 0 \leq p \leq 127$

Spezifikation:

$$s = ((A|Q|AQ|Z)([L][K]|[R])[U][B])| \\ (ZLR [U][B])|B|'0'$$

Variante 1: $s = (A|Q|AQ)[U][R]$

W2. Rechtsschift in A bzw. Q bzw.
 in A und in Q gesondert.

Ist die Tk im angegebenen Register
 ≤ 1 und gilt $s = (A|Q|AQ)[R]$,
 so werden Bits gleich $\langle A \rangle_1$ bzw.
 $\langle Q \rangle_1$ nachgezogen.

Ist die Tk im angegebenen Register
 ≥ 2 oder gilt $s = (A|Q|AQ)U[R]$,
 so werden Nullbits nachgezogen.

sinnvoll ist nur
 $0 \leq p \leq 96$

A,Q: Register einzeln
 Z: Register zusammen
 K: im Kreise
 L: nach links
 R: mit Rundung
 U: unabhängig von
 Typenkennung
 B: mit Zählung der
 L - Bits

Kurze Rechtsschifte

Zahlworte

$$\langle \rangle_t = 0 \text{ wie } \langle \rangle_t = 1$$

Nichtzahlwort oder unab-
 hängiger Schift

W3. Ist die Tk im angegebenen Register ≤ 1 und gilt $s = (A|Q|AQ)R$ und ist $p > 0$, so wird in Abhängigkeit vom zuletzt hinausgeschifteten Bit gerundet.

W4. Wurde in W3. $\langle A \rangle$ gerundet und gilt danach $\langle A \rangle_1 \neq \langle A \rangle_2$, dann A-Alarm.

nur bei ursprünglich $|\langle A \rangle_{1,48}| = 0L.....L$
und Befehl
SH $A[Q]R$ 1

Variante 2: $s = Z[U][R]$

Lange Rechtsschifte

Fall A: $s = Z[R]$ und
 $\langle A \rangle_t$ und $\langle Q \rangle_t \leq 1$

Zahlworte

W2. falls $\langle A \rangle_1 \neq \langle Q \rangle_1$:
 $\langle Q \rangle_{1,48} := \overline{\langle Q \rangle_{1,48}}$

Invertierung aller Bits,
auch bei $\langle Q \rangle_t = 0$

W3. Rechtsschift mit Nachziehen von Bits gleich $\langle A \rangle_1$. Dabei ist $\langle A \rangle_{48}$ mit $\langle Q \rangle_3$ verbunden, die Bits $\langle Q \rangle_{1,2}$ werden um-
umschiftet

rechte Bits gehen
verloren

W4. $\langle Q \rangle_2 := \langle Q \rangle_1$

W5. falls R in s enthalten und $p > 0$:
In Abhängigkeit vom zuletzt hinaus-
geschifteten Bit wird $\langle A, Q \rangle$ gerundet.

über Rundung siehe ent-
sprechend Befehl NRM

W6. falls in W5 gerundet wurde und danach
 $\langle A \rangle_1 \neq \langle A \rangle_2$:
A-Alarm

Fall B: $(s = Z[R] \text{ und } \langle A \rangle_t \text{ oder } \langle Q \rangle_t \geq 2)$
oder $s = ZU[R]$

Nichtzahlworte oder
unabhängiger Schift
R bedeutungslos

W2. Rechtsschift mit Nachziehen von 0-Bits,
dabei ist $\langle A \rangle_{48}$ mit $\langle Q \rangle_1$ verbunden.

rechte Bits gehen
verloren

Variante 3: $s = (A|Q|AQ)L[U]$

W2. Linksschift in A bzw. Q bzw. in A und Q gesondert.
Ist die Tk im angegebenen Register ≤ 1 und gilt $s = (A|Q|AQ)L$,
so werden Bits gleich $\langle A \rangle_1$ bzw. $\langle Q \rangle_1$ nachgezogen.

Ist die Tk im angegebenen Register ≥ 2 oder gilt $s = (A|Q|AQ)LU$,
so werden Nullbits nachgezogen.

W3. Ist die Tk im angegebenen Register ≤ 1 und gilt $s = A[Q]L$ und war nach irgendeinem der Schifftschriffe $\langle A \rangle_2$ ungleich dem ursprünglichen $\langle A \rangle_1$,
oder ist die Tk im angegebenen Register ≤ 1 und gilt $s = A[Q]L$ und wurde in A ein ursprünglich untergelaufenes Zahlwort um mindestens 5 Stellen geschifft,
dann A-Alarm.

Variante 4: $s = ZL[U][R]$

Fall C: $s = ZL[R]$ und $\langle A \rangle_t$ und $\langle Q \rangle_t \leq 1$

W2. falls $\langle A \rangle_1 \neq \langle Q \rangle_1$:
 $\langle A \rangle_{1,48} := \overline{\langle A \rangle_{1,48}}$

W3. $\langle Q \rangle_2 := \langle Q \rangle_1$

Kurze Linksschiffe

Zahlworte

Nichtzahlworte
oder unabhängiger
Schift

Lange Linksschiffe

Zahlworte

Invertierung aller Bits,
auch bei $\langle A \rangle_t = 0$

- W4. Linksschift mit Nachziehen von Bits gleich $\langle Q \rangle_1$. Dabei ist $\langle Q \rangle_3$ mit $\langle A \rangle_{48}$ verbunden, die Bits $\langle Q \rangle_{1,2}$ werden umschiftet.
- W5. Falls R in s enthalten:
In Abhängigkeit von $\langle Q \rangle_3$ wird gerundet auf einfache Wortlänge.
- W6. Falls R in s enthalten:
 $\langle Q \rangle := +0$
- W7. falls nach irgendeinem der Schiftschritte $\langle A \rangle_2 \neq \langle Q \rangle_1$ oder falls eine nach W2 untergelaufene Zahl um mindestens 5 Stellen geschiftet wurde oder falls in W5 gerundet wurde und danach $\langle A \rangle_1 \neq \langle A \rangle_2$:
A-Alarm

Jedoch kein Alarm bei
Befehl SH ZLR 0 mit ursprünglich
($\langle A \rangle_{1,48} = 0L\dots L$ und $\langle Q \rangle_3 = L$)
oder
($\langle A \rangle_{1,48} = 10\dots 0$ und $\langle Q \rangle_3 = 0$)
obwohl nach W5 $\langle A \rangle_1 \neq \langle A \rangle_2$

Fall D: ($s = ZL [R]$ und $\langle A \rangle_t$ oder $\langle Q \rangle_t \geq 2$)
oder $s = ZLU [R]$

Nichtzahlworte oder
unabhängiger Schift
R bedeutungslos

- W2. Linksschift mit Nachziehen von 0-Bits, dabei ist $\langle Q \rangle_1$ mit $\langle A \rangle_{48}$ verbunden.

linke Bits gehen
verloren

Variante 5: $s = (A|Q|AQ) [L] K [U]$

Kurze Kreisschifte, U
und Tk bedeutungslos

- W2. Kreisschift, wobei
 $\langle A \rangle_1$ mit $\langle A \rangle_{48}$ bzw.
 $\langle Q \rangle_1$ mit $\langle Q \rangle_{48}$ verbunden ist

Variante 6: $s = Z [L] K [U]$

Lange Kreisschifte
U und Tk bedeutungslos

- W2. Kreisschift, wobei
 $\langle A \rangle_{48}$ mit $\langle Q \rangle_1$ und
 $\langle Q \rangle_{48}$ mit $\langle A \rangle_1$ verbunden ist.

Variante 7: s = '0'

kein Register

W2. entfällt

Variante 8: s enthält B

Schift mit Bitzählung

Wie Varianten ohne Spezifikation B,
zusätzliche Wirkung:

$\langle Y \rangle :=$ Anzahl der aus $\langle A \rangle$ links
oder rechts hinausge-
schifteten L-Bits.

Bemerkung: zu allen Varianten

Falls $128 \leq p \leq 254$:
fehlerhafte Ausführung

Ausführungszeit:

$[0,10 (q+r) + 0,25] \mu s$, falls s kein bedeutungsvolles
R enthält

$[0,10 (q+r) + 0,50] \mu s$, falls s bedeutungsvolles
R enthält

wobei q und r sich ergeben aus

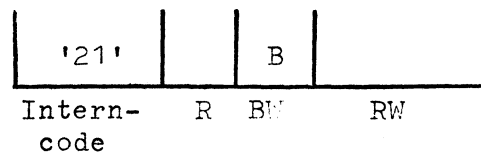
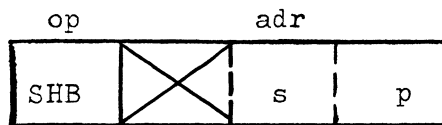
$p = 4q+r$ und

$r = 0, 1, 2, 3$

p, q ganzzahlig

SHB

SCHIFTE IN B



Adreßteil: s : L|R
 +0 ≤ p ≤ 24

Wirkung: W1. Der Inhalt von B wird je nach
 Spezifikation um die angegebene
 Zahl von Bit - Stellen nach
 rechts oder nach links geschiftet.
 Dabei werden entsprechend von
 links oder von rechts 0 - Bits
 nachgezogen.

Bemerkung: Falls 25 ≤ p ≤ 255 :
 := +0

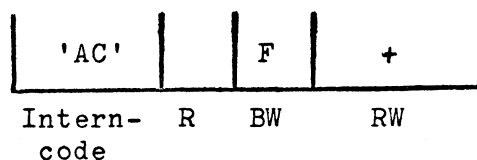
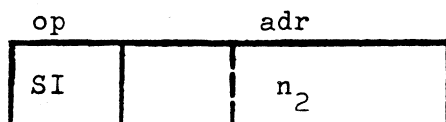
s: Schiftrichtung
 L: links
 R: rechts

alle ursprünglichen
 Stellen sind hinaus-
 geschoben.

Ausführungszeit: (0,2 n + 0,3) µs
 wobei n die Anzahl der Schiftschritte ist.
 Es wird höchstens solange geschiftet, bis := +0.

SI

SPRINGE, WENN IDENTISCH



Sprungbedingung SB: $\langle A \rangle = \langle H \rangle$

[1]

Wirkung: W1. $n_2 := n_2 + \text{mod}2$
 falls $\langle A \rangle_t = \langle H \rangle_t = 0$
 zusätzlich
 $\langle A \rangle := \langle A \rangle$ normalisiert
 falls $\langle A \rangle_t = \langle H \rangle_t = 0$ und $\langle H \rangle_1 = \langle H \rangle_2$
 zusätzlich
 $\langle H \rangle := \langle H \rangle$ normalisiert
 Falls in A der Exponent +0 ist,
 wird er in -0 geändert.

Gleitkommazahlen

Fall A: SB ist erfüllt

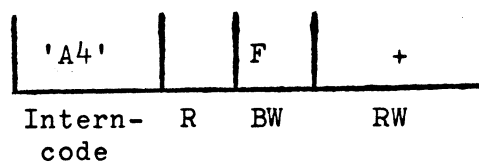
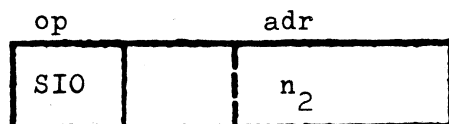
W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

W2. entfällt

Ausführungszeit: 0,35 μ s

[1] siehe Abschnitt 4.1

SPRINGE, WENN IDENTISCH 0Sprungbedingung SB: $\langle A \rangle = \pm 0$

SB ist erfüllt, wenn bei

 $\langle A \rangle_t = 0$: alle Bits $\langle A \rangle_1$ bis $\langle A \rangle_{40}$ gleich sind $\langle A \rangle_t = 1$: $\langle A \rangle = \pm 0$ $\langle A \rangle_t \geq 2$: $\langle A \rangle = + 0$

Mantisse

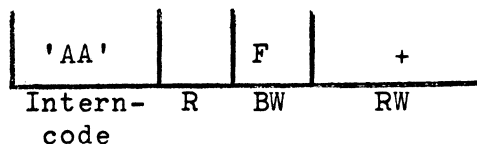
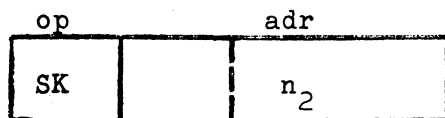
Wirkung: W1. $n_2 := n_2 + \text{mod}2$ Fall A: SB ist erfülltW2. $\langle F \rangle_{9,24} := n_2$ Fall B: SB ist nicht erfüllt

W2. entfällt

Ausführungszeit: 0,05 μ s

SK

SPRINGE, WENN KLEINER



Sprungbedingung SB: $\langle A \rangle < \langle H \rangle$

[1]

Wirkung:

W1. $n_2 := n_2 + \text{mod}2$

falls $\langle A \rangle_t = \langle H \rangle_t = 0$

zusätzlich

$\langle A \rangle := \langle A \rangle$ normalisiert

falls $\langle A \rangle_t = \langle H \rangle_t = 0$ und $\langle H \rangle_1 = \langle H \rangle_2$

zusätzlich

$\langle H \rangle := \langle H \rangle$ normalisiert.

