Programmieren in der IBM/360 ASSEMBLER-Sprache

IBM

ASS-PU

Text 4

Programmierte Unterweisung

IBM Schule für Datenverarbeitung

Programmieren in der IBM / 360 Assembler-Sprache (ASS-PU)

Eine Programmierte Unterweisung

Text 4

Programmiert von:

Gerhard K. Bulling
Helmut Hertweck
Heinz Laubner
Rosa Maria Schartz

Herausgegeben von:

IBM Deutschland GmbH
7032 Sindelfingen
Abt. DP-Unterrichtsmethoden

Copyright by IBM Deutschland 1968
Alle Rechte, auch die des Nachdrucks, der Herstellung von Kopien und der Übersetzung vorbehalten.

DP-Unterrichtsmethoden
Gerhard K. Bulling/Lch
Änderungen des Textes bleiben vorbehalten.
Zweite Auflage August 1970

Dieser Band ist Teil der IBM Form R12-1731

INHALT

		Seite
Kapitel I:	Die Übertragungsoperationen MVC, MVZ, MVN und MVI	7
Kapitel II:	Vergleichs- und Verzweigungsoperationen	77
Kapitel III:	Die Operationen PACK, UNPK, AP, SP und ZAP	145
Kapitel IV:	Die Operationen MP, DP, MVO und CP	251
Kapitel V:	Die ED-Operation für die Druckaufbereitung	359
Kapitel VI:	Die Operationen BALR, A, AH, AR, LR, L und LA	431
Anhang:	Die Zuordnung der Instruktionen zu den Instruktionsformaten	503

Vorbemerkung zum Aufbau dieses Bandes

Dieser Band umfaßt 6 Kapitel.

In jedem Kapitel sind eine oder mehrere Instruktionen behandelt.
Bei jeder Instruktion finden Sie folgende Gliederung:

1. Die linke Seite enthält als Abbildung die Beschreibung der Instruktion. Diese Beschreibung ist Ihre Informationsquelle über die jeweilige Instruktion.

2. Der Inhalt der Beschreibung wird in Lernelementen abgefragt und gegebenenfalls an Anwendungsbeispielen gefestigt und erweitert.

3. In anschließenden Übungen können Sie Ihr Wissen überprüfen und einüben.

Jedem Kapitel sind FRAGEN bzw. ÜBUNGEN angefügt. Die nachfolgenden Blätter enthalten die richtigen Lösungen.

KAPITEL I

DIE ÜBERTRAGUNGSOPERATIONEN MVC, MVZ, MVN UND MVI

Lernstoff:

MVC - Übertragen Zeichen

MVZ - Übertragen Zonen

MVN - Übertragen Ziffern

MVI - Übertragen Direktzeichen

MVC

ÜBERTRAGEN ZEICHEN

(MoVe Characters)

Diese Operation überträgt Zeichen von einem Speicherfeld in ein anderes.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVC	AFELD,EFELD

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel EFELD) wird in die Speicherstellen (Bytes) des ersten Operanden (im Beispiel AFELD) übertragen.

Der bisherige Inhalt des ersten Operanden wird zerstört.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die Länge der Operation ist variabel. Es gibt aber eine obere Grenze: Eine MVC-Operation kann maximal den Inhalt von 256 Bytes übertragen.

Durchführung der Operation:

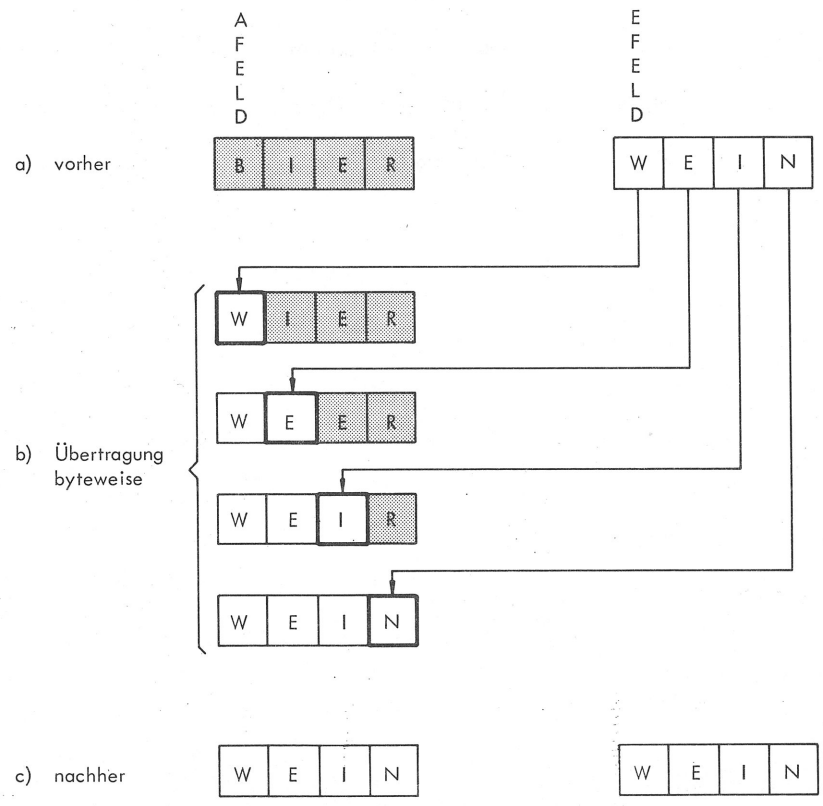


Abb. 1

Beachten Sie bitte zum Aufbau dieses Bandes:

Bevor Sie sich den Lernelementen zu einer Instruktion zuwenden, müssen Sie die dazugehörende Abb. 1 durchgearbeitet haben!

1 In Abb. 1 sehen Sie die Beschreibung einer Operation, die Ihnen nicht mehr ganz unbekannt ist.

Sie hat im Englischen die Bezeichnung "Move Characters".
Im Deutschen heißt sie "___(WW)___".

Der Operationsschlüssel ist von der englischen Bezeichnung abgeleitet und heißt _____.

Übertragen Zeichen

MVC

2 Betrachten Sie das Beispiel einer Instruktion in Abb. 1.

a) Wieviele Operanden hat eine Operation "Übertragen Zeichen"?

b) Welche symbolischen Namen haben die beiden Operanden im Beispiel?

2

AFELD und EFELD

3 Einer der beiden Operanden gibt an, WOHIN Zeichen übertragen werden und der andere sagt, WOHER die Zeichen kommen.

Benutzen Sie die Beschreibung in Abb. 1 und entscheiden Sie:

WOHIN Zeichen übertragen werden, gibt der Operand mit dem Namen _____ an.
WOHER die Zeichen kommen, sagt der Operand mit dem Namen _____.

AFELD

EFELD

MVC

ÜBERTRAGEN ZEICHEN

(MoVe Characters)

Diese Operation überträgt Zeichen von einem Speicherfeld in ein anderes.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVC	AFELD,EFELD

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel EFELD) wird in die Speicherstellen (Bytes) des ersten Operanden (im Beispiel AFELD) übertragen.

Der bisherige Inhalt des ersten Operanden wird zerstört.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die Länge der Operation ist variabel. Es gibt aber eine obere Grenze: Eine MVC-Operation kann maximal den Inhalt von 256 Bytes übertragen.

Durchführung der Operation:

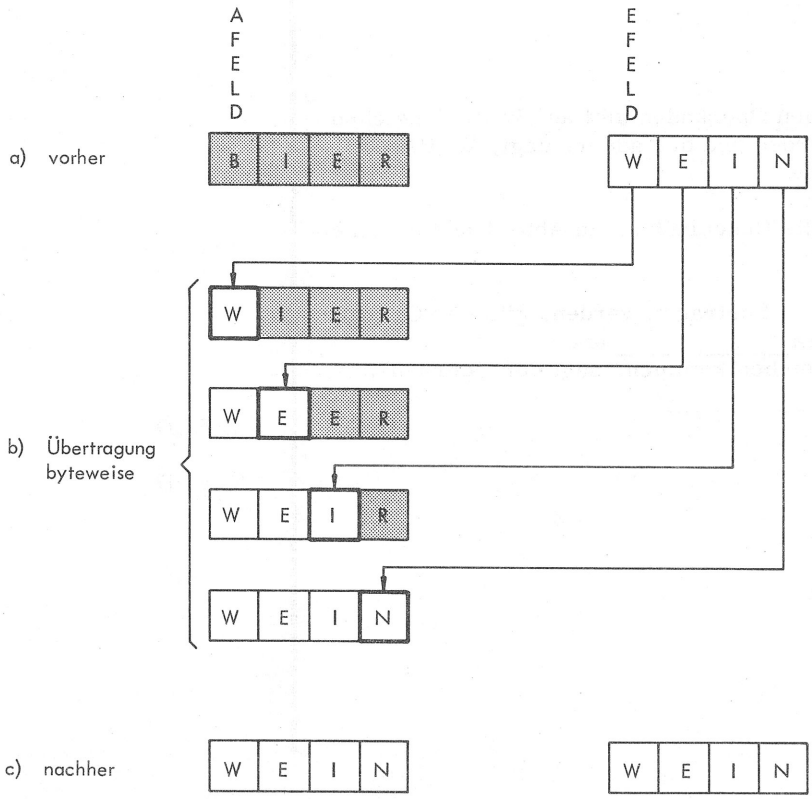


Abb. 1

4 Für die Reihenfolge der Operanden gilt kurz:

1. Operand - WOHIN

2. Operand - WOHER

Bei der Übertragung wird der Inhalt des _____ Operanden verändert.
Der Inhalt des zweiten Operanden dagegen bleibt erhalten. Er kann daher wiederholt übertragen werden.

ersten

5 Nach der Durchführung der Operation sind die übertragenen Zeichen (im ersten Operanden, im zweiten Operanden, in beiden Operanden) _____ enthalten.

in beiden Operanden

6 In Abb. 1 ist auch dargestellt, wie eine Übertragung im DVS durchgeführt wird.

Der Inhalt des Feldes EFELD wird (bitweise, byteweise, auf einmal) _____ übertragen.

byteweise

7 Im ersten Übertragungsschritt wird der Inhalt des (linken, rechten) _____ Bytes eines Feldes in das (linke, rechte) _____ Byte des anderen Feldes übertragen.

linken

linke

8 Es wird übertragen

byteweise

von links nach rechts

in der Länge des e r s t e n Operanden.

Hat bei einem Übertragungsbefehl der erste Operand eine Länge von zwei Bytes und der zweite Operand eine Länge von vier Bytes, so werden _____ Bytes übertragen.

zwei

MVC

ÜBERTRAGEN ZEICHEN

(MoVe Characters)

Diese Operation überträgt Zeichen von einem Speicherfeld in ein anderes.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVC	AFELD,EFELD

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel EFELD) wird in die Speicherstellen (Bytes) des ersten Operanden (im Beispiel AFELD) übertragen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Der bisherige Inhalt des ersten Operanden wird zerstört.

Die Länge der Operation ist variabel. Es gibt aber eine obere Grenze: Eine MVC-Operation kann maximal den Inhalt von 256 Bytes übertragen.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Durchführung der Operation:

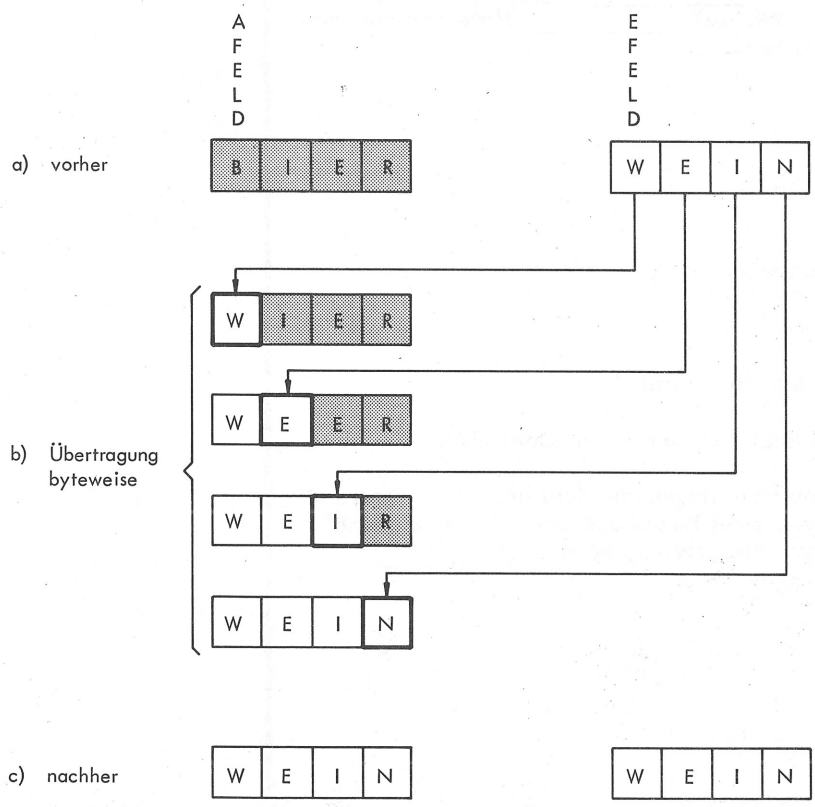
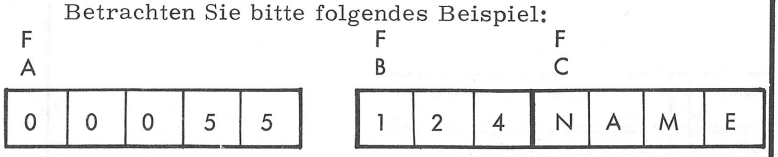


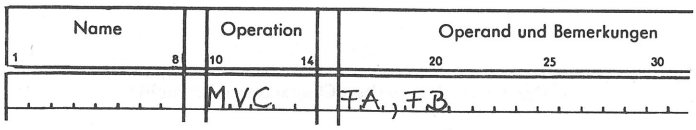
Abb. 1

9 Es wird also übertragen in der Länge des e r s t e n Operanden.

Diese Regel gilt in jedem Fall, ganz gleich, ob der erste Operand genauso lang, kürzer oder länger ist als der zweite Operand. Die Länge des z w e i t e n Operanden hat keinerlei Einfluß auf die Ausführung einer Übertragungsoperation:

Betrachten Sie bitte folgendes Beispiel:

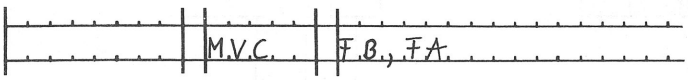




a) Wieviel Bytes werden hier in ihrem Inhalt verändert?

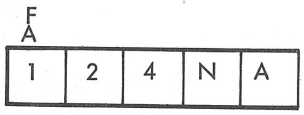
b) Welchen Inhalt hat das Feld FA nach der Ausführung dieser Operation?

c) Welchen Inhalt haben die beiden Felder FB und FC, wenn folgende MVC-Operation ausgeführt wurde?
(Beachten Sie bitte, daß die beiden Operanden nun vertauscht sind!)

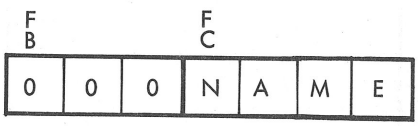


a) 5 Bytes

b)



c)



MVC

ÜBERTRAGEN ZEICHEN

(MoVe Characters)

Diese Operation überträgt Zeichen von einem Speicherfeld in ein anderes.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVC	AFELD,EFELD

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel EFELD) wird in die Speicherstellen (Bytes) des ersten Operanden (im Beispiel AFELD) übertragen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Der bisherige Inhalt des ersten Operanden wird zerstört.

Die Länge der Operation ist variabel. Es gibt aber eine obere Grenze: Eine MVC-Operation kann maximal den Inhalt von 256 Bytes übertragen.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Durchführung der Operation:

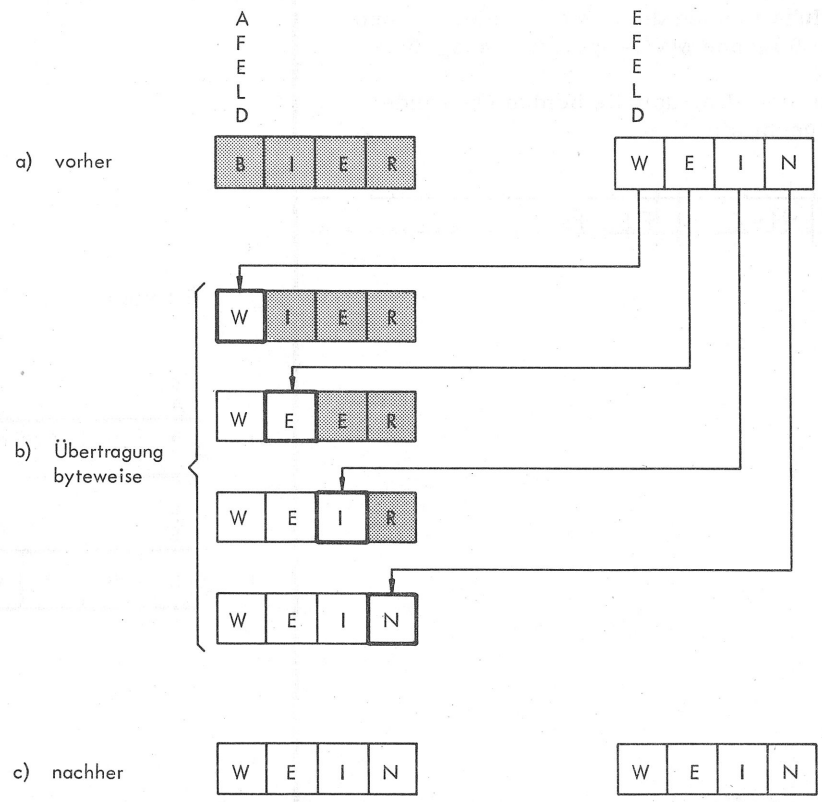


Abb. 1

Übungen (Teil 1)

Entnehmen Sie Ihren Arbeitsmitteln Übung 401, Seite 1, 2 und 3.

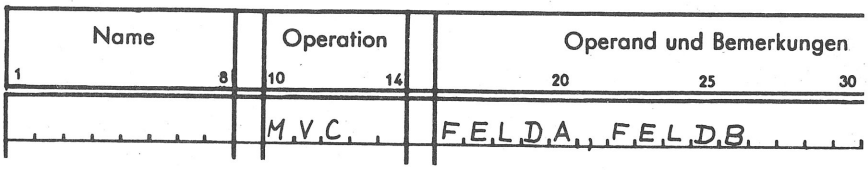
Tragen Sie bitte in die dafür vorgesehenen leeren Felder den Inhalt der Operanden nach der Übertragung ein.

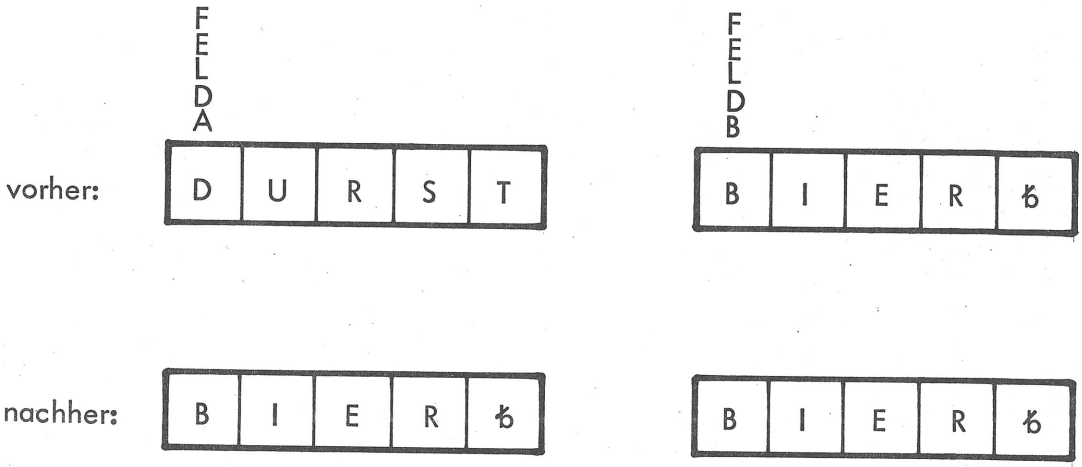
Vergleichen Sie Ihre Eintragungen mit den Lösungen auf den folgenden Seiten, sobald Sie ein Blatt vollständig bearbeitet haben.

Als Hilfsmittel können Sie Abb. 1 benutzen.

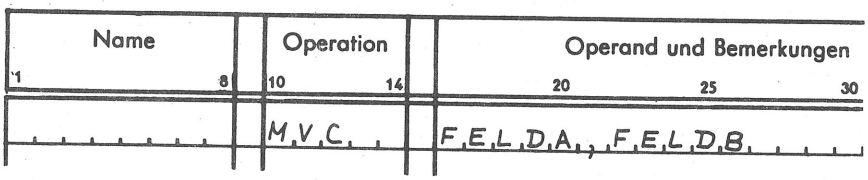
Lösungen zu den Übungen (Teil 1)

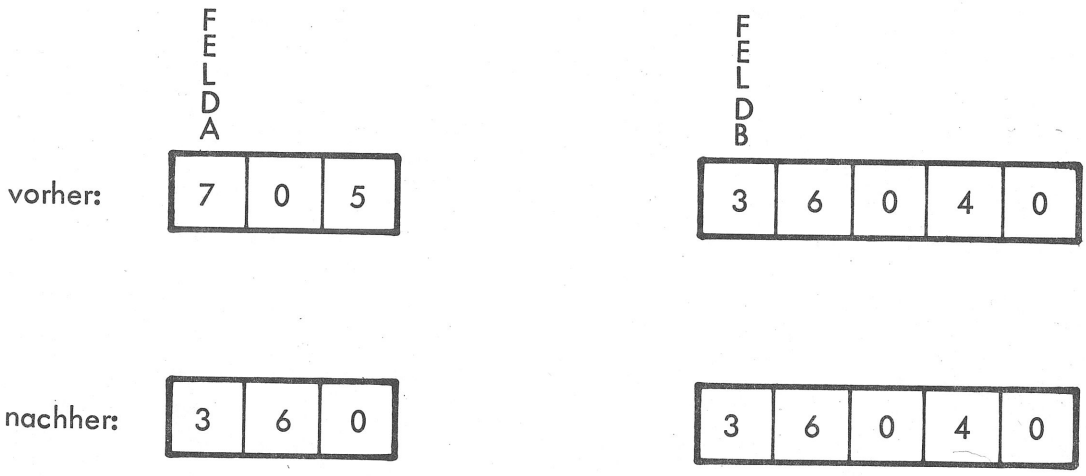
1.

