

5.4 Vergleichsbefehle

5.4.1 CMD

TO	Reg. Nr.	Reg. Nr.
<b>9.1</b>	<b>R<sub>1</sub></b>	<b>R<sub>2</sub></b>
	00 - 99	00 - 99
1. Byte	2. Byte	3. Byte

Dauer: 1,75 ms =  
 2,10 ms ≠

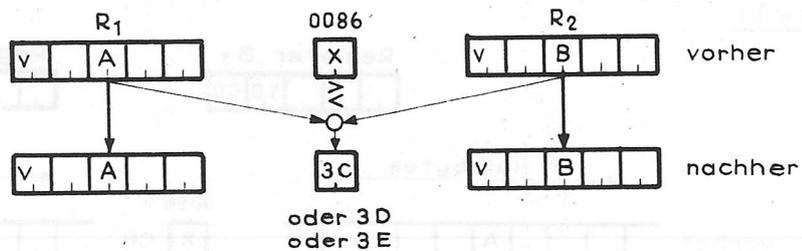
Algebraischer Vergleich des Inhaltes zweier Register (R<sub>1</sub> und R<sub>2</sub>). Das Vergleichsergebnis steht in Stelle 0086 im internen Code.

3C wenn (R<sub>1</sub>) < (R<sub>2</sub>)

3D wenn (R<sub>1</sub>) = (R<sub>2</sub>)

3E wenn (R<sub>1</sub>) > (R<sub>2</sub>)

Beispiel:



5.4.2 CMDD

TO	Reg. Nr.	Reg. Nr.
<b>9.2</b>	<b>R<sub>1</sub></b>	<b>R<sub>2</sub></b>
	01 - 99	01 - 99
1. Byte	2. Byte	3. Byte

Dauer: 2,08 ms =  
 2,8 ms ≠

Vergleich entsprechend der CMD, jedoch mit zwei Doppelregistern.

*Compare Characters*

5.4.3 CMC

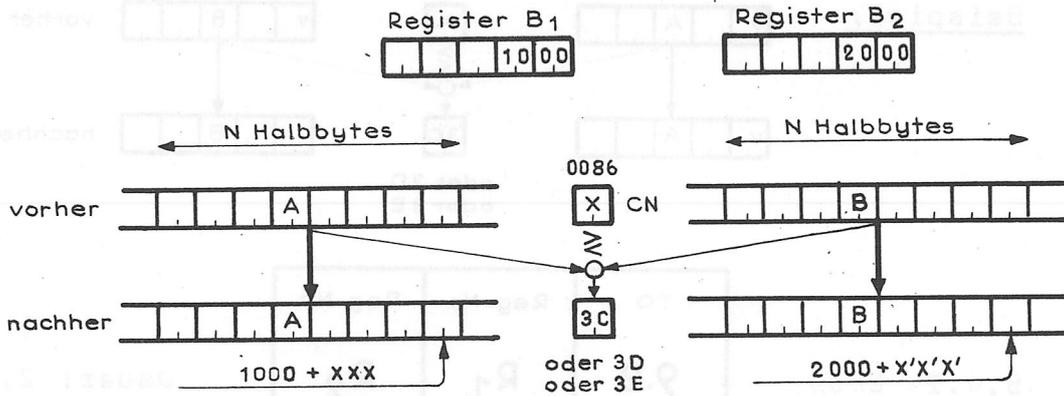
Dauer:  $2 + 0,07ms =$   
 $23 + 0,1Nms \neq$

TO	Anzahl der Halbytes	Basis	Versetzung		Basis	Versetzung	
<b>8.8</b>	<b>N</b>	<b>B<sub>1</sub></b>	<b>X</b>	<b>X X</b>	<b>B<sub>2</sub></b>	<b>X'</b>	<b>X' X'</b>
	00-99	0-9	000	- 999	0-9	000	- 999
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte		

Vergleich von 2 Worten gleicher Länge in Basis-  
adresse B<sub>1</sub> xxx und B<sub>2</sub>'x'x' .  
Das Resultat steht in Stelle 0086 im internen  
Code.

- 3C = <
- 3D = =
- 3E = >

Beispiel:



Sonderfall:

Wenn N gleich Null ist, enthält die Stelle  
0086, 3D (=)

5.5 Logische Befehle

5.5.1 NI  
*K in  
 78Code*

TO		Basis	Versetzung	
<b>6.1</b>	<b>K</b>	<b>B</b>	<b>X</b>	<b>X X</b>
	00-FE*	0-9	000	- 999
1. Byte	2. Byte	3. Byte	4. Byte	

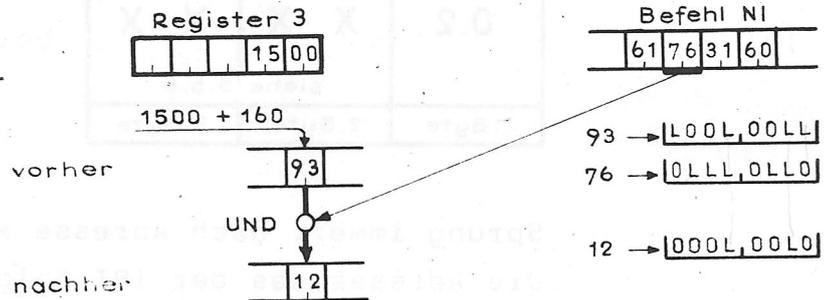
Dauer: 1,6 ms

*add immediate*

Zusammenfassung von 2 Zeichen, nach dem Prinzip des logischen "UND".

