

Beispiele:

a) SRD

72|02|21

vorher:

nachher:

00....03|75|45

00....00|03|75

0196

0205

0196

0205

375.45 → 375

b) SRD

72|14|21

99|24|50.....00|00|00

99|99|99.....99|24|50

0196

0205

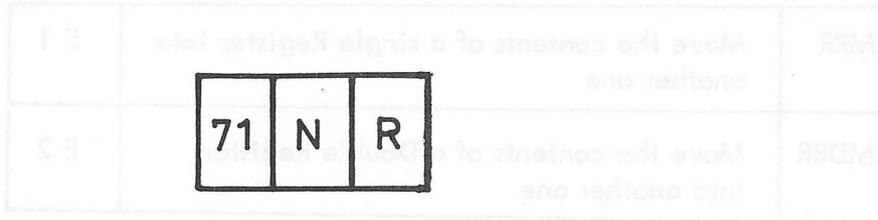
0196

0205

3.3.3.3. Rundungen

**ROUND**

Round



N = 00 - 99

R = Doppelregister

Die Zahl N wird ohne Berücksichtigung des Vorzeichens auf den Inhalt des Doppelregisters R addiert. Anschließend wird das Resultat, das sich in R befindet, um zwei Dezimalstellen (1 Byte) nach rechts versetzt, wobei die zwei rechten Stellen verloren gehen.

Die Operation erfolgt ohne Berücksichtigung des Vorzeichens.

Anmerkung:

Bei N = 00 erhält man eine Division durch 100 unter Vernachlässigung der Dezimalstellen.

Enthält das Register keine numerischen Werte, so wird das nicht angezeigt. Das Resultat ist jedoch falsch.

Das Resultat der Addition darf 19 Stellen nicht überschreiten, damit das Vorzeichen nicht zerstört wird. Kapazitätsüberschreitung wird jedoch nicht angezeigt.

Beispiele:

Round **71|50|85**

vorher:

nachher:

- a) ...00 07 25      00 00 07 = + 7.25 → 7
- b) ...00 07 85      00 00 08 = + 7.85 → 8
- c) ...99 92 75      99 99 93 = - 7.25 → 7
- d) ...99 92 15      99 99 92 = - 7.85 → 8

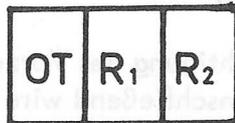
0525

0525

### 3.3.4. Übertragungsbefehle

#### 3.3.4.1. Überträge in einfacher und doppelter Länge

MRR	Move the contents of a single Register into another one	E 1
MDRR	Move the contents of a Double Register into another one	E 2



OT = siehe oben

R<sub>1</sub> = Abgebendes Register (einfach oder doppelt je nach OT)

R<sub>2</sub> = Empfangendes Register (einfach oder doppelt je nach OT)

Der Inhalt des Registers R<sub>1</sub> wird in das Register R<sub>2</sub> übertragen, nachdem dessen Inhalt gelöscht wurde. Der übertragene Wert erfährt keine Veränderung. Handelt es sich um verschiedene Register, wird der Inhalt von R<sub>1</sub> nicht verändert.

#### Länge:

- MRR : 5 Bytes oder 10 Dezimalziffern (R<sub>1</sub> + R<sub>2</sub> einfach)
- MDRR : 10 Bytes oder 20 Dezimalziffern (R<sub>1</sub> + R<sub>2</sub> doppelt)

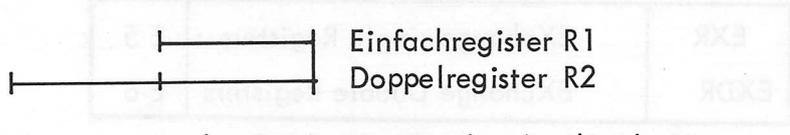
Bei MDRR:

- Wenn  $R_1 = R_2 - 1$  erfolgt der Übertrag normal. Das kommt auf das gleiche hinaus, als wenn man den Inhalt des Registers 1 um 10 Dezimalstellen nach rechts verschoben hätte.
- Wenn  $R_1 = R_2 + 1$  erhält man dreimal den Inhalt der 10 Dezimalstellen von rechts des Registers 1 in den Einfachregistern: R<sub>1</sub>, R<sub>2</sub> und R<sub>2</sub> - 1.





