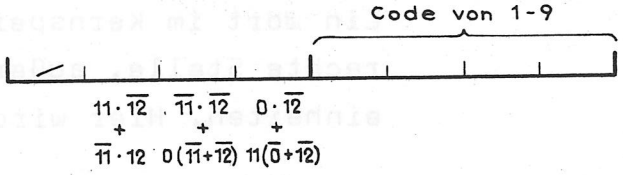
3.6 ISO-Code in internen Code (ISO)

Symbol	Kartencode	ISO	Symbol	Kartencode	ISO
Esp.		20	@	12-	40
!	0 - 1	21	A	12- 1	41
"	0 - 2	22	B	12- 2	42
#	0 - 3	23	C	12- 3	43
$	0 - 4	24	D	12- 4	44
%	0 - 5	25	E	12- 5	45
&	0 - 6	26	F	12- 6	46
'	0 - 7	27	G	12- 7	47
(	0 - 8	28	H	12- 8	48
)	0 - 9	29	I	12- 9	49
*	0 - 8 - 2	2A	J	12- 8-2	4A
,	0 - 8 - 4	2C	L	12- 8-4	4C
-	0 - 8 - 5	2D	M	12- 8-5	4D
.	0 - 8 - 6	2E	N	12- 8-6	4E
/	0 - 8 - 7	2F	O	12- 8-7	4F
Ø	0	30	P	11-	50
1	1	31	Q	11- 1	51
2	2	32	R	11- 2	52
3	3	33	S	11- 3	53
4	4	34	T	11- 4	54
5	5	35	U	11- 5	55
6	6	36	V	11- 6	56
7	7	37	W	11- 7	57
8	8	38	X	11- 8	58
9	9	39	Y	11- 9	59
:	8 - 2	3A	Z	11- 8-2	5A
;	8 - 3	3B	[	11- 8-3	5B
<	8 - 4	3C	CS2	11- 8-4	5C
=	8 - 5	3D	]	11- 8-5	5D
>	8 - 6	3E	^	11- 8-6	5E
?	8 - 7	3F		11- 8-7	5F



4. Allgemeine Befehle

4.1 Definition

Dies sind die Elementarbefehle, die der Programmierer in die Maschine eingeben kann.

4.2 Zusammensetzung der Befehle

Operationstyp (TO)

Er ist auf 2 verschiedene Arten dargestellt:

- TO symbolisch oder mnemonisch. Er ist dargestellt in einem Code, der für die Maschine nicht lesbar und für den Befehl zu lang ist.
- TO intern. Er besetzt ein Byte im Zentralspeicher und besteht aus 2 Zeichen, hexadezimal codiert.

Konstante

Wenn in dem Befehl eine Konstante vorkommt, so besetzt diese das zweite Byte (außer C).
In der allgemeinen Darstellungsart eines Befehls steht das Zeichen:

K -als Symbol für ein Zeichen außer F1, F2, F3 und FF.
N -für 2 Ziffern im Format 2 (00 - 99), welche die Zeichenanzahl bei einer Übertragung, einem Vergleich und einer Versetzung angeben.
C -für den Anschlußcode einer Randeinheit.

Achtung: Dieses Zeichen besetzt das dritte Byte oder wenn mehrere Randeinheiten angeschlossen werden, eventuell das fünfte oder siebte Byte.

4.3 Adressierung

Ein Wort im Kernspeicher ist adressiert durch seine rechte Stelle, außer bei einem Befehl für die Randeinheiten. Hier wird die rechte Seite des Wortes

durch eine Trennmarke (F4) begrenzt. Die Adressierung ist je nach Befehl direkt oder durch eine Basis.

Direkte Adresse: Sie kann 2 Formen haben:

.XX.XX. die reelle Adresse ist dargestellt in 2 Bytes durch 4 Ziffern in Format 2.

.R. die Registernummer (00 - 99) steht in einem Byte, in Format 2.

Basisadresse mit Angabe des Basisregisters

Sie ist dargestellt durch 4 Ziffern in Format 2. Plaziert in 2 Bytes.

.BX.XX.

.B . . im 4. Halbbyte ist die Registernummer (0 - 9) angegeben, in der sich die Basis befindet. Diese Basis besteht aus 4 Ziffern, in den 4 rechten Halbbytes des Registers.

. X.XX. Diese 3 Ziffern geben die Versetzung der reellen zur Basisadresse an. Man findet die Realadresse durch Addition dieser Ziffer (000 - 999) auf die Realadresse.

Beispiel: Das Register 8 enthält 1525.

In .BX.XX. steht 8015, dann ist die Realadresse 1525 + 015 = 1540.

Im Operationstyp enthaltene Adressen.