Falls in A der Exponent +0 ist,
wird er in -0 geändert.

Gleitkommazahlen

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

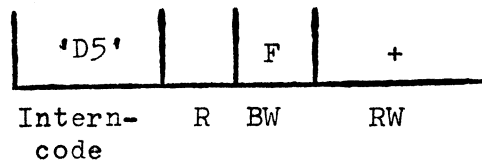
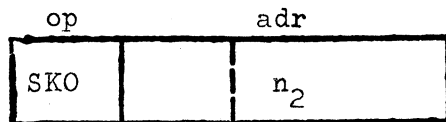
W2. entfällt

Ausführungszeit: 0,35 μ s

[1] siehe Abschnitt 4.1

SK0

SPRINGE, WENN KLEINER 0



Sprungbedingung SB: $\langle A \rangle < 0$

SB ist erfüllt, wenn bei

$\langle A \rangle_t = 0 : \langle A \rangle_1 = L$ und nicht alle Bits $\langle A \rangle_1$ bis $\langle A \rangle_{40}$ gleich Mantisse

$\langle A \rangle_t = 1 : \langle A \rangle_1 = L$ und $\langle A \rangle \neq \pm 0$

bei

$\langle A \rangle_t \geq 2 : \text{nie}$

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB ist erfüllt

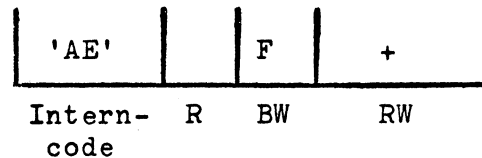
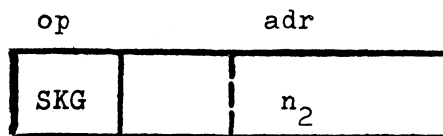
W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

W2. entfällt

Ausführungszeit: 0,05 μs

SPRINGE, WENN KLEINER GLEICH



Sprungbedingung SB: $\langle A \rangle \leq \langle H \rangle$

[1]

Wirkung:

W1. $n_2 := n_2 + \text{mod}2$
 falls $\langle A \rangle_t = \langle H \rangle_t = 0$,
 zusätzlich
 $\langle A \rangle := \langle A \rangle$ normalisiert
 falls $\langle A \rangle_t = \langle H \rangle_t = 0$ und $\langle H \rangle_1 = \langle H \rangle_2$
 zusätzlich
 $\langle H \rangle := \langle H \rangle$ normalisiert.
 Falls in A der Exponent +0 ist,
 wird er in -0 umgeändert.

Gleitkommazahlen

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

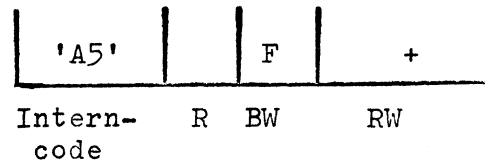
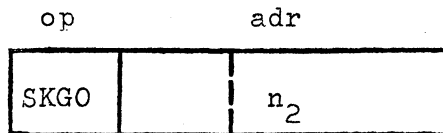
Fall B: SB ist nicht erfüllt

W2. entfällt

Ausführungszeit: 0,35 μ s

[1] siehe Abschnitt 4.1

SPRINGE, WENN KLEINER GLEICH 0



Sprungbedingung SB: $\langle A \rangle \leq 0$

SB ist erfüllt, wenn bei

$\langle A \rangle_t = 0$: $\langle A \rangle_1 = L$ oder alle Bits $\langle A \rangle_1$ bis $\langle A \rangle_{40}$ gleich

$\langle A \rangle_t = 1$: $\langle A \rangle_1 = L$ oder $\langle A \rangle = +0$

$\langle A \rangle_t \geq 2$: $\langle A \rangle = +0$

Mantisse

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

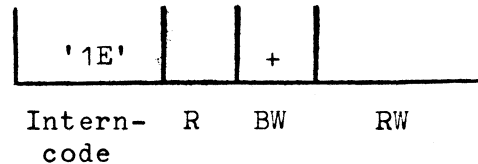
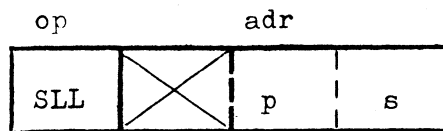
Fall B: SB ist nicht erfüllt

W2. entfällt

Ausführungszeit: 0,05 μs

SL

SPRINGE, WENN MERKLICHT



Adreßteil p: $-127 \leq p \leq 127$
 s: 0 v 1 v 2 v 3 ... v 8

Merklichter

Sprungbedingung SB: $s \neq 0$ und mindestens eines
 der durch s adressierten
 Merklichter ist gesetzt

Wirkung:

Fall A: SB ist erfüllt

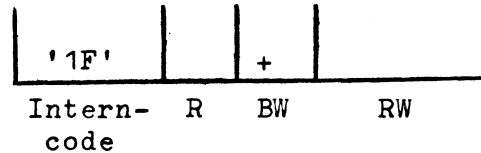
W1. $\langle F \rangle_{9,24} := \langle F \rangle + p$

Fall B: SB ist nicht erfüllt

W1. entfällt

Ausführungszeit: 0,45 μ s ,falls SB erfüllt
 0,05 μ s ,falls SB nicht erfüllt

SPRINGE, WENN MERKLICHT UND LÖSCHE



Merklichter

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := \langle F \rangle + p$

W2. die adressierten Merklichter
 werden gelöscht

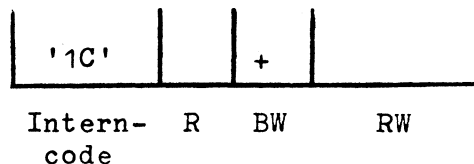
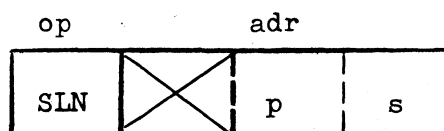
Fall B: SB ist nicht erfüllt

W1. entfällt

Ausführungszeit: 0,45 μ s , falls SB erfüllt
0,05 μ s , falls SB nicht erfüllt

SLN

SPRINGE, WENN MERKLICHT NICHT



Adreßteil p: $-127 \leq p \leq 127$
s: 0 v 1 v 2 v ... v 8

Merklichter

Sprungbedingung SB: s \neq 0 und keines durch s adressierten Merklichter ist gesetzt

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := \langle F \rangle + p$

Fall B: SB ist nicht erfüllt

W1. entfällt

Ausführungszeit: 0,45 μ s ,falls SB erfüllt
0,05 μ s ,falls SB nicht erfüllt

SM

SPRINGE, WENN MARKE

op	adr
SM	n_2

'34'		F	M
Intern- code	R	BW	RW

Sprungbedingung SB: $\langle M \rangle = L$

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$
 $\langle M \rangle := 0$

Fall B: SB ist nicht erfüllt

W2. entfällt

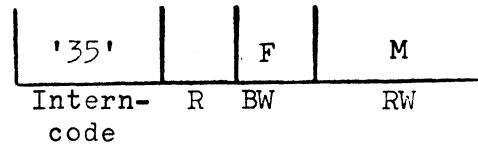
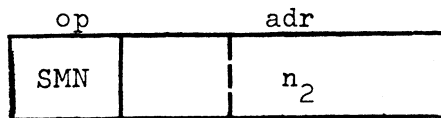
Marke wird
gelöscht

Bemerkung: Der Befehl SM belegt nicht das
Rechenwerk

Ausführungszeit: 0,05 μs

SMN

SPRUNGE, WENN MARKE NICHT



SPRUNGBEDINGUNG SB: $\langle M \rangle = 0$

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

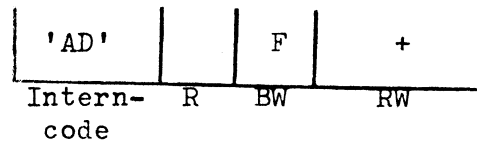
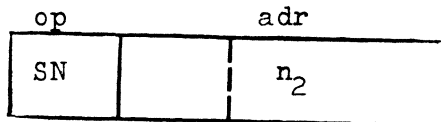
W2. $\langle M \rangle := 0$

Marke wird
gelöscht

Bemerkung: Der Befehl SMN belegt nicht
das Rechenwerk

Ausführungszeit: 0,05 μs

SPRINGE, WENN NICHT IDENTISCH



Sprungbedingung SB: $\langle A \rangle \neq \langle H \rangle$

[1]

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Falls $\langle A \rangle_t = \langle H \rangle_t = 0$:

zusätzlich

$\langle A \rangle := \langle A \rangle$ normalisiert.

Falls $\langle A \rangle_t = \langle H \rangle_t = 0$ und $\langle H \rangle_1 = \langle H \rangle_2$

zusätzlich

$\langle H \rangle := \langle H \rangle$ normalisiert.

Gleitkommazahlen

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

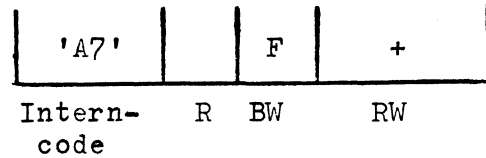
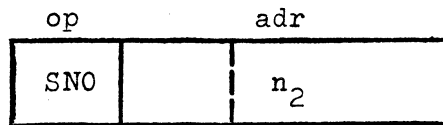
W2. entfällt

Ausführungszeit: 0,35 μ s

[1] siehe Abschnitt 4.1

SN0

SPRINGE, WENN NICHT 0



Sprungbedingung SB: $\langle A \rangle \neq 0$

SB ist erfüllt, wenn bei

$\langle A \rangle_t = 0$: nicht alle Bits $\langle A \rangle_1$ bis $\langle A \rangle_{40}$ gleich sind

$\langle A \rangle_t = 1$: $\langle A \rangle \neq 0$

$\langle A \rangle_t \geq 2$: $\langle A \rangle \neq 0$

Mantisse

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

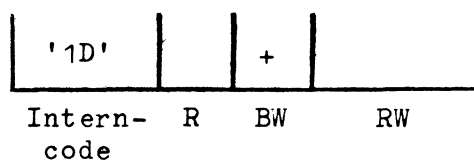
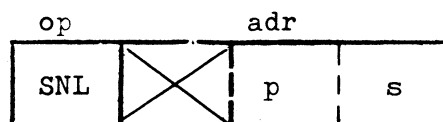
Fall B: SB nicht erfüllt

W2. entfällt

Ausführungszeit: 0,05 μs

SNL

SPRINGE, WENN MERKLICHT NICHT, SONST LÖSCHE



Adreßteil p: $-127 \leq p \leq 127$
s: 0 v 1 v 2 v 3 v ... v 8

Merklichter

Sprungbedingung: s \neq 0 und keines der durch s adressierten Merklichter ist gesetzt

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := \langle F \rangle + p$

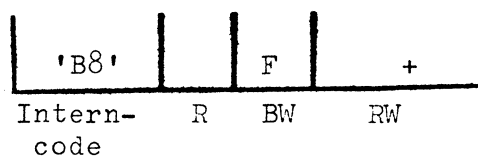
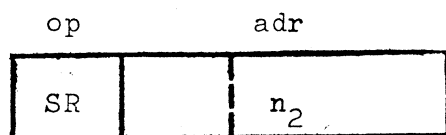
Fall B: SB ist nicht erfüllt

W1. die durch s angegebenen Merklichter werden gelöscht

Ausführungszeit: 0,45 μ s ,falls SB erfüllt
0,05 μ s ,falls SB nicht erfüllt

SR

SPRUNGE, WENN RECHTES BIT IN A GESETZT



Sprungbedingung SB: $\langle A \rangle_{48} = L$

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

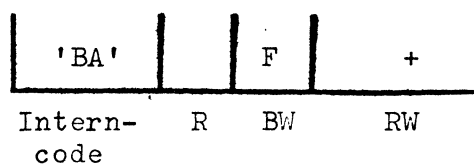
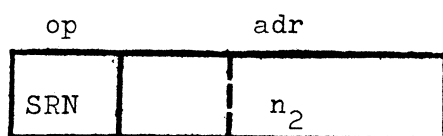
Fall B: SB ist nicht erfüllt

W2. entfällt

Ausführungszeit: 0,05 μ s

SRN

SPRUNGE, WENN RECHTES BIT IN A NICHT GESETZT



Sprungbedingung SB: $\langle A \rangle_{48} = 0$

Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

W2. entfällt

Ausführungszeit: 0,05 μ s

SPRINGE INS SYSTEM UND RESERVIERE

Wirkung: W1. $\text{adr} := \text{adr} + \text{mod}2$
Adreßteil: $+0 \leq p_L \leq 255 ; +0 \leq p_R \leq 255$