b) MSRDR | E3 | 17 | 18 |



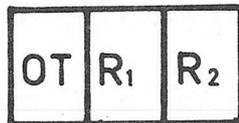
vorher : | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |

nachher : | 9 9 | 9 9 | 9 9 | 9 9 | 9 9 | 9 9 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |

Zahl negativ

3.3.4.3. Registeraustausch

EXR	EXchange single Registers	E 5
EXDR	EXchange Double Registers	E 6



OT = siehe oben  
 R<sub>1</sub>, R<sub>2</sub> = Einfach- oder Doppelregister je nach OT

Dieser Befehl vollzieht den Austausch des Inhaltes der Register R<sub>1</sub> und R<sub>2</sub>. Am Ende der Operation befindet sich im Register 1 der vorige Inhalt von R<sub>2</sub> und in R<sub>2</sub> der vorige Inhalt von R<sub>1</sub>.

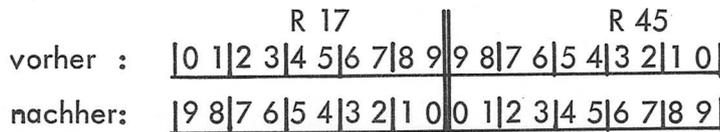
Größen:

- bei EXR : 5 Bytes oder 10 Dezimalstellen
- bei EXDR : 10 Bytes oder 20 Dezimalstellen

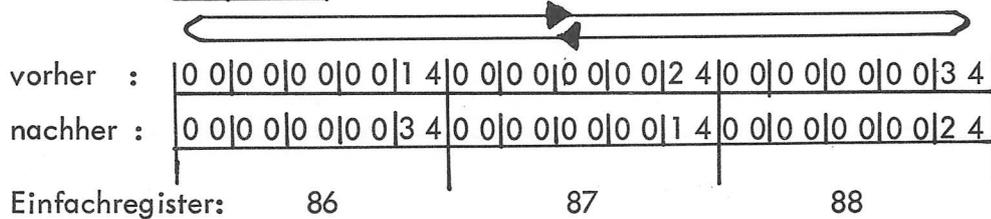
Bei EXDR ist R<sub>1</sub> = R<sub>2</sub> - 1 möglich. In diesem Fall erhält man einen rechtsläufigen Ringtausch des Inhaltes der drei Einfachregister.

Beispiele:

a) EXR | E5 | 17 | 45 |



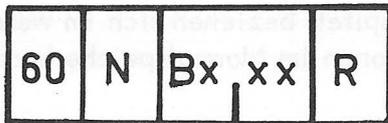
b) EXDR | E6 | 87 | 88 |



3.3.4.4. Laden eines Registers

LRN

Load Register N



N = 01 - 10

Bxxx = Adresse des zu übertragenden Wertes

R = empfangendes Einfach- oder Doppelregister

Mit diesem Befehl kann man in ein Register R eine Zahl von N Bytes (2N Dezimalziffern) übertragen, die in algebraischer Form gepackt im Normal-speicherbereich an der Adresse Bxxx steht.

Der Übertrag erfolgt in ein Einfachregister, wenn  $N \leq 5$  ist, und in ein Doppelregister, wenn  $N > 5$  ist. Der vorherige Inhalt des Registers wird gelöscht.

Die Zahl wird rechtsbündig in das Register eingesetzt. Wenn nicht alle Stellen des Registers belegt werden ( $N < 5$  oder 10), wird die äußerste linke Ziffer bis zum linken Ende des Registers wiederholt. Diese Ziffer bestimmt das Vorzeichen: Es muß unbedingt 0 oder 9 sein. Anderenfalls ist die Zahl im Register ungleich der ursprünglichen Zahl .

Anmerkung:

Es werden nur die Ziffern der Einer bei N berücksichtigt; die Ziffer 0 wird wie 10 behandelt.