Dieser Adresstyp wird nur bei Übertragungen, in Verbindung mit den Ein- Ausgabezonen, verwendet. Entsprechend dem Operationstyp nimmt man den Inhalt eines Spezialregisters (4 rechte Halbbytes) als Basisadresse.

Diese Spezialregister sind:

Register 06: bestimmt die Eingangszone A, die für den Kartenleser reserviert ist.
Die Versetzung wird im Befehl unter D_6 angegeben.

Register 07: bestimmt die Eingangszone B.
Die Versetzung wird im Befehl unter D_7 angegeben.

Register 08: bestimmt die Ausgangszone C, die für den Drucker (MB1) reserviert ist.
Die Versetzung wird im Befehl unter D_8 angegeben.

Register 09: bestimmt die Ausgangszone D, die für den Stanzer reserviert ist.
Die Versetzung wird im Befehl unter D_9 angegeben.

Wenn die Versetzung ≤ 99 ist, wird sie in einem Byte in Format 2, gespeichert. Z.B. | 8 | 7 |

Ist die Versetzung >99, (max. 999) wird sie durch 3 Ziffern, in Format 2, in 2 Bytes dargestellt.
Im ersten Halbbyte dieser beiden Bytes steht ein A.
Z.B. | A | 1 | 7 | 5 |

Dieses A im 1. Halbbyte zeigt der Maschine, daß die Versetzung in 2 Bytes gespeichert ist.

4.4 Länge der Befehle

Die Länge der Befehle ist variabel, besteht aber immer aus ganzen Bytes. (keine Halbbytes)
Der kürzeste Befehl besetzt 1 Byte und der längste Befehl besetzt 9 Bytes.

4.5 Anzahl der Befehle

Es gibt 67 Befehle, die entsprechend ihrem Zweck, in folgende Gruppen unterteilt sind:

- Übertragungsbefehle
- Rechenbefehle
- Befehle für Randeinheiten
- Sprungbefehle
- Logische Operationen
- Versetzungsbefehle
- Spezialbefehle für Programmerstellung
- Befehle für Multiprogrammierung.

5. Darstellung der Befehle

5.1 Übertragung

5.1.01 MRR

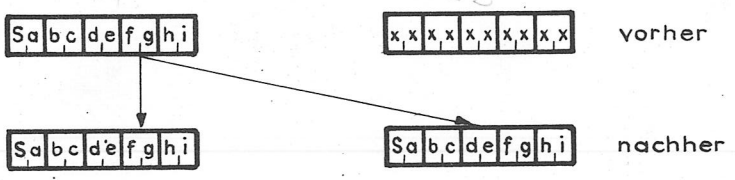
TO	Reg. Nr.	Reg. Nr.
E1	R1	R2
	00 - 99	00 - 99
1. Byte	2. Byte	3. Byte

Dauer: 1.6 ms

Übertragung des Inhalts eines Einfachregisters R1 in ein Einfachregister R2.

Die Übertragung erfolgt ohne Wechsel des Formats. Der Inhalt von R1 bleibt erhalten.

Beispiel



5.1.02 MDRR

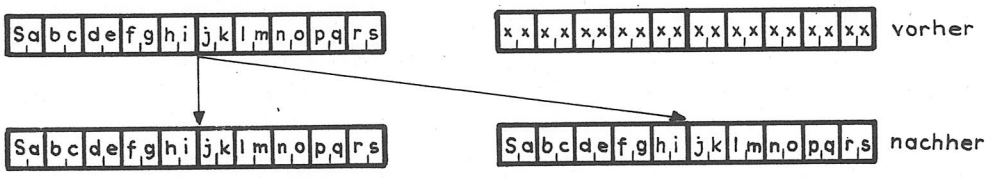
E2	R1	R2
	01 - 99	01 - 99
1. Byte	2. Byte	3. Byte

Dauer: 1.9 ms

Übertragung des Inhaltes eines Doppelregisters R1 in ein Doppelregister R2.

Die Übertragung erfolgt ohne Wechsel des Formats.

Beispiel



5.1.03 MSRDR

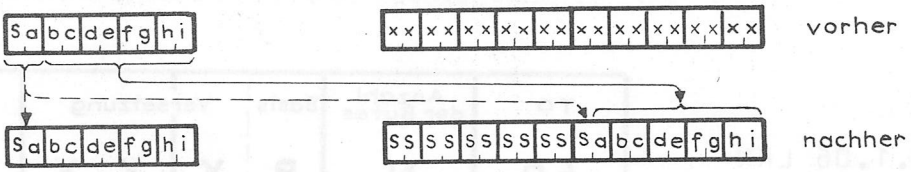
TO	Reg. Nr.	Reg. Nr.
E3	R1	R2
	00-99	00-99
1. Byte	2. Byte	3. Byte

Dauer 1.8 ms

Übertragung des Inhaltes von Einfachregister R1 (in Format 2) in das Doppelregister R2. Das äußerste linke Halbbyte von R1 wird 10 x in R2 - 1 dupliziert.