Das 1. Zeichen steht in Adresse Bxxx. Das 2. Zeichen ist das 2. Byte des Befehls.

Beispiel:



Anwendungsbeispiel:

wenn K = 01: Resultat = 00, (Bxxx) war PAARIG  
 wenn K = 01: Resultat = 01, (Bxxx) war UNPAARIG.

\* K darf nicht F1, F2, F3 oder FF sein.

5.5.2 OI

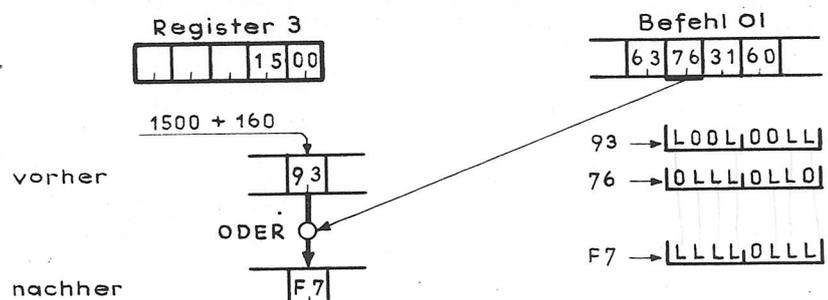
*Or immediate*

TO		Basis	Versetzung	
<b>6.3</b>	<b>K</b>	<b>B</b>	<b>X</b>	<b>X X</b>
	00-FE*	0-9	000	- 999
1. Byte	2. Byte	3. Byte	4. Byte	

Dauer: 1,1 ms

Zusammenfassung von 2 Zeichen, nach dem Prinzip des logischen "ODER".

Beispiel:



5.6 Sprungbefehle

5.6.1 NOP

TO
<b>0.0</b>
1. Byte

Dauer: 0,35 ms

Sprung niemals. Es wird systematisch der nächste Befehl gelesen.

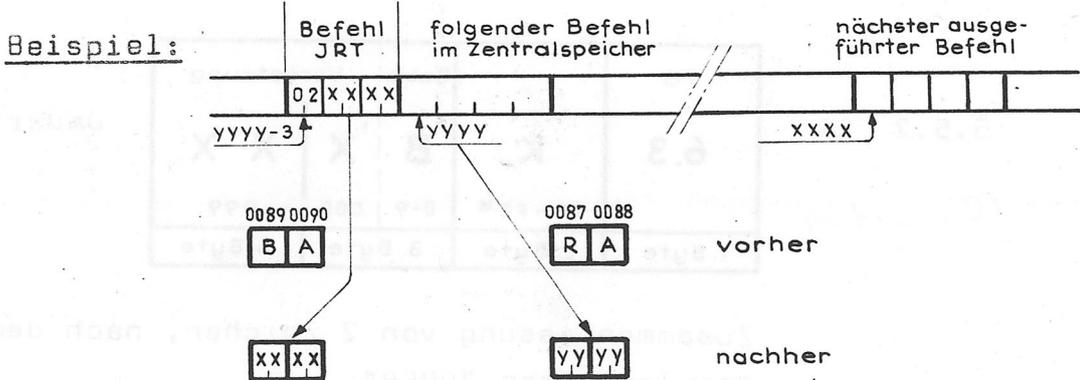
5.6.2 IRT

*Jump, save Address for Return*

TO	Sprungadresse	
<b>0.2</b>	X X	X X
	siehe 5.5.6	
1. Byte	2. Byte	3. Byte

Dauer: 0,8 ms

Sprung immer, nach Adresse xxxx. Die Adresse des der IRT folgenden Befehls, wird in Stelle 0087 + 0088 (Rücksprungadresse) gespeichert.



*Jump if immediate octet Equal to store octet.  
 save address for return*

5.6.3 JIERT

Dauer: bei Sprung  
 1,5 ms  
 ohne Sprung  
 1,3 ms

TO		Basis	Versetzung		Sprungadresse	
0.4	K	B	X	X X	X X	X X
	OO-FE*	0-9	000	- 999	siehe 5.5.6	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	

Ein Sprung wird durchgeführt, wenn das Zeichen, das sich in Adresse Bxxx befindet, gleich dem Zeichen K ist. ~~XXXX~~



In diesem Fall verhält sich JIERT wie JRT.