Beispiel:

LRN	<u>60</u>   <u>03</u>   <u>71,52</u>   <u>14</u>	Basis 7 = 2000
	Normalspeicher	Reg. 14
vorher :	<u>2 0,9 5,4 2,1 3,2 0</u>	<u>0 0   0 0   0 0   0 0   0</u>
nachher :	<u>2 0,9 5,4 2,1 3,2 0</u>	<u>9 9   9 9   9 5   4 2   1 3</u>

### 3.4. Operationen auf Zeichenbasis

#### 3.4.1. Allgemeines

Die Befehle dieses Kapitels beziehen sich im wesentlichen auf die Bearbeitung von Informationen im Normalspeicherbereich des Zentralspeichers und zwar sind es:

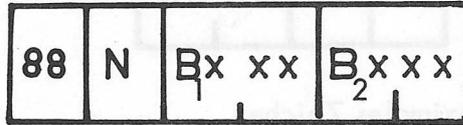
- logische Befehle
- Übertragungsbefehle.

Sie arbeiten in variabler Länge auf 8 Bit-Zeichen. Die Länge der bearbeiteten Worte wird durch die Zahl des Bytes spezifiziert, außer, ein Befehl bearbeitet nur ein Zeichen. Das trifft bei einigen logischen Befehlen zu. Die benutzte Adressierung ist die Basisadressierung.

3.4.2. Logische Befehle

3.4.2.1. Alphanumerischer Vergleich

**CMC** CoMpare Characters



N = Die Zahl der zu vergleichenden Bytes von 00 - 99  
 Bxxx = Adresse der rechten Stelle der zu vergleichenden Worte

Mit diesem Befehl können zwei Worte gleicher Länge N verglichen werden: Das Wort, das sich an B<sub>1</sub>xxx befindet, wird verglichen mit dem Wort an B<sub>2</sub>xxx. Die beiden Operanden werden durch die Operation nicht verändert. Der Vergleich verarbeitet Zeichen, die sich aus 8 Bits zusammensetzen. Die Worte können daher beliebige Zeichen des internen Codes enthalten.

Das Vergleichsergebnis beruht auf der Binärdarstellung des internen Codes und wird im Vergleichregister in Form eines hexadezimalen Zeichens gespeichert.

- 3 C bei  $B_1xxx < B_2xxx$
- 3 D bei  $B_1xxx = B_2xxx$
- 3 E bei  $B_1xxx > B_2xxx$

Bei N = 0 werden beide Worte wie "Nichts" behandelt. Das Register enthält das Ergebnis "Gleich" (3 D).

Beispiel:

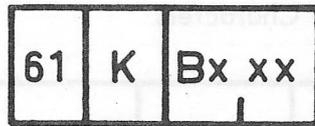
CMC	88   06   6006   7006		Basis 6 = 1500	
			Basis 7 = 2000	
	Zone 6	Zone 7		VG
vorher :	3 0   4 5   3 1   3 8   5 4   4 F   3 4	3 0   4 C   3 0   3 3   4 2   5 6   3 9		00
nachher :	3 0   4 5   3 1   3 8   5 4   4 F   3 4	3 0   4 C   3 0   3 3   4 2   5 6   3 9		3C
	1500	1505	2000	2005
	$B_1xxx < B_2xxx$			

3.4.2.2. Logisches UND

NI

aNd

Immediat



K = hexadezimaler Zeichen

Bxxx = Adresse des zu bearbeitenden Zeichens

Dieser Befehl setzt nach den Regeln des logischen UND das Zeichen K auf das Zeichen, das sich an Adresse Bxxx befindet.

Beispiel:

NI | 61 | 77 | 6004 | Basis = 1800

K (Binärdarstellung) | 0 1 1 1, 0 1 1 1 | (77)

Stelle 1804 vorher: | 1 1 1 1, 1 0 1 1 | (FB)

nachher: | 0 1 1 1, 0 0 1 1 | (73)

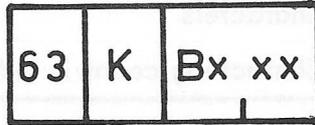
*Wenn = 1  
≠ 0*

*all 0*

3.4.2.3. Logisches ODER

$\emptyset I$

Or      Immediat



K = hexadezimaler Zeichen  
 Bxxx = Adresse des zu bearbeitenden Zeichens

Dieser Befehl setzt nach den Regeln des logischen ODER das Zeichen K auf das Zeichen, das sich an Adresse Bxxx befindet.

Das Ergebnis steht an dieser Adresse zur Verfügung. Das Zeichen, das ursprünglich dort stand, ist verändert worden.