Anmerkung: Man kann R1 = R2 oder R1 = R2 - 1 haben.

Beispiel



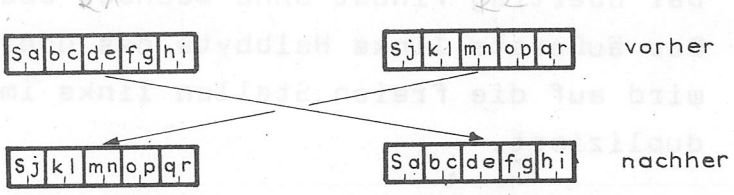
5.1.04 EXR

TO	Reg. Nr.	Reg. Nr.
E5	R1	R2
	00-99	00-99
1. Byte	2. Byte	3. Byte

Dauer: 1.7 ms

Austausch des Inhaltes zweier Einfachregister. Das Format wird nicht gewechselt.

Beispiel

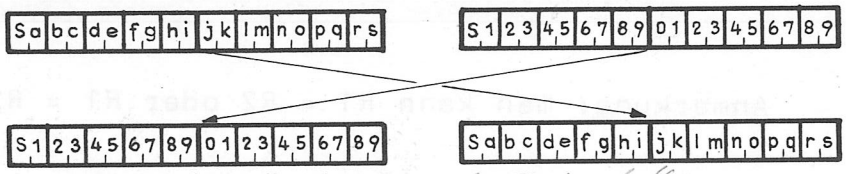


5.1.05 EXDR

TO	Reg. Nr.	Reg. Nr.
E6	R1 01 - 99	R2 01 - 99
1. Byte	2. Byte	3. Byte

Dauer 2,2 ms

Austausch des Inhaltes zweier Doppelregister.
R1 - R2 - 1 ist möglich. Es findet dann ein Ringtausch dreier Einfachregister statt.



5.1.06 LRN

TO	Anzahl der Bytes	Basis	Versetzung		Reg. Nr.
6.0	N 00 - 10	B 0-9	X 000	X X - 999	R 00 - 99
1. Byte	2. Byte	3. Byte		4. Byte	5. Byte

Dauer
1,4 +
0,08 Nms
N = pro Byte

Übertrag eines Wortes von N (Anzahl) Bytes von der Basisadresse Bx.xx in das Register R.

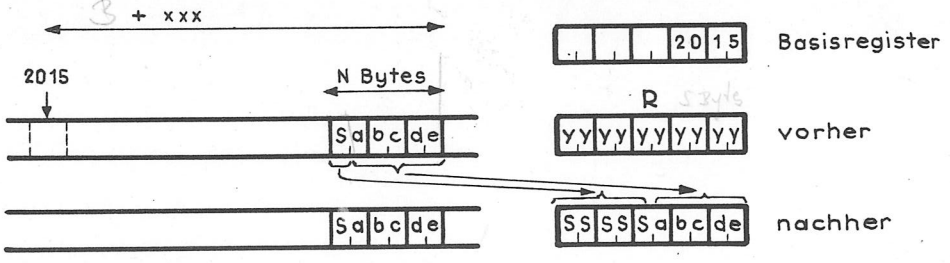
R = Einfachregister, wenn N ≤ 5

R = Doppelregister, wenn N > 5

N = 0 wird wie 10 behandelt.

Wenn N > 10, werden nur die Einerstellen des angegebenen Wertes beachtet.

Der Übertrag findet ohne Wechsel des Formats statt. Das äußerste linke Halbbyte des übertragenden Wortes wird auf die freien Stellen links im Register dupliziert.



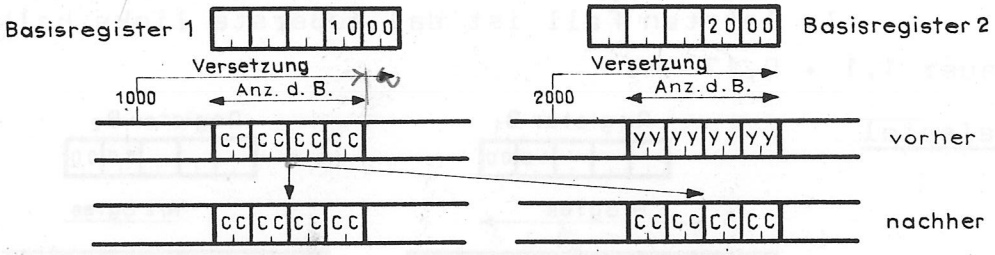
5.1.07 MVC

Move Caracters

TO	Anzahl der Bytes	1. Byte abgebender Basis	1. Byte abgebender Versetzung		1. Byte empfangender Basis	1. Byte empfangender Versetzung	
8.0	N	B_1	X	X X	B_2	X'	X' X'
	00 - 99	0 - 9	000	- 999	0 - 9	000	- 999
1. Byte	2. Byte	3. Byte		4. Byte	5. Byte		6. Byte