Fall A: Der Rechner läuft im Normalmodus
(BEBN=L, BEBY=0, BEB ϕ =0)

W2. $\langle (\langle \text{BL} \rangle \cdot 2^8 + 21)_2 \rangle :=$ Steuerbits 25 bis 48 Reservierung

W3. Steuerbits 25 bis 42 und 44 bis 45 := 0
 BEB ϕ := L Abwicklermodus wird
 eingestellt

W4. $\langle (\langle \text{BL} \rangle \cdot 2^8 + 18)_2 \rangle := \langle \text{B} \rangle$
 $\langle (\langle \text{BL} \rangle \cdot 2^8 + 19)_2 \rangle := \text{adr}$
 $\langle (\langle \text{BL} \rangle \cdot 2^8 + 20)_2 \rangle := \langle \text{F} \rangle + 1$
 $\langle \langle \text{BL} \rangle \cdot 2^8 + 18 \rangle_+ := 3$
 $\langle \langle \text{BL} \rangle \cdot 2^8 + 20 \rangle_+ := 3$

W5. $\langle \text{LZ1} \rangle := \langle \langle \text{BL} \rangle \cdot 2^8 + 65 \rangle_{8,12}$ Abwickler Δ werden
 $\langle \text{LZ2} \rangle := \langle \langle \text{BL} \rangle \cdot 2^8 + 65 \rangle_{3,7}$ eingestellt

W6. $\langle \text{F} \rangle := \langle (\langle \text{BL} \rangle \cdot 2^8 + 6)_2 \rangle$

W7. $\langle \text{B} \rangle := \text{adr}$

Fall B: Der Rechner läuft im Abwicklermodus
(BEBN=L, BEBY=0, BEB ϕ =L)

W2. BEFE := L Eingriffssperre wird
 gesetzt

W3. $\langle (\langle \text{BL} \rangle \cdot 2^8 + 13)_2 \rangle :=$ Steuerbits
 25 bis 48

W4. Steuerbits 25 bis 42 und 44
 bis 45 := 0

W5. $\langle (\langle BL \rangle \cdot 2^8 + 10)_2 \rangle := \langle B \rangle$
 $\langle (\langle BL \rangle \cdot 2^8 + 11)_2 \rangle := \text{adr}$
 $\langle (\langle BL \rangle \cdot 2^8 + 12)_2 \rangle := \langle F \rangle + 1$
 $\langle \langle BL \rangle \cdot 2^8 + 10 \rangle_t := 3$
 $\langle \langle BL \rangle \cdot 2^8 + 12 \rangle_t := 3$
 W6. $\langle F \rangle := \langle 6_2 \rangle$

 W7. $\langle B \rangle := \text{adr}$

Fall C: Der Rechner läuft im Spezial-oder Systemmodus.

(BEBN=beliebig, BEBY=L, BEB=beliebig)

W2. $\langle 13_2 \rangle := \text{Steuerbits 25 bis 48}$
 W3. Steuerbits 25 bis 42 und 44 bis 45:=0
 BEBN := 0

Systemmod. wird eingest.

W4. $\langle 10_2 \rangle := \langle B \rangle$
 $\langle 11_2 \rangle := \text{adr}$
 $\langle 12_2 \rangle := \langle F \rangle + 1$

 $\langle 10 \rangle_t := 3$
 $\langle 12 \rangle_t := 3$

W5. Die Indexzellen, für die das Rück-
 speicherungsbit = L ist, werden
 zurückgespeichert.

W6. $\langle XB \rangle := \langle 4_2 \rangle$

W7. $\langle F \rangle := \langle 6_2 \rangle$

W8. $\langle B \rangle := \text{adr}$

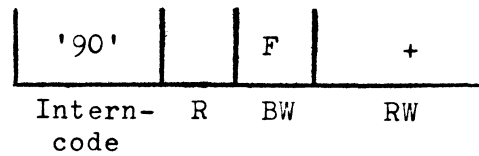
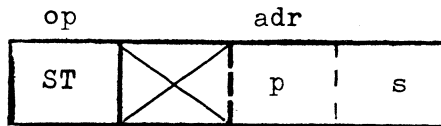
Ausführungszeit: Fall A: 2,40 μs
 Fall B: 1,60 μs
 Fall C: $(2,65 + 0,85x) \mu\text{s}$
 wobei x = Anzahl der
 zurückgespeicherten
 Indexregister

Register Flip-Flop	Bei Speicherschutzalarm bei der Berechnung der Index- basis	Bei Speicherschutzalarm beim Auslesen des rech- ten Befehlshalbwortes
BA, BC, BB, BF	wie vor Beginn der Ausführungsphase des VPU	
BL	wie nach Ausführung von W3	
BLZ1, BLZ2	$\Delta 1$ und $\Delta 2$, mit denen die Indexbasis berechnet werden sollte	wie nach Ausführung von W7
Seitenadr.- Register	wie nach Ausführung von W2	
Index- register	wie nach Ausführung von W2	
BXB, BXBZ	undefiniert	wie nach Ausführung von W6
RS	<p style="text-align: center;">TK = 3</p> <p>RS_{linkes HW} : alarmerzeugende Adresse</p> <p>= relative Indexbasis</p> <p>RS_{rechtes HW} : Adressteil des VPU</p>	
Steuerbits	<p>BEBO: = L Indexbasis wurde mit Δ's des Ab- wicklers berech- net</p> <p>= 0 Indexbasis wurde mit Δ's des Ope- rators berechnet</p> <p>BEIC = L BEEC = L BEMB = BEMU = 0</p> <p>alle übrigen Flip-Flop blei- ben unverändert.</p>	<p>BEBO: entspr. $\langle I+4 \rangle_{46}$</p> <p>BEIC = 0 BEEC = L BEMB = BEMU = BEML = 0</p> <p>alle übrigen Flip-Flop bleiben unverändert.</p>

Ausführungszeit: 7,5 μ s

ST

SPRINGE, WENN TYPENKENNUNG



Wirkung: W1. $\text{adr} := \text{adr} + \text{mod}2$

Adreßteil p: $-127 \leq p \leq 127$

$s = s_1 s_2$
 $s_1 = 0|1|2|3$
 $s_2 = 0|A|Q|D|H$

Sprungbedingung SB: $s_2 \neq 0$ und $\langle s_2 \rangle_t = s_1$

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := \langle F \rangle + p$

Fall B: SB ist nicht erfüllt

W2. entfällt

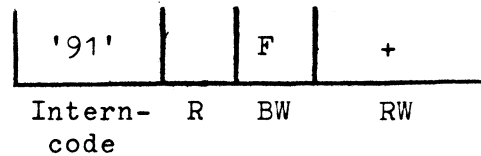
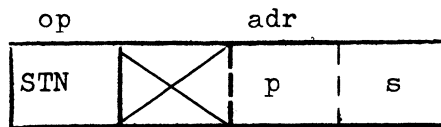
Typenkennung
Register

Die Tk im angegebenen Register
stimmt mit s_1
überein

Ausführungszeit: Fall A: 0,50 μ s
Fall B: 0,15 μ s

STN

SPRINGE, WENN TYPENKENNUNG NICHT



Wirkung: W1. $\text{adr} := \text{adr} + \text{mod}2$

Adreßteil p: $-127 \leq p \leq 127$

s: $s = s_1 s_2$

$s_1 = 0|1|2|3$

$s_2 = 0|A|Q|D|H$

Typenkennung
Register

Sprungbedingung SB: $\langle s_2 \rangle \neq 0$ und $\langle s_2 \rangle_t \neq s_1$

Fall A: SB ist erfüllt

W2. $\langle F \rangle_{9,24} := \langle F \rangle + p$

Fall B: SB ist nicht erfüllt

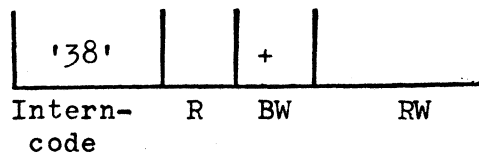
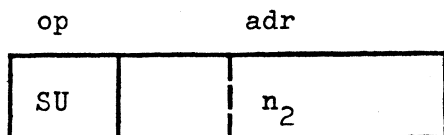
W2. entfällt

Die Tk im angegebenen Register
stimmt nicht mit
 s_1 überein

Ausführungszeit: Fall A: 0,50 μs

Fall B: 0,15 μs

SPRINGE IN UNTERPROGRAMM



Wirkung: W1. $n_2 := n_2 + \text{mod}2$

Fall A: $FE \neq \langle U \rangle \neq FF$

W2. $\langle U \rangle := \langle U \rangle + 1$

W3. $\langle \langle U \rangle \rangle := \langle F \rangle + 1$

W4. $\langle F \rangle_{9,24} := n_2$

Fall B: $\langle U \rangle = FF$

W2. $\langle U \rangle := +0$

W3. $\langle \langle U \rangle \rangle := \langle F \rangle + 1$

W4. $\langle F \rangle_{9,24} := n_2$

Fall C: $\langle U \rangle = FE$

W2. $\langle U \rangle := FF$

W3. $\langle \langle U \rangle \rangle := \langle F \rangle + 1$

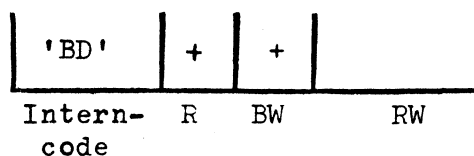
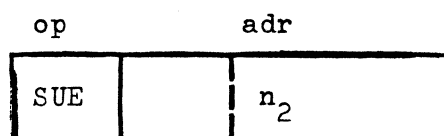
W4. $\langle F \rangle_{9,24} := n_2$

W5. U-Alarm

Normalfall

Ausführungszeit: 0,40 μs

SPRINGE IN UNTERPROGRAMM NACH ERSETZUNG



Wirkung:

Fall A: $FE \neq \langle U \rangle \neq FF$

- W1. $\langle U \rangle := \langle U \rangle + 1$
- W2. $\langle \langle U \rangle \rangle := \langle F \rangle + 1$
- W3. $\langle F \rangle := \langle n_2 \rangle + \text{mod}2$

Fall B: $\langle U \rangle = FF$

- W1. $\langle U \rangle := + 0$
- W2. $\langle \langle U \rangle \rangle := \langle F \rangle + 1$
- W3. $\langle F \rangle := \langle n_2 \rangle + \text{mod}2$

Fall C: $\langle U \rangle = FE$

- W1. $\langle U \rangle := FF$
- W2. $\langle \langle U \rangle \rangle := \langle F \rangle + 1$
- W3. $\langle F \rangle := \langle n_2 \rangle + \text{mod}2$
- W4. U-Alarm

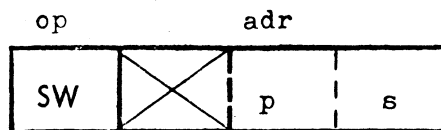
Normalfall

$\langle n_2 \rangle_t$ ohne Bedeutung

Bemerkung: Adreßteil wie beim Befehl B2 . (Falls nicht modifiziert, nur Großseite 0.)