Beispiel:

$\emptyset I$       |63|34|6081|      Basis 6 = 2300

K (Binärdarstellung)      |0 0 1 1, 0 1 0 0|      (34)

Stelle 2381 vorher:      |1 1 1 0, 0 0 0 0|      (D0)

nachher:      |1 1 1 1, 0 1 0 0|      (F4)

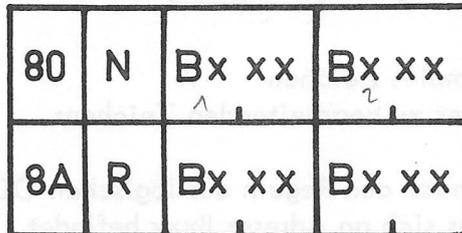
*alle 1*

*Wenn  $\neq \rightarrow 1$   
 $= = 0$  oder  $1$*

3.4.3. Übertragungsbefehle im Normalspeicherbereich

3.4.3.1. Einfachüberträge

MVC	MoVe Characters
MVIC	MoVe Characters; count is Indirect



N = Länge des Übertrages in Anzahl Bytes

R = Nummer des Registers, das die Länge N enthält.

Die Länge N kann 00 - 99 sein.

B<sub>1</sub>xxx = rechte Adresse der abgebenden Zone

B<sub>2</sub>xxx = rechte Adresse der empfangenden Zone

Die Adresse B<sub>2</sub>xxx muß  $\geq$  als Adresse B<sub>1</sub>xxx oder  $\leq$  Adresse B<sub>2</sub>xxx - N sein.

Das Wort mit der Länge von N Bytes, das an Adresse B<sub>1</sub>xxx steht, wird nach Adresse B<sub>2</sub>xxx übertragen.

Der alte Inhalt der empfangenden Zone wird gelöscht. Der Inhalt der abgebenden Zone bleibt unverändert, wenn beide Felder unterschiedliche Adressen haben.

Die Länge des Wortes wird durch den Wert von N bestimmt:

- bei MVC innerhalb des Befehls
- bei MVIC durch den Inhalt des rechten Bytes des Registers R.

Der Übertrag erfolgt Stelle für Stelle von rechts nach links.

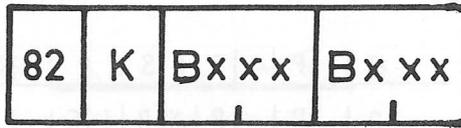
Anmerkung:

Befindet sich die Adresse B<sub>2</sub>xxx innerhalb der Adressen B<sub>1</sub>xxx - N und B<sub>1</sub>xxx, werden nur die Stellen zwischen den Adressen B<sub>2</sub>xxx und B<sub>1</sub>xxx richtig übertragen; sie wiederholen sich aber innerhalb des Empfangsfeldes (vgl. Beispiel -b)).



3.4.3.2. Einsetzen eines Zeichens

INC      IN sert Character



- K = einzusetzendes Zeichen
- B<sub>1</sub>xxx = Adresse der 1. Stelle, die belegt werden soll (rechte Adresse)
- B<sub>2</sub>xxx = Adresse der letzten Stelle - 1, die belegt werden soll (linke Adresse)

Die Adresse B<sub>1</sub>xxx muß > B<sub>2</sub>xxx sein.

Das Zeichen K wird auf jede Stelle einer Speicherzone eingesetzt, die durch die beiden Adressen begrenzt wird. Das Einsetzen erfolgt von rechts nach links von Stelle B<sub>1</sub>xxx einschließlich bis Stelle B<sub>2</sub>xxx ausschließlich.

Wichtig:

Ist die Adresse B<sub>1</sub>xxx ≤ Adresse B<sub>2</sub>xxx, erfolgt die Belegung bis Stelle 0000 des Speichers und vom Ende des Speichers bis Adresse B<sub>2</sub>xxx. Der Inhalt der speziellen Zonen, der Register und des Programmes werden zerstört.