Übertrag eines Wortes von N Bytes von Basisadresse B1x.xx nach B2x.xx
Der Übertrag erfolgt ohne Wechsel des Formats.
Bezeichnet A die abgebende und E die empfangende Adresse, so muß E $\leq$ A oder E $>$ A-N sein.

Dauer: 1,9 + 0,05 N ms



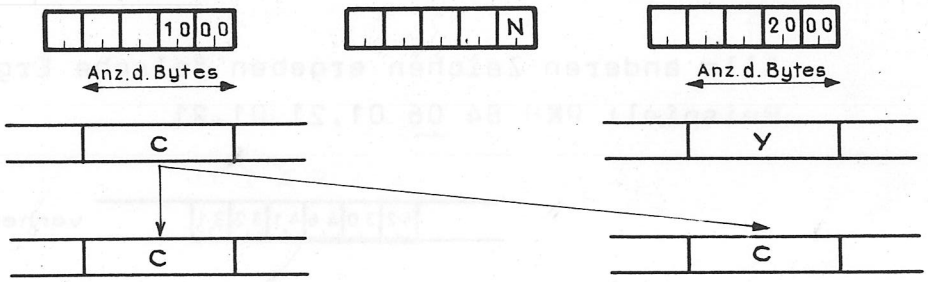
5.1.08 MVIC

Move Caracters
Count is indirect

8A	R	B_1 XX	XX	B_2 X'	X' X'
	00 - 99				
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

Der Befehl entspricht MVC. Die Länge des zu übertragenden Wortes befindet sich in Register R und ist $<$ 99.

Dauer 2,2 + 0,05 ms



5.1.09 PKH

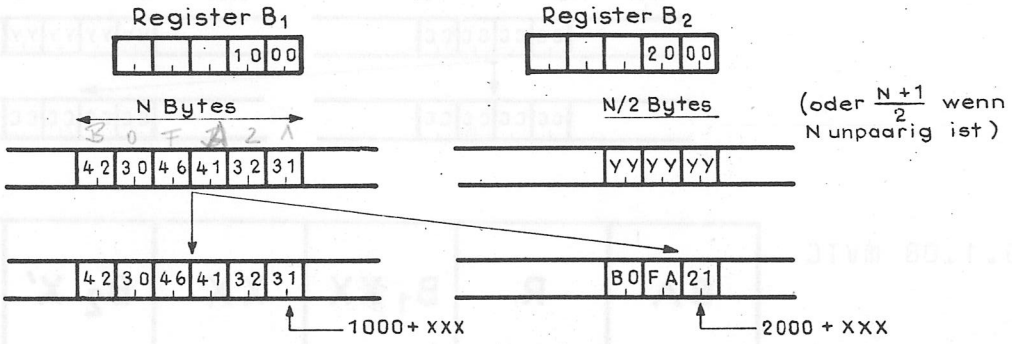
Pack Hexodecimal

84	Nr. des abgebenden Byte: N (Anzahl der Bytes) 00 - 99	1. Byte abgebender: Basis B_1 (0-9)	Versetzung X (000 - 999)	1. Byte empfangender: Basis B_2 (0-9)	Versetzung X' (000 - 999)
1.Byte	2.Byte	3.Byte	4.Byte	5.Byte	6.Byte

Übertrag mit Verdichtung eines Wortes mit der Länge von N Bytes in Format 1, welches durch die Basisadresse B1x xx bestimmt wird, nach B2x' x'x' . Das Resultat hat im Format 2 die Länge 1, die sich wie folgt errechnet:
$1 = \frac{N}{2}$ wenn N gerade ist. $1 = \frac{N+1}{2}$ wenn N ungerade ist.
Im letzten Fall ist das äußerste linke Halbbyte Null.