Ausführungszeit: 0,40 μs

SPRINGE, WENN WAHLSCHALTER



Intern-
code

R

BW

RW

Voraussetzung: BEBY=L oder BEBT=BEWA=L,
sonst Makro

System-Spezial- oder
Wartungsmodus

Adreßteil p: $-127 \leq p \leq 127$
s: 0 v 1 v 2 v 3 ...v 8

Wahlschalter

Sprungbedingung SB: s \neq 0 und mindestens einer
der durch s adressierten
Wahlschalter ist gesetzt

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := \langle F \rangle + p$

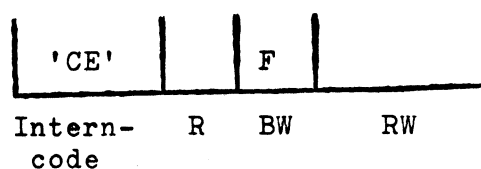
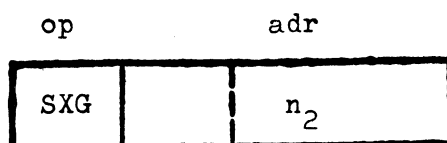
Fall B: SB ist nicht erfüllt

W1. entfällt

Ausführungszeit: 0,45 μ s ,falls SB erfüllt
0,05 μ s ,falls SB nicht erfüllt

SXG

SPRINGE, WENN INDEX GRÖßER 0



Sprungbedingung SB: $\langle B \rangle > +0$

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

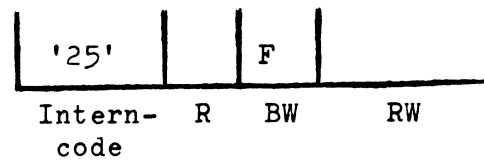
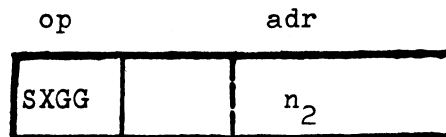
W1. entfällt

Wirkung wie Null-
befehl

Ausführungszeit: 0,05 μ s

SXGG

SPRINGE, WENN INDEX GRÖßER GLEICH 0



Sprungbedingung SB: $\langle B \rangle \geq \pm 0$

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

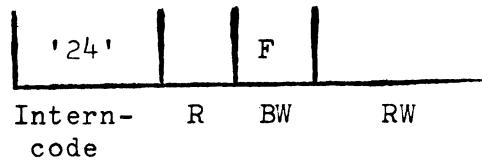
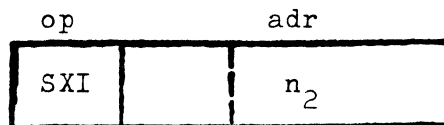
W1. entfällt

Wirkung wie Null-
befehl

Ausführungszeit: 0,05 μ s

SXI

SPRINGE, WENN INDEX IDENTISCH 0



Sprungbedingung SB: $\langle B \rangle = \pm 0$

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

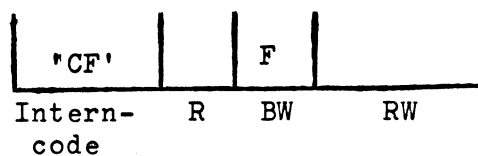
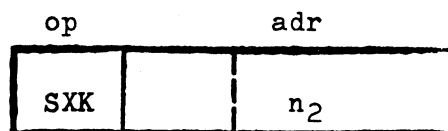
W1. entfällt

Wirkung wie Null-
befehl

Ausführungszeit: 0,05 μ s

SXK

SPRUNGE, WENN INDEX KLEINER 0



Sprungbedingung SB: $\langle B \rangle < -0$

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

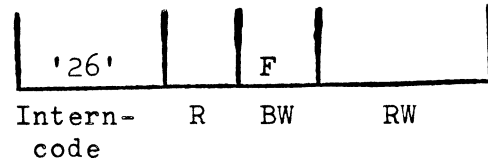
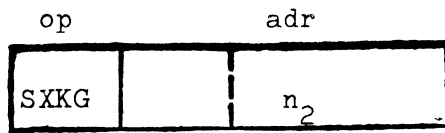
W1. entfällt

Wirkung wie Null-
befehl

Ausführungszeit: 0,05 μ s

SXKG

SPRUNGE, WENN INDEX KLEINER GLEICH 0



Sprungbedingung SB: $\langle B \rangle \leq \pm 0$

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

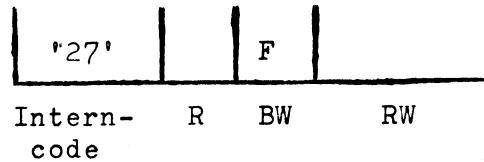
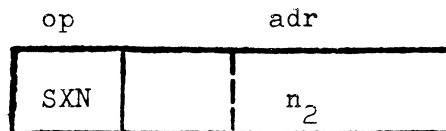
W1. entfällt

Wirkung wie Null-
befehl

Ausführungszeit: 0,05 μ s

SXN

SPRINGE, WENN INDEX NICHT IDENTISCH 0



Sprungbedingung SB: $\langle B \rangle \neq 0$

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := n_2$

Fall B: SB ist nicht erfüllt

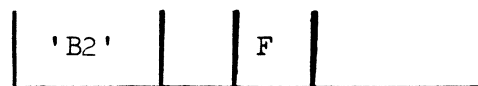
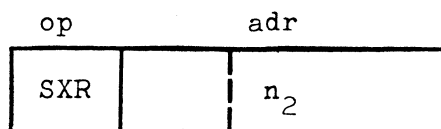
W1. entfällt

Wirkung wie Null-
befehl

Ausführungszeit: 0,05 us

SXR

SPRINGE, WENN INDEXGRÖSSE RECHTES BIT = L



Intern-
code R BW RW

Sprungbedingung SB: $\langle B \rangle_{24} = L$

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := n_2$

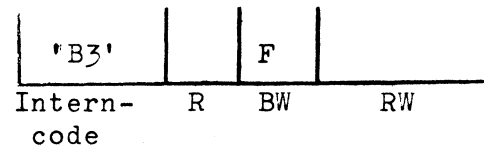
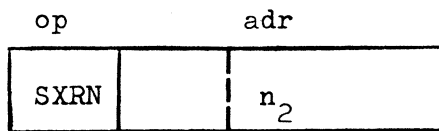
Fall B: SB ist nicht erfüllt

W1. entfällt

Ausführungszeit: 0,05 µs

SXRN

SPRUNGE, WENN INDEXGRÖSSE RECHTES BIT NICHT 1



Sprungbedingung SB: $\langle B \rangle_{24} = 0$

Wirkung:

Fall A: SB ist erfüllt

W1. $\langle F \rangle_{9,24} := n_2$

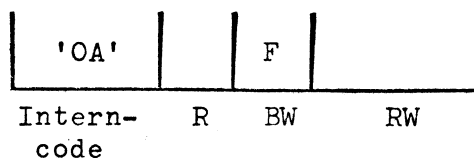
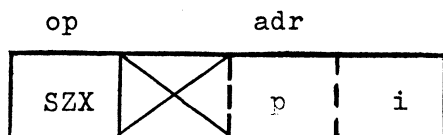
Fall B: SB ist nicht erfüllt

W1. entfällt

Ausführungszeit: 0,05 μ s

SZX

SPRINGE UND ZÄHLE, WENN INDEX KLEINER 0



Wirkung:

Fall A: $\langle i \rangle < \pm 0$

W1. $\langle F \rangle_{9,24} := \langle F \rangle + p$
 $-127 \leq p \leq +127$

W2. $\langle i \rangle := \langle i \rangle + 1$

Fall B: $\langle i \rangle \geq \pm 0$

W1. und W2. entfallen

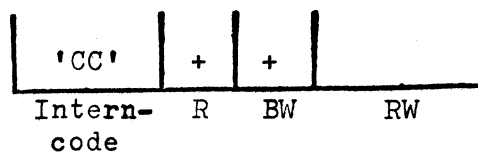
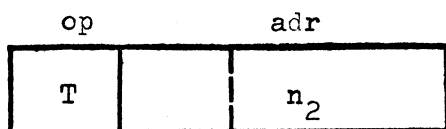
letzter ausgeführter Sprung
bei ursprünglich $\langle i \rangle = -1$

Wirkung wie
NULL-Befehl

Ausführungszeit: Fall A: 0,55 μ s
Fall B: 0,10 μ s

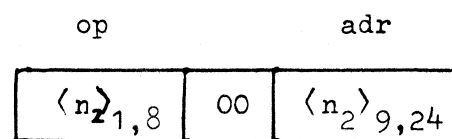
T

TU



Wirkung: W1. op := $\langle n_2 \rangle_{1,8}$
 adr_{1,8} := 0
 adr_{9,24} := $\langle n_2 \rangle_{9,24}$
 W2. Ausgang 2

Definition des Befehls

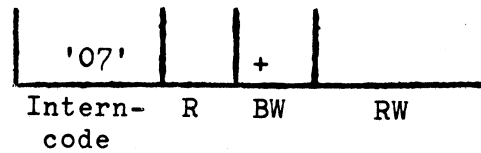
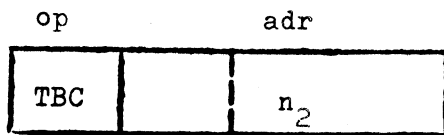


Bemerkung: $\langle n_2 \rangle_t$ beliebig

Ausführungszeit: 0,55 μ s

TBC

TRANSPORT AUS B NACH SPEICHER

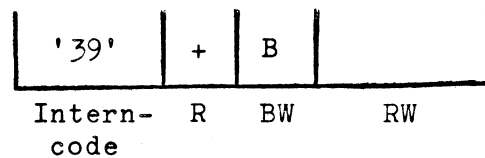
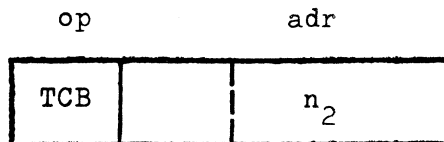


Wirkung: W1. $\langle n_2 \rangle := \langle B \rangle$

Ausführungszeit: 0,35 μ s

TCB

TRANSPORT AUS SPEICHER NACH B



Wirkung: W1. $n_2 := n_2 + \text{mod}2$
 W2. $\langle B \rangle := \langle n_2 \rangle$

Ausführungszeit: 0,10 μ s

TDM

TABELLE DURCHSUCHEN MIT DEHNUNG UND MASKE



Eingangsgrößen :

$\langle D \rangle$ = Suchwort

$\langle H \rangle$ = Maske

$\langle B \rangle$ = Dehnungswert $\neq \pm 0$. Falls $\langle B \rangle = \pm 0$:
Wirkung wie NULL-Befehl

$\langle B \rangle$ darf negativ und ungerade sein

Suchkriterium:

$$\langle n + k\langle B \rangle \rangle \wedge \langle H \rangle = \langle D \rangle \wedge \langle H \rangle$$

$$k = 0, 1, 2, \dots$$

Wirkung:

W1. Die Tabelle $\langle n \rangle, \langle n + \langle B \rangle \rangle, \dots, \langle n + k\langle B \rangle \rangle, \dots$ wird mit $\langle D \rangle$ verglichen, bis ein Wort auftritt, das dem Suchkriterium genügt, oder dessen T_k ungleich $\langle D \rangle_t$ ist, in diesem Fall zusätzlich mit T_k -Alarm.

W2. $\langle B \rangle := n^+$ (Adresse des in W1 zuletzt untersuchten Wortes)

$$\langle A \rangle := \langle H \rangle \wedge \langle n^+ \rangle$$

$$\langle A \rangle_t := \langle H \rangle_t$$

Identität in den von der Maske ausgeblendetten Stellen

D

X....X	.. $\langle D \rangle_{i..}$
--------	------------------------------

n

N....N	.. $\langle n \rangle_{i..}$
--------	------------------------------

H

L....L	O....O
--------	--------

A

N....N	O....O
--------	--------

TLD

TABELLE DURCHSUCHEN MIT DEHNUNG

op	adr
TLD	n

'EB'		B	A D
Intern- code	R	BW	RW

Eingangsgrößen:

$\langle D \rangle$ = Suchwort; falls $\langle D \rangle_t = 0, 1$, muß sein

$$\langle D \rangle_1 = \langle D \rangle_2$$

$\langle B \rangle$ = Dehnungswert $\neq \pm 0$. Falls $\langle B \rangle = \pm 0$:

Wirkung wie NULL-Befehl

$\langle B \rangle$ darf negativ
und ungerade sein

Suchkriterium:

$\langle n + k \langle B \rangle \rangle \geq \langle D \rangle$. Der Vergleich wird wie bei
SGG ausgeführt, ist also Tk-abhängig.

Falls $\langle n + k \langle B \rangle \rangle_t \leq 1$, ist $\langle n + k \langle B \rangle \rangle_m$ ohne Bedeutung.

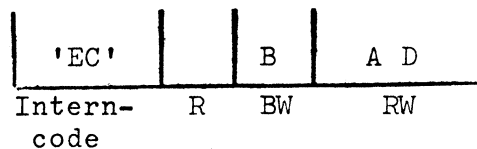
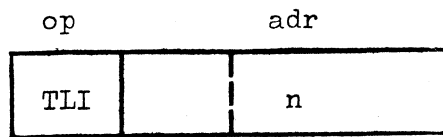
Wirkung: W1. Die Tabelle $\langle n \rangle, \langle n + \langle B \rangle \rangle, \dots, \langle n + k \langle B \rangle \rangle, \dots$
wird mit $\langle D \rangle$ verglichen.
Falls $\langle D \rangle_t = 0$ und $\langle n \rangle_t = 0$ zusätzlich:
 $\langle D \rangle := \langle D \rangle$ normalisiert

Der Prozeß wird abgebrochen, wenn in der Tabelle
ein Wort auftritt, das dem Suchkriterium genügt,
oder dessen Tk ungleich $\langle D \rangle_t$ ist, in diesem Fall
mit Tk-Alarm.