Beispiel:

INC      |82|20|7117|7108|      Basis 7 = 1200

vorher : |39|39|37|36|35|34|33|32|31|30|20|5D|4D|

nachher : |39|38|37|20|20|20|20|20|20|20|20|5D|4D|

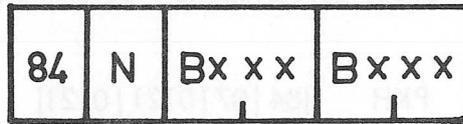
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17  
 1306                      1310                      1315

(1308)  
 1309  
 1312

3.4.3.3. Packen hexadezimal

PKH

Pack Hexadezimal



- N = Zahl der zu packenden Bytes von 00-99
- B<sub>1xxx</sub> = rechte Adresse der abgebenden Zone
- B<sub>2xxx</sub> = rechte Adresse der empfangenden Zone

Die Adresse B<sub>2xxx</sub> muß  $\geq$  als B<sub>1xxx</sub> oder  $\leq$  B<sub>1xxx</sub> - N sein.

Dieser Befehl packt auf N Halbbytes ein Wort von N Bytes, das im ISO-Code ausgedrückt ist und nur die folgenden Werte enthält: Leerstelle, 0 - 9 und A - F. Die Umwandlung von Byte auf Halbbyte ist in der Tabelle dargestellt:

Zeichen	Leer- stelle	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ISO-Code (Byte)	20	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46
gepackt (Halb- byte)	0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Jede andere Kombination des ISO-Codes in den ursprünglichen Bytes gibt ein nicht auswertbares, sinnloses Ergebnis.

Das Wort von N Bytes, das gepackt werden soll, steht an Adresse B<sub>1xxx</sub>. Es wird in gepackter Form nach Adresse B<sub>2xxx</sub> übertragen und belegt dort N Bytes. Ist N ungerade, wird auf volle Bytes aufgerundet. In diesem 2 Fall enthält das äußerste linke Halbbyte Null.

Der alte Inhalt der empfangenden Zone wird gelöscht. Bei unterschiedlichen Feldern bleibt der Inhalt des abgebenden Feldes unverändert. Die Operation packt Stelle für Stelle von rechts nach links.

Anmerkung:

Befindet sich die Adresse B<sub>2xxx</sub> innerhalb der Adressen B<sub>1xxx</sub> - N und B<sub>1xxx</sub>, ist das Ergebnis nicht voraussehbar und hat keinerlei Wert.

Der Befehl PKH dient nicht dazu, Zahlen, mit denen gerechnet werden soll, von ungepackter in gepackte Form zu bringen. Diese Aufgabe wird von den Mehrfachübertragungsbefehlen übernommen. (s. 3.5.) Im wesentlichen dient PKH dazu, geschriebene Programme durch das Ladeprogramm zu verdichten.

Beispiel:                    PKH    84 | 07 | 0121 | 0121                    Basis 0 = 2500

vorher :    45 | 42 | 30 | 46 | 41 | 32 | 31 | 20 | 20

nachher:    45 | 42 | 30 | 0E | B0 | FA | 21 | 20 | 20

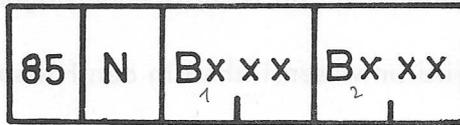
2621

Zeichen	ISO-Codes (byte)	gepackt (Half-byte)
0	30	0
1	31	1
2	32	2
3	33	3
4	34	4
5	35	5
6	36	6
7	37	7
8	38	8
9	39	9
A	40	A
B	41	B
C	42	C
D	43	D
E	44	E
F	45	F

3.4.3.4. Entpacken hexadezimal

UPH

Un Pack Hexadezimal



N = Zahl der zu erzielenden Bytes von 00 - 99  
 B<sub>1</sub>xxx = rechte Adresse der abgebenden Zone  
 B<sub>2</sub>xxx = rechte Adresse der empfangenden Zone

Die Adresse B<sub>2</sub>xxx muß:

$$\geq B_{1xxx} + \frac{N}{2} \text{ oder } \leq B_{1xxx} - \frac{N}{2} \text{ sein.}$$

Ist N ungerade, wird  $\frac{N}{2}$  auf volle Bytes aufgerundet.