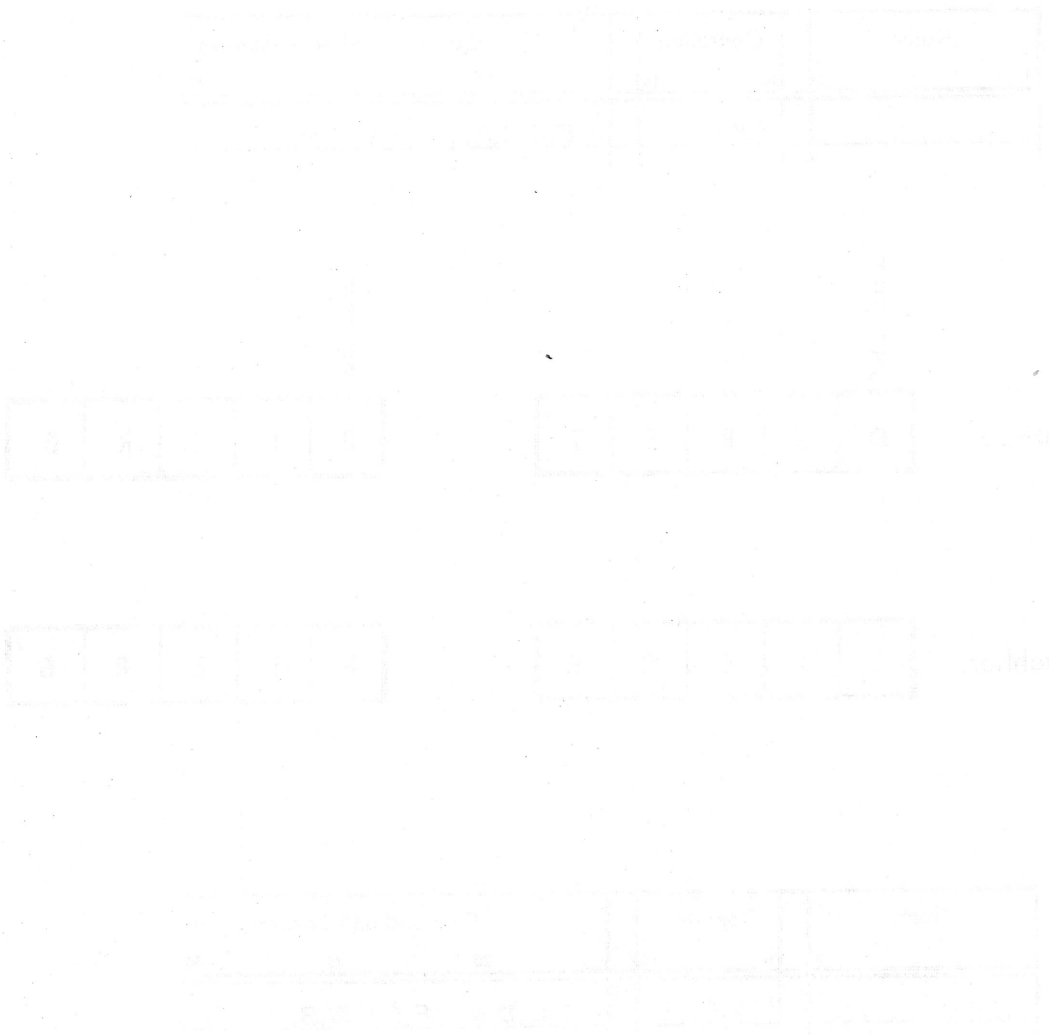




2.

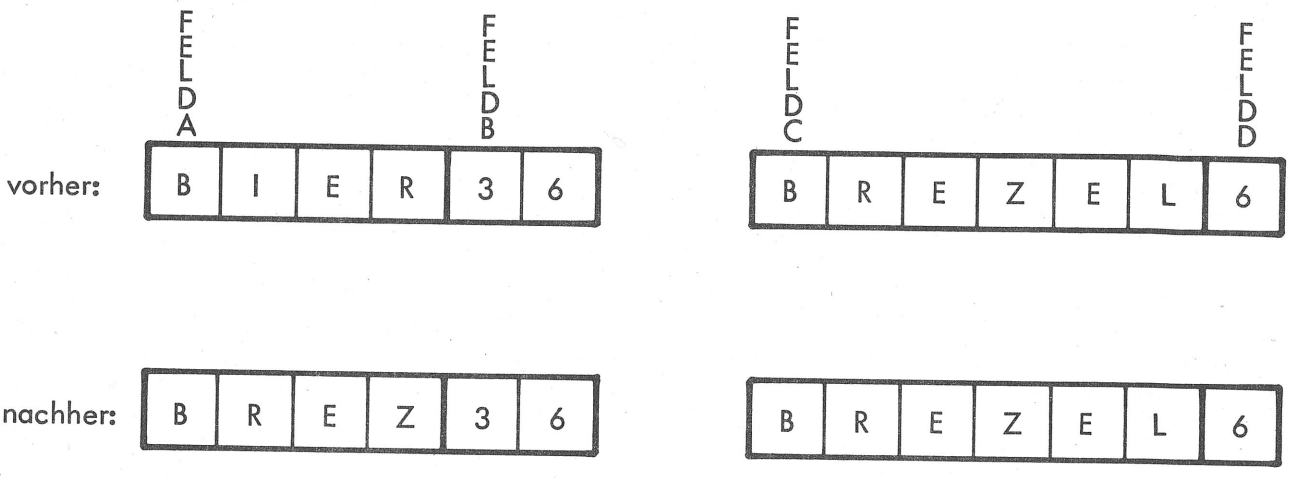






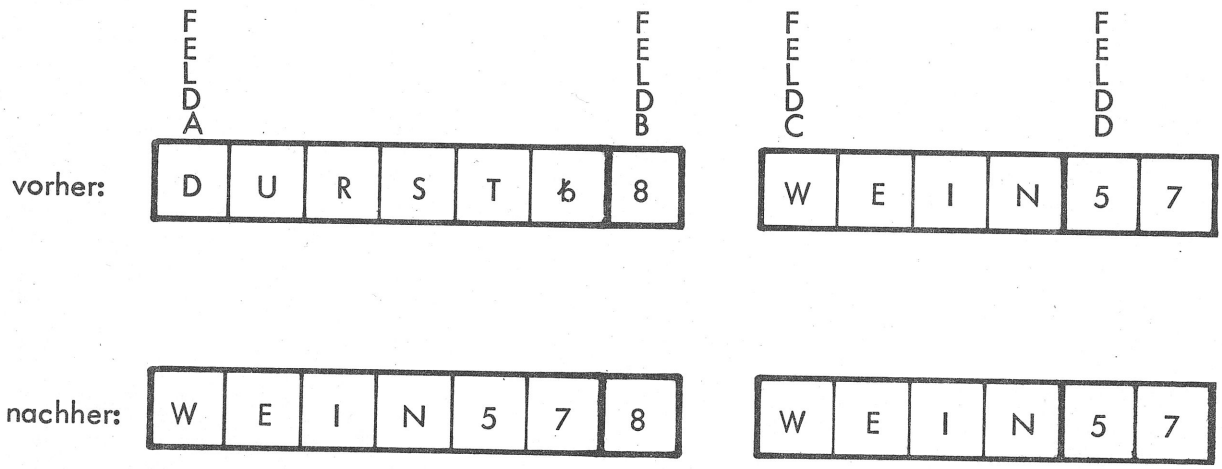
3.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,C, ,	F,E,L,D,A,,,F,E,L,D,C, , , , ,

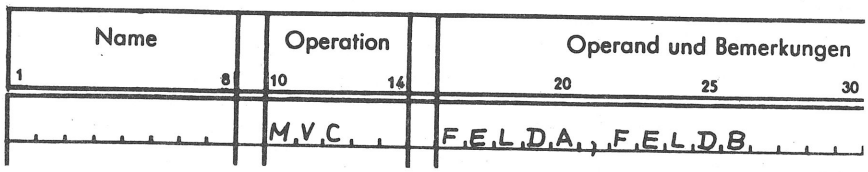


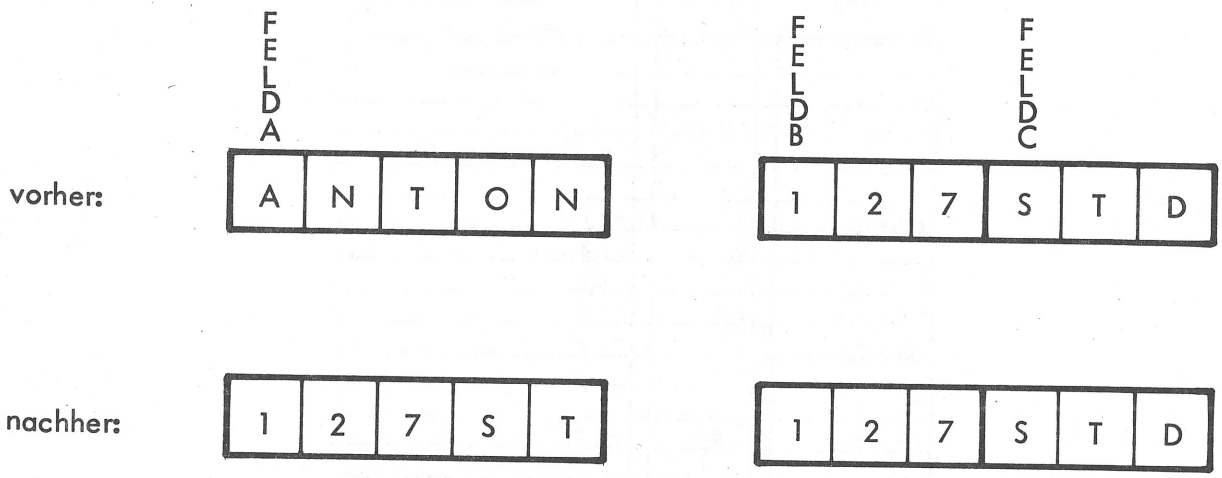
4.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,C, ,	F,E,L,D,A,,,F,E,L,D,C, , , , ,



5.





Berichtigen Sie gegebenfalls Ihre Lösungen, damit die Ihnen verbleibenden Unterlagen fehlerfrei sind!

Merkblatt 401 enthält die Beschreibung der Operation "Übertragen Zeichen" (vgl. Abb. 1).



Band 1, Seite 45

Name (1 … 8)	Operation (10 … 14)	Operand und Bemerkungen (20 … 25 … 30)
*, D,E,F,I,N,I,T	I,O,N,E,N	
A,B,E,R,	D,S,	Ø,C,L,1,1,
	D,S,	C,L,6,
A,W,E,R,T,	D,S,	C,L,5,
F,E,L,D,1,	D,S,	C,L,4,
F,E,L,D,2,	D,S,	C,L,5,
F,E,L,D,3,	D,S,	C,L,7,
F,E,L,D,4,	D,S,	C,L,6,
K,O,N,1,	D,C,	C,',K,6,',
K,O,N,2,	D,C,	C,',W,E,R,T,',

Abb. 2

Übungen (Teil 2)

Sie kennen die Technik der Adreßrechnung und der expliziten Längenangabe. Die folgenden Übungen beziehen sich auf die Anwendung dieser Technik für MVC-Instruktionen.

Entnehmen Sie Ihren Arbeitsmitteln Übung 402, Seite 1, 2 und 3.

Tragen Sie Ihre Lösungen in die Übungsblätter ein. Verlangt sind entweder das Ergebnis der Übertragung oder die Codierung der Instruktion.

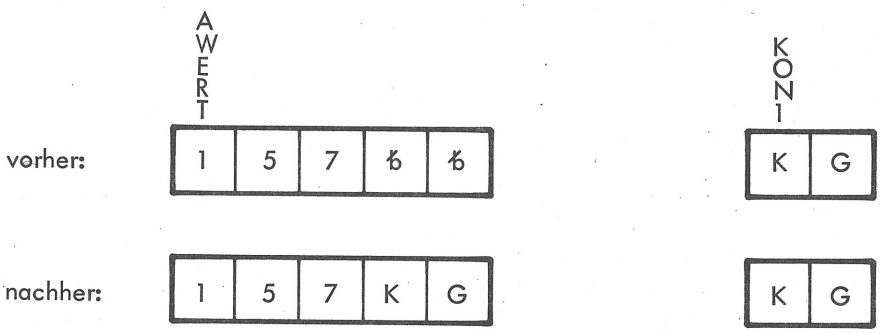
Abb. 2 enthält die Definitionen der angesprochenen Speicherfelder.

Vergleichen Sie Ihre Eintragungen mit den Lösungen auf den folgenden Seiten, sobald Sie ein Blatt vollständig bearbeitet haben.

Lösungen zu den Übungen (Teil 2)

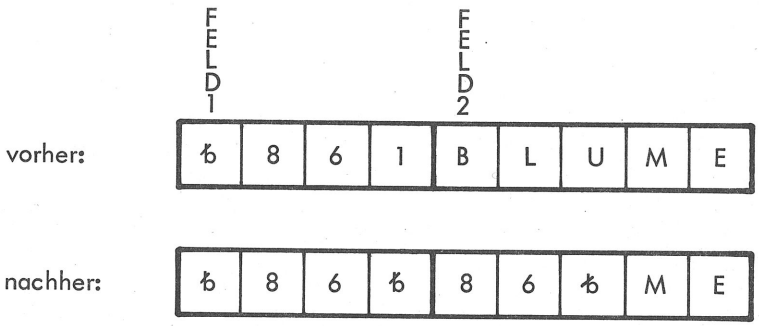
1.

Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	M,V,C, ,	A,W,E,R,T,+,3,(,2,),,K,O,N,1, ,

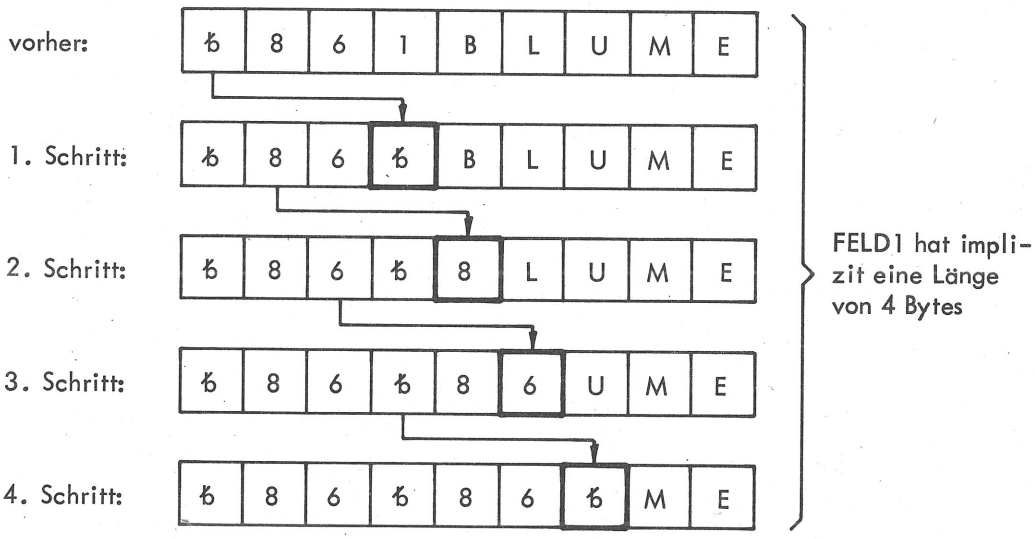


2.

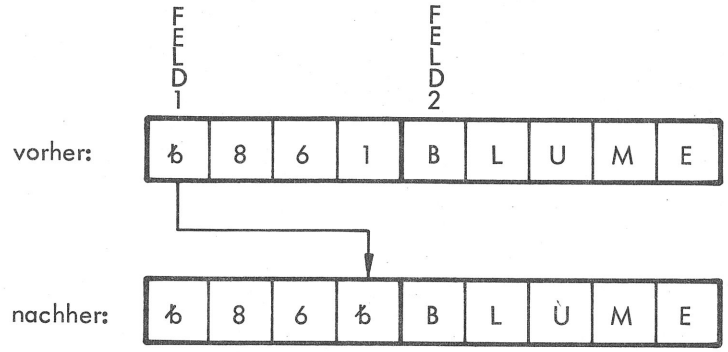
Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	M,V,C, ,	F,E,L,D,1,+,3,,F,E,L,D,1, , , ,



Im Einzelnen:



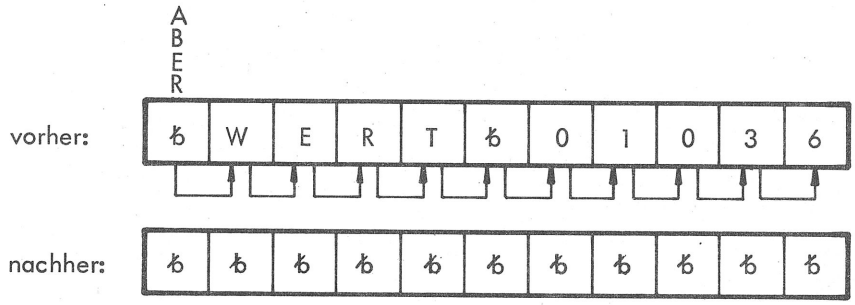
3.



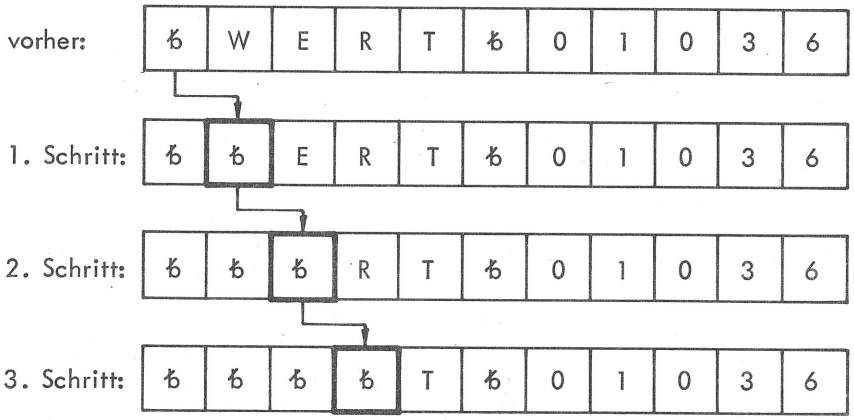
Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,C, ,	F,E,L,D,1,+,3,(,1,),,,F,E,L,D,1,
oder:		
	M,V,C, ,	F,E,L,D,2,-,1,(,1,),,,F,E,L,D,1,

4.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,C, ,	A,B,E,R,+,1,(,1,Ø,),,,A,B,E,R, ,

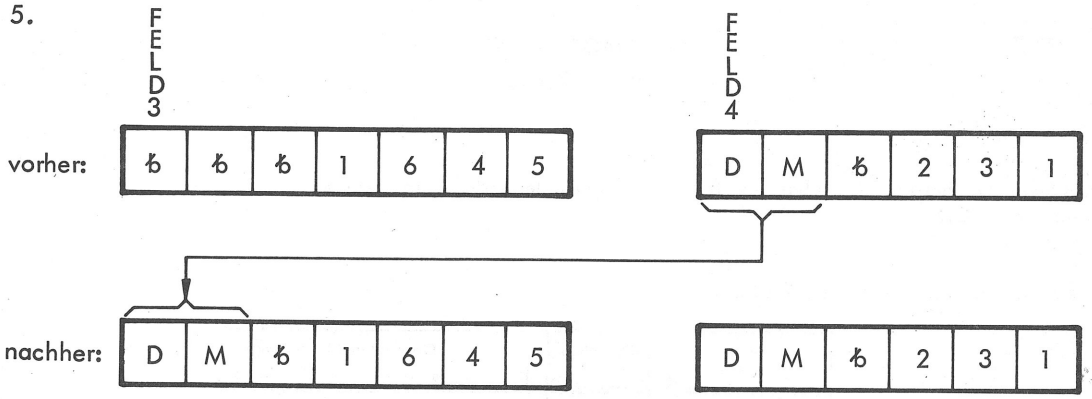


Im Einzelnen:



usw., insgesamt 10 Übertragungsschritte

5.