W2. $\langle B \rangle :=$ Adresse des zuletzt untersuchten Wortes
 $\langle A \rangle := \langle D \rangle$
 $\langle A \rangle_t := \langle D \rangle_t$

Ausführungszeit: $(0,725p + 0,525) \mu s$

TABELLE DURCHSUCHEN AUF IDENTITÄT



Eingangsgrößen:

$\langle D \rangle$ = Suchwort. Falls $\langle D \rangle_t = 0, 1$ muß gelten $\langle D \rangle_1 = \langle D \rangle_2$,
sonst falsche Befehlsausführung.

Suchkriterium:

$\langle n + 2k \rangle = \langle D \rangle$. Der Vergleich wird wie bei SI
ausgeführt, ist daher Tk-abhängig
Falls $\langle n + 2k \rangle_t \leq 1$, ist $\langle n + 2k \rangle_m$ ohne Bedeutung.

Wirkung:

W1. Die Tabelle $\langle n \rangle, \langle n + 2 \rangle, \dots, \langle n + 2k \rangle, \dots$
wird mit $\langle D \rangle$ verglichen.
Falls $\langle D \rangle_t = 0$ und $\langle n \rangle_t = 0$ zusätzlich:
 $\langle D \rangle := \langle D \rangle$ normalisiert

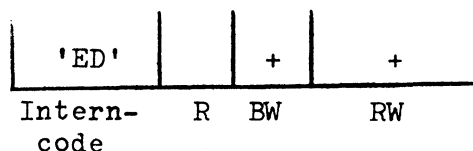
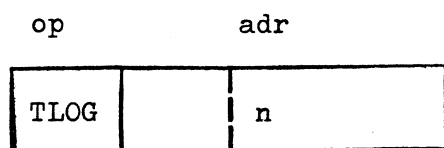
Der Prozeß wird abgebrochen, wenn in der
Tabelle ein Wort auftritt, das dem Such-
kriterium genügt, oder dessen Tk ungleich
 $\langle D \rangle_t$ ist, in diesem Fall zusätzlich mit
Tk-Alarm.

W2. $\langle B \rangle :=$ Adresse des in W1 zuletzt untersuchten Wortes
 $\langle A \rangle := \langle D \rangle$
 $\langle A \rangle_t := \langle D \rangle_t$

Ausführungszeit: $(0,75 p + 0,50) \mu s$
wobei p = Anzahl der untersuchten Wörter

TLOG

TABELLE DURCHSUCHEN LOGARITHMISCH



Eingangsgrößen:

$\langle A \rangle$ = Tabellenlänge 1 in Ganzworten

$\langle D \rangle$ = Suchwort $2 \leq 1 \leq 2^{20}$

$\langle H \rangle$ = Maske

stets Tk beliebig

Es muß gelten $\langle n \rangle \wedge \overline{\langle H \rangle} < \langle D \rangle \wedge \overline{\langle H \rangle}$

(Vergleich und Verknüpfung wie bei Tk 2 oder 3)

Bei geeignetem Aufbau der Tabelle kann der TLOG auch bei Tabellen mit Dehnung verwendet werden

Suchkriterium: $\langle n+2k \rangle \wedge \overline{\langle H \rangle} \geq \langle D \rangle \wedge \overline{\langle H \rangle}$

Der duale Wert in dem dem Nullfeld der Maske entsprechenden Teil des Tabellenwortes muß \geq dem des Suchwortes sein. Alle Operationen verlaufen wie bei Tk 2 oder Tk 3

Wirkung:

W1. Die geordnete Tabelle $\langle n \rangle, \langle n+2 \rangle, \dots, \langle n+2k \rangle, \dots, \langle n+2(\langle A \rangle - 1) \rangle$ wird mit $\langle D \rangle$ nach dem Intervallschachtelverfahren ("logarithmisch") durchsucht, bis ein Wort auftritt, das dem Suchkriterium genügt.

W2. $\langle B \rangle := n^+$, Adresse des gefundenen Wortes, wobei n^+ die kleinstmögliche Adresse ist, oder des ersten Wortes hinter der Tabelle

$\langle D \rangle := \langle D \rangle \wedge \overline{\langle H \rangle}$

Fall A: Suchkriterium erfüllt mit $\langle n^+ \rangle \wedge \overline{\langle H \rangle} = \langle D \rangle \wedge \overline{\langle H \rangle}$

Identität im
Nullfeld

$$\begin{aligned} W3. \quad \langle A \rangle, \langle Q \rangle &:= \langle H \rangle \wedge \langle n^+ \rangle \\ \langle A \rangle_t, \langle Q \rangle_t &:= \langle n^+ \rangle_t \end{aligned}$$

Fall B: Suchkriterium erfüllt mit $\langle n^+ \rangle \wedge \overline{\langle H \rangle} > \langle D \rangle \wedge \overline{\langle H \rangle}$

$$\begin{aligned} W3. \quad \langle A \rangle, \langle Q \rangle &:= \langle H \rangle \wedge \langle n^+ \rangle \\ \langle A \rangle_t, \langle Q \rangle_t &:= \langle n^+ \rangle_t \end{aligned}$$

$$W4. \quad \langle F \rangle_{9,24} := \langle F \rangle + 2$$

übernächster
Befehl

Fall C: Suchkriterium nicht erfüllt d. h.
es wurde kein Wort gefunden

$$\begin{aligned} W3. \quad \langle A \rangle, \langle Q \rangle &:= +0 \\ \langle A \rangle_t, \langle Q \rangle_t &:= 3 \end{aligned}$$

$$W4. \quad \langle F \rangle_{9,24} := \langle F \rangle + 2$$

$$W5. \quad A\text{-Alarm}$$

übernächster Befehl
verbunden mit
A-Alarm

Bemerkung: Falls $1 \leq l$ oder $l > 2^{20}$

W1. entfällt

$$\begin{aligned} W2. \quad \langle B \rangle &:= +0 \\ \langle D \rangle &:= \langle D \rangle \wedge \overline{\langle H \rangle} \end{aligned}$$

W3 bis W5 wie Fall C

Ausführungszeit für $2 \leq l \leq 2^{20}$:
(1,15 p + 2,30) μ s
wobei
 $(2^{\log l} - 1) \leq p < 2^{\log l}$
p ganzzahlig

TMAX

TABELLE DURCHSUCHEN AUF MAXIMUM

op	adr
TMAX	n

'EF'		B	AQD
Intern- code	R	BW	RW

Eingangsgrößen :

$\langle H \rangle$ = Maske

$\langle B \rangle$ = Dehnungswert $\neq \pm 0$. Falls $\langle B \rangle = \pm 0$:
Wirkung wie NULL-Befehl

$\langle B \rangle$ darf negativ
und ungerade sein

Wirkung:

- W1. Die Tabelle $\langle n \rangle, \langle n + \langle B \rangle \rangle, \dots, \langle n + k \langle B \rangle \rangle, \dots$ für $k = 0, 1, 2, \dots$ wird in dem dem Nullfeld der Maske entsprechenden Teil ($\langle H \rangle_1 = 0$) auf Bitmuster-Maximum durchsucht.
- W2. $\langle B \rangle := n^+$ (Adresse des in W1 gefundenen Wortes)
 $\langle D \rangle := \overline{\langle H \rangle} \wedge \langle n^+ \rangle$
 $\langle D \rangle_t := \langle n^+ \rangle_t$
- W3. $\langle A \rangle, \langle Q \rangle := \langle H \rangle \wedge \langle n^+ \rangle$
 $\langle A \rangle_t, \langle Q \rangle_t := \langle n^+ \rangle_t$

Bemerkung:

Die Tabelle besteht nur aus den Worten, für die gilt: $\langle n + k \langle B \rangle \rangle_t = \langle n \rangle_t$

Bei mehreren Maxima gilt:

n^+ = Adresse des in W1 zuerst gefundenen Maximums

Ausführungszeit: $(0,85 p + 0,20) \mu s$

wobei p = Anzahl der Worte der Tabelle

TMIN

TABELLE DURCHSUCHEN AUF MINIMUM



Eingangsgroößen :

$\langle H \rangle$ = Maske

$\langle B \rangle$ = Dehnungswert $\neq \pm 0$. Falls $\langle B \rangle = \pm 0$:
Wirkung wie NULL-Befehl

$\langle B \rangle$ darf negativ
und ungerade sein

Wirkung:

W1. Die Tabelle $\langle n \rangle, \langle n + \langle B \rangle \rangle, \dots, \langle n + k \langle B \rangle \rangle, \dots$ für $k = 0, 1, 2, \dots$ wird in dem dem Nullfeld der Maske entsprechenden Teil ($\langle H \rangle_i = 0$) auf Bitmuster-Minimum durchsucht.

W2. $\langle B \rangle := n^+$ (Adresse des in W1 gefundenen Wortes)

$\langle D \rangle := \overline{\langle H \rangle} \wedge \langle n^+ \rangle$

$\langle D \rangle_t := \langle n^+ \rangle_t$

W3. $\langle A \rangle, \langle Q \rangle := \langle H \rangle \wedge \langle n^+ \rangle$

$\langle A \rangle_t, \langle Q \rangle_t := \langle n^+ \rangle_t$

Bemerkung: Die Tabelle besteht nur aus den Worten, für die gilt: $\langle n + k \langle B \rangle \rangle_t = \langle n \rangle_t$

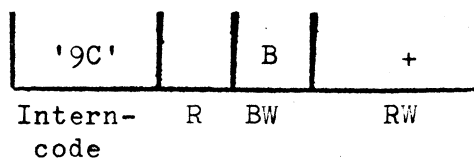
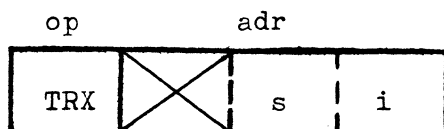
Bei mehreren Minima gilt:

n^+ = Adresse des in W1. zuerst gefundenen Minimums

Ausführungszeit: $(0,85 p + 0,20) \mu s$
wobei p = Anzahl der Worte der Tabelle

TRANSPORT AUS RECHENWERK NACH INDEXZELLE

TRX



Spezifikation: $s = s_1 [N]$
 $s_1 = (A|Q|D|H)$

Negation
Register

Wirkung:

Fall A: s enthält N nicht

W1. $\langle i \rangle := \langle s_1 \rangle_{25,48}$
 $\langle B \rangle := \langle s_1 \rangle_{25,48}$

Fall B: s enthält N

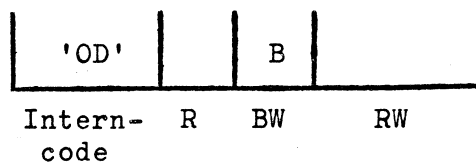
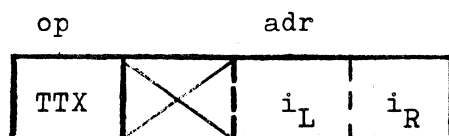
W1. $\langle i \rangle := \overline{\langle s_1 \rangle}_{25,48}$
 $\langle B \rangle := \overline{\langle s_1 \rangle}_{25,48}$

Bemerkung: Falls s_1 leer: Wirkung wie oben, jedoch
+0 anstelle von $\langle s_1 \rangle_{25,48}$
-0 anstelle von $\overline{\langle s_1 \rangle}_{25,48}$

Ausführungszeit: 0,30 μ s

TTX

TAUSCH-TRANSPORT IN INDEXZELLEN

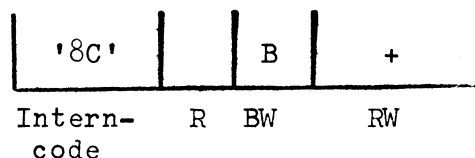
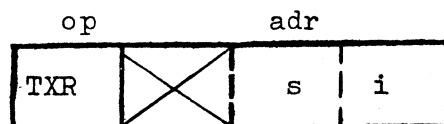


Wirkung: W1. $\langle B \rangle := \langle i_L \rangle$
 W2. $\langle i_R \rangle ::= \langle i_L \rangle$

Ausführungszeit: 0,55 μ s

TXR

TRANSPORT AUS INDEXZELLE NACH RECHENWERK



Spezifikation: $s = (A \vee Q \vee D \vee H) [N]$

A, Q, H, D : Register
N: Negation

Wirkung: W1. Die angegebenen Register
bekommen Tk 1

Fall A: s enthält N nicht

W2. Inhalte der angegebenen
Register :v= $\langle i \rangle$

W3. $\langle B \rangle := \langle i \rangle$

Fall B: s enthält N

W2. Inhalte der angegebenen
Register :v= $\overline{\langle i \rangle}$

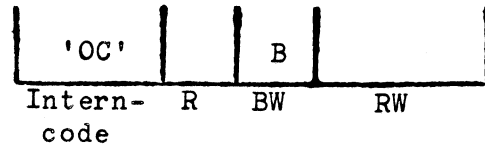
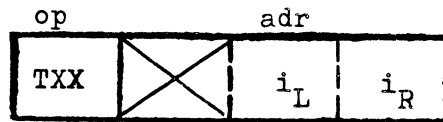
W3. $\langle B \rangle := \overline{\langle i \rangle}$

Bemerkung: Falls $s = [N]$, entfallen W1 und W2

Ausführungszeit: 0,25 μ s

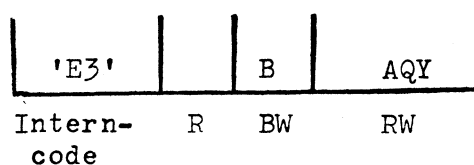
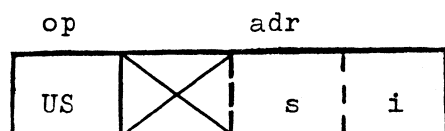
TXX

Transport aus INDEXzelle nach INDEXzelle



Wirkung: W1. $\langle i_L \rangle := \langle i_R \rangle$
 $\langle B \rangle := \langle i_R \rangle$

Ausführungszeit: 0,20 μ s

UMSCHLÜSSELN

Spezifikation: $s = (6|8|C) (E|G)$

Wirkung: W1. $\langle B \rangle := \langle i \rangle + \text{mod}2$

W2. Adresse der Umschlüsselungstabelle
(Halbwortadresse) $:= \langle B \rangle$

Fall A: $s = (6|8|C)E$

W3. Das in A rechtsbündig stehende 6-, 8- oder 12- Bit-Zeichen a wird umgeschlüsselt, d.h. durch das a-te Zeichen der Umschlüsselungstabelle ersetzt. A wird links mit Nullen aufgefüllt.