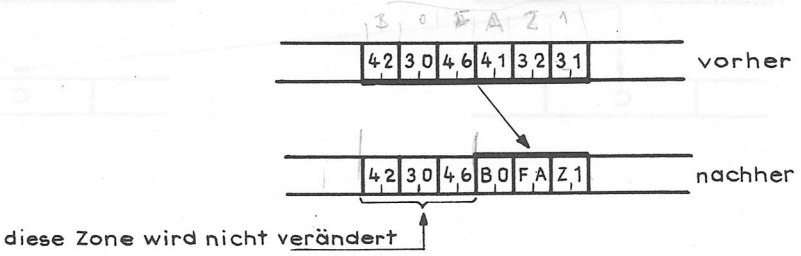
Dauer 1.1 + 0,17 Nms

Beispiel



Wenn B1x xx = B2x' x'x' ist, wird der linke Teil des Feldes durch den Übertrag nicht gelöscht.
Die Zeichen, die verdichtet, bzw. erweitert werden können, sind normalerweise:
Die Zahlen 30 - 39 in hexadezimal (1 - 9)
Die Buchstaben 41 - 46 in hexadezimal (A - F)
Der Spaltensprung 20 in hexadezimal

Alle anderen Zeichen ergeben falsche Ergebnisse.
Beispiel: PKH 84 06 01.21 01.21



5.1.10 UPH

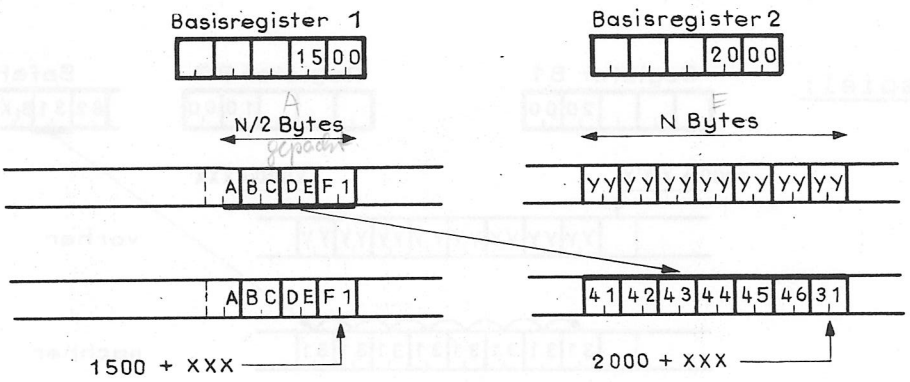
Un Pack Hexadecimal

TO	Nr. des abgebenden Byte	1. Byte abgebender Basis	1. Byte abgebender Versetzung		1. Byte empfangender Basis	1. Byte empfangender Versetzung	
85	N (anzahl der abgeb. Halb-byte)	B_1	X	XX	B_2	X'	X'X'
	00-99	0-9	000 - 999		0-9	000 - 999	
1.Byte	2.Byte	3.Byte	4.Byte		5.Byte	6.Byte	

Übertrag mit Erweitern eines Wortes, das sich im Format 2 in der Basisadresse B1x xx befindet, auf ein Feld, das mit der Basisadresse B2x' x'x' beginnt. Das Resultat ist in Format 1 nach dem Erweitern von N:2 Zeichen des abgebenden Feldes.

Dauer: 06 + 0,2 N ms

Beispiel



A sei die abgebende und E die empfangende Adresse.

Dann ist:

$E > A + \frac{N}{2}$ $\quad (A + \frac{N+1}{2}$ wenn N ungerade ist)

oder

$E < A + \frac{N}{2}$ $\quad (A - \frac{N+1}{2}$ wenn N gerade ist)

K = Kennzahl und Hexadezimal Binärwert oder 2 ungepackte hexadezimale Ziffern.

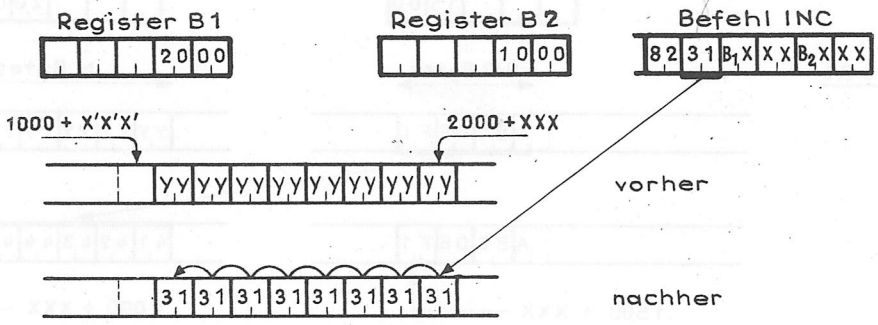
5.1.11 INC

TO		1. Byte der Zone			1. Byte nach der Zone		
		Basis	Versetzung		Basis	Versetzung	
82	K	B_1	X	XX	B_2	X'	X'X'
	00 - FE*	0-9	000	- 999	0-9	000	- 999
1.Byte	2.Byte	3.Byte		4.Byte	5.Byte		6.Byte

Eingabe des 2. Bytes des Befehls (K) von der Basis B1x xx (einschließlich) bis zur Basis B2 x'x'x' (ausschließlich).