Die Sprungadresse befindet sich in xxxx.

Bei Ungleichheit wird nächster Befehl ausgeführt.

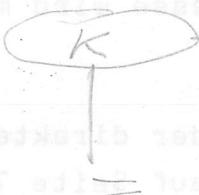
*Jump if immediate octet is unequal to store octet.  
 save address for return.*

5.6.4 JIURT

Dauer: bei Sprung  
 1,5 ms  
 ohne Sprung  
 1,3 ms

TO		Basis	Versetzung		Sprungadresse	
0.5	K	B	X	X X	X X	X X
	OO-FE*	0-9	000	- 999	siehe 5.5.6	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	

Ein Sprung wird ausgeführt, wenn das Zeichen, das sich in Adresse Bxxx befindet, ungleich dem Zeichen K ist. ~~XXXX~~



In diesem Fall, Sprung auf Adresse xxxx.

Bei Gleichheit wird nächster Befehl ausgeführt.

\* K darf nicht F1, F2, F3 oder FE sein.

### 5.6.5 Verwendung von JIERT und JIURT

In den meisten Fällen werden diese Befehle dazu benutzt, Vergleichsergebnisse auszuwerten und entsprechend dem Vergleichsergebnis den Programmablauf zu steuern. Man hat dann  $K = 3C (<)$ ,  $3D (=)$  oder  $3E(>)$  und  $Bxxx = 0086 (CN)$

Es sind dann folgende Fälle möglich:

Sprung bei	=	wenn eine	<b>JIERT</b>	und K	=	<b>3D</b>
	≠	"	JIURT	"	K	= <b>3D</b>
	<	"	<b>JIERT</b>	"	K	= <b>3C</b>
	≥	"	JIURT	"	K	= <b>3C</b>
	>	"	<b>JIERT</b>	"	K	= <b>3E</b>
	≤	"	JIURT	"	K	= <b>3E</b>

### 5.6.6

LEVEL

TO	'
<b>F.2</b>	<b>X X</b>
	00 - FD*
1. Byte	2. Byte

\* außer F1, F2 und F3

Dieser Befehl wird nur zur Programmaufstellung verwendet. Die Karte mit diesem Befehl liegt sofort vor der Karte des Befehls, auf den gesprungen werden soll. Der entsprechende Sprungbefehl (JRT, JIERT oder JIURT) hat dann nicht mehr die reelle Adresse xxxx, sondern die symbolische Adresse F1 xx.

xx entspricht dem xx der LEVEL-Karte.

Es ist also nicht mehr erforderlich, die direkte Adresse des Sprungs zu berechnen. Man legt an dieser Stelle eine LEVEL-Karte ein. Die Realadresse wird mit Programm errechnet.

Anmerkung: Der Befehl für das Errechnen der direkten Adresse ist RLA. Dieser wird auf Seite 73 beschrieben.