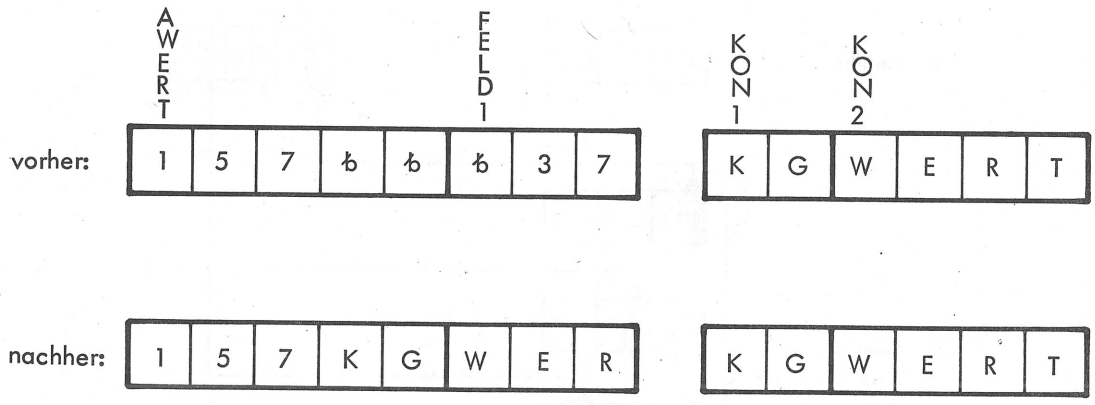
Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	M,V,C, ,	F,E,L,D,3,(,2,),,F,E,L,D,4, ,

Anmerkung zur Lösung:

Die Länge der Operation richtet sich nach der Länge des ersten Operanden. Der zweite Operand darf keine explizite Längenangabe haben.

6.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	M,V,C, ,	A,W,E,R,T,+,3,,K,O,N,1, , , ,





Band 1, Seite 109

MVZ

ÜBERTRAGEN ZONEN
(MoVe Zones)

Diese Operation überträgt den Inhalt von linken Halbbytes. Im übrigen arbeitet sie genauso wie MVC.

B e i s p i e l e i n e r I n s t r u k t i o n:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	M,V,Z,	W,E,R,T,A,, W,E,R,T,B, ...

W i r k u n g s w e i s e:

Der Inhalt der l i n k e n H a l b b y t e s des zweiten Operanden (im Beispiel WERTB) wird in die linken Halbbytes des ersten Operanden (im Beispiel WERTA) übertragen.

Der bisherige Inhalt der linken Halbbytes des ersten Operanden wird zerstört.

Verarbeitet wird b y t e w e i s e von l i n k s nach r e c h t s in der Länge des e r s t e n Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die m a x i m a l e Länge der Operanden ist 256 Bytes.

D u r c h f ü h r u n g d e r O p e r a t i o n:

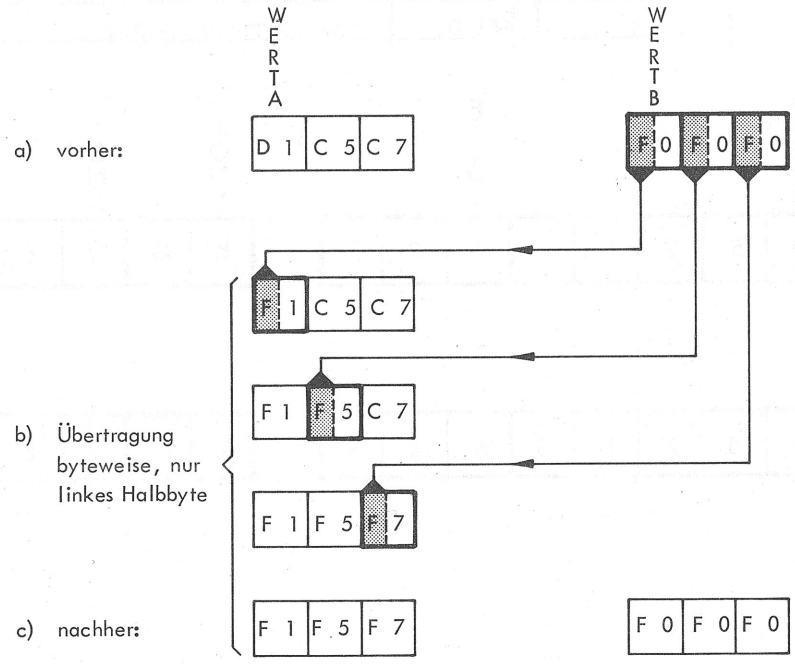


Abb. 1

1 In Abb. 1 ist die Operation " (WW) " beschrieben. Der Operationsschlüssel ist aus der englischen Bezeichnung "Move Zones" abgeleitet und heißt _____.

Übertragen Zonen

MVZ

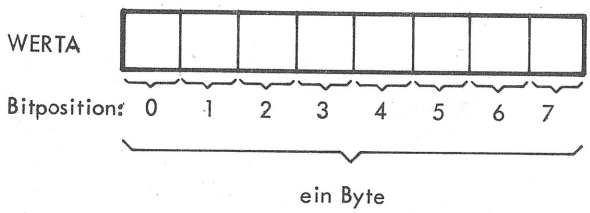
2 Es wird byteweise übertragen.
Die Übertragungs-Operation beginnt im (linken, rechten) _____ Byte der Felder.

Ist der erste Operand 3 Bytes groß, so wird die Übertragung in (Z) Schritten durchgeführt.

linken

3

3



Hier ist das linke Byte des Feldes WERTA dargestellt.

Der erste Übertragungsschritt verändert den Inhalt der Bitpositionen (ZZ) in diesem Byte.

0, 1, 2 und 3

4 Die rechten 4 Bits bleiben bei allen angesprochenen Bytes unverändert.

Neuen Inhalt erhalten bei der Instruktion "Übertragen Zonen" die (linken, rechten) _____ Halbbytes oder _____-teile des ersten Operanden. Unverändert dagegen bleiben die _____ Halbbytes oder _____-teile.

linken

Zonen

rechten

Ziffern

MVZ

ÜBERTRAGEN ZONEN
(MoVe Zones)

Diese Operation überträgt den Inhalt von linken Halbbytes. Im übrigen arbeitet sie genauso wie MVC.

B e i s p i e l e i n e r I n s t r u k t i o n:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	M,V,Z, .	W,E,R,T,A,,W,E,R,T,B, ...

W i r k u n g s w e i s e:

Der Inhalt der l i n k e n H a l b b y t e s des zweiten Operanden (im Beispiel WERTB) wird in die linken Halbbytes des ersten Operanden (im Beispiel WERTA) übertragen.

Der bisherige Inhalt der linken Halbbytes des ersten Operanden wird zerstört.

Verarbeitet wird b y t e w e i s e von l i n k s nach r e c h t s in der Länge des e r s t e n Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die m a x i m a l e Länge der Operanden ist 256 Bytes.

D u r c h f ü h r u n g d e r O p e r a t i o n:

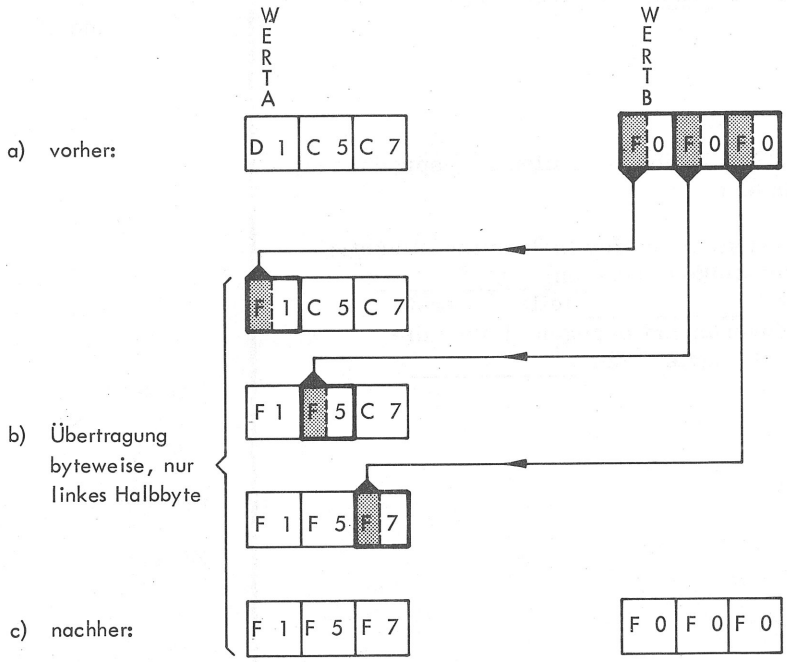


Abb. 1

5 Es kann vorkommen, daß die Operanden verschieden groß sind.

Die Länge der Übertragungsoperation richtet sich nach der Länge des ___(WW)___.

ersten Operanden

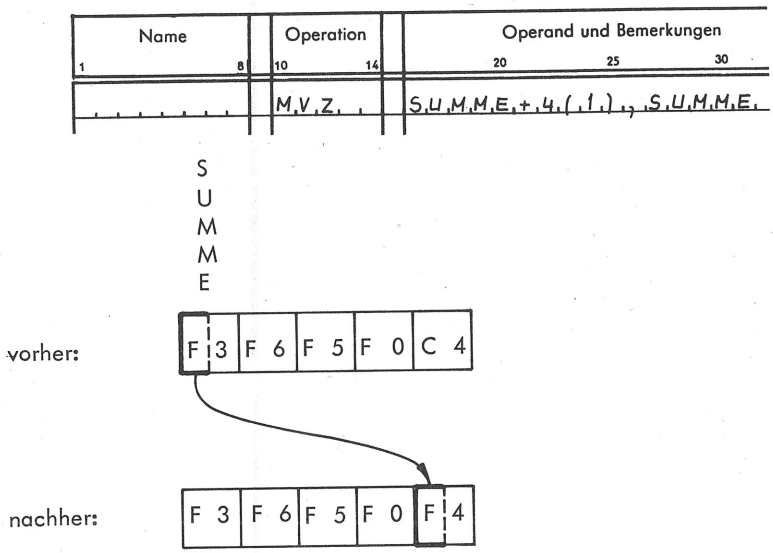

Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
M,V,Z, ,
S,U,M,M,E,+,4,(,1,),,S,U,M,M,E,
S
U
M
M
E
vorher:
F 3 F 6 F 5 F 0 C 4
nachher:
F 3 F 6 F 5 F 0 F 4

Abb. 2

6 Das Beispiel in Abb. 2 zeigt eine häufige Anwendung für den Befehl "Übertragen Zonen".
In diesem Befehl hat der erste Operand die Adresse _____ und der zweite Operand die Adresse _____.

SUMME+4

SUMME

7 Das Feld SUMME hat eine implizite Länge von 5 Bytes.
Für den ersten Operanden wird diese Länge durch eine explizite Längenangabe auf __(Z)__ Byte(s) beschränkt.

Es werden also die Zonenverschlüsselungen von (1, 5) _____ Byte(s) übertragen.

1

1

8 Aus dem Byte mit der Adresse "SUMME" wird die Hexadezimalziffer F in das Byte mit der Adresse "SUMME+4" übertragen.

"F" ersetzt die Hexadezimalziffer "__(Z)__" im Zonenteil dieses Bytes.

C

9 Wird nach dieser Übertragung die 5-stellige Summe in eine Karte gestanzt, so erhält die Einerstelle (eine 12er-Zone, keine Zonenlochung) _____.

keine Zonenlochung

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
*, D,E,F,I ,N,I	T I ,O,N,E,N	
W,E,R,T,A, . . .	D,S, . . .	C,L,1, ...
W,E,R,T,B, . . .	D,S, . . .	C,L,1, ...
W,E,R,T,C, . . .	D,S, . . .	C,L,6, ...
K,O,N,1, ...	D,C, . . .	C,',S,U,M,M,E,', ...
W,E,R,T,D, . . .	D,S, . . .	C,L,4, ...
W,E,R,T,E, . . .	D,S, . . .	C,L,5, ...

Abb. 3

Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 403, Seite 1 und 2.

Tragen Sie Ihre Lösungen ein. Verlangt sind entweder das Übertragungsergebnis oder die Übertragungs-Instruktion.

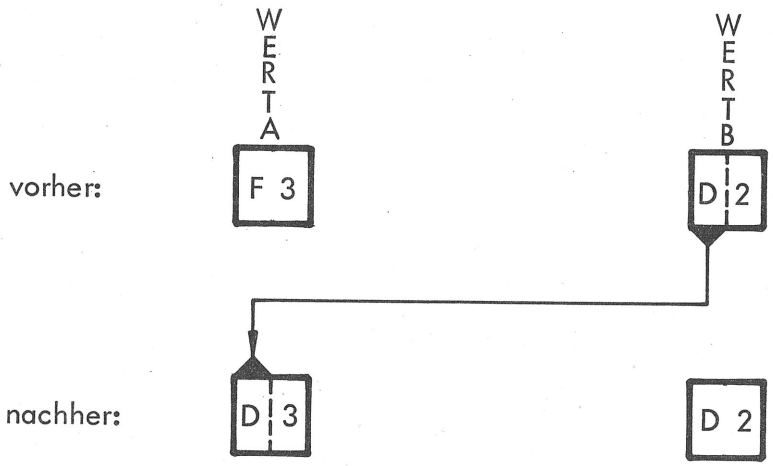
Die Definitionen der angesprochenen Speicherfelder finden Sie in Abb. 3.

Vergleichen Sie Ihre Eintragungen mit den Lösungen auf den folgenden Seiten, sobald Sie ein Blatt vollständig bearbeitet haben.

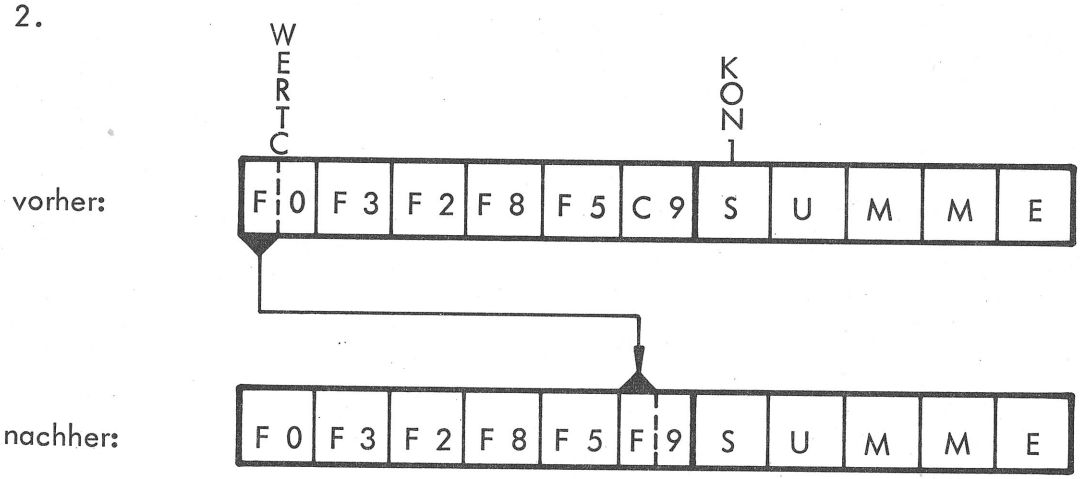
Lösungen zu den Übungen

1.

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	M,V,Z,	W,E,R,T,A,,,W,E,R,T,B,



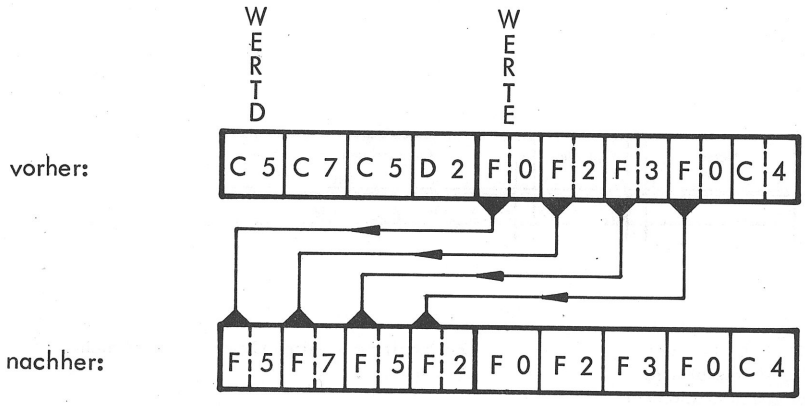
2.



Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	M,V,Z,	W,E,R,T,C,+,5,(,1,),,,W,E,R,T,C,
oder		
	M,V,Z,	K,O,N,1,-,1,(,1,),,,W,E,R,T,C,

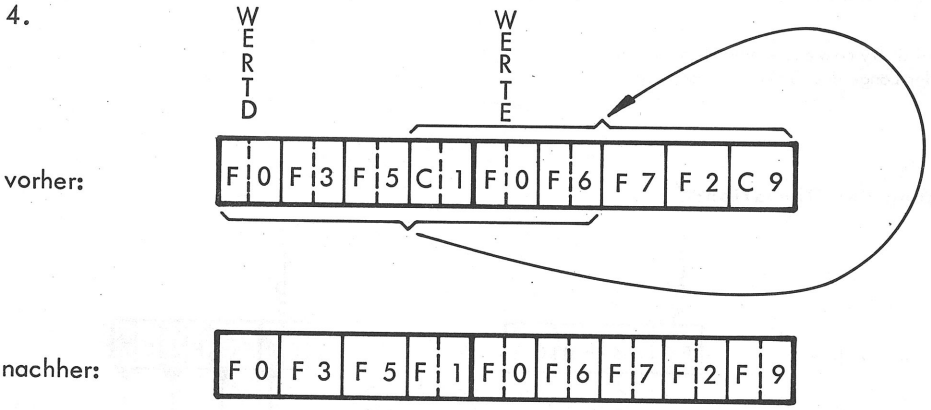
3.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,Z,	W,E,R,T,D,,,W,E,R,T,E,



Anmerkung zur Lösung:
WERTD hat implizit eine Länge von 4 Bytes.

4.



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,Z,	W,E,R,T,D,+,3,(,6,),,W,E,R,T,D,
oder		
	M,V,Z,	W,E,R,T,E,-,1,(,6,),,W,E,R,T,D,

Tragen Sie ein:

a) den Inhalt der 6 rechten Bytes nach dem Übertragen der linken Halbbytes,

b) die MVZ-Instruktion.

Merkblatt 402 enthält die Beschreibung der Operation "Übertragen Zonen" (vgl. Abb. 1).

MVN

ÜBERTRAGEN ZIFFERN

(MoVe Numerics)

Diese Operation überträgt den Inhalt von rechten Halbbytes. Im übrigen arbeitet sie wie MVC.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVN	FELDA,FELDB

Wirkungsweise:

Der Inhalt der rechten Halbbytes des zweiten Operanden (im Beispiel FELDB) wird in die rechten Halbbytes des ersten Operanden (im Beispiel FELDA) übertragen.

Der bisherige Inhalt der rechten Halbbytes des ersten Operanden wird zerstört.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die maximale Länge der Operanden ist 256 Bytes.

Durchführung der Operation:

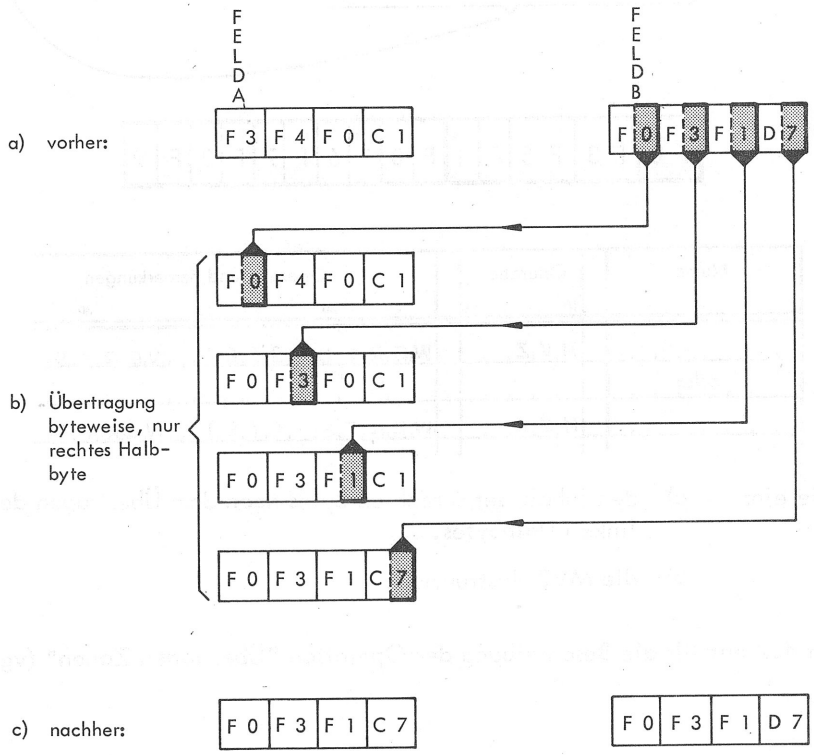


Abb. 1

1 Sie lernen nun eine dritte Übertragungsoperation kennen. Sie heißt "____(WW)____".

Nach der englischen Bezeichnung "Move Numerics" hat sie den Operationsschlüssel "_____".

Übertragen Ziffern

MVN

2 Vergleichen Sie die Operation "Übertragen Ziffern" mit der Operation "Übertragen Zonen".

Formulieren Sie bitte den Unterschied der beiden Befehle in einem Satz.

Sinngemäß:

Mit dem Befehl "Übertragen Zonen" wird der Inhalt von linken Halbbytes übertragen, während mit dem Befehl "Übertragen Ziffern" der Inhalt von rechten Halbbytes übertragen wird.

3 Das rechte Halbbyte hat auch die Bezeichnung "Ziffernteil".

Die Operation heißt "Übertragen Ziffern", weil der Inhalt von _____ übertragen wird.