Dauer: 0,8 + 0,2 N ms

Beispiel:



Anmerkung:

- B_1 xxx muß größer als B_2 x'x'x' sein, da sonst die logische Zone verändert wird.
- Wenn K = 00 entspricht dieser Befehl einem Löschbefehl.
- K darf nicht F1, F2, F3 oder FF sein, da dies die Programmeingabe stört.

5.2 Rechenbefehle

5.2.1 ADD

TO	Reg.-Nr.	Reg.-Nr.
A1	R_1 00 - 99	R_2 00 - 99
1.Byte	2.Byte	3.Byte

Dauer: 2,2 ms

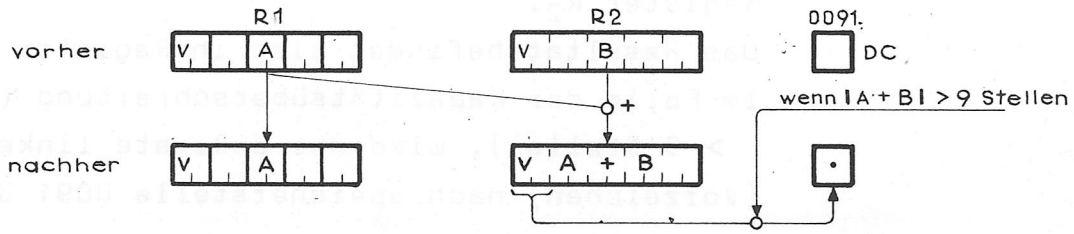
Addition Decimal single length

Algebraische Addition der beiden Register R_1 und R_2.
Das Ergebnis steht im Register R_2.

Im Falle der Kapazitätsüberschreitung (Resultat > 9 Ziffern), wird das äußerste linke Halbbyte (Vorzeichen) nach Speicherstelle 0091 übertragen.

Die Auswertung der Stelle 0091 kann der Bedienung, durch Leuchtanzeige oder Abdruck, den Fehler anzeigen.

Beispiel:



Sonderfall: Wenn $R_1 = R_2$ wird der Inhalt der Register verdoppelt.

5.2.2 ADDD

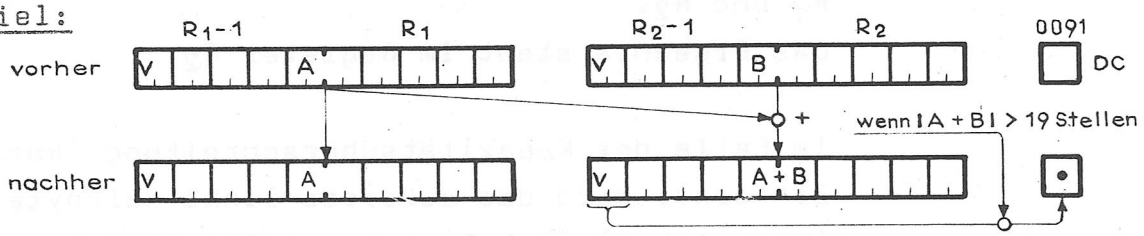
Addition Decimal Double length

TO	Reg.-Nr.	Reg.-Nr.
A2	R_1 01 - 99	R_2 01 - 99
1.Byte	2.Byte	3.Byte

Dauer: 2,9 ms

Dieser Befehl ist gleich der ADD, jedoch werden hierbei 2 Doppelregister verwendet.

Beispiel:



5.2.3 SBD

Subtract Decimal single length

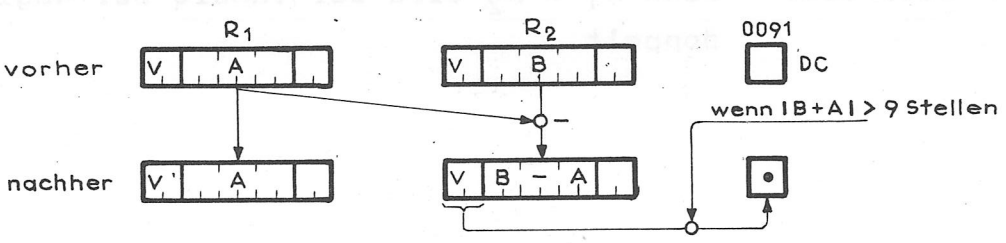
TO	Reg.-Nr.	Reg.-Nr.
B1	R_1 00 - 99	R_2 00 - 99
1.Byte	2.Byte	3.Byte

Dauer: 2,2 ms

Algebraische Subtraktion des Register R_1 von Register R_2.
Das Resultat befindet sich in Register R_2.
Im Falle der Kapazitätsüberschreitung (Resultat > 9 Stellen), wird das äußerste linke Halbbyte (Vorzeichen) nach Speicherstelle 0091 übertragen.