## 5.7 Übertragungen für Ein- und Ausgabezonen

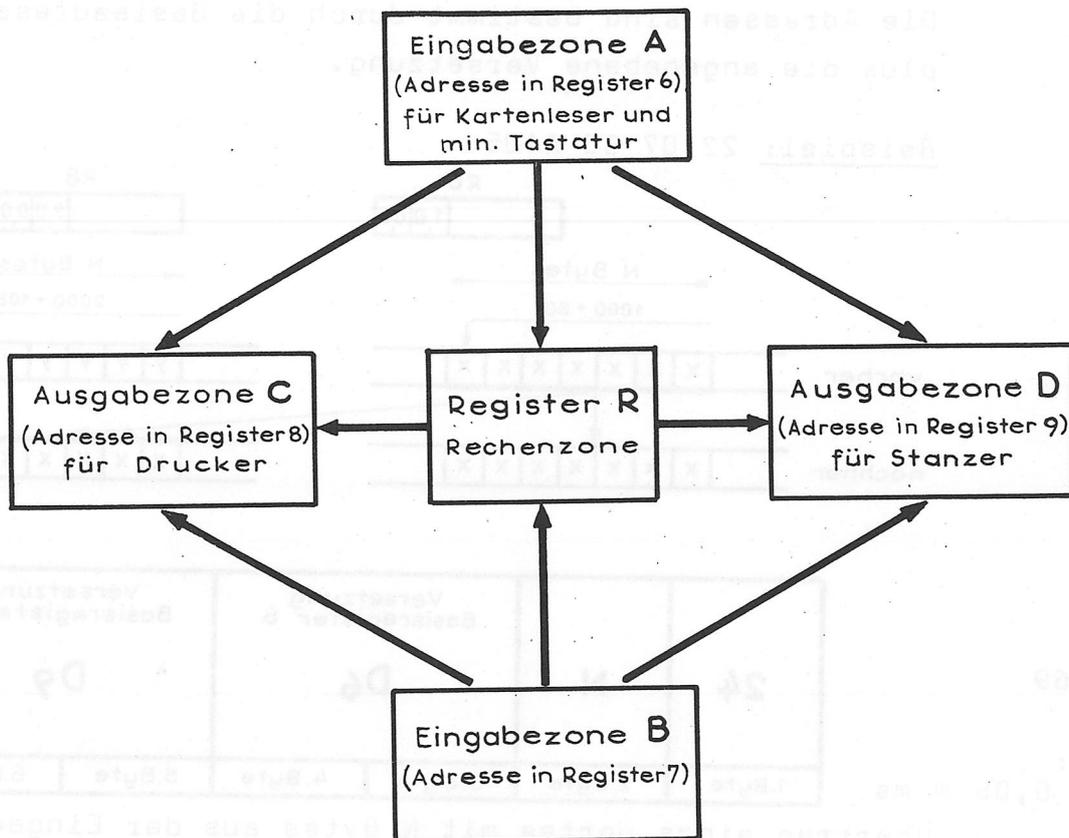
### 5.7.1 Allgemeines

Diese Befehle erlauben die simultane Übertragung eines Wortes von N Bytes, von

- R - einem Register
- A - Eingabezone (Basisadressregister 6)
- B - Eingabezone (Basisadressregister 7)

nach

- R - einem Register
- C - Ausgabezone (Basisadressregister 8)
- D - Ausgabezone (Basisadressregister 9)



5.7.2 Übertragungen von der Eingangszone zu den Ausgangs-  
zonen

Allgemeine Merkmale:

Diese Operation wird ausgeführt wie ein normaler Übertrag:

Es gibt keinen Wechsel des Formats.

Der Vorzeichenspeicher wird nicht verändert.

M 68

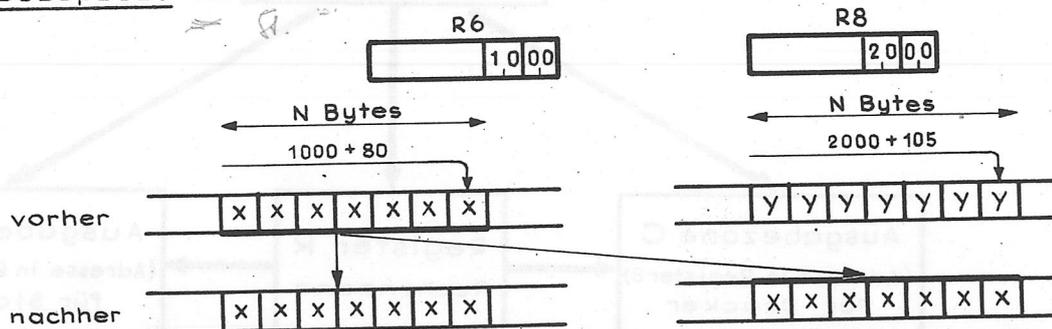
Dauer:  
3,5 + 0,05 N ms

TO	Anz. d. Bytes	Versetzung Basisregister 6		Versetzung Basisregister 8	
<b>22</b>	<b>N</b> 00 - 99	<b>D6</b>		<b>D8</b>	
1.Byte	2.Byte	3.Byte	4.Byte	5.Byte	6.Byte

Übertrag eines Wortes mit N Bytes aus der Eingangszone A in die Ausgangszone C.

Die Adressen sind bestimmt durch die Basisadressen plus die angegebene Versetzung.

Beispiel: 22 07 80 A105



M 69

Dauer:  
3,5 + 0,05 N ms

		Versetzung Basisregister 6		Versetzung Basisregister 9	
<b>24</b>	<b>N</b>	<b>D6</b>		<b>D9</b>	
1.Byte	2.Byte	3.Byte	4.Byte	5.Byte	6.Byte

Übertrag eines Wortes mit N Bytes aus der Eingangszone A in die Ausgangszone D.

Die Operation ist die gleiche wie bei M 68; nur die Zone C wird durch die Zone D ersetzt.

M 689  
 Dauer:  
 4 + 0,1 N ms

TO		Versetzung Basisregister 6		Versetzung Basisregister 8		Versetzung Basisregister 9	
26	N	D6		D8		D9	
	00-99						
1.Byte	2.Byte	3.Byte	4.Byte	5.Byte	6.Byte	7.Byte	8.Byte

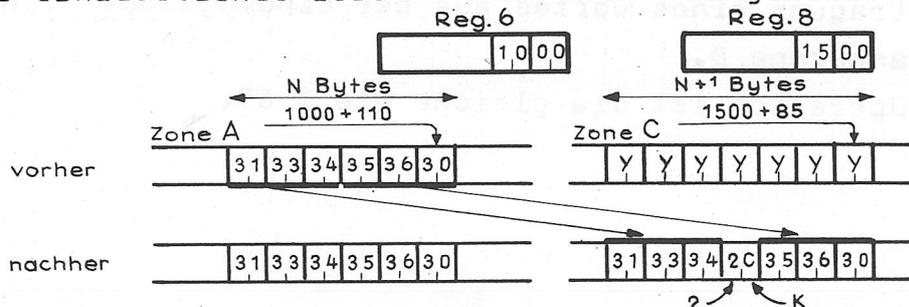
Gleichzeitige Übertragung eines Wortes von N Bytes aus Register 6 in Register 8 und 9. Die Übertragung erfolgt Byte pro Byte.