W4. $\langle A \rangle_t :=$ Typenkennung des Tabellenwortes, in dem das umgeschlüsselte Zeichen steht.

Fall B: $s = (6|8|C)G$

W3. Die 8, 6 oder 4 in A stehenden 6-, 8- oder 12- Bit-Zeichen werden umgeschlüsselt, d.h. durch die entsprechenden Zeichen der Umschlüsselungstabelle ersetzt.

6: 6-Bit-Zeichen
8: 8-Bit-Zeichen
C: 12-Bit-Zeichen
E: Einzelzeichen
G: gesamter $\langle A \rangle$

W4. $\langle A \rangle_t$:= Typenkennung des Tabellenwortes,
das das in A linksstehende Zeichen enthält.

W5. $\langle Q \rangle$:= $\langle A \rangle$
 $\langle Q \rangle_t$:= $\langle A \rangle_t$

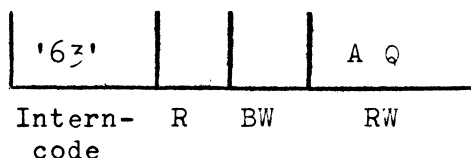
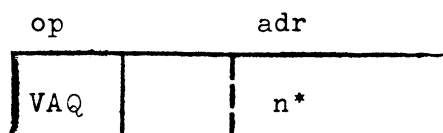
W6. $\langle Y \rangle$:= Anzahl der umgeschlüsselten
Zeichen.

Bemerkung: Die Umschlüsselungstabelle besteht maximal aus
64, 256 bzw. 4098 Viertelwörtern. Tk beliebig.

Ausführungszeit: Fall A: 1,60 μ s
Fall B: s = 6G: 8,80 μ s
s = 8G: 6,80 μ s
s = CG: 4,80 μ s

VAQ

VORZEICHENANGLEICH ZWISCHEN A UND Q



Wirkung: W1. $n^* := n^* + \text{mod}2$

Fall A: $\langle A \rangle_t = 1$ und $\langle A \rangle \neq \pm 0$
 und $\langle Q \rangle \neq \pm 0$ und $\langle Q \rangle_1 = \langle Q \rangle_2$

W2. $\langle Q \rangle_t := 1$
 $\langle A, Q \rangle := \langle A \rangle + 2^{-46} \langle Q \rangle$
 wobei
 $\langle Q \rangle_{1,2} := \langle A \rangle_1$

Fall B: $\langle A \rangle_t = 1$ und $\langle Q \rangle_1 = \langle Q \rangle_2$ und
 $\langle Q \rangle = \pm 0$

W2. $\langle Q \rangle_t := 1$
 $\langle Q \rangle_{1,48} := \langle A \rangle_1$

Fall C: $\langle A \rangle_t = 1$ und $\langle Q \rangle_1 = \langle Q \rangle_2$
und $\langle A \rangle = \pm 0$ und $\langle Q \rangle \neq \pm 0$

W2. $\langle Q \rangle_t := 1$
 $\langle A \rangle_{1,48} := \langle Q \rangle_1$

Der Adreßteil ist
bedeutungslos

in Q keine über-
oder untergelaufene
Zahl

Auch bei Vorzeichen-
verschiedenheit von
 $\langle A \rangle$ und $\langle Q \rangle$ bleibt
der Wert der doppelt
langen Zahl beim Vor-
zeichenangleich er-
halten, gemäß
 $a-q = (a-1)+(1-q)$ bzw.
 $-a+q = -(a-1)-(1-q)$

Fall D: $\langle A \rangle_t = 1$ und $\langle Q \rangle_1 \neq \langle Q \rangle_2$

über-oder unter-
gelaufene Zahl

W2. $\langle Q \rangle_t := 1$
 $\langle A \rangle := \text{undefiniert}$
 $\langle Q \rangle := \text{undefiniert}$

Fall E: $\langle A \rangle_t \neq 1$

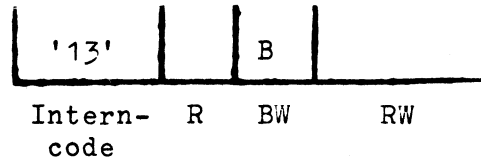
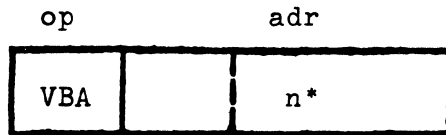
Keine Festkommazahl

W2. Tk-Alarm

Ausführungszeit: 0,50 μ s

VBA

VERMINDERE B UM ADRESSTEIL

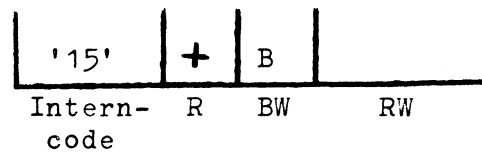
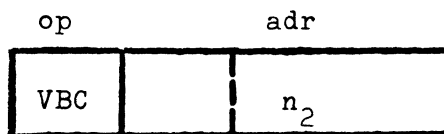


Wirkung: W1. $\langle B \rangle := \langle B \rangle - n^*$

Ausführungszeit: 0,425 μ s

VBC

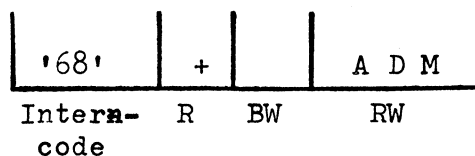
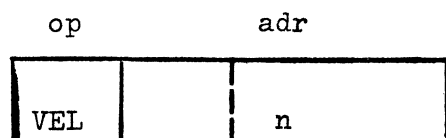
VERMINDERE B UM SPEICHER



Wirkung: W1. $\langle B \rangle := \langle B \rangle - \langle n_2 \rangle$

Ausführungszeit: 0,425 μ s

VEL



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle n \rangle_t \geq 2$

W2. $\langle A \rangle := \langle A \rangle \vee \langle n \rangle$
 $\langle A \rangle_t := \text{Max} (\langle A \rangle_t, \langle n \rangle_t)$

W3. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

Fall B: $\langle n \rangle_t \leq 1$

W2. $\langle A \rangle_1 := \langle A \rangle_1 \vee \langle n \rangle_2$
 $\langle A \rangle_k := \langle A \rangle_k \vee \langle n \rangle_k, \quad 2 \leq k \leq 48$
 $\langle A \rangle_t := \text{Max} (\langle A \rangle_t, \langle n \rangle_t)$
 $\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

W3. $\langle D \rangle := \langle n \rangle$
 $\langle D \rangle_t := \langle n \rangle_t$

Zahlwort

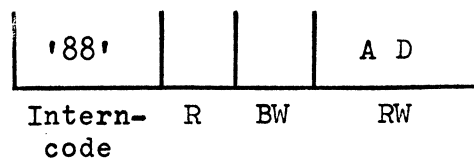
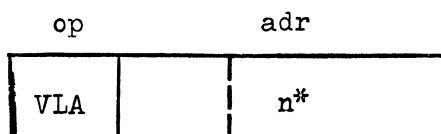
Markenstelle

Marke

Ausführungszeit: 0,25 μ s

VLA

VEL ADRESSTEIL



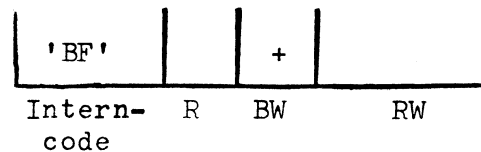
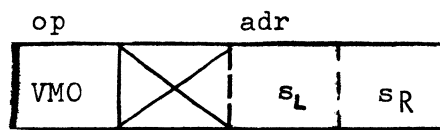
Wirkung:

W1. $n^* := n^* + \text{mod}2$

W2. $\langle A \rangle := \langle H \rangle \vee n^*$
 $\langle A \rangle_t := \langle H \rangle_t$

W3. $\langle D \rangle := n^*$
 $\langle D \rangle_t := 1$

Ausführungszeit: 0,40 μ s

VERÄNDERE MODUS

Voraussetzung: Der Rechner läuft im System-oder Spezialmodus.
(Sonst Makro)

Wirkung: W1. adr := adr + mod2

Adreßteil: adr = a₁ a₂ a₃ a₂₄

a₁ bis a₈ ohne Bedeutung

Spezifikationen nur intern angebbbar; alle 2¹⁶ Kombinationen sind erlaubt

W2. Falls a₂₄ = L : <F> :=

Falls a₂₃ = L :

Das Eingriffswerk meldet solche anstehenden Eingriffe, die infolge der Stellung der Bits in den Kanal-Zuordnungszellen bisher nicht durchkommen konnten, jetzt (d.h. nachdem eine Änderung der Bits der Kanal-Zuordnungs-Liste vorgenommen wurde) aber gemeldet werden können.

Falls a₂₂ = L : BEFE := L

Falls a₂₁ = L : := <BG>

Falls a₂₀ = L : <TV> := _{13,24}

Falls a₁₉ = L : <BW> := _{9,24}

Falls a₁₈ = L : <BT> :=

Falls a₁₇ = L : <BLZ2> := _{3,7}

<BLZ1> := _{8,12}

Die Seitenadreibregister werden ungültig gesetzt

Falls a₁₆ = L : <BL> := _{1,16}

Die Seitenadreibregister und die Indexregister werden ungültig gesetzt.

Die Indexregister werden jedoch nicht zurückgespeichert.

Falls a₁₅ = L : BEBO := a₁₀

Eingriffssperre

Uhr BG ablesen

Vergleichsregister TV

Wecker BW

Prüfregister BT

Δ ändern

Leitadresse

Abwicklerbit

GR/EP1
I/12/66
1267

29. Juli 1968

Falls $a_{15} = L, a_{13} = L$: BEBY:=L; BEBN:=0
 Falls $a_{15} = L, a_{13} = 0, a_{12} = L$: BEBY:=L; BEBN:=L
 Falls $a_{15} = L, a_{13} = 0, a_{12} = 0, a_{11} = L$: BEBY:=0; BEBN:=L
 Falls $a_{14} = L$: BEBA : = a_9

Systemmodus
 Spezialmodus
 Normal-oder
 Abwicklermodus
 Modus 24 oder 16

Ausführungszeit: 0,15 μs , falls $a_{21} = 0$
 1,30 μs , falls $a_{21} = L$

VPU

VERLASSE PROGRAMMUNTERBRECHUNG



Wirkung: W1. adr := adr + mod2

Adreßteil:

$$+0 \leq p \leq 255$$

$$s = m_1 m_2 m_3 m_4 m_5 m_6 m_7 m_8$$

nur intern angebbar

Voraussetzung: Der Rechner läuft im Spezial- oder Systemmodus, (sonst Makro).

Fall I: (Normalfall) bei der Ausführung des VPU tritt kein Speicherschutz-Alarm auf

W2. Alle Indexregister, für die das Rückspeicherungsbit gesetzt ist, werden zurückgespeichert.
Die Seitenadreib- und Indexregister werden ungültig gesetzt.