Ziffernteilen

4 Die Operationen MVC, MVZ und MVN stimmen in der Wirkungsweise vollständig überein bis auf die folgenden Unterschiede:

Die Operation
"_____" überträgt linke Halbbytes,
"_____" überträgt rechte Halbbytes und
"_____" überträgt ganze Bytes.

Übertragen Zonen/MVZ

Übertragen Ziffern/MVN

Übertragen Zeichen/MVC

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
*, ,D,E,F,I,N,I	T	I,O,N,E,N
W,E,R,T,A,		D,S, C,L,1,
W,E,R,T,B,		D,S, C,L,2,
W,E,R,T,C,		D,S, C,L,3,
W,E,R,T,D,		D,S, C,L,1,
W,E,R,T,E,		D,S, C,L,3,

Abb. 2

Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 404, Seite 1 und 2.

Tragen Sie Ihre Lösungen ein.

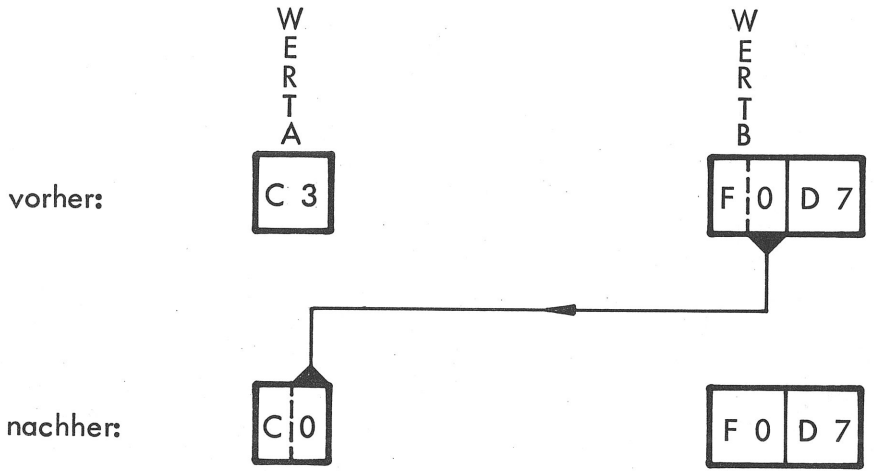
Die Definitionen der angesprochenen Speicherfelder finden Sie in Abb. 2.

Bearbeiten Sie erst ein Blatt vollständig, bevor Sie Ihre Eintragungen mit den Lösungen auf den folgenden Seiten vergleichen.

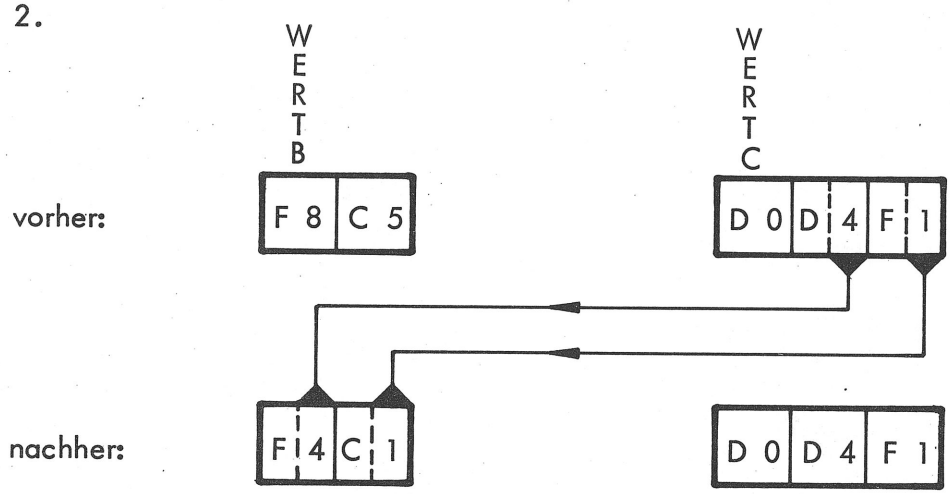
Lösungen zu den Übungen

1.

Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	MVN	WERTA,WERTB

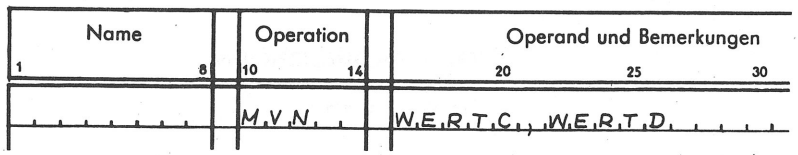


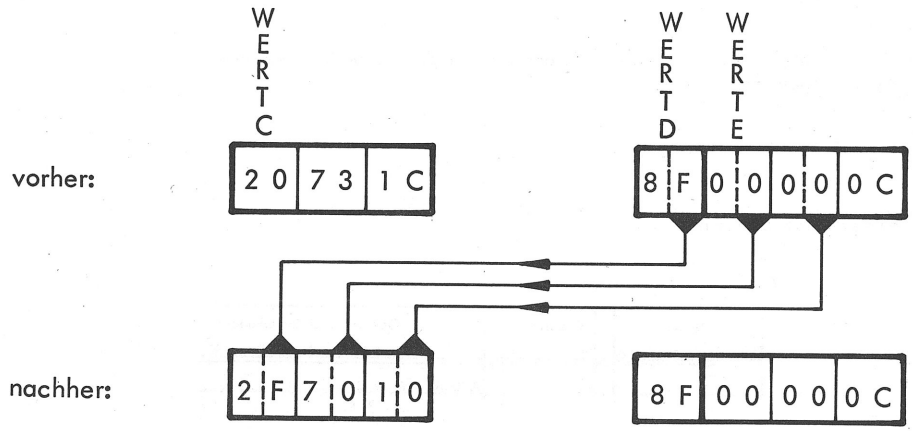
2.



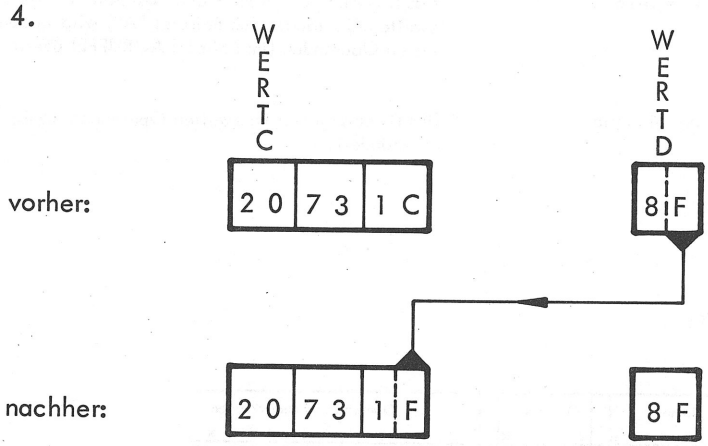
Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	MVN	WERTB,WERTC+1

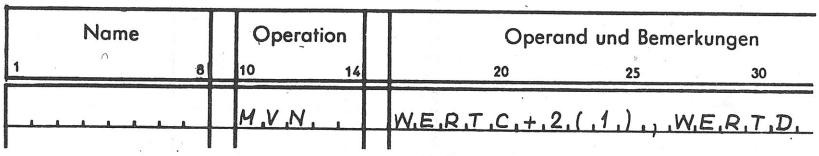
3.





4.





Merkblatt 403 enthält die Beschreibung der Operation "Übertragen Ziffern" (vgl. Abb. 1).

Band 1, Seite 139

MVI

ÜBERTRAGEN DIREKTZEICHEN

(MoVe Immediate)

Diese Operation überträgt ein einzelnes Zeichen, das direkt in der Instruktion angegeben wird, in ein Byte im Speicher.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVI	AGRUPPE,C'A'

Wirkungsweise:

Der zweite Operand ist ein Direktoperand (im Beispiel C'A').

Das Direktzeichen - auch Direktdate - des zweiten Operanden (im Beispiel "A") wird in den ersten Operanden (im Beispiel AGRUPPE) übertragen.

Die Operation bezieht sich immer auf ein Byte.

Das Direktzeichen im zweiten Operanden bleibt unverändert.

Durchführung der Operation:

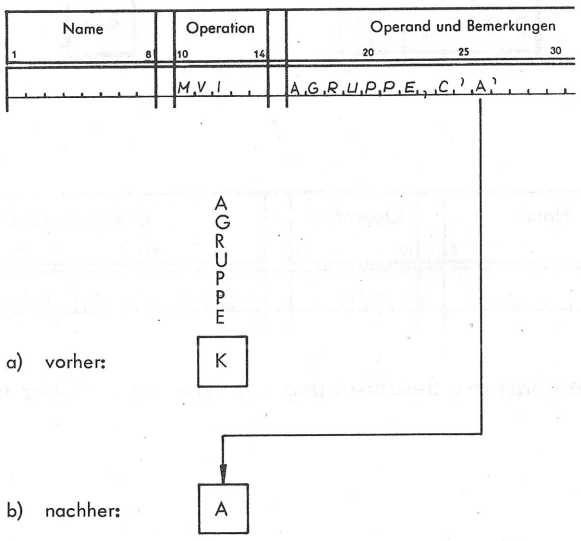


Abb. 1

1 Die Übertragungsoperation, die wir hier besprechen, heißt "___(WW)___".

Der Operationsschlüssel _____ ist abgeleitet von der englischen Bezeichnung "Move Immediate".

Übertragen Direktzeichen

MVI

2 Charakteristisch für diesen Übertragungsbefehl ist der 2. Operand.

Bei den anderen Übertragungsoperationen enthält die Instruktion die symbolischen Adressen der beiden Operanden.

In der Instruktion mit dem Operationsschlüssel MVI ist nur der (erste, zweite) _____ Operand durch seine symbolische Adresse angegeben.

erste

3 Für den zweiten Operanden einer Operation "Übertragen Direktzeichen" ist eine Adreßangabe nicht nötig.

Er steht d i r e k t in der Instruktion. Darum heißt er "_____".

Direktoperand

4 Der Direktoperand enthält ein konstantes Zeichen, das sogenannte _____.

Direktzeichen

5 Jedes beliebige Zeichen kann als Direktzeichen eingesetzt werden.

Genauer gesagt: als Direktzeichen kann jede Bitkombination in (einem Bit, einem Halbbyte, einem Byte, mehreren Bytes) _____ verwendet werden.

einem Byte

MVI

ÜBERTRAGEN DIREKTZEICHEN

(MoVe Immediate)

Diese Operation überträgt ein einzelnes Zeichen, das direkt in der Instruktion angegeben wird, in ein Byte im Speicher.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVI	AGRUPPE,C'A'

Wirkungsweise:

Der zweite Operand ist ein Direktoperand (im Beispiel C'A').

Das Direktzeichen - auch Direktdate - des zweiten Operanden (im Beispiel "A") wird in den ersten Operanden (im Beispiel AGRUPPE) übertragen.

Die Operation bezieht sich immer auf ein Byte.

Das Direktzeichen im zweiten Operanden bleibt unverändert.

Durchführung der Operation:

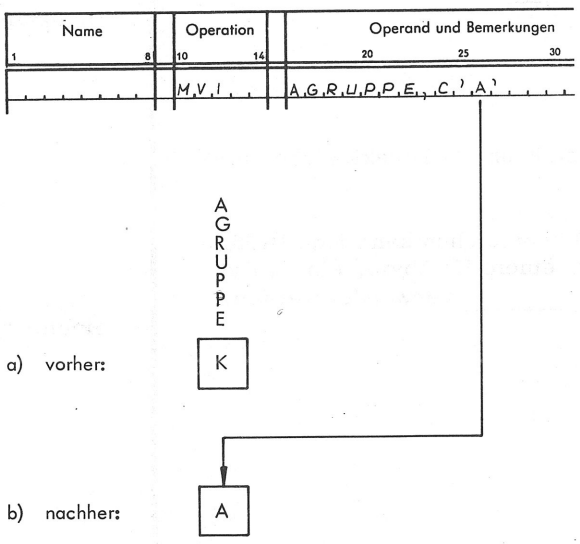


Abb. 1

6 Eine MVI-Instruktion kann z.B. die Bit-Kombination 1100 0001 im ersten Operanden erzeugen.
Betrachten Sie die folgende Tabelle.

Byte-Inhalt	A	
Halbbyte-Inhalt	C	1
Bit-Kombination	1 1 0 0	0 0 0 1

Der Direktoperand der MVI-Instruktion in Abb. 1 könnte auch mit Hilfe des Konstantentyps X gebildet werden.
Er hätte dann die Form _____.

X' C1'

7 Als Direktzeichen kann auch eine gepackte Dezimalzahl auftreten.

In diesem Fall enthält das linke Halbbyte eine (Ziffernverschlüsselung, Zonenverschlüsselung)
_____ und das rechte Halbbyte eine
_____.

Ziffernverschlüsselung

Zonenverschlüsselung (Vorzeichen)

8 Eine gepackte Dezimalzahl in einem Byte hat (1, 2)
_____ Ziffernstelle(n).

1

9 Wir nehmen an, eine gepackte positive "8" tritt als Direktzeichen auf.
Als Zonenverschlüsselung soll die Hexadezimalziffer C dienen. Als Konstantentyp können Sie P und X verwenden.

Wie heißt der Direktoperand

a) beim Konstantentyp P

b) beim Konstantentyp X?

P'+8' /P' 8'

X' 8C'

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
* D.E.F.I.N.I.T.I.O.N.E.N		
Z.A.E.H.L.E.R.	D.S.	C.L.1.
A.N.U.M.M.E.R.	D.S.	C.L.3.
F.E.L.D.1.	D.S.	C.L.7.

Abb. 2

Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 405, Seite 1 und 2.

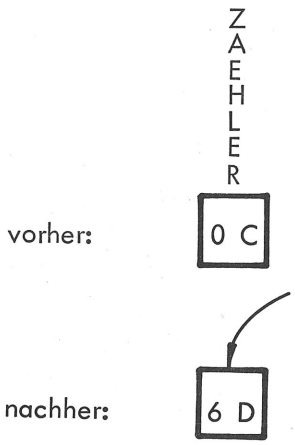
Tragen Sie Ihre Lösungen ein.

Die Definitionen der angesprochenen Speicherfelder finden Sie in Abb. 2.

Bearbeiten Sie erst ein Blatt vollständig, bevor Sie Ihre Eintragungen mit den Lösungen auf den folgenden Seiten vergleichen.

Lösungen zu den Übungen

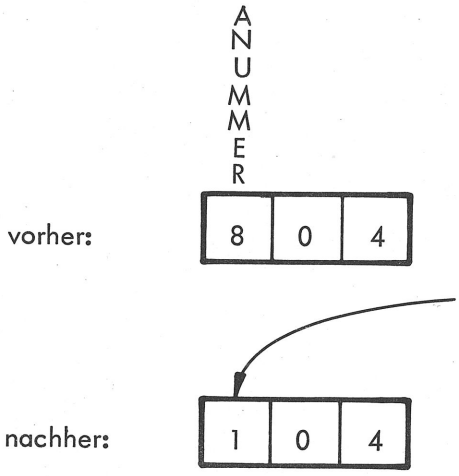
1.



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVI	ZAEHLER,X'6D'
oder		
	MVI	ZAEHLER,P'-6'

2.

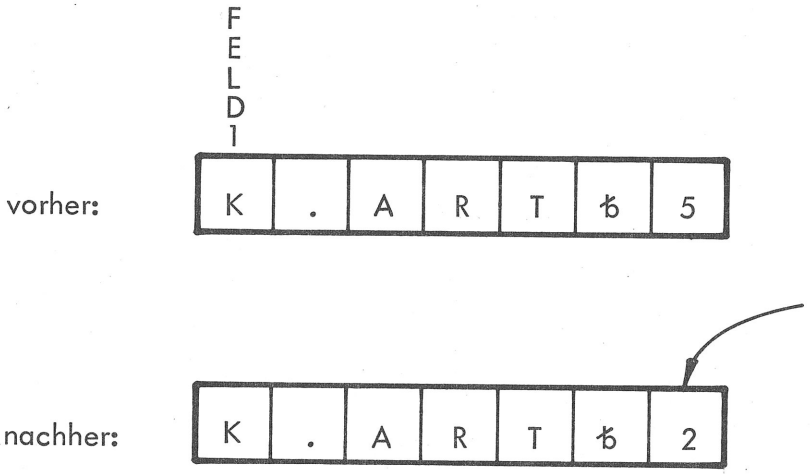
Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVI	ANUMMER,C'1'



Anmerkung zur Lösung:

Der erste Operand wird immer ohne explizite Längenangabe geschrieben, da die Operation "Übertragen Direktzeichen" immer nur 1 Byte anspricht.

3.



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVI	F,E,L,D,1,+,6,,C,',2,'
oder		
	MVI	F,E,L,D,1,+,6,,X,',F2,'

Merkblatt 404 enthält die Beschreibung der Operation "Übertragen Direktzeichen" (vgl. Abb. 1).

FRAGEN ZUM KAPITEL I

Überprüfen Sie anhand der folgenden Fragen Ihre Kenntnisse bezüglich der Übertragungsoperationen.
Schreiben Sie die Lösungen auf ein Extrablatt bzw. auf die dafür vorgesehenen Übungsblätter.

Einige der folgenden Fragen enthalten Auswahlantworten. Entscheiden Sie sich für eine dieser Antworten und notieren Sie sich den entsprechenden Buchstaben.

Vergleichen Sie Ihre Eintragungen mit den Lösungen auf den folgenden Seiten, sobald Sie alle Fragen beantwortet haben.

1. Mit den Operationsschlüsseln MVC, MVN und MVZ wird byteweise übertragen.

   A Die Übertragung beginnt bei den 3 Operationen in den rechten Bytes der beiden Felder.

   B Die Übertragung beginnt bei den 3 Operationen in den linken Bytes der beiden Felder.

   C Es gilt teils A und teils B.

2. Jede Übertragungsoperation hat 2 Operanden.

   A Der erste Operand gibt an, WOHIN die Daten übertragen werden.

   B Der erste Operand gibt an, WOHER die Daten kommen.

3. A Bei einer Übertragungsoperation wird der Inhalt des ersten Operanden verändert.

   B Bei einer Übertragungsoperation wird der Inhalt des zweiten Operanden verändert.

   C Bei einer Übertragungsoperation wird der Inhalt beider Operanden verändert.

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
* DEFINITIONEN		
FELD	DS	CL4
ART	DS	CL1
KONS	DC	X'F5'
ZAEHLER	DS	CL1
SUMME	DS	CL5
KONØ	DC	X'FØFØFØFØFØ'
BFELD	DS	CL5
ZAHL	DS	CL2
EFELD	DS	CL6
FELD1	DS	CL4
FELD2	DS	CL3

Abb. 1

4. Welcher der beiden Operationsschlüssel MVN und MVZ überträgt linke Halbbytes?

5. Wieviel Bytes können bei den einzelnen Übertragungsoperationen

   a) Übertragen Zeichen,

   b) Übertragen Ziffern,

   c) Übertragen Zonen,

   d) Übertragen Direktzeichen

   maximal angesprochen werden?

6. bis 14.

Zur Lösung der nächsten Fragen entnehmen Sie bitte Ihren Arbeitsmitteln Übung 406, Seite 1 bis 4.

Tragen Sie Ihre Lösungen ein.

Die Definitionen der benutzten Speicherfelder finden Sie in Abb. 1.

ANTWORTEN ZUM KAPITEL I

Überprüfen Sie nun Ihre Antworten auf ihre Richtigkeit. Beseitigen Sie erst alle Unklarheiten, ehe Sie fortfahren. Wiederholen Sie gegebenenfalls den in Klammern angegebenen Lernstoff.

1. B (Seiten 8, 30, 42)

2. A (Seiten 8, 30, 42, 50)

3. A (Seiten 8, 30, 42, 50)

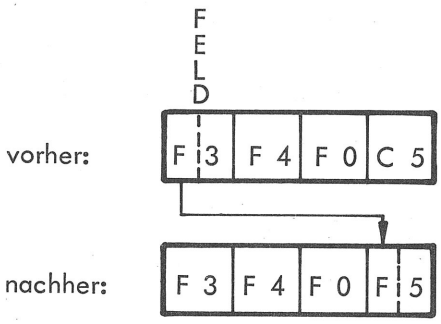
4. MVZ (Seiten 30, 42)

5. a) 256 Bytes (Seite 8)
   b) 256 Bytes (Seite 30)
   c) 256 Bytes (Seite 42)
   d) 1 Byte (Seite 50)

6. bis 14. auf den folgenden Seiten

6.

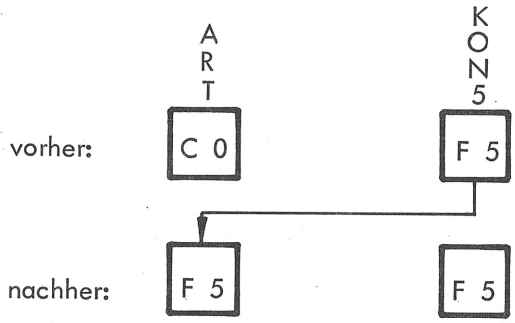
Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVZ	FELD+3(1),FELD



(Seite 35, LE 6 - 9)

7.