Sonderfall: Wenn $R_1 = R_2$ wird R_2 gelöscht.

Beispiel:



5.2.4 SBDD

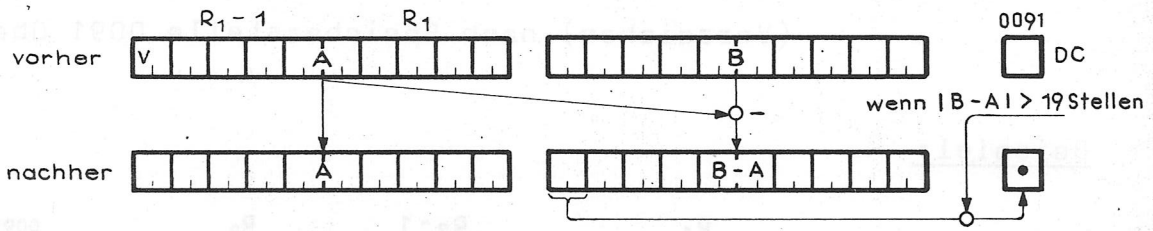
TO	Reg.-Nr.	Reg.-Nr.
B2	R_1 01 - 99	R_2 01 - 99
1.Byte	2.Byte	3.Byte

Dauer: 2,9 ms

Subtract Decimal Double length

Dieser Befehl ist gleich der SBD, jedoch werden hierbei 2 Doppelregister verwendet.

Beispiel:



5.2.5 RUUND

TO	Reg.-Nr.	Reg.-Nr.
7.1	K 00 - 99	R 01 - 99
1.Byte	2.Byte	3.Byte

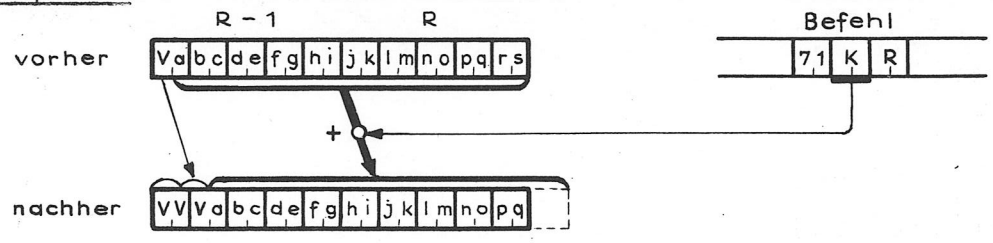
Dauer: 5,4 ms

Algebraische Addition des Zeichens K (2. Byte des Befehls) mit dem Doppelregister R.

2.R +50

Nach der Addition, Versetzung um 1 Byte nach rechts, mit Duplizierung des Vorzeichens und Wegfall der 1. zwei Ziffern rechts.

Beispiel:



5.2.6 MPD

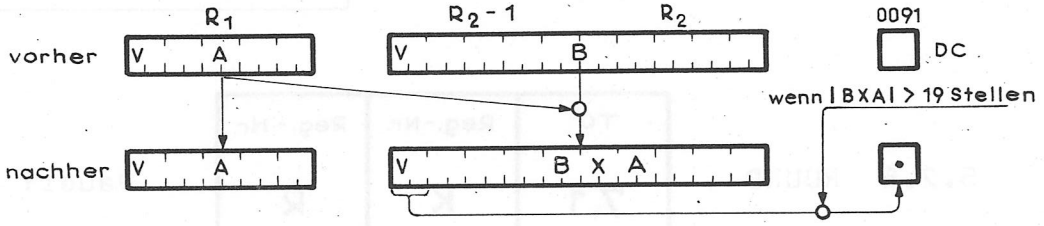
Multiply Decimal

TO	Register-Nr. Multiplikator	Register-Nr. Multiplikator
C1	R_1 00-99	R_2 01-99
1.Byte	2.Byte	3.Byte

Dauer: 50 (+.+) ms
52 (+.-) ms

Multiplikation des Inhaltes von Doppelregister R_2 (Multiplikant) mit dem Inhalt von Einfachregister R_1 (Multiplikator). Resultat in Doppelregister R_2.
Im Falle der Kapazitätsüberschreitung (Resultat > 10 Stellen) wird das äußerste linke Halbbyte (Vorzeichen) nach Speicherstelle 0091 übertragen.

Beispiel:



Sonderfall:
- Bei $R_1 = R_2$ wird der Inhalt des Registers R_2 zum Quadrat erhoben, unter der Bedingung, daß er kleiner als 10 Stellen ist.
- $R_1 = R_2 - 1$ ist möglich.