M68 G  
 Dauer:  
 4 + 0,05 N ms

*06      A 110      25      C 3*

TO						Anz.d. Dezi- malst.
2E	N	D6		D8		K   N'
						0-E   0-9
1.Byte	2.Byte	3.Byte	4.Byte	5.Byte	6.Byte	7.Byte

Die Operation ist ähnlich wie bei M68. Der Unterschied besteht darin, daß in die Ausgangszone C ein Zeichen (Komma, Punkt usw. unter Berücksichtigung der Dezimalstellen eingesetzt werden kann. Das <sup>letzte</sup> 7. Byte enthält in seiner linken Hälfte das einzusetzende Zeichen und in der rechten Hälfte die Anzahl der Dezimalstellen. Die Zeichen, welche eingesetzt werden können, sind hexadezimal verschlüsselt. Es sind folgende: 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E. Die 2 wird immer von der Maschine forciert, so daß in K nur das einzusetzende Zeichen ohne die 2 eingesetzt wird.



Wenn N = 00, findet kein Übertrag statt, aber das Komma ist auf seine Stelle in der Zone C gesetzt worden. Wenn der Wert des Wortes kleiner als 1 ist, wird links vor das eingesetzte Zeichen eine Null gesetzt. Alle im Befehl vorgesehenen Stellen hinter dem eingesetzten Zeichen werden geschrieben.

*Auch wenn es eine Leerpalte ist wird eine '0' im ISO-Wort = 30 eingesetzt.*

M 78

Dauer: 3,5 + 0,05 N ms

TO	Anz. d Bytes	Versetzung Basisregister 7	Versetzung Basisregister 8		
<b>32</b>	<b>N</b> 00 - 99	<b>D7</b>	<b>D8</b>		
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

Übertragung eines Wortes aus der Eingangszone B in die Ausgangszone C.

Die Operation ist die gleiche wie M68.

M 79

Dauer: 3,5 + 0,05 N ms

TO	Anz. d Bytes	Versetzung Basisregister 7	Versetzung Basisregister 9		
<b>34</b>	<b>N</b> 00 - 99	<b>D7</b>	<b>D8</b>		
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

Übertragung eines Wortes aus der Eingangszone B in die Ausgangszone D.

Die Operation ist die gleiche wie M 69.

TO	Anz.d. Bytes	Versetzung Basisregister 7		Versetzung Basisregister 8		Versetzung Basisregister 9	
36	N 00 - 99	D7		D8		D9	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte

M 789

Dauer:  $4 + 0,1 \cdot N$  ms

Übertragung eines Wortes aus der Eingangszone B in die Ausgangszonen C und D.

Die Operation ist die gleiche wie M 689

TO	Anz.d. Bytes	Versetzung Basisregister 7		Versetzung Basisregister 8		Anz.d. Dezimalst.
3E	N 00 - 99	D7		D8		K N'
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

M 78G

Dauer:  $4 + 0,05 \cdot N$  ms

Übertragung eines Wortes aus der Zone B in die Zone C mit Einsetzen eines Zeichens (20 bis 2E) entsprechend des 7. Halbbytes dieser Instruktion.

Diese Operation ist die gleiche wie M68G.

5.7.3 Übertragung vom Register zur Ausgangszone

Diese Übertragungen geben entsprechend ihrem Wert in den Vorzeichenspeicher (MS 0085) ein:

2D wenn der Wert < 0 ist, d.h. eine 9 im linken Halbyte des Registers

20 wenn der Wert ≥ 0 ist, d.h. eine 0 im linken Halbyte des Registers.

Beim Übertrag vom Register in die Ausgabezone, wird der Inhalt des Registers, wenn dieser negativ ist, ins Komplement gesetzt. *Zu 10<sup>10</sup>  
10<sup>20</sup>*

Zahlen, die sich im Register in Format 2 befinden, erscheinen in der Ausgabezone im Format 1.