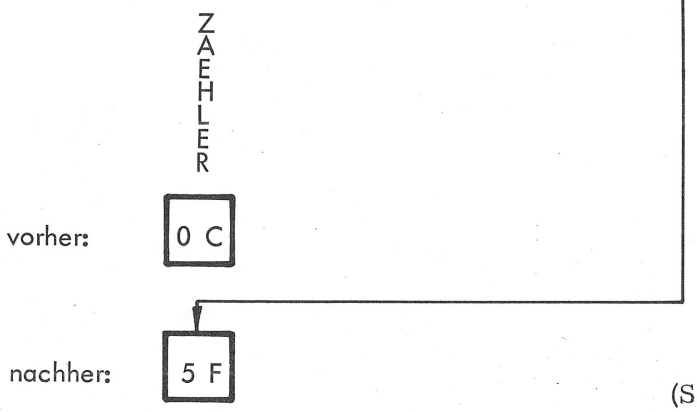
Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVC	ART,KON5



(Seite 8)

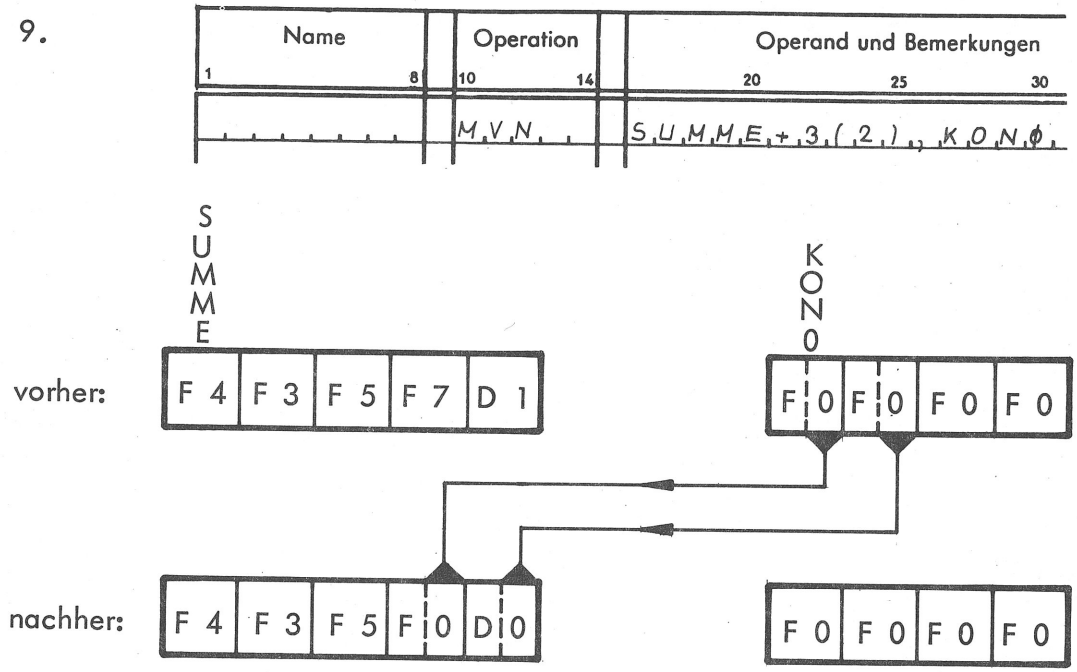
8.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVI	Z(AEHLER),X'5F'



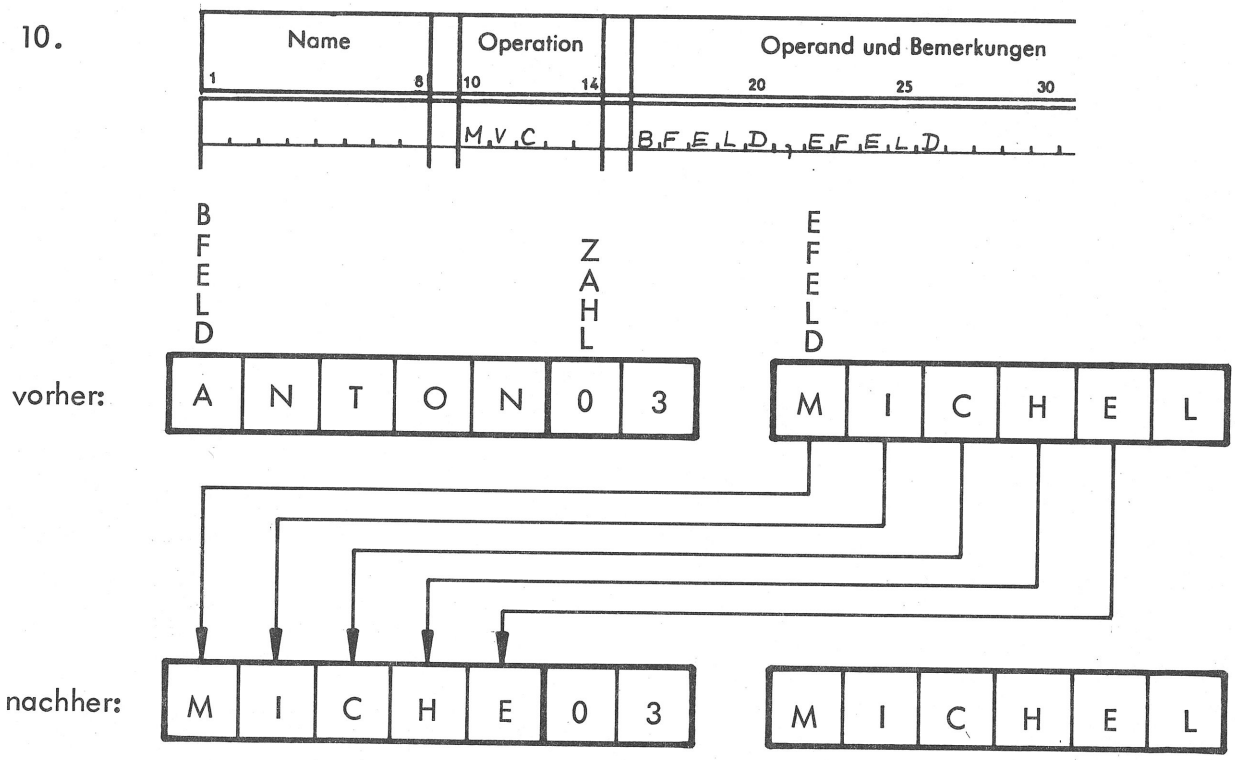
(Seite 53, LE 6 - 9)

9.



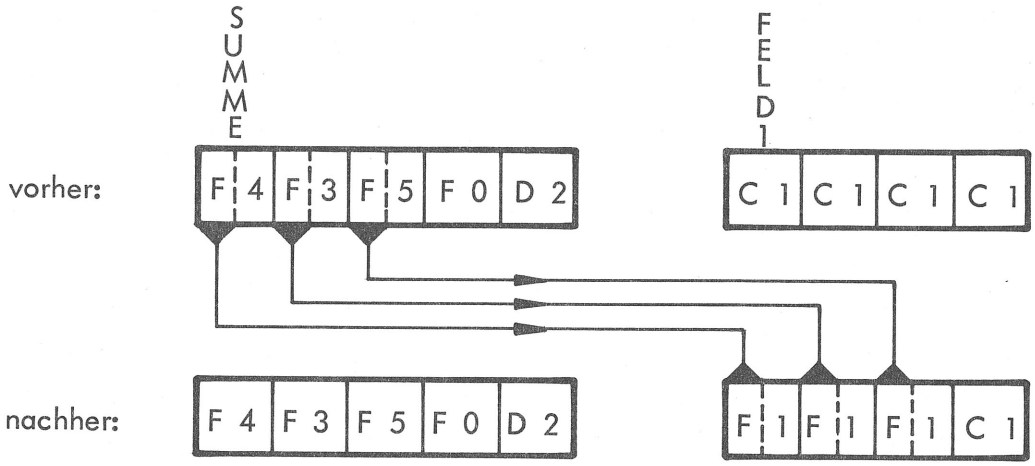
(Seite 43, LE 1 - 4)

10.



(Seite 11, LE 8 - 9)

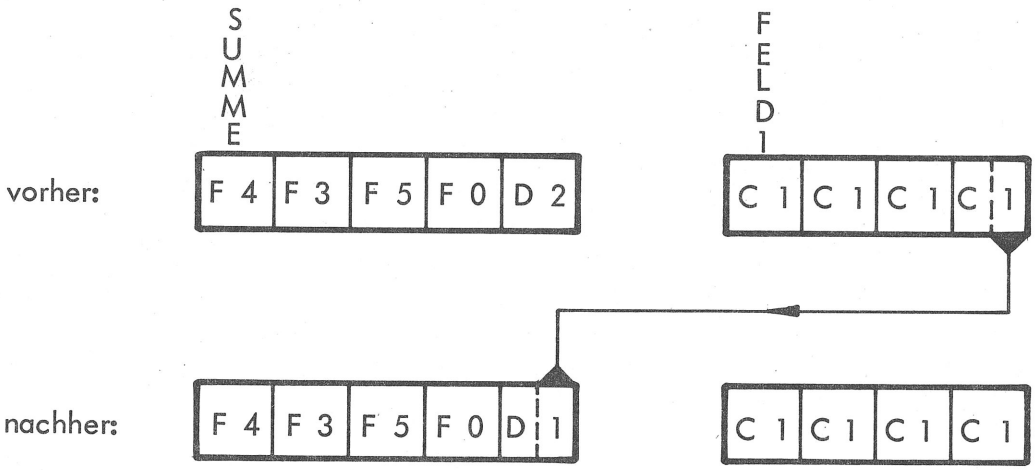
11.



(Seite 30 ff.)

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,Z, ,	F,E,L,D,1,(,3,),,S,U,M,M,E, , ,

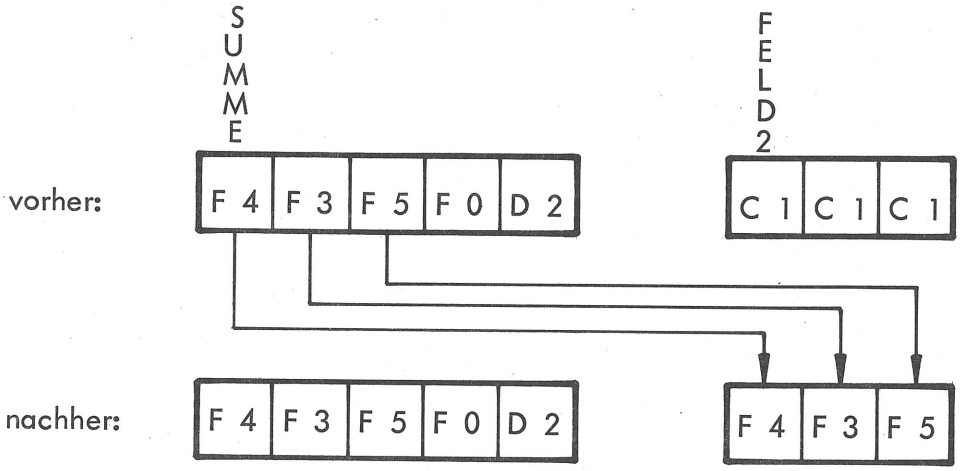
12.



(Seite 42 ff.)

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,N, ,	S,U,M,M,E,+,4,(,1,),,F,E,L,D,1,+,3,

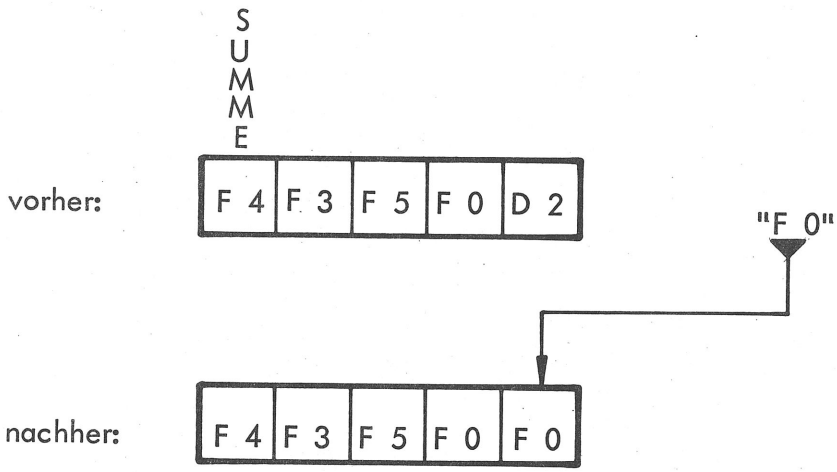
13.



(Seite 8 ff.)

Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	MVC	FELD2,SUMME

14.



(Seite 50 ff.)

Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	MVI	SUMME+4,X'F0'

Sie haben jetzt 4 verschiedene Übertragungsoperationen kennengelernt.
Die wesentlichen Eigenschaften dieser Operationen sind in Merkblatt 405 zusammengestellt.

Entnehmen Sie Ihren Arbeitsmitteln dieses Merkblatt und sehen Sie es sich an.



Seite 98 (CLI)

KAPITEL II

VERGLEICHS- UND VERZWEIGUNGSOPERATIONEN

Lernstoff:

CLC - Vergleichen Logisch Zeichen

CLI - Vergleichen Logisch Direktzeichen

Verzweigungsoperationen (bedingte und unbedingte)

Das Ergebnis einer Vergleichsoperation kann "größer", "kleiner" oder "gleich" lauten. Das Ergebnis einer Rechenoperation kann "positiv", "negativ" oder "null" sein. Das jeweilige Ergebnis einer solchen Operation steht für eine eventuelle Abfrage zur Verfügung.

In der Zentraleinheit gibt es hierfür einen "Anzeiger", den Bedingungsschlüssel. Die Anzeige oder Stellung dieses Bedingungsschlüssels bezieht sich immer auf die letzte Operation, die eine Änderung des Bedingungsschlüssels hervorgerufen hat.

Sie lernen nun eine der Operationen kennen, die den Bedingungsschlüssel beeinflussen.

CLC

VERGLEICHEN LOGISCH ZEICHEN

(Compare Logical Characters)

Diese Operation vergleicht den Inhalt zweier Speicherfelder und zeigt das Ergebnis im Bedingungsschlüssel an.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	CLC	A,NR,,E,NR,

Wirkungsweise:

Der Inhalt des ersten Operanden (im Beispiel ANR) wird mit dem Inhalt des zweiten Operanden (im Beispiel ENR) verglichen.

Das Vergleichsergebnis wird im Bedingungsschlüssel angezeigt. Es bezieht sich immer auf den ersten Operanden.

Folgende Vergleichsergebnisse sind möglich:

gleich	(erster Operand gleich zweiter Operand)
kleiner/niedrig	(erster Operand kleiner als zweiter Operand)
größer/hoch	(erster Operand größer als zweiter Operand)

Verarbeitet wird Bit für Bit von links nach rechts in der Länge des ersten Operanden.

Der Vergleich endet bei der ersten Ungleichheit.

Jede der 256 möglichen Bit-Kombinationen in einem Byte ist erlaubt.

Der Inhalt beider Operanden bleibt unverändert.

Die Operanden können maximal 256 Bytes lang sein.

Durchführung der Operation:

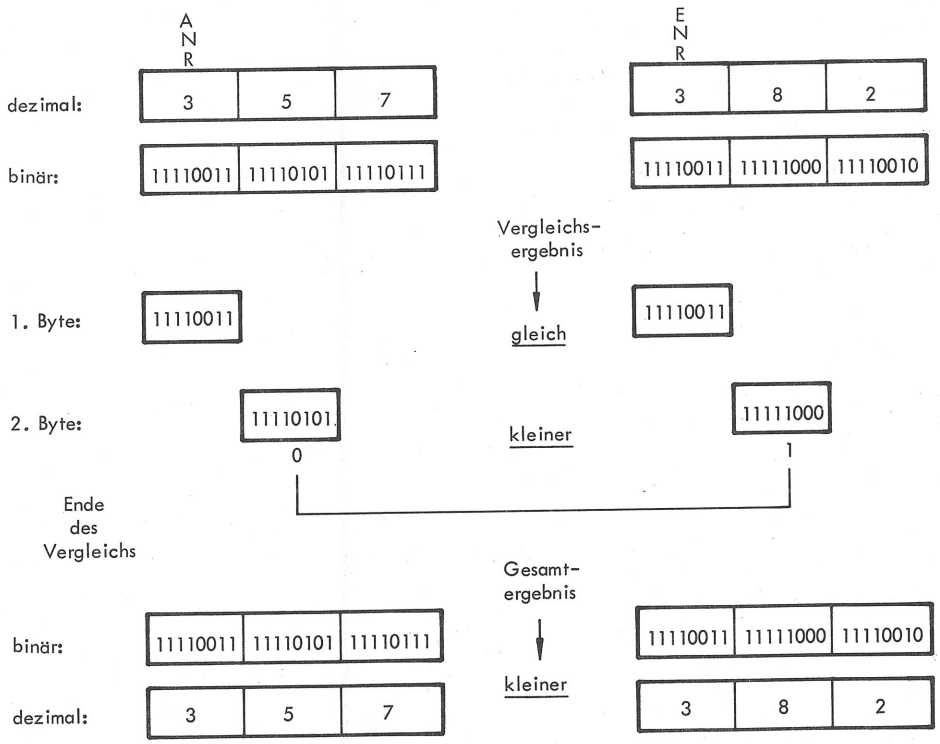


Abb. 1

1 Die Vergleichsoperation, mit der wir uns hier beschäftigen, heißt "__(WW)__".

Sie hat den symbolischen Operationsschlüssel _____.

Vergleichen Logisch Zeichen

CLC

2 Der Vergleich erfolgt von links nach rechts und zwar (bit-, byte-) _____weise.

bit- (weise)

3 Die beiden Operanden werden durch die Vergleichsoperation (verändert, nicht verändert) _____.

nicht verändert

4 Das Ergebnis des Vergleichs wird im _____ festgehalten.

Bedingungsschlüssel

5 Der Bedingungsschlüssel zeige z.B. das Vergleichsergebnis "kleiner" an.

Dann ist der (erste, zweite) _____ Operand kleiner als der (erste, zweite) _____ Operand.

erste

zweite

6 Ist der erste Operand größer als der zweite Operand, so heißt das Vergleichsergebnis _____.

größer/hoch

7 Wir nehmen an, der erste Operand umfasse die Speicherstellen 398 bis 401 und der zweite Operand die Speicherstellen 533 bis 536.

In diesem Fall beginnt die Vergleichsoperation bei den Bytes mit den Adressen __(Z)__ und __(Z)__.

398 (und) 533

CLC

VERGLEICHEN LOGISCH ZEICHEN

(Compare Logical Characters)

Diese Operation vergleicht den Inhalt zweier Speicherfelder und zeigt das Ergebnis im Bedingungsschlüssel an.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	C,L,C, ,	A,N,R,,E,N,R,

Wirkungsweise:

Der Inhalt des ersten Operanden (im Beispiel ANR) wird mit dem Inhalt des zweiten Operanden (im Beispiel ENR) verglichen.

Das Vergleichsergebnis wird im Bedingungsschlüssel angezeigt. Es bezieht sich immer auf den ersten Operanden.

Folgende Vergleichsergebnisse sind möglich:

gleich	(erster Operand gleich zweiter Operand)
kleiner/niedrig	(erster Operand kleiner als zweiter Operand)
größer/hoch	(erster Operand größer als zweiter Operand)

Verarbeitet wird Bit für Bit von links nach rechts in der Länge des ersten Operanden.

Der Vergleich endet bei der ersten Ungleichheit.

Jede der 256 möglichen Bit-Kombinationen in einem Byte ist erlaubt.

Der Inhalt beider Operanden bleibt unverändert.

Die Operanden können maximal 256 Bytes lang sein.

Durchführung der Operation:

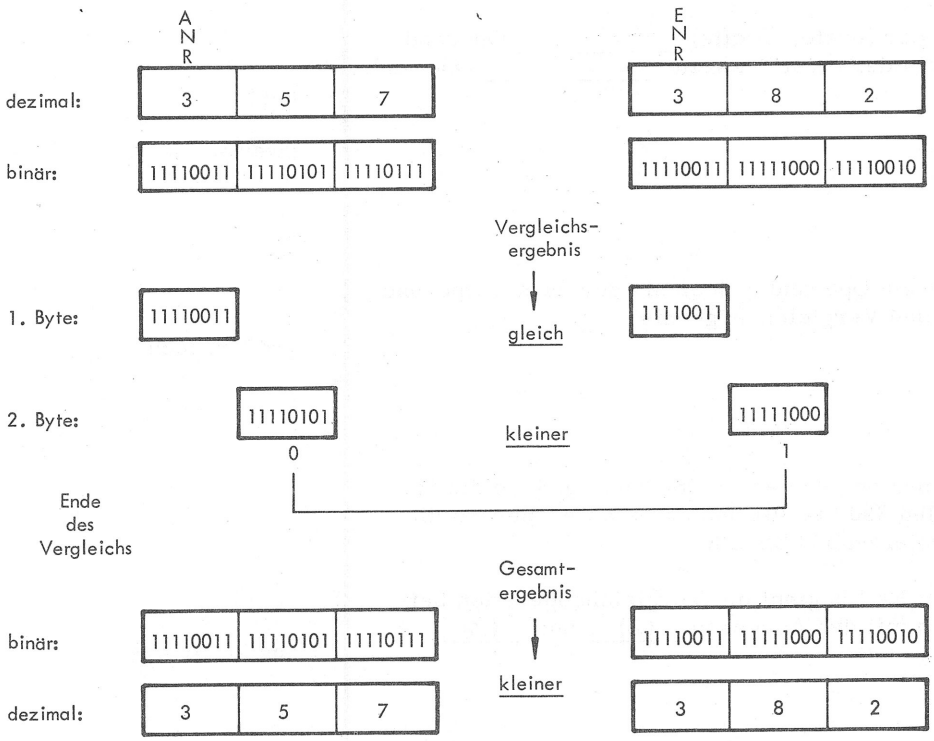
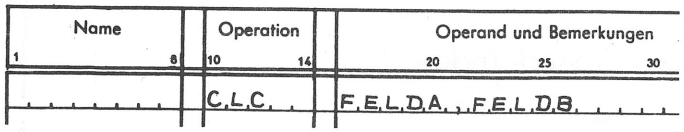


Abb. 1

8





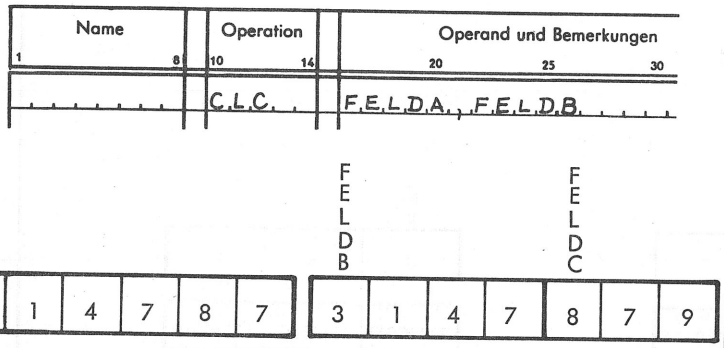
FELDA und FELDB sollen verglichen werden. Die Operation endet beim Vergleich des (ersten, zweiten, dritten, vierten) _____ Bytes von links.