W3. Falls $m_1 = L$ ist:
 $\langle BL \rangle := \langle B \rangle_{1,16}$

Leitadreibregister neu laden

W4. Es wird eine Abholadresse I gebildet:
Falls $m_2 = L$: $I := \langle BL \rangle \cdot 2^8 + p$
Falls $m_2 = 0$: $I := p$

W5. $\langle B \rangle, \text{adr} := \langle I + 2 \rangle$
 $\langle F \rangle, \text{Steuerbits} := \langle I + 4 \rangle$
(Die Stelle 43 wird dabei nicht in das Flip-Flop BEFE (Eingriffssperre) übernommen).
Falls $m_3 = 0$ (Rückkehr nach Eingriff oder Alarm):

$$\langle S \rangle := \langle I \rangle$$

$$\text{Code, Steuerbits, } \langle BH \rangle := \langle I + 6 \rangle$$

I ₂	
$\langle S \rangle$	
$\langle B \rangle$	adr
$\langle F \rangle$	Steuerbits
Code, Steuerbits	$\langle BH \rangle$

- W6. Falls $m_4 = 0$ ist, entfällt W6
 Falls $m_4 = L$ ist, wird die Index-
 basis neu gesetzt.

Fall A: neuer Modus ist Systemmodus:

$$\langle XB \rangle := \langle 4_2 \rangle$$

Fall B: neuer Modus ist nicht

Systemmodus:

$$\text{falls } \langle \langle BL \rangle \cdot 2^8 + 4 \rangle_t = 3 \text{ (oder 1):}$$

$$\langle XB \rangle := \langle (\langle BL \rangle \cdot 2^8 + 4)_2 \rangle$$

Die Berechnung der absoluten Index-
 basis geschieht mit dem $\Delta 1$ und $\Delta 2$
 des Operators.

Indexbasis des
 Operators

$$\text{Falls } \langle \langle BL \rangle \cdot 2^8 + 4 \rangle_t = 2 \text{ (oder 0):}$$

$$\langle XB \rangle := \langle (\langle BL \rangle \cdot 2^8 + 4)_2 \rangle$$

Die Berechnung der absoluten Index-
 basis geschieht mit dem $\Delta 1$ und $\Delta 2$
 des Abwicklers.

Indexbasis des
 Abwicklers

- W7. Falls im neuen Modus Abwickler-
 Flip-Flop nicht gesetzt:

$$\langle LZ1 \rangle := \langle (\langle BL \rangle \cdot 2^8 + 64)_2 \rangle_{8,12}$$

$$\langle LZ2 \rangle := \langle (\langle BL \rangle \cdot 2^8 + 64)_2 \rangle_{3,7}$$

Δ des Operators
 werden geladen

- Falls im neuen Modus Abwickler-
 Flip-Flop gesetzt:

$$\langle LZ1 \rangle := \langle (\langle BL \rangle \cdot 2^8 + 65)_2 \rangle_{8,12}$$

$$\langle LZ2 \rangle := \langle (\langle BL \rangle \cdot 2^8 + 65)_2 \rangle_{3,7}$$

Δ des Abwicklers
 werden geladen

- W8. Falls $m_5 = L$:

Die Sperren werden gelockert wie bei VSS

- W9. Falls $m_3 = L$ (Rückkehr nach SSR oder
 MAKRO):

Der VPU wirkt wie ein Sprungbefehl.
 Es wird auf den Befehl gesprungen,
 der über den einzustellenden Befehls-
 zählerstand erreicht wird.

Falls $m_3 = 0$ (Rückkehr nach Eingriff
 oder Alarm):

Fall A: Nach Speicherschutz-Alarm in der Ausführungsphase eines VPU, BCI, ZI, BL oder PDP: Die Rückkehr erfolgt am Ende der Abrufphase des betreffenden VPU, BCI, ZI, BL oder PDP, der betreffende Befehl wird wiederholt.

Fall B: Übrige Fälle:

Der VPU kehrt an die Unterbrechungsstelle zurück, falls der abgelegte Befehlswerksregister-Block unverändert übernommen wird mit Ausnahme von abgehandelten und gelöschten Alarmen. Alle darüberhinausgehenden Änderungen in den abgelegten Befehlswerks-Register führen zu einer der vorgenommenen Änderung entsprechenden Fortsetzung. Einzelheiten sind dann nur unter Zuhilfenahme der angesprochenen Mikroprogramme zu klären.

W10. STØP-Alarm:

Bei Rückkehr nach Stop-Alarm (BEEF) aufgrund von BEBE mit gesetzten Stop-Alarm-Steuerbits BEBF oder BEBE führt der neue Stop-Alarm erst nach Abruf- und Ausführungsphase des am Anfang seiner Abrufphase unterbrochenen Befehls zum Einsprung in das Alarm-Mikroprogramm.

Bei Rückkehr nach Eingriff oder Alarm in allen übrigen Fällen mit gesetzten Stop-Alarm-Steuerbits BEBF oder BEBE

führt der hierdurch erzeugte Stop-Alarm erst nach Ausführung des am Ende seiner Abrufphase unterbrochenen Befehls zum Aussprung in das Alarm-Mikroprogramm.

Fall II: (Alarmfall) Bei der Ausführung des VPU tritt Speicherschutz-Alarm auf (bei Bildung der absoluten Indexbasis oder bei Auslesen eines rechten Befehlshalbwortes - falls überhaupt erneut ausgelesen wird).

neu ausgelesen
wird nur, wenn
BF ungerade
 \wedge BEMP=0
 \wedge $m_3=0$ ist

Wirkung: Folgende Register-und Flip-Flop-Inhalte werden hinterlassen:
(Hinweise auf die Wirkungen W1, W2... beziehen sich auf Fall I) .

Register Flip-Flop	Bei Speicherschutzalarm bei der Berechnung der Index- basis	Bei Speicherschutzalarm beim Auslesen des rech- ten Befehlshalbwortes
BA, BC, BB, BF	wie vor Beginn der Ausführungsphase des VPU	
BL	wie nach Ausführung von W3	
BLZ1, BLZ2	$\Delta 1$ und $\Delta 2$, mit denen die Indexbasis berechnet werden sollte	wie nach Ausführung von W3
Seitenadr.- Register	wie nach Ausführung von W2	
Index- register	wie nach Ausführung von W2	
BXB, BXBZ	undefiniert	wie nach Ausführung von W6
RS	<p style="text-align: center;">TK = 3</p> <p>RS_{linkes HW} : alarmerzeugende Adresse</p> <p>= relative Indexbasis</p> <p>RS_{rechtes HW} : Adressteil des VPU</p>	
Steuerbits	<p>BEBO: = L Indexbasis wurde mit Δ's des Ab- wicklers berech- net</p> <p>= 0 Indexbasis wurde mit Δ's des Ope- rators berechnet</p> <p>BEIC = L BEEC = L BEMB = BEMU = 0</p> <p>alle übrigen Flip-Flop blei- ben unverändert.</p>	<p>BEBO: entspr. $\langle I+4 \rangle_{46}$</p> <p>BEIC = 0 BEEC = L BEMB = BEMU = BEML = 0</p> <p>alle übrigen Flip-Flop bleiben unverändert.</p>

VERÄNDERE SPERRE UND SPRINGE

op	adr
VSS	n ₂

'37'		+	
Intern- code	R	BW	RW

Voraussetzung: Der Rechner läuft im System- oder Spezialmodus.

(Sonst Makro.)

Wirkung:

W1. n₂ := n₂ + mod2

W2: ⟨F⟩ := n₂

Fall A: BEFB = L

W3. BEFB := 0

Alarmsperre 2 wird aufgehoben.

Fall B: BEFB = 0 und BEFA = L

W3. BEFA := 0

Alarmsperre 1 wird aufgehoben

Fall C: BEFB = 0 und BEFA = 0 und BEFE = L

W3. BEFE := 0

Eingriffssperre wird aufgehoben.

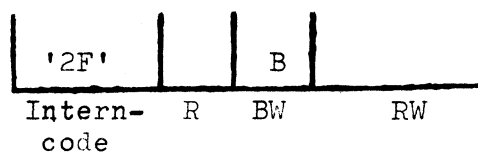
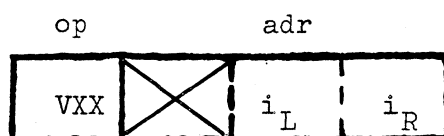
Fall D: BEFB = BEFA = BEFE = 0

W3. entfällt.

Ausführungszeit: 0,05 µs

VXX

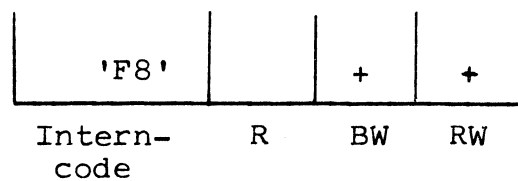
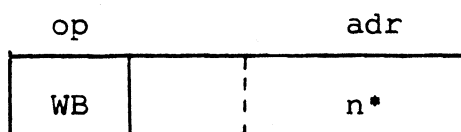
VERMINDERE INDEXZELLE UM INDEXZELLE



Wirkung: $W1. \langle B \rangle, \langle i_R \rangle := \langle i_R \rangle - \langle i_L \rangle$

Ausführungszeit: 0,725 μ s

WARTEBEFEHL



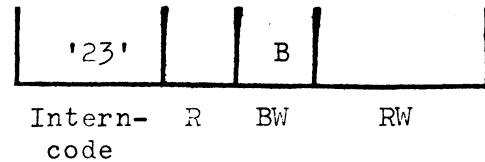
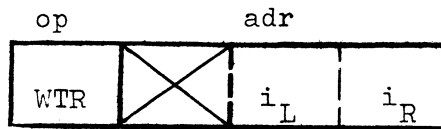
Wirkung: keine Wirkung außer Warten abhängig vom Adreßteil

Bemerkung: Der Adreßteil wird alle 10 μ s (Uhrsignal) um 1 erniedrigt, bis auf +0.

Ausführungszeit: $(10n^* - 5) \mu$ s für $1 \leq n^* \leq 2^{24} - 1$
 0,15 μ s für $n^* = +0$

WTR

WORTGRUPPENTRANSPORT RÜCKWÄRTS



Eingangsinformation

- $\langle B \rangle$ = Länge der Wortgruppe in Ganzworten = $p \gg 0$
 $\langle i_R \rangle$ = höchste Adresse des Quellbereiches
 $\langle i_L \rangle$ = höchste Adresse des Zielbereiches

Wirkung:

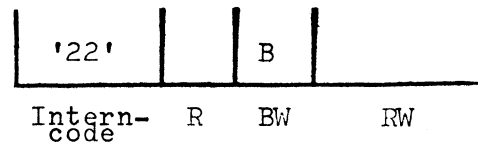
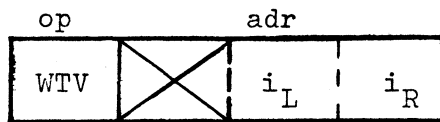
- W1. $\langle \langle i_L \rangle - 2k \rangle := \langle \langle i_R \rangle - 2k \rangle$
 wortweise zeitlich nacheinander
 für $k = 0, 1, 2, \dots, \langle B \rangle - 1$
 W2. $\langle B \rangle := \langle i_L \rangle - 2 (\langle B \rangle - 1)$

Bemerkung: Falls $p = +0$, entfallen W1. und W2.

Ausführungszeit: $(1,05 p + 0,50) \mu s$

WTV

WORTGRUPPENTRANSPORT VORWÄRTS



Eingangsinformation:

- $\langle B \rangle$ = Länge der Wortgruppe in Ganzworten = $p \gg 0$
 $\langle i_R \rangle$ = niederste Adresse des Quellbereiches
 $\langle i_L \rangle$ = niederste Adresse des Zielbereiches

Wirkung:

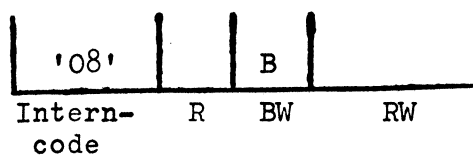
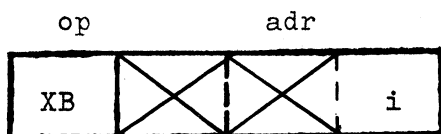
- W1. $\langle \langle i_L \rangle + 2k \rangle := \langle \langle i_R \rangle + 2k \rangle$
 wortweise zeitlich nacheinander
 für $k = 0, 1, 2, \dots, \langle B \rangle - 1$
 W2. $\langle B \rangle := \langle i_L \rangle + 2 (\langle B \rangle - 1)$

Bemerkung: Falls $p = +0$, entfallen W1. und W2.

Ausführungszeit: $(1,05 p + 0,50) \mu s$

XB

INDEX : BRINGE



Wirkung: W1. $\langle B \rangle := \langle i \rangle$

Bemerkung: Interne Darstellung von XB :

Die Linksadresse muß aus folgenden

Bits bestehen: OLOX XXXX (X beliebig).

Andernfalls falsche Befehlsausführung.