Die Nullen links werden unterdrückt und durch Spaltensprünge ersetzt. (20)

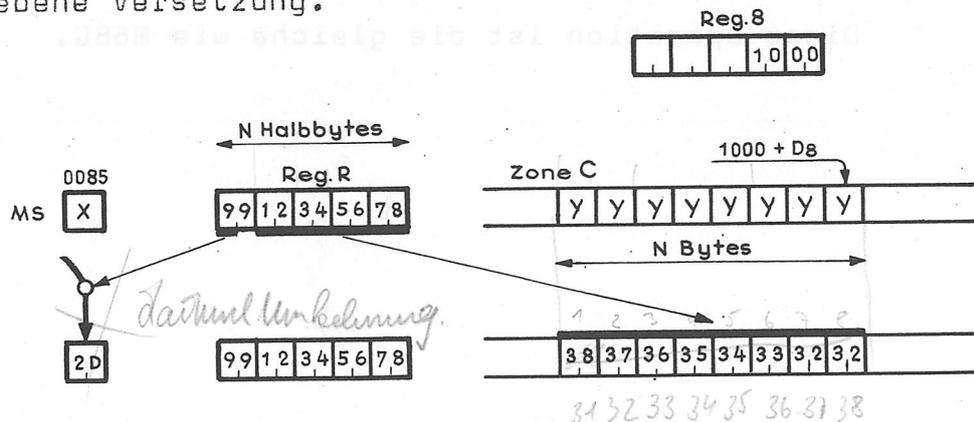
MR8

TO	Anzahl der Halbytes	Register-Nr.	Versetzung Basisregister 8	
12	N	R	D8	
	00 - 19	00 - 99		
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte

Dauer:  
5 + 0,2 N  
(+ 0,4 wenn -)ms

Übertragung des Inhaltes von N ~~rechten~~ Halbytes von Register R in die Zone C.

Die Zone C ist definiert durch Reg. 8 und die angegebene Versetzung.



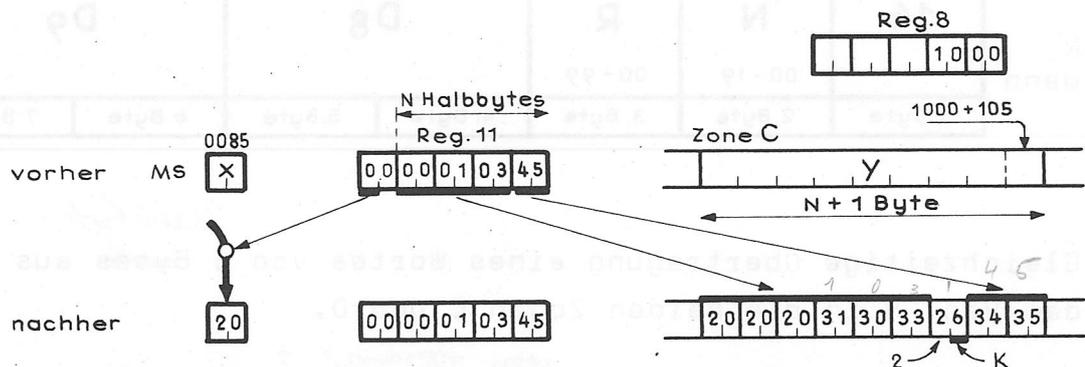
MR8G ✓

Dauer:  
 5,2 + 0,2 N ms  
 (+ 0,4 wenn -)

TO	Anzahl der Halbytes	Register-Nr.	Versetzung Basisregister 8		Anz.d. Dezimalst.
<b>1E</b>	<b>N</b>	<b>R</b>	<b>D8</b> <i>auf Adressen in Reg. 8</i>		<b>K N'</b>
	00-99	00-99			0-E 0-9
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte

Die Operation ist ähnlich wie bei MR8. Der Unterschied besteht darin, daß in die Zone C ein Zeichen (Komma, Punkt usw.) unter Berücksichtigung der Dezimalstellen eingesetzt werden kann. Das 7. Byte enthält in seiner linken Hälfte das einzusetzende Zeichen und in der rechten Hälfte die Anzahl der Dezimalstellen. Die Zeichen, die eingesetzt werden können, sind hexadezimal verschlüsselt. Es sind folgende:  
 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 2A, 2B, 2C, 2D, 2E.  
 Die 2 wird immer von der Maschine forciert, so daß in K nur das einzusetzende Zeichen steht.

Beispiel: 1E.08.11.A105.C2



Wenn N = 00, findet kein Übertrag statt, aber das Komma ist auf seine Stelle in der Zone C gesetzt worden.

Wenn der Wert des Wortes kleiner als 1 ist, wird links vor das eingesetzte Zeichen eine Null gesetzt.

Alle im Befehl vorgesehenen Stellen hinter dem eingesetzten Zeichen werden geschrieben.

MR9

 Dauer:  $5 + 0,2N$   
 (+ 0,4 wenn -)ms

TO	Anzahl der Halbytes	Register-Nr.	Versetzung Basisregister 9	
<b>14</b>	<b>N</b>	<b>R</b>	<b>D9</b>	
	00 - 19	00 - 99		
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte

Übertragung des Inhaltes von N rechten Halbytes aus dem Reg. R in die Zone D.

Die Instruktion ist der von MR8 ähnlich. Anstelle der Zone C tritt hier die Zone D.