Das Vergleichsergebnis heißt "_____".

zweiten

größer

9



FELDA und FELDB sollen verglichen werden.

Die Länge der Operation "Vergleichen Logisch Zeichen" richtet sich nach der Länge des (ersten, zweiten) _____ Operanden.

Es werden also __(Z)__ Bytes verglichen. Das Vergleichsergebnis heißt "_____".

ersten

6

gleich

CLC

VERGLEICHEN LOGISCH ZEICHEN

(Compare Logical Characters)

Diese Operation vergleicht den Inhalt zweier Speicherfelder und zeigt das Ergebnis im Bedingungsschlüssel an.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 ... 14	20 ... 25 ... 30
	C,L,C,	A,N,R,,E,N,R,

Wirkungsweise:

Der Inhalt des ersten Operanden (im Beispiel ANR) wird mit dem Inhalt des zweiten Operanden (im Beispiel ENR) verglichen.

Das Vergleichsergebnis wird im Bedingungsschlüssel angezeigt. Es bezieht sich immer auf den ersten Operanden.

Folgende Vergleichsergebnisse sind möglich:

gleich	(erster Operand gleich zweiter Operand)
kleiner/niedrig	(erster Operand kleiner als zweiter Operand)
größer/hoch	(erster Operand größer als zweiter Operand)

Verarbeitet wird Bit für Bit von links nach rechts in der Länge des ersten Operanden.

Der Vergleich endet bei der ersten Ungleichheit.

Jede der 256 möglichen Bit-Kombinationen in einem Byte ist erlaubt.

Der Inhalt beider Operanden bleibt unverändert.

Die Operanden können maximal 256 Bytes lang sein.

Durchführung der Operation:

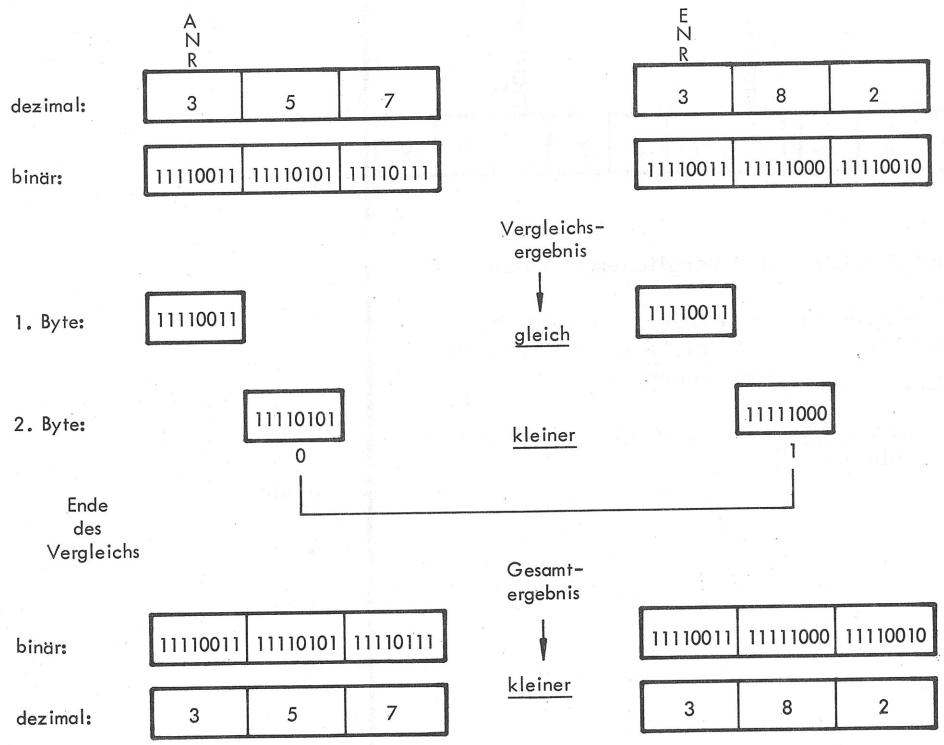


Abb. 1

10 Betrachten Sie den Teil "Durchführung der Operation" in Abb. 1.

Es werden 2 drei Bytes große Felder verglichen. Der Vergleich endet beim (ersten, zweiten, dritten) _____ Byte von links.

zweiten

11 Beim Vergleich der ersten Bytes der beiden Felder ANR und ENR wird (eine, keine) _____ Ungleichheit festgestellt.

In den zweiten Bytes tritt die erste Ungleichheit in Bitposition (0, 1, 2, 3, 4, 5, 6, 7) _____ auf.

keine

4

12 Im Feld ANR enthält diese Bitposition die Binärziffer __(Z)__ und im Feld ENR die Binärziffer __(Z)__.

0

1

13 Die Binärziffer 0 ist kleiner als die Binärziffer 1. Das Vergleichsergebnis heißt also: ANR ist __(WW)__ ENR.

Die weiter rechts stehenden Bits mit niedrigerem Stellenwert können dieses Ergebnis (ändern, nicht mehr ändern) _____.

kleiner als

nicht mehr ändern

14 Der logische Vergleich bringt also das richtige Gesamtergebnis: Die Zahl 357 ist kleiner als die Zahl 382.

Der Bedingungsschlüssel zeigt nach diesem Vergleich "_____" an.

kleiner

CLC

VERGLEICHEN LOGISCH ZEICHEN

(Compare Logical Characters)

Diese Operation vergleicht den Inhalt zweier Speicherfelder und zeigt das Ergebnis im Bedingungsschlüssel an.

Beispiel einer Instruktion:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–25–30)
	CLC	ANR, ENR

Wirkungsweise:

Der Inhalt des ersten Operanden (im Beispiel ANR) wird mit dem Inhalt des zweiten Operanden (im Beispiel ENR) verglichen.

Das Vergleichsergebnis wird im Bedingungsschlüssel angezeigt. Es bezieht sich immer auf den ersten Operanden.

Folgende Vergleichsergebnisse sind möglich:

gleich	(erster Operand gleich zweiter Operand)
kleiner/niedrig	(erster Operand kleiner als zweiter Operand)
größer/hoch	(erster Operand größer als zweiter Operand)

Verarbeitet wird Bit für Bit von links nach rechts in der Länge des ersten Operanden.

Der Vergleich endet bei der ersten Ungleichheit.

Jede der 256 möglichen Bit-Kombinationen in einem Byte ist erlaubt.

Der Inhalt beider Operanden bleibt unverändert.

Die Operanden können maximal 256 Bytes lang sein.

Durchführung der Operation:

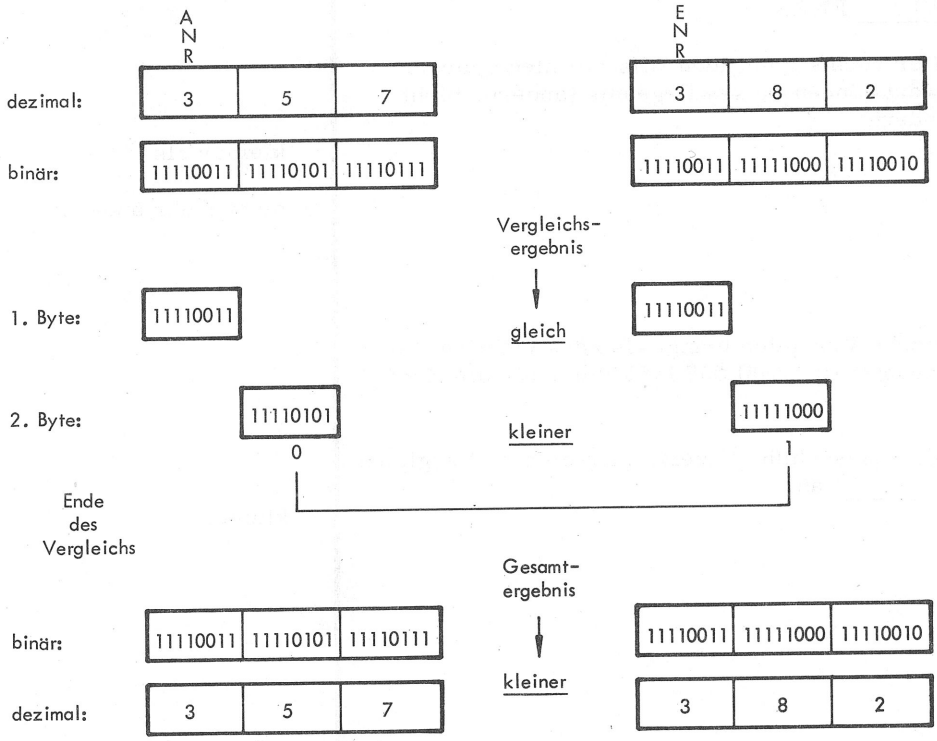


Abb. 1

15 Mit der Operation "Vergleichen Logisch Zeichen" können nicht nur Ziffern verglichen werden.

Jeder beliebige Inhalt in einem Byte kann mit irgendeinem anderen Inhalt in einem anderen Byte verglichen werden.

Das Ergebnis lautet entweder _____ oder _____ oder _____.

gleich

kleiner/niedrig

größer/hoch

16 Bezogen auf ein solches Vergleichsergebnis sagt man: Ein bestimmtes Zeichen, z.B. der Buchstabe M, ist g r ö ß e r oder k l e i n e r als ein anderes Zeichen.

Ein Zeichen ist größer als ein anderes, wenn der Vergleich _____ ergibt.

größer

EBCD-CODE

(Erweiterter 8-Bit IBM-BCD-CODE)

linkes Halbbyte (Zonenteil)		rechtes Halbbyte (Ziffernteil)															
hexadezimal →		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
↓	binär →	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000																
1	0001																
2	0010																
3	0011																
4	0100	blank											.	¤	(	+	
5	0101	&											$	*	)		
6	0110	-	/										,	%			
7	0111												#	@	'	=	
8	1000																
9	1001																
A	1010																
B	1011																
C	1100		A	B	C	D	E	F	G	H	I						
D	1101		J	K	L	M	N	O	P	Q	R						
E	1110			S	T	U	V	W	X	Y	Z						
F	1111	0	1	2	3	4	5	6	7	8	9						

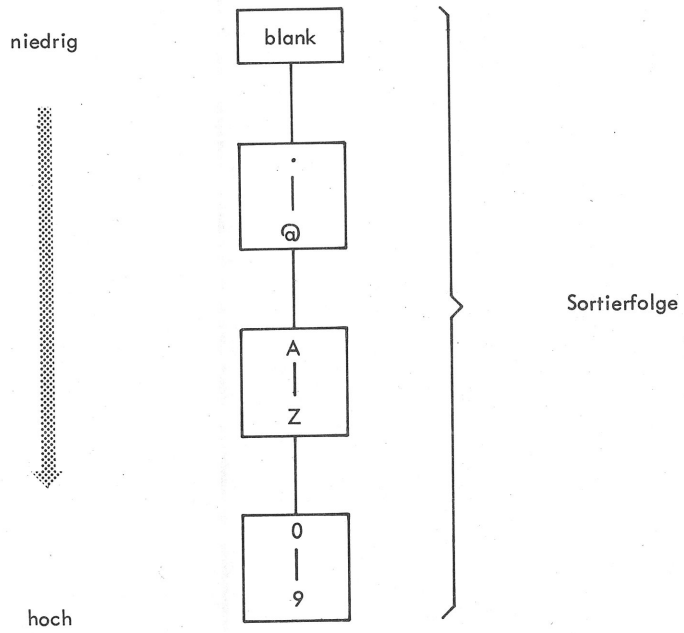


Abb. 2

17 Man kann die Zeichen nach diesen Vergleichsergebnissen ihrer Größe nach anordnen. Diese Reihenfolge ist die Sortierfolge der Zeichen.

Abb. 2 (unterer Teil) zeigt die Sortierfolge der Ihnen bekannten Zeichen im EBCD-Code.

In dieser Folge ist das kleinste Zeichen das _____ und das größte Zeichen die Ziffer _____.

blank

9

18 In der Sortierfolge sind alle Ziffern _____ als die Buchstaben.

Die Buchstaben sind _____ als die Sonderzeichen.

größer

größer

19 Abb. 2 zeigt im oberen Teil die Verschlüsselung der Zeichen.

Die Sortierfolge der Zeichen entnimmt man der Tabelle wie folgt:

Man liest zeilenweise von oben nach unten und innerhalb der Zeilen von _____ nach _____.

links (nach) rechts

20 In den folgenden Lernelementen werden wir anhand des logischen Vergleichs diese Sortierfolge überprüfen.

Die Zeichen "blank" und "." haben im Speicher folgende Verschlüsselung:

"blank" 0100 0000

"." 0100 1011

Beim Vergleich tritt die erste Ungleichheit in Bitposition _____ auf.

Das Zeichen "." ist _____ als "blank".

4

größer

EBCD-CODE

(Erweiterter 8-Bit IBM-BCD-CODE)

linkes Halbbyte (Zonenteil)		rechtes Halbbyte (Ziffernteil)															
hexadezimal →		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
↓	binär →	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000																
1	0001																
2	0010																
3	0011																
4	0100	blank											.	¤	(	+	
5	0101	&											$	*	)		
6	0110	-	/										,	%			
7	0111												#	@	'	=	
8	1000																
9	1001																
A	1010																
B	1011																
C	1100		A	B	C	D	E	F	G	H	I						
D	1101		J	K	L	M	N	O	P	Q	R						
E	1110			S	T	U	V	W	X	Y	Z						
F	1111	0	1	2	3	4	5	6	7	8	9						

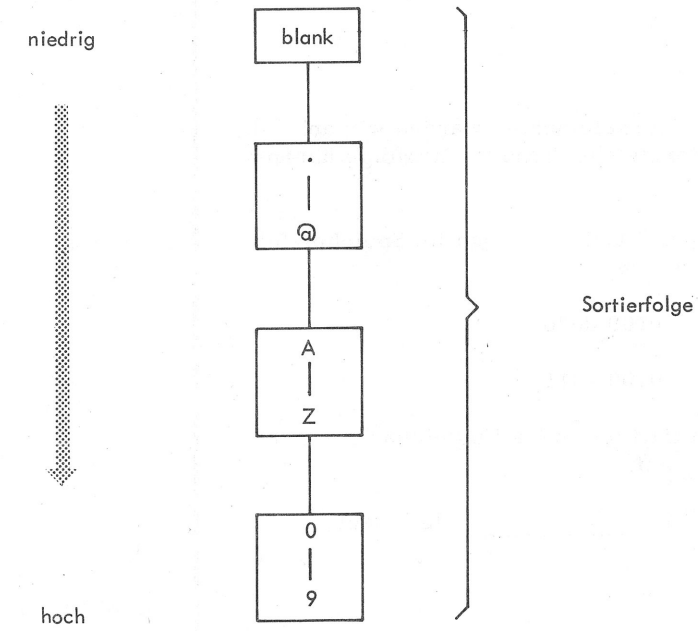


Abb. 2

21 Die Zeichen "@" und "A" haben folgende Verschlüsselung:

"@" 0111 1100

"A" 1100 0001

Das Vergleichsergebnis entscheidet sich in Bitposition (Z).

"0" ist kleiner als "@".

22 Wir vergleichen noch den Buchstaben "Z" und die Ziffer "0".

"Z" 1110 1001

"0" 1111 0000

Durch die Ungleichheit in Bitposition (Z) ist die Ziffer "0" größer als der Buchstabe "Z".

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
*, ,D,E,F,I,N,I	T	I,O,N,E,N
F,E,L,D,A, , ,	D,S, , ,	C,L,5,
F,E,L,D,B, , ,	D,S, , ,	C,L,5,

Abb. 3

Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 407, Seite 1 und 2.

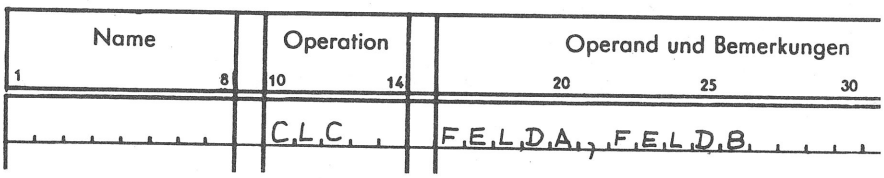
Tragen Sie Ihre Lösungen ein.

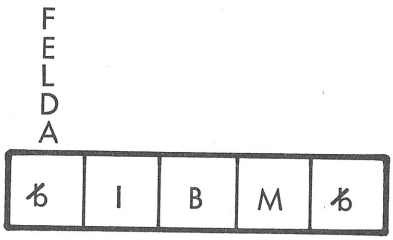
Die Definitionen zu den Speicherfeldern sind in Abb. 3 enthalten.

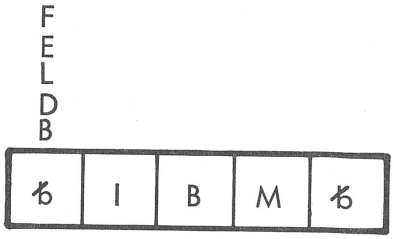
Vergleichen Sie Ihre Eintragungen mit den Lösungen auf den folgenden Seiten.

Lösungen zu den Übungen

1.





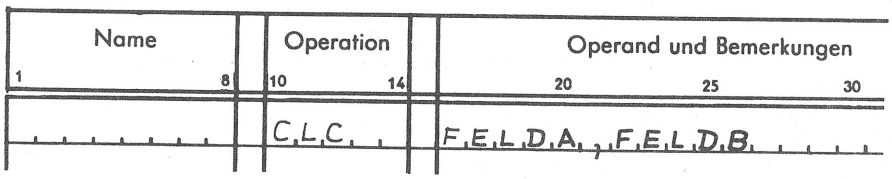


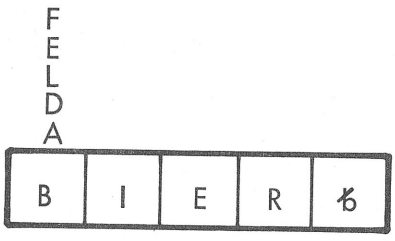
Der Bedingungsschlüssel zeigt das Vergleichsergebnis "gleich" an.

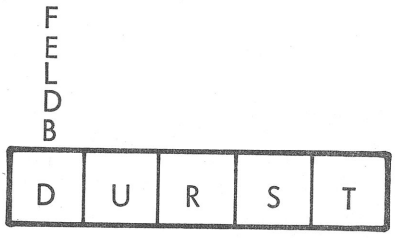
FELDA ist gleich FELDB.

Der Vergleich endet nach dem fünften Byte von links.

2.







Der Bedingungsschlüssel zeigt das Vergleichsergebnis "kleiner" an.

FELDA ist kleiner als FELDB.

Der Vergleich endet nach dem ersten Byte von links.

3. Wie muß die Vergleichsinstruktion lauten, wenn bei folgendem Inhalt der Felder FELDA und FELDB das Ergebnis "größer" heißen soll?

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	CLC	FELDB,FELDA

Anmerkung zur Lösung:

Beachten Sie bitte, daß das Vergleichsergebnis sich immer auf den ersten Operanden bezieht.

Merkblatt 406 enthält die Beschreibung der Operation "Vergleichen Logisch Zeichen" (vgl. Abb. 1).



Seite 111 (Verzweigungsoperationen)

CLI VERGLEICHEN LOGISCH DIREKTZEICHEN

(Compare Logical Immediate)

Diese Operation vergleicht den Inhalt eines Bytes im Speicher mit einem Direktzeichen. Das Ergebnis wird im Bedingungsschlüssel angezeigt.

Beispiel einer Instruktion:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
	CLI	ART,C'L'

Wirkungsweise:

Der zweite Operand ist ein Direktoperand (im Beispiel C'L').

Der Inhalt des ersten Operanden (ein Byte, im Beispiel ART) wird mit dem Direktzeichen (auch Direktdate, im Beispiel L) des zweiten Operanden verglichen.

Das Vergleichsergebnis wird im Bedingungsschlüssel angezeigt. Es bezieht sich immer auf den ersten Operanden.

Folgende Vergleichsergebnisse sind möglich:

gleich	(erster Operand gleich Direktzeichen)
kleiner/niedrig	(erster Operand kleiner als Direktzeichen)
größer/hoch	(erster Operand größer als Direktzeichen)

Verarbeitet wird 1 Byte und zwar Bit für Bit von links nach rechts.

Der Vergleich endet bei der ersten Ungleichheit.