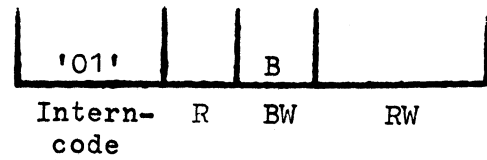
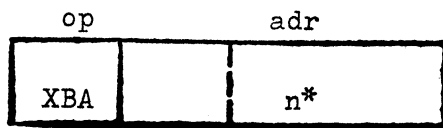
Ausführungszeit: 0,05 μ s

GR/EP1
12/66
201267

29. Juli 1968

XBA

INDEX : BRINGE ADRESSTEIL

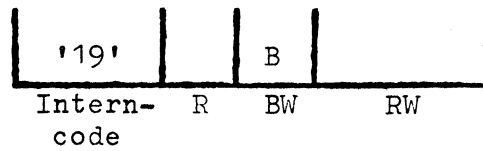
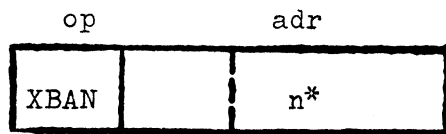


Wirkung: W1. $\langle B \rangle := n^*$

Ausführungszeit: 0,05 μ s

XBAN

INDEX : BRINGE ADRESSTEIL NEGATIV

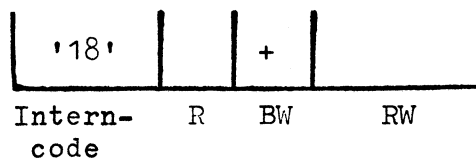
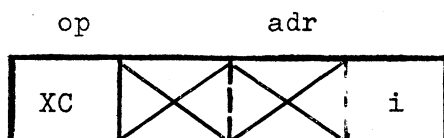


Wirkung: W1. $\langle B \rangle := -n^*$

Ausführungszeit: 0,10 μ s

XC

INDEX : SPEICHERE



Wirkung W1. $\langle i \rangle := \langle B \rangle$

Bemerkung: Interne Darstellung von XC :

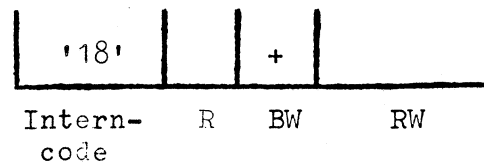
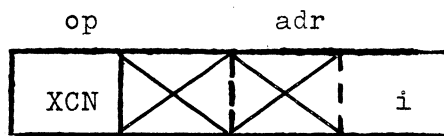
Das erste Bit der Linksadresse ist 0
(zur Unterscheidung von XCN).

Die restlichen 7 Bit der Linksadresse
sind ohne Bedeutung.

Ausführungszeit: 0,15 μ s

XCN

INDEX : SPEICHERE NEGATIV



Wirkung: W1. $\langle i \rangle := -\langle B \rangle$

Bemerkung: Interne Darstellung von XCN :

Das erste Bit der Linksadresse ist L
(zur Unterscheidung von XC).

Die restlichen 7 Bit der Linksadresse
sind ohne Bedeutung.

Ausführungszeit: 0,25 μ s

SETZE INDEXBASIS



Wirkung:

Fall A. Der Rechner läuft im Systemmodus

- W1. $\langle 4_2 \rangle := \langle n_2 \rangle$
- W2. $\langle 4 \rangle_t := 3$ falls Abwickler-
flip-flop = 0
- $\langle 4 \rangle_t := 2$ falls Abwickler-
flip-flop = L
- W3. $\langle XB \rangle := \langle 4_2 \rangle$

Reservieren der Indexbasis

Setzen der Indexbasis

Fall B. Der Rechner läuft nicht im Systemmodus

- W1. $\langle (\langle BL \rangle \cdot 2^8 + 4)_2 \rangle := \langle n_2 \rangle$
- W2. $\langle \langle BL \rangle \cdot 2^8 + 4 \rangle_t := 3$ falls
Abwicklerflipflop = 0
- $\langle \langle BL \rangle \cdot 2^8 + 4 \rangle_t := 2$ falls
Abwicklerflipflop = L
- W3. $\langle XB \rangle := \langle (\langle BL \rangle \cdot 2^8 + 4)_2 \rangle$

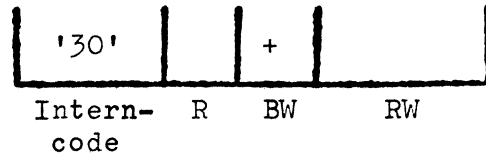
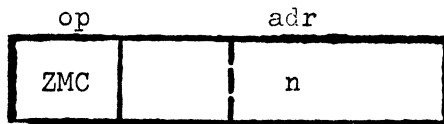
Reservieren der
rel. Indexbasis

Setzen der Indexbasis

Ausführungszeit: $(0,85 p + 2,90 \mu s)$

wobei p = Anzahl der zurückzuspeichernden Indexregister

SETZE MARKE IM SPEICHER



Wirkung: W1. $n := n + \text{mod}2$

Fall A: $\langle n \rangle_t \leq 1$

W2. $\langle n \rangle_1 := L$

Fall B: $\langle n \rangle_t \geq 2$

W2. Tk-Alarm

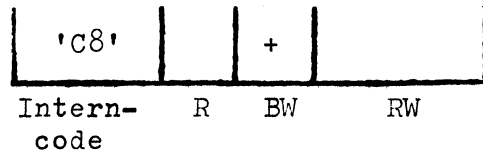
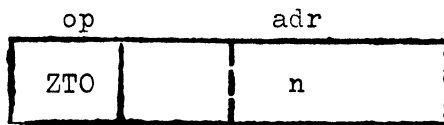
Zahlwort

Markenstelle

Ausführungszeit: Fall A = 0,35 μ s
 Fall B = 0,05 μ s

ZT0

SETZE TYPENKENNUNG 0



Wirkung: W1. $n := n + \text{mod}2$

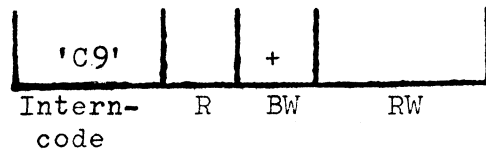
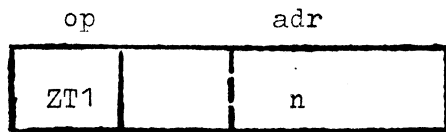
W2. $\langle n \rangle_t := 0$

Bemerkung: Alle 48 Informationsbits bleiben erhalten.

Ausführungszeit: 0,65 μs

ZT1

SETZE TYPENKENNUNG 1



Wirkung: W1. $n := n + \text{mod}2$

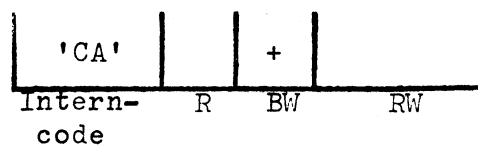
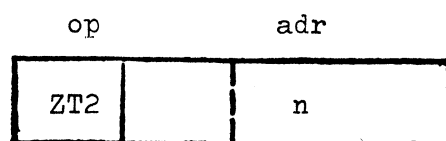
W2. $\langle n \rangle_t := 1$

Bemerkung: Alle 48 Informationsbits bleiben erhalten.

Ausführungszeit: 0,65 μ s

ZT2

SETZE TYPENKENNUNG 2



Wirkung: W1. $n := n + \text{mod}2$

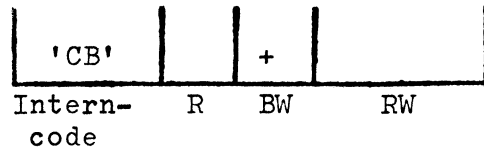
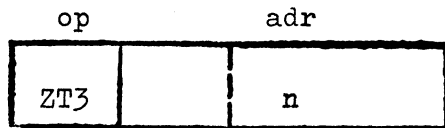
W2. $\langle n \rangle_t := 2$

Bemerkung: Alle 48 Informationsbits bleiben erhalten.

Ausführungszeit: 0,65 μ s

ZT3

SETZE TYPENKENNUNG 3



Wirkung: W1. $n := n + \text{mod}2$

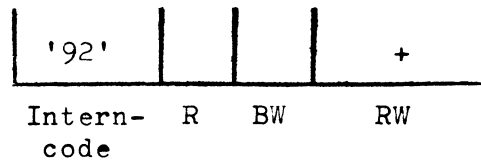
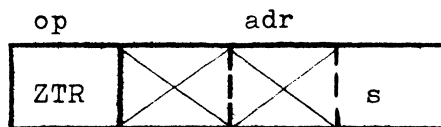
W2. $\langle n \rangle_t := 3$

Bemerkung: Alle 48 Informationsbits bleiben erhalten.

Ausführungszeit: 0,65 μs

SETZE TYPENKENNUNG IM REGISTER

ZTR



Wirkung: W1. $s := \text{adr} + \text{mod}2$

Adreßteil: $s = [s_1 s_2] [M]$

$s_1 = 0|1|2|3$

$s_2 = A|Q|D|H$

W2. Falls $s = s_1 s_2 [M]$:

$\langle s_2 \rangle_t := s_1$

W3. Falls $s = [s_1 s_2] M$:

$\langle M \rangle := L$

M: Markenregister

Typenkennung

Register

Das angegebene Register bekommt die angegebene Tk

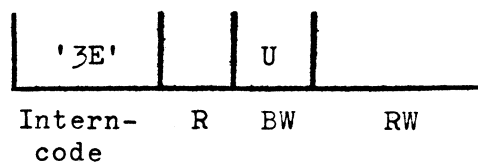
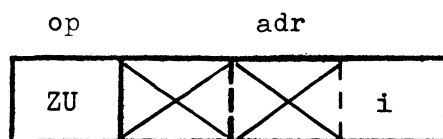
Markenregister wird gesetzt

Bemerkung: Falls $s = s_1 [M]$, ist s_1 ohne Bedeutung

Ausführungszeit: 0,25 μ s

ZU

SETZE U NTERPROGRAMMORDNUNGSSZÄHLER



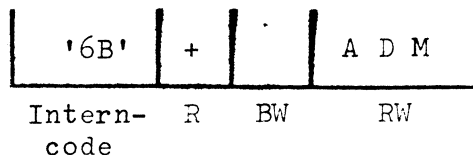
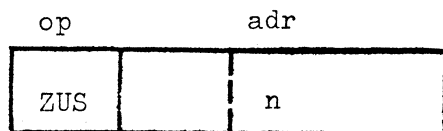
Wirkung: W1. $\text{adr} := \text{adr} + \text{mod}2$

W2. $\langle U \rangle := i$

Ausführungszeit: 0,05 μs

ZUS

SETZE ZUSAMMEN



Wirkung: W1. $n := n + \text{mod}2$

W2. $\langle A \rangle_t := \text{Max} (\langle A \rangle_t, \langle n \rangle_t)$

$\langle D \rangle := \langle n \rangle$

$\langle D \rangle_t := \langle n \rangle_t$

Fall A: $\langle n \rangle_t \geq 2$

W3. $\langle A \rangle_i := \langle n \rangle_i$ falls $\langle H \rangle_i = L$

$\langle A \rangle_i := \langle A \rangle_i$ falls $\langle H \rangle_i = 0$

für $1 \leq i \leq 48$

Fall B: $\langle n \rangle_t \leq 1$

W3. $\langle A \rangle_1 := \langle n \rangle_2$ falls $\langle H \rangle_1 = L$

$\langle A \rangle_i := \langle n \rangle_i$ falls $\langle H \rangle_i = L$

für $2 \leq i < 48$

$\langle A \rangle_i := \langle A \rangle_i$ falls $\langle H \rangle_i = 0$

für $1 \leq i < 48$

$\langle D \rangle_1 := \langle D \rangle_2$

$\langle M \rangle := \langle M \rangle \vee \langle n \rangle_m$

A AAAAAAAAAA

N NNNNNNNNNN

H

ZUS
A NNNAAAAANN

Zahlwort

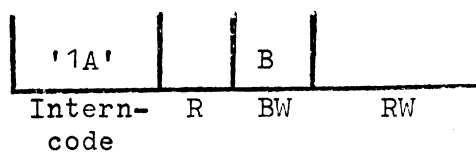
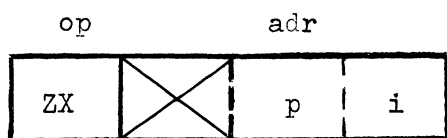
Markenstelle

Marke

Ausführungszeit: 0,25 μ s

ZX

SETZE INDEX



Wirkung: W1. $\langle i \rangle : \mathbf{v} = p$
 $\langle B \rangle : \mathbf{v} = p$
 $-127 \leq p \leq 127$

Ausführungszeit: 0,30 μ s