*Von Format 2 (gepackt)  
in Format 1 ungepackt.*

MR89

 Dauer:  
 $6 + 0,25N$   
 (+ 0,4, wenn -)  
 ms

TO	Anzahl der Halbytes	Register-Nr.	Versetzung Basisregister 8	Versetzung Basisregister 9		
<b>16</b>	<b>N</b>	<b>R</b>	<b>D8</b>	<b>D9</b>		
	00 - 19	00 - 99				
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

Gleichzeitige Übertragung eines Wortes von N <sup>Halbytes</sup> Bytes aus dem Reg. R in die beiden Zonen C und D.

*von Format 2*

*in Format 1*

5.7.4 Übertragung der Eingangszone in ein Register

Bei dieser Übertragung ist der algebraische Wert entsprechend dem Inhalt des MS (0085 im MC).

- 20 oder 00 - der Wert ist  $\geq 0$  *Memoriesigne -*
- 2D - der Wert ist  $< 0$

Es ist also eventuell eine Änderung des Inhaltes von MS vor der Übertragung erforderlich.

Am Ende der Operation ist der MS auf 00.

Umsetzen von *780 Code* Format 1 in *780 Code* Format 2.

Vor der Übertragung ist der Wert normal codiert, also in Format 1.

Nach der Übertragung steht der Wert in seiner algebraischen Form also in Format 2.

TO	Anzahl der Bytes	Versetzung Basisregister 6	Register-Nr.
21	N	D6	R
	00 - 19		00 - 99
1. Byte	2. Byte	3. Byte	5. Byte

M6R

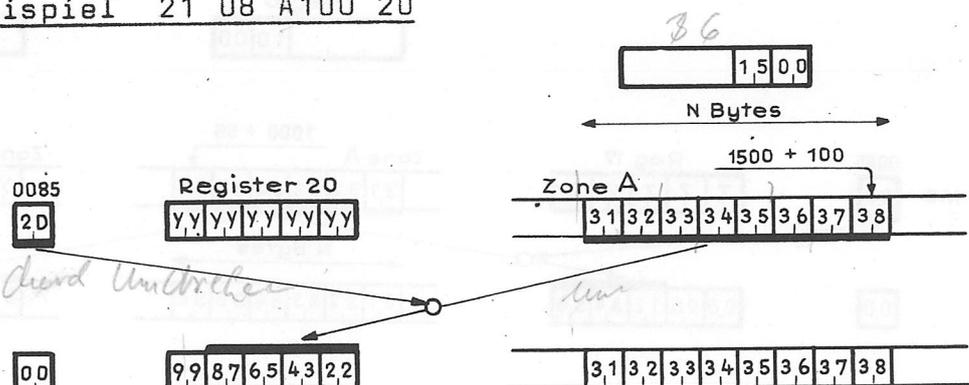
Dauer: 4,7 + 0,2 N ms

Übertragung eines Wortes mit N Bytes aus der Eingangszone in ein Register R.

Einfachregister wenn  $N < 10$

Doppelregister wenn  $N \geq 10$

Beispiel 21 08 A100 20



TO	Anzahl der Bytes	Versetzung Basisregister 7		Reg.-Nr.
<b>31</b>	<b>N</b>	<b>D7</b>		<b>R</b>
	00 - 19			00 - 99
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte

M7R

Dauer: 4,7+0,2 N ms

Übertragung eines Wortes mit N Bytes aus der Eingangszone in ein Register R.

Die Operation ist ähnlich wie M6R. Anstelle Zone A tritt hier die Zone B.

5.7.5 Übertragung der Eingangszone in ein Register und eine Ausgangszone

Dieser Befehl erlaubt zwei verschiedene Übertragungen simultan, wenn die abgebende Zone gemeinsam ist:

- Eine Übertragung Eingangszone zum Register (siehe 5.7.4)
- Eine Übertragung Eingangszone zur Ausgangszone (s. 5.7.3)

TO N Da 2 Dc

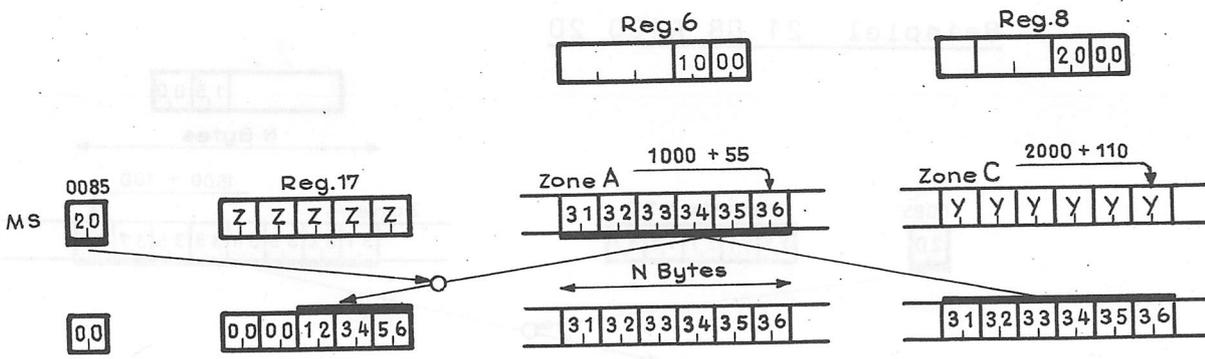
TO	Anz. d. Bytes	Versetzung Basisregister 6		Reg.-Nr.	Versetzung Basisregister 8	
<b>23</b>	<b>N</b>	<b>D6</b>		<b>R</b>	<b>D8</b>	
	00 - 19			00 - 99	Dc	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