Jede der 256 Bit-Kombinationen in einem Byte ist erlaubt.

Der Inhalt beider Operanden bleibt unverändert.

Durchführung der Operation:

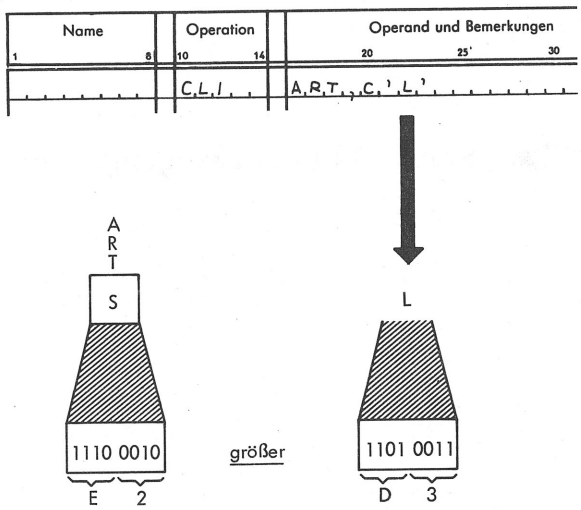


Abb. 1

1 Wir besprechen jetzt die Operation mit dem Namen "___(WW)___" und dem Operationsschlüssel _____.

Vergleichen Logisch Direktzeichen

CLI

2 Der zweite Operand dieser Operation ist ein _____.

Direktoperand

3 Der Direktoperand enthält ein _____.

Das Direktzeichen wird als Konstante in der Instruktion selbst gespeichert.

Direktzeichen

4 Der Direktoperand hat im Instruktionsbeispiel in Abb. 1 die Form C'L'. Er könnte auch mit X'D3' angegeben sein.

In beiden Fällen ist die gleiche Bitkombination __(ZZ)__ in einem Byte gemeint.

1101 0011

5 Wir nehmen an, daß der erste Operand einer Operation "Vergleichen Logisch Direktzeichen" die implizite Länge 3 Bytes hat und in den Speicherstellen 769, 770, 771 steht.

Auf welche dieser Speicherstellen bezieht sich dann der Vergleich mit dem Direktzeichen?

Speicherstelle 769

6 In Abb. 1 sehen Sie, wie eine Operation "Vergleichen Logisch Direktzeichen" durchgeführt wird.
Die 8 Bits werden Bit für Bit von _____ nach _____ verglichen.

Das Vergleichsergebnis steht fest, sobald zwei Binärziffern (gleich, ungleich) _____ sind.

links (nach) rechts

ungleich

CLI

VERGLEICHEN LOGISCH DIREKTZEICHEN

(Compare Logical Immediate)

Diese Operation vergleicht den Inhalt eines Bytes im Speicher mit einem Direktzeichen. Das Ergebnis wird im Bedingungsschlüssel angezeigt.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	C,L,I	A,R,T,,C,'L,'

Wirkungsweise:

Der zweite Operand ist ein Direktoperand (im Beispiel C'L').

Der Inhalt des ersten Operanden (ein Byte, im Beispiel ART) wird mit dem Direktzeichen (auch Direktdate, im Beispiel L) des zweiten Operanden verglichen.

Das Vergleichsergebnis wird im Bedingungsschlüssel angezeigt. Es bezieht sich immer auf den ersten Operanden.

Folgende Vergleichsergebnisse sind möglich:

gleich	(erster Operand gleich Direktzeichen)
kleiner/niedrig	(erster Operand kleiner als Direktzeichen)
größer/hoch	(erster Operand größer als Direktzeichen)

Verarbeitet wird 1 Byte und zwar Bit für Bit von links nach rechts.

Der Vergleich endet bei der ersten Ungleichheit.

Jede der 256 Bit-Kombinationen in einem Byte ist erlaubt.

Der Inhalt beider Operanden bleibt unverändert.

Durchführung der Operation:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	C,L,I	A,R,T,,C,'L,'

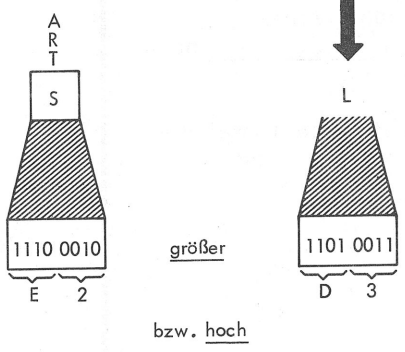


Abb. 1

7 Im Beispiel in Abb. 1 entscheidet sich das Vergleichsergebnis "größer" in Bitposition ___(Z)___ der beiden Bytes.

"S" ist _____ als "L".

2

größer

8 Sie haben vorher einen anderen logischen Vergleich mit dem Operationsschlüssel CLC kennengelernt.

Bei beiden Operationen wird binär von links nach rechts verglichen.
Darum ist die Sortierfolge der Zeichen bei beiden Operationen (gleich, verschieden) _____.

gleich

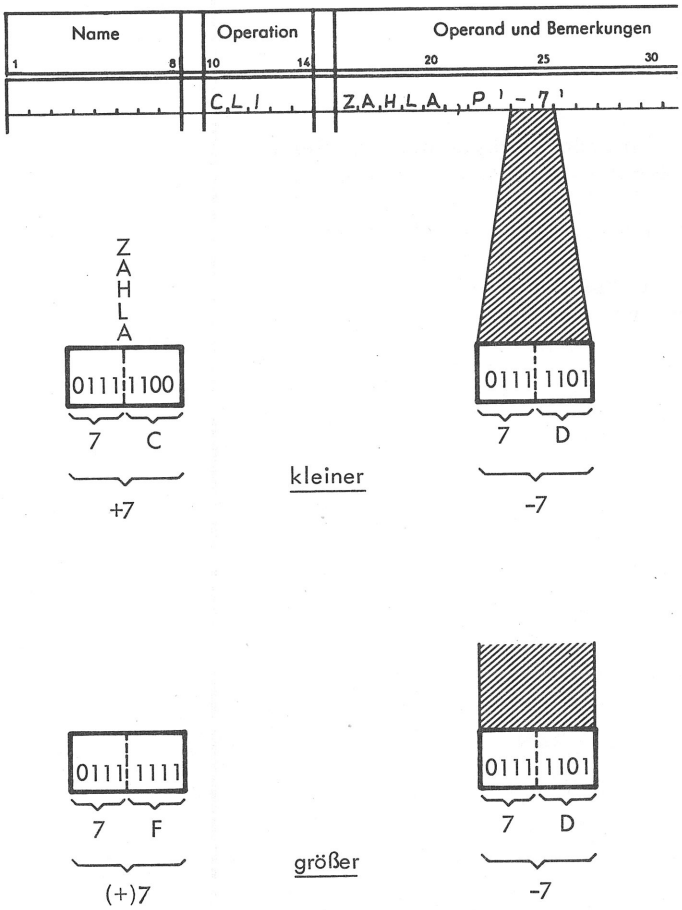

Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
C,L,I
Z,A,H,L,A,,,P,',-,7,'
ZAHLA
0111 1100
7 C
+7
kleiner
0111 1101
7 D
-7
0111 1111
7 F
(+)7
größer
0111 1101
7 D
-7

Abb. 2

9 Das Direktzeichen kann auch eine Zahl mit Vorzeichen sein, wie z.B. die Zahl "-7" in Abb. 2.
Der Direktoperand hat dann die Form _____.

P'-7'/X'7D'

10 Der erste Operand "ZAHLA" in Abb. 2 enthält eine positive "7".
Die Operation "Vergleichen Logisch Direktzeichen" bringt nur dann das Ergebnis: "+7" größer als "-7", wenn in ZAHLA das positive Vorzeichen von der Hexadezimalziffer (C, F) _____ dargestellt wird.

F

Sie sehen:

Die Operation CLI eignet sich nur unter Vorbehalten für den Vergleich von Zahlen mit Vorzeichen. Diese Einschränkung gilt auch für CLC, da beide Operationen völlig gleich arbeiten - bis auf die Form des zweiten Operanden und die Längenbeschränkung auf ein Byte bei CLI.

Um auch Zahlen mit beliebig verschlüsselten Vorzeichen richtig vergleichen zu können, steht den logischen Vergleichsoperationen CLC und CLI ein arithmetischer Vergleich gegenüber.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
*, D,E,F,I,N,I,T,I,O,N,E,N		
A,R,T,	D,S,	C,L,1,
N,U,M,M,E,R,	D,S,	C,L,1,
A,W,E,R,T,	D,S,	C,L,1,

Abb. 3

Übungen

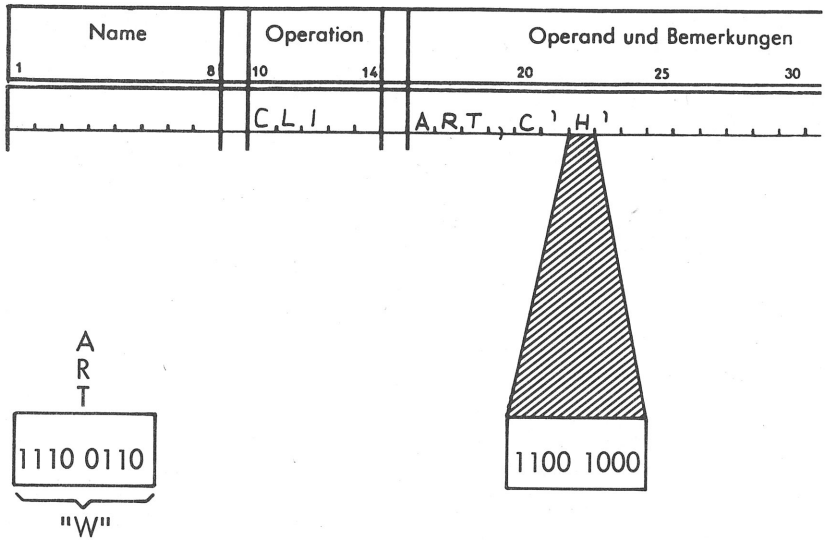
Entnehmen Sie Ihren Arbeitsmitteln Übung 408, Seite 1 und 2.

Tragen Sie Ihre Lösungen ein.

Abb. 3 enthält die Definitionen der benutzten Speicherfelder.

Vergleichen Sie Ihre Eintragungen mit den Lösungen auf den folgenden Seiten, sobald Sie ein Blatt vollständig bearbeitet haben.

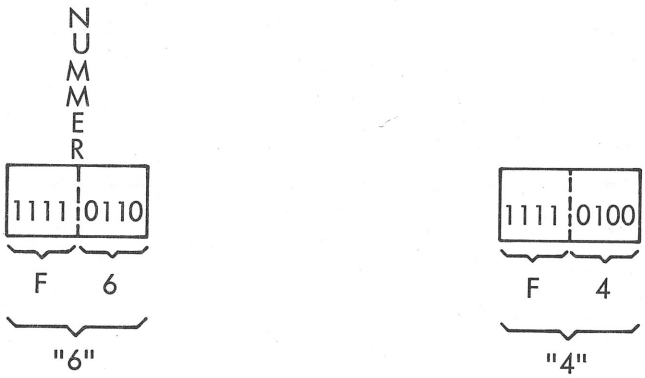
1.



Das Vergleichsergebnis heißt größer.

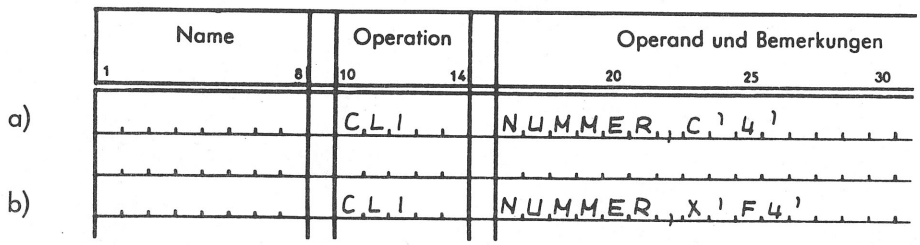
Die erste Ungleichheit ergibt sich in Bitposition 2.

2.

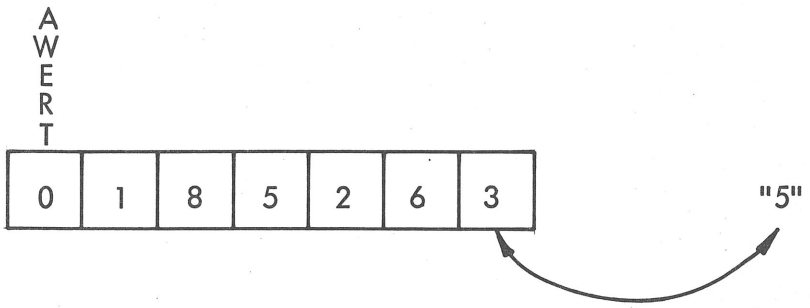


Codieren Sie bitte die Instruktion zum Vergleich des Feldes NUMMER mit dem Direktzeichen "4".
Benutzen Sie

a) den Konstantentyp C,
b) den Konstantentyp X.



3.



Das rechte Byte des Feldes AWERT soll darauf geprüft werden, ob sein Inhalt größer, kleiner oder gleich 5 ist.
Codieren Sie die entsprechende Instruktion mit dem Operationsschlüssel CLI.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	CLI	AWERT+6,C'5'
oder		
	CLI	AWERT+6,X'F5'

Merkblatt 407 enthält die Beschreibung der Operation "Vergleichen Logisch Direktzeichen" (vgl. Abb. 1).



Band 1, Seite 411

Verzweigungsoperationen

Im folgenden beschäftigen wir uns mit Verzweigungsoperationen.

Es gibt bedingte und unbedingte Verzweigungsbefehle. Die bedingten Verzweigungsoperationen fragen den Bedingungsschlüssel ab, der z.B. durch einen CLC-Vergleich gesetzt wurde.

Wenn Sie den folgenden Abschnitt durchgearbeitet haben, kennen Sie folgende Operationen:

a) die unbedingte Verzweigung,

b) die bedingten Verzweigungen im Anschluß an Vergleichsoperationen,

c) die bedingten Verzweigungen im Anschluß an Rechenoperationen,

d) die Operation "NOP".

Zum Schluß erhalten Sie ein Merkblatt, in dem alle Verzweigungsoperationen zusammengefaßt sind. (Sie brauchen also die Operationsschlüssel nicht auswendig zu lernen.)

B

VERZWEIGEN UNBEDINGT

(Branch)

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	B,	A,3,

Wirkungsweise:

Diese Operation bewirkt eine Verzweigung des Programms zur Instruktion mit der Hauptspeicheradresse, die im Operandenfeld angegeben ist (im Beispiel A3).

Es wird unbedingt, d.h. unabhängig von der Stellung des Bedingungsschlüssels verzweigt.

Durchführung der Operation:

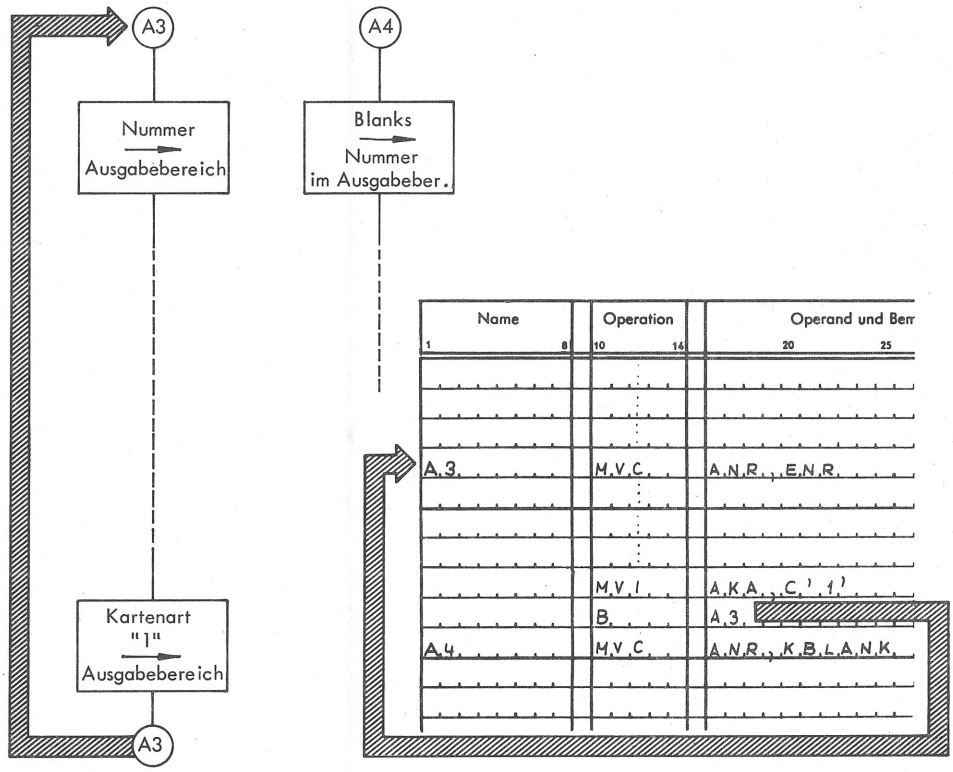


Abb. 1

Die unbedingte Verzweigung

1 Die Verzweigungsoperation in Abb. 1 heißt " (WW) " und hat den symbolischen Operationsschlüssel _____.

Verzweigen Unbedingt

B

2 Beim unbedingten Verzweigungsbefehl enthält das Operandenfeld (einen, zwei) _____ Operanden.

einen

3 Der Operand des Verzweigungsbefehls ist eine (symbolische, echte) _____ Hauptspeicheradresse.

Es ist die Adresse (eines Datenfeldes, einer Konstanten, einer Instruktion) _____.

symbolische

einer Instruktion

4 Abb. 1 zeigt einen Ausschnitt aus einem Programmablaufplan und die entsprechende Codierung.

Der Befehl "B A3" bewirkt, daß als nächster Programmschritt nicht der Befehl mit dem Namen _____, sondern der Befehl mit dem Namen _____ ausgeführt wird.

A4

A3

B

VERZWEIGEN UNBEDINGT

(Branch)

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	B,	A,3,

Wirkungsweise:

Diese Operation bewirkt eine Verzweigung des Programms zur Instruktion mit der Hauptspeicheradresse, die im Operandenfeld angegeben ist (im Beispiel A3).

Es wird unbedingt, d.h. unabhängig von der Stellung des Bedingungsschlüssels verzweigt.

Durchführung der Operation:

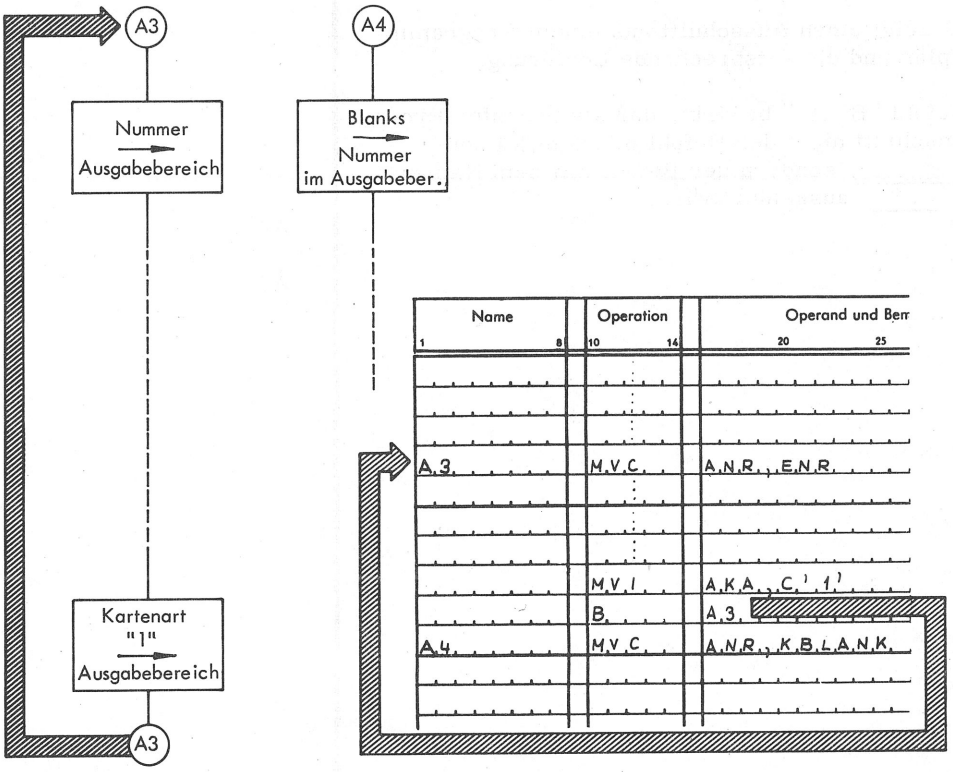


Abb. 1

5 Durch den Befehl "B A3" ist die codierte Instruktionsfolge unterbrochen.
Der Befehl A3 wird ausgeführt. Danach folgt (der Befehl A4, der auf A3 folgende Befehl) _____.

der auf A3 folgende Befehl

6 Wir nehmen an, daß zur Zeit der Ausführung der Instruktion "B A3" der Bedingungsschlüssel das Vergleichsergebnis "hoch" enthält.

In diesem Fall wird als nächster der Programmschritt (A3, A4) _____ ausgeführt.

A3

7 Die Durchführung dieser Verzweigungsoperation ist nicht abhängig von einer bestimmten Stellung des Bedingungsschlüssels.

Die Operation heißt "Verzweigen _____".