5.2.7 DVD

Divide Decimal!

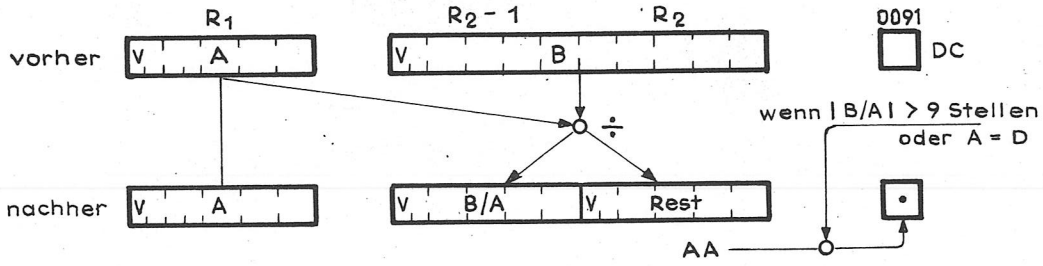
TO	Register - Nr. Multiplikator	Register - Nr. Multiplikator
D.1	R_1	R_2
	00 - 99	01 - 99
1.Byte	2.Byte	3.Byte

Dauer: 40 - 200 ms

Division des Inhaltes von Doppelregister R_2 (Dividend) durch den Inhalt des Einfachregisters R_1 (Divisor).
Am Ende der Operation enthält das Doppelregister:
- im linken Register den Quotient und
- im rechten Register den Rest.

Hat der Quotient mehr als 9 Stellen oder ist der Divisor gleich Null, so enthält die Stelle 0091, AA.

Beispiel:



Sonderfall:
- $R_1 = R_2$, Eingabe von 1 für Quotient, wenn $R_2 < 10$ Stellen ist.
- Ein Divisor Null ändert die Inhalte der Register 1 und 2 nicht.

5.3 Versetzungen

5.3.1 SLD

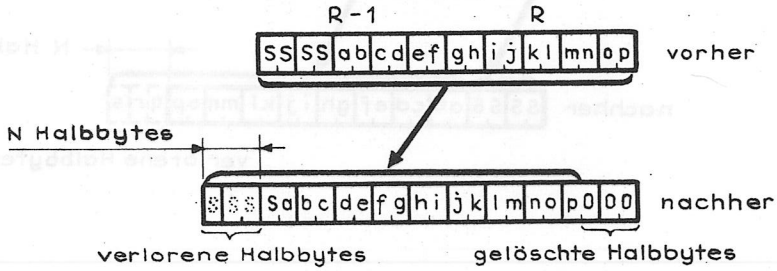
Shift left decimal

TO	Anzahl der Halbbytes	Reg. Nr.
7.0	N 00-19	R 00-99
1.Byte	2.Byte	3.Byte

Dauer: 1 + N ms

Versetzung des Doppelregisters R um N Halbbytes.
Nach der Versetzung sind:
- die rechten N Halbbytes gelöscht
- die linken N Halbbytes verloren.

Beispiel:



Anmerkung: Wenn es sich bei der Versetzung um eine Zahl handelt, die nicht zerstört werden darf, so muß ihre Stellenzahl $< (20 - N)$ sein.

Sonderfall: Die Zahl der durchgeführten Versetzungen ist gleich dem Rest der Division N : 20, wenn $N \geq 20$.

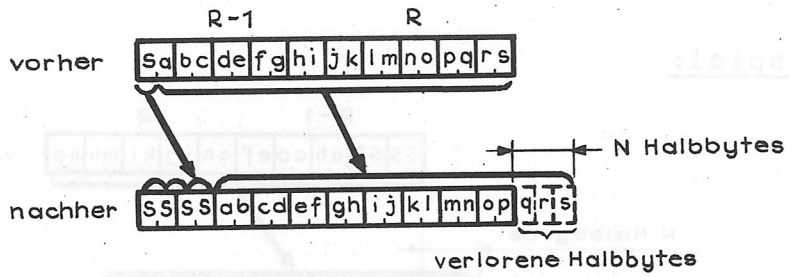
5.3.2 SRD

TO	Anzahl der Halbbytes	Reg.Nr.
7.2	N 00-19	R 01-99
1.Byte	2.Byte	3.Byte

Dauer: 1 + N ms

Versetzung des Doppelregisters R um N Halbbytes.
Das äußerste linke Halbbyte wird dupliziert.
Die rechten Halbbytes sind verloren.

Beispiel:



Sonderfall: Wenn N ≥ 20 ist, ist die Zahl der durchgeführten Versetzungen gleich dem Rest der Division N : 20.