M6R8 ✓

Dauer: 5,8+0,25N ms

Übertragung eines Wortes mit N Bytes aus der Eingangszone A in das Register R und die Ausgangszone C.

Beispiel 23 06 55 17 A110



TO	Anz. d. Bytes	Versetzung Basisregister 6		Reg.-Nr.	Versetzung Basisregister 9	
<b>25</b>	<b>N</b> 00 - 19	<b>D6</b>		<b>R</b> 00 - 99	<b>D9</b>	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

M6R9

Dauer:  $5,8 + 0,25N$  ms

Übertragung eines Wortes mit N Bytes aus der Eingangszone A in das Register R und in die Ausgangszone D. Die Instruktion ist ähnlich wie M6R8, die Zone C wird durch die Zone D ersetzt.

TO	Anz. d. Bytes	Versetzung Basisregister 6		Reg.-Nr.	Versetzung Basisregister 8		Versetzung Basisregister 9	
<b>27</b>	<b>N</b> 00 - 19	<b>D6</b>		<b>R</b> 00 - 99	<b>D8</b>		<b>D9</b>	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte	9. Byte

M6R89

Dauer:  $7 + 0,3N$  ms

Übertragung eines Wortes mit N Bytes aus der Eingangszone A in ein Register R und die Ausgangszonen C und D.

M7R8

Dauer:  $5,8 + 0,25N$  ms

TO	Anz. d. Bytes	Versetzung Basisregister 7		Reg.-Nr.	Versetzung Basisregister 8	
<b>33</b>	<b>N</b>	<b>D7</b>		<b>R</b>	<b>D8</b>	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

Übertragung eines Wortes mit N Bytes aus der Zone B in das Register R und die Ausgangszone C.

Diese Instruktion ist ähnlich wie M6R8.

Anstelle der Eingangszone A tritt die Eingangszone B.

M7R9

Dauer:  $5,8 + 0,25N$  ms

TO	Anz. d. Bytes	Versetzung Basisregister 7		Reg.-Nr.	Versetzung Basisregister 9	
<b>35</b>	<b>N</b>	<b>D7</b>		<b>R</b>	<b>D9</b>	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte

Übertragung eines Wortes mit N Bytes aus der Eingangszone B in ein Register R und die Ausgangszone D.

Diese Instruktion ist ähnlich wie M6R9.

Anstelle der Eingangszone A tritt die Eingangszone B.

M7R89

Dauer:  $7 + 0,3N$  ms

TO	Anz. d. Bytes	Versetzung Basisregister 7		Reg.-Nr.	Versetzung Basisregister 8		Versetzung Basisregister 9	
<b>37</b>	<b>N</b>	<b>D7</b>		<b>R</b>	<b>D8</b>		<b>D9</b>	
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte	9. Byte

Übertragung eines Wortes mit N Bytes aus der Eingangszone B in ein Register R und die Ausgangszonen C und D.

Diese Instruktion ist ähnlich wie bei M6R89.

Anstelle der Eingangszone A tritt die Eingangszone B.



5.8 Anschlußbefehle für Randeinheiten

		1. Randeinheit		2. Randeinheit		n. Randeinheit (<7)	
IOC	Anzahl d. Rand- einheiten	C <sub>1</sub>	R <sub>1</sub>	C <sub>2</sub>	R <sub>2</sub>	C <sub>n</sub>	R <sub>n</sub>
	53	N					
	1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte
							8. Byte

N = Anzahl der anzuschließenden Randeinheiten (N < 7)

R = Nummer des Registers, welches die Adresse der Zone enthält, mit der die Randeinheit korrespondiert.

C = Anschlußcode entsprechend folgender Tafel:

Kanal	abgebende Randeinheit			annehmende Randeinheit		
	MHZ	OHZ*		MHZ	OHZ*	
1	92	95	Pufferspeicher	9A	9D	Pufferspeicher
1		96	Kartenleser			
1	94	97	α-Tastatur			
2				AA	AD	Stanzer
3				CA	CD	Drucker

In einer IOC dürfen nicht 2 Randeinheiten an den selben Kanal angeschlossen sein.

\* MHZ = mit Halt pro Zeichen, OHZ = ohne Halt pro Zeichen.

Diese Unterscheidung ist interessant bei Multiprogrammierung. Bei Monoprogramm kann das eine oder andere verwendet werden.