Dieser Befehl drückt im internen ISØ-Code ein Wort auf N Bytes aus, das auf N Halbbytes gespeichert ist. Die 16 hexadezimalen Werte, die jedes Halbbyte darstellen kann, ergeben folgende Zeichen ISØ-Codes:

Inhalt des Halbbytes	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
ISØ-Code (1 Byte)	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46		
Zeichen	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		

Das an Adresse B<sub>1</sub>xxx stehende, gepackte Wort wird in den ISØ-Code übersetzt und auf N Bytes an Adresse B<sub>2</sub>xxx übertragen.

Die abgebende Zone enthält  $\frac{N}{2}$  Stellen, aufgerundet auf volle Stellen, wenn N ungerade ist. In diesem Fall wird das äußerste linke Halbbyte dieser Zone nicht bearbeitet.

Der alte Inhalt der empfangenden Zone wird gelöscht. Bei verschiedenen Feldern bleibt der Inhalt der abgebenden Zone unverändert. Die Operation entpackt Stelle für Stelle von rechts nach links.

Anmerkung:

Befindet sich die Adresse B<sub>2</sub>xxx zwischen den Adressen  $B_{1xxx} + \frac{N}{2}$  und  $B_{1xxx} - \frac{N}{2}$  ist das Ergebnis nicht vorhersehbar und ohne Bedeutung.

Dieser Befehl dient nicht dazu, algebraische Zahlen aus gepackter in ungepackte Darstellung zu bringen.

Diese Aufgabe übernehmen die Mehrfachübertragungsbefehle (s. 3.5.).

Der Befehl UPH wird im wesentlichen in den Hilfsprogrammen benutzt.

Beispiel:

UPH            85 | 05 | 0150 | 0153            Basis 0 = 1000

vorher :    AB | 4C | 0B | 1E | 20 | 20 | 20 | A1

nachher:    AB | 4C | 43 | 30 | 42 | 31 | 45 | A1

1150

Zeichen	150-Code (1 Byte)	150-Code	150-Code
0	30	31	32
1	34	35	36
2	39	3A	3B
3	3C	3D	3E
4	3F	40	41
5	43	44	45
6	48	49	4A
7	4B	4C	4D
8	4E	4F	
9			
A			
B			
C			
D			
E			
F			

### 3.5. Mehrfachüberträge

#### 3.5.1. Darstellung

Die Mehrfachübertragungsbefehle gestatten es, ein Wort variabler Länge von einer Zone nach 1,2 oder 3 anderen Zonen gleichzeitig zu übertragen.

Die abgebende Stelle ist entweder ein numerisches Register oder eine von zwei Normalspeicherzonen; im allgemeinen Eingabezonen von Randeinheiten.

Die empfangenden Stellen sind: ein numerisches Register und/oder eine oder 2 Normalspeicherzonen; im allgemeinen Ausgabezonen für Randeinheiten.

Im Befehl kann nur ein Register entweder als abgebend oder als empfangend angegeben werden. Der Befehl erkennt die Normalspeicherzonen an den Adressen, die in den Basisregistern 6 - 9 stehen. Die Basisregister 6 und 7 entsprechen immer abgebenden Zonen; die Basisregister 8 und 9 immer empfangenden Zonen. Durch Vereinbarung und zur Unterscheidung bezeichnet man die Zonen mit der Nummer des ihnen zugeordneten Basisregisters.

Die verschiedenen Kombinationen der zu erzielenden Überträge, die durch Variation der abgebenden und empfangenden Zonen entstehen, ergeben eine Anzahl Befehle, die in Abschnitt 3.5.4 zusammengestellt sind.

Die Abschnitte 3.5.2 und 3.5.3 beschreiben die gemeinsamen Eigenschaften.

Drei Befehle geben die Möglichkeit, ein Zeichen in eine Zahl, die nach Zone 8 übertragen wurde, einzusetzen. Sie sind beschrieben in Abschnitt 3.5.5.

#### 3.5.2. Allgemeiner Aufbau und Darstellung

##### Symbolischer Operationstyp

Jedem Mehrfachübertragungsbefehl entspricht ein symbolischer Schlüssel der sich zusammensetzt aus dem Buchstaben M und 2, 3 oder 4 Zeichen, die von links nach rechts bedeuten:

die abgebende Stelle und die empfangende(n) Stelle(n). Die empfangenden Zonen stehen in folgender Reihenfolge:

numerisches Register, Zone 8, Zone 9.