Unbedingt

OPERATIONS- Bezeichnung	Schlüssel	Es wird verzweigt, wenn der Bedingungsschlüssel anzeigt:
Verzweigen Unbedingt (Branch)	B	(Unabhängig von der Anzeige. Es wird immer verzweigt.)
Nach Vergleichs-Operationen:		
Verzweigen bei Größer (Branch on High)	BH	größer (hoch)
Verzweigen bei Kleiner (Branch on Low)	BL	kleiner (niedrig)
Verzweigen bei Gleich (Branch on Equal)	BE	gleich
Verzweigen bei Nicht Gleich (Branch on Not Equal)	BNE	größer oder kleiner
Verzweigen bei Nicht Kleiner (Branch on Not Low)	BNL	größer oder gleich
Verzweigen bei Nicht Größer (Branch on Not High)	BNH	kleiner oder gleich

Abb. 1

Die bedingten Verzweigungen im Anschluß an Vergleichsoperationen

Im Unterschied zur unbedingten Verzweigung wird bei den folgenden Operationen nur dann verzweigt, wenn der Bedingungsschlüssel die entsprechende Bedingung anzeigt.

Die Tabelle in Abb. 1 zeigt neben der bereits bekannten unbedingten Verzweigung die im Anschluß an Vergleichsoperationen möglichen bedingten Verzweigungen. Machen Sie sich bitte mit dieser Tabelle vertraut und wenden Sie sich dann den folgenden Lernelementen zu.

1 Nach einer Vergleichsoperation steht der Bedingungsschlüssel auf _____ oder _____ oder _____.

größer/hoch

kleiner/niedrig

gleich

2 Ein bedingter Verzweigungsbefehl bewirkt nur dann eine Verzweigung, wenn der Bedingungsschlüssel die entsprechende Bedingung (anzeigt, nicht anzeigt) _____.

anzeigt

3 Aus Abb. 1 ersehen Sie:

Die Stellung des Bedingungsschlüssels ("größer", "kleiner" oder "gleich") können wir durch (Z) bedingte Verzweigungsoperationen abfragen.

6

OPERATIONS- Bezeichnung	Schlüssel	Es wird verzweigt, wenn der Bedingungsschlüssel anzeigt:
Verzweigen Unbedingt (Branch)	B	(Unabhängig von der Anzeige. Es wird immer verzweigt.)
Nach Vergleichs-Operationen:		
Verzweigen bei Größer (Branch on High)	BH	größer (hoch)
Verzweigen bei Kleiner (Branch on Low)	BL	kleiner (niedrig)
Verzweigen bei Gleich (Branch on Equal)	BE	gleich
Verzweigen bei Nicht Gleich (Branch on Not Equal)	BNE	größer oder kleiner
Verzweigen bei Nicht Kleiner (Branch on Not Low)	BNL	größer oder gleich
Verzweigen bei Nicht Größer (Branch on Not High)	BNH	kleiner oder gleich

Abb. 1

4 Zu den Verzweigungen bei "größer", "kleiner" und "gleich" (BH, BL, BE) gibt es je eine Verneinung (entgegengesetzte Abfrage).

So stehen sich gegenüber:

BH und _____,

BL und _____,

BE und _____.

BNH

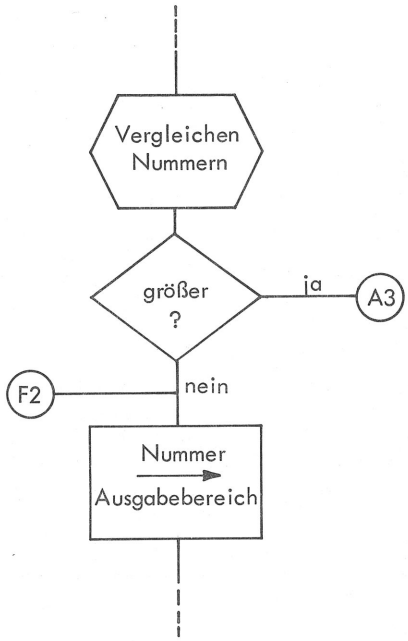
BNL

BNE

BH

VERZWEIGEN BEI GRÖSSER

(Branch on High)



Name	Operation	Operand und Be
1 8	10 14	20 25
	C,L,C,	A,N,R,,E,N,R,
	B,H,	A,3,
F,2,	M,V,C,	A,N,R,,E,N,R,
A,3,	M,V,C,	A,B,E,R,,K,B,

Abb. 2

5 Wir beschäftigen uns zunächst mit der Operation "Verzweigen bei Größer".

Sie hat den Operationsschlüssel _____ (vgl. Abb. 2).

BH

6 Die Operation "BH" bewirkt nur dann eine Verzweigung, wenn der Bedingungsschlüssel "_____" anzeigt.

größer/hoch

7 Wir nehmen den Fall an, daß nach dem Befehl "BH A3" in Abb. 2 die Instruktion mit der symbolischen Adresse F2 ausgeführt wird.

Welche Stellung muß der Bedingungsschlüssel in diesem Fall haben?

gleich oder kleiner

8 Ergibt der vorangehende Nummernvergleich weder "gleich" noch "kleiner", so wird nach dem Verzweigungsbefehl die Instruktion mit der symbolischen Adresse _____ ausgeführt.

A3

9 Der Vergleichsbefehl in Abb. 2 hat die beiden Operanden ANR und ENR.

Eine Verzweigung zum Befehl A3 findet nur statt, wenn (ANR, ENR) _____ größer ist.

ANR

OPERATIONS-Bezeichnung	Schlüssel	Es wird verzweigt, wenn der Bedingungsschlüssel anzeigt:
Verzweigen Unbedingt (Branch)	B	(Unabhängig von der Anzeige. Es wird immer verzweigt.)
Nach Vergleichs-Operationen:		
Verzweigen bei Größer (Branch on High)	BH	größer (hoch)
Verzweigen bei Kleiner (Branch on Low)	BL	kleiner (niedrig)
Verzweigen bei Gleich (Branch on Equal)	BE	gleich
Verzweigen bei Nicht Gleich (Branch on Not Equal)	BNE	größer oder kleiner
Verzweigen bei Nicht Kleiner (Branch on Not Low)	BNL	größer oder gleich
Verzweigen bei Nicht Größer (Branch on Not High)	BNH	kleiner oder gleich

Abb. 1

10 Abb. 1 zeigt außer "BH" noch 5 weitere bedingte Verzweigungsoperationen.

Soll z.B. bei dem Vergleichsergebnis "gleich" verzweigt werden, bei "kleiner" oder "größer" dagegen nicht, so muß die Operation "___(WW)___" verwendet werden.

Sie hat den Operationsschlüssel _____.

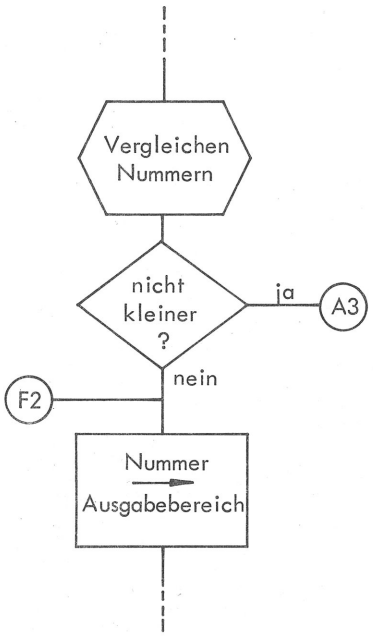
Verzweigen bei Gleich

BE

BNL

VERZWEIGEN BEI NICHT KLEINER

(Branch on Not Low)



Name	Operation	Operand und Beme
1 8	10 14	20 25
	C,L,C,	A,N,R,,,E,N,R,
	B,N,L,	A,3,
F,2,	M,V,C,	A,N,R,,,E,N,R,
A,3,	M,V,C,	A,B,E,R,,,K,B,

Abb. 3

11 Eine weitere Verzweigungsoperation heißt "Verzweigen bei Nicht Kleiner" (vgl. Abb. 3).

Zur angegebenen Hauptspeicheradresse wird verzweigt, wenn ANR "___(WW)___" als ENR ist.

nicht kleiner

12 Der Bedingungsschlüssel kann "größer", "kleiner" oder "gleich" anzeigen.

Bei welcher Stellung bzw. welchen Stellungen des Bedingungsschlüssels ist die Bedingung "nicht kleiner" erfüllt?

Sinngemäß:

Bei der Stellung "größer" und bei der Stellung "gleich".

13 Ist ANR größer als ENR oder gleich ENR, so wird nach der Verzweigungsoperation in Abb. 3 der Befehl mit dem Namen _____ durchgeführt.

A3

14 Anders ausgedrückt:
Es wird immer dann nach "A3" verzweigt, wenn die Bedingung "_____" n i c h t erfüllt ist.

kleiner

15 Das Programm wird nur dann mit dem Befehl "F2" fortgeführt, wenn ANR ___(WW)___ ENR ist.

kleiner als

Nach diesen Beispielen zu bedingten Verzweigungen nach Vergleichsoperationen wenden wir uns einer weiteren Gruppe von Verzweigungen zu.

OPERATIONS-Bezeichnung	Schlüssel	Es wird verzweigt, wenn der Bedingungsschlüssel anzeigt:
Verzweigen Unbedingt (Branch)	B	(Unabhängig von der Anzeige. Es wird immer verzweigt.)
Nach Vergleichs-Operationen:		
Verzweigen bei Größer (Branch on High)	BH	größer (hoch)
Verzweigen bei Kleiner (Branch on Low)	BL	kleiner (niedrig)
Verzweigen bei Gleich (Branch on Equal)	BE	gleich
Verzweigen bei Nicht Gleich (Branch on Not Equal)	BNE	größer oder kleiner
Verzweigen bei Nicht Kleiner (Branch on Not Low)	BNL	größer oder gleich
Verzweigen bei Nicht Größer (Branch on Not High)	BNH	kleiner oder gleich
Nach arithmetischen Operationen:		
Verzweigen bei Positiv (Branch on Plus)	BP	positiv (größer als 0)
Verzweigen bei Negativ (Branch on Minus)	BM	negativ (kleiner als 0)
Verzweigen bei Null (Branch on Zero)	BZ	null (gleich 0)
Verzweigen bei Nicht Null (Branch on Not Zero)	BNZ	positiv oder negativ
Verzweigen bei Nicht Negativ (Branch on Not Minus)	BNM	positiv oder null
Verzweigen bei Nicht Positiv (Branch on Not Plus)	BNP	negativ oder null
Verzweigen bei Überlauf (Branch on Overflow)	BO	Überlauf

Abb. 1

Die bedingten Verzweigungen im Anschluß an Rechenoperationen

Der Bedingungsschlüssel wird nicht nur durch Vergleichsoperationen, sondern auch bei bestimmten arithmetischen Operationen, z.B. Addition und Subtraktion, neu eingestellt.

Betrachten Sie die Erweiterung der Tabelle in Abb. 1.

Mit den ersten 3 Operationen werden die Rechenergebnisse auf "positiv", "negativ" und "null" abgefragt.
Die folgenden 3 Operationen stellen die Verneinung der ersten dar. Außerdem gibt es noch eine Operation zur Abfrage auf "Überlauf".

Die Operationen "BP" bis "BNP" entsprechen in ihrer Funktion den Operationen "BH" bis "BNH" - bis auf den einzigen Unterschied, daß sie sich auf Rechenergebnisse beziehen. Wir behandeln diese Verzweigungsoperationen deshalb nur exemplarisch in einigen Lernelementen.

1 "BZ" ist der Operationsschlüssel für die Operation "__(WW)__".

Es wird verzweigt, wenn bei der vorangehenden Rechenoperation das Ergebnis _____ ist.

Verzweigen bei Null

null

2 Wie lautet die Instruktion für "Verzweigen bei Null zu dem Befehl mit der symbolischen Adresse H4".

Name	Operation	Oper
1 8	10 14	20
	B,Z,	H,4,

3

Name	Operation	Oper
1 8	10 14	20
	B,Z,	H,4,

Dieser Befehl bewirkt k e i n e Verzweigung nach "H4", wenn das vorangehende Rechenergebnis __(WW)__ ist.

Sinngemäß:

nicht null
(größer oder kleiner als 0)

OPERATIONS-Bezeichnung	Schlüssel	Es wird verzweigt, wenn der Bedingungsschlüssel anzeigt:
Verzweigen Unbedingt (Branch)	B	(Unabhängig von der Anzeige. Es wird immer verzweigt.)
Nach Vergleichs-Operationen:		
Verzweigen bei Größer (Branch on High)	BH	größer (hoch)
Verzweigen bei Kleiner (Branch on Low)	BL	kleiner (niedrig)
Verzweigen bei Gleich (Branch on Equal)	BE	gleich
Verzweigen bei Nicht Gleich (Branch on Not Equal)	BNE	größer oder kleiner
Verzweigen bei Nicht Kleiner (Branch on Not Low)	BNL	größer oder gleich
Verzweigen bei Nicht Größer (Branch on Not High)	BNH	kleiner oder gleich
Nach arithmetischen Operationen:		
Verzweigen bei Positiv (Branch on Plus)	BP	positiv (größer als 0)
Verzweigen bei Negativ (Branch on Minus)	BM	negativ (kleiner als 0)
Verzweigen bei Null (Branch on Zero)	BZ	null (gleich 0)
Verzweigen bei Nicht Null (Branch on Not Zero)	BNZ	positiv oder negativ
Verzweigen bei Nicht Negativ (Branch on Not Minus)	BNM	positiv oder null
Verzweigen bei Nicht Positiv (Branch on Not Plus)	BNP	negativ oder null
Verzweigen bei Überlauf (Branch on Overflow)	BO	Überlauf

Abb. 1

4 Es soll zu einer Instruktion mit dem Namen "C4" verzweigt werden, wenn das Ergebnis einer vorangehenden Rechenoperation nicht positiv ist.

Wie muß die Codierung hierzu lauten?

Name	Operation	Oper
1 8	10 14	20
	B N P	C 4

5 Bei welcher der unten aufgeführten Rechenergebnisse bewirkt der Operationsschlüssel "BNP" eine Verzweigung?

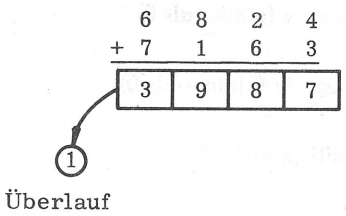
+367 -411 0 -1 +3,4

-411 0 -1

Nach arithmetischen Operationen können Rechenergebnisse nicht nur auf "positiv", "negativ" oder "null", sondern außerdem auch auf "Überlauf" abgefragt werden.

Ein Überlauf tritt auf, wenn ein Rechenfeld zu klein ist, d.h. wenn ein Rechenergebnis - abgesehen von führenden Nullen - darin nicht vollständig Platz hat. Die links nicht hineinpassenden Ziffern "laufen über".

Beispiel:



Überlauf

Zur Abfrage eines Überlaufs gibt es die Operation "BO".

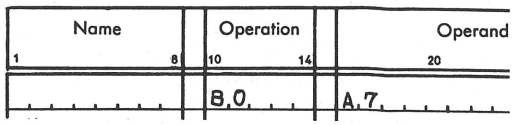
OPERATIONS-Bezeichnung	Schlüssel	Es wird verzweigt, wenn der Bedingungsschlüssel anzeigt:
Verzweigen Unbedingt (Branch)	B	(Unabhängig von der Anzeige. Es wird immer verzweigt.)
Nach Vergleichs-Operationen:		
Verzweigen bei Größer (Branch on High)	BH	größer (hoch)
Verzweigen bei Kleiner (Branch on Low)	BL	kleiner (niedrig)
Verzweigen bei Gleich (Branch on Equal)	BE	gleich
Verzweigen bei Nicht Gleich (Branch on Not Equal)	BNE	größer oder kleiner
Verzweigen bei Nicht Kleiner (Branch on Not Low)	BNL	größer oder gleich
Verzweigen bei Nicht Größer (Branch on Not High)	BNH	kleiner oder gleich
Nach arithmetischen Operationen:		
Verzweigen bei Positiv (Branch on Plus)	BP	positiv (größer als 0)
Verzweigen bei Negativ (Branch on Minus)	BM	negativ (kleiner als 0)
Verzweigen bei Null (Branch on Zero)	BZ	null (gleich 0)
Verzweigen bei Nicht Null (Branch on Not Zero)	BNZ	positiv oder negativ
Verzweigen bei Nicht Negativ (Branch on Not Minus)	BNM	positiv oder null
Verzweigen bei Nicht Positiv (Branch on Not Plus)	BNP	negativ oder null
Verzweigen bei Überlauf (Branch on Overflow)	BO	Überlauf

Abb. 1

6 Wir nehmen an, daß nach einer Addition über den Drucker eine Nachricht geschrieben werden soll, falls das Ergebnisfeld nicht groß genug ist für die Summe.

Das Drucken der Nachricht geschieht im Programmzweig "A7".

Codieren Sie auf einem Programmschema die nötige Verzweigungs-Instruktion.



7 Zu der Operation "Verzweigen bei Überlauf" gehört der gleiche Instruktionsaufbau wie zu den übrigen Verzweigungen.

Zur Instruktion gehört

a) im Feld "Name" (immer, manchmal, nie) _____ ein symbolischer Name,

b) im Feld "Operation" ein _____,

c) im Feld "Operand und Bemerkungen" eine _____.

manchmal

Operationsschlüssel

symbolische Adresse/Verzweigadresse

NOP

KEINE OPERATION

(No OPeration)

Wirkungsweise:

Der Bedingungsschlüssel wird nicht abgefragt.

Es wird nicht zur angegebenen Hauptspeicheradresse (im Beispiel A3) verzweigt, sondern die nächstfolgende Instruktion ausgeführt.

Beispiel:

Name (1–8)	Operation (10–14)	Operand und Be (20–25)
	C,L,C,	A,N,R,,E,N,R,
	N,O,P,	A,3,
H,2,	M,V,C,	A,N,R,,E,N,R,
A,3,	M,V,C,	A,B,E,R,,K,B,

Abb. 1

Die Operation "NOP"

Lesen Sie bitte die Beschreibung der Operation in Abb. 1 genau durch.

Obwohl eine NOP-Instruktion weder eine Verzweigung noch sonst irgendetwas bewirkt, kann es für den Programmierer sinnvoll sein, sie zu benutzen. Der Programmierer hat nämlich die Möglichkeit, aus der NOP-Instruktion während der Ausführung des Programms eine bedingte oder unbedingte Verzweigung zu machen.

Betrachten wir dazu die NOP-Operation im Beispiel der Abb. 1. Würde diese Operation während des Programmablaufs in eine unbedingte Verzweigung umgewandelt, so würde das Programm nach dem Vergleich der Nummern nicht mehr den Schritt "H2" ausführen, sondern zum Schritt "A3" verzweigen.

Wie eine NOP-Operation in eine bedingte oder unbedingte Verzweigung umgewandelt wird, erfahren Sie im Folgelehrgang.

Auf der Rückseite finden Sie die vollständige Tabelle der hier behandelten Verzweigungsoperationen (Abb. 2).

OPERATIONS-Bezeichnung	Schlüssel	Es wird verzweigt, wenn der Bedingungsschlüssel anzeigt:
Verzweigen Unbedingt (Branch)	B	(Unabhängig von der Anzeige. Es wird immer verzweigt.)
Nach Vergleichs-Operationen:		
Verzweigen bei Größer (Branch on High)	BH	größer (hoch)
Verzweigen bei Kleiner (Branch on Low)	BL	kleiner (niedrig)
Verzweigen bei Gleich (Branch on Equal)	BE	gleich
Verzweigen bei Nicht Gleich (Branch on Not Equal)	BNE	größer oder kleiner
Verzweigen bei Nicht Kleiner (Branch on Not Low)	BNL	größer oder gleich
Verzweigen bei Nicht Größer (Branch on Not High)	BNH	kleiner oder gleich
Nach arithmetischen Operationen:		
Verzweigen bei Positiv (Branch on Plus)	BP	positiv (größer als 0)
Verzweigen bei Negativ (Branch on Minus)	BM	negativ (kleiner als 0)
Verzweigen bei Null (Branch on Zero)	BZ	null (gleich 0)
Verzweigen bei Nicht Null (Branch on Not Zero)	BNZ	positiv oder negativ
Verzweigen bei Nicht Negativ (Branch on Not Minus)	BNM	positiv oder null
Verzweigen bei Nicht Positiv (Branch on Not Plus)	BNP	negativ oder null
Verzweigen bei Überlauf (Branch on Overflow)	BO	Überlauf
Keine Operation (No OPeration)	NOP	(Unabhängig von der Anzeige. Es wird nie verzweigt.)

Abb. 2

Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 409, Seite 1 und 2.

Tragen Sie Ihre Lösungen ein.

Als Hilfsmittel können Sie die Zusammenstellung der Verzweigungsoperationen in Abb. 2 benutzen.

Vergleichen Sie zum Schluß Ihre Eintragungen mit den Lösungen auf den folgenden Seiten.

Lösungen zu den Übungen

1. Sowohl durch Vergleichs- als auch durch Rechenoperationen wird der Bedingungsschlüssel gesetzt.

2.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	:	
	C,L,C,	A,N,R, , E,N,R,
	:	

Nach dieser Vergleichsoperation soll zur Instruktion "A4" verzweigt werden, wenn ANR kleiner als ENR ist.

Codieren Sie den entsprechenden Verzweigungsbefehl.

	B,L,	A,4,

3.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	:	
	C,L,C,	A,N,R, , E,N,R,
	B,N,H,	C,7,
	:	

Wenn die Verzweigung nach C7 durchgeführt wird, ist ANR gleich oder kleiner als ENR.

4. Wir nehmen an, daß eine Rechenoperation durchgeführt wurde, die den Bedingungsschlüssel neu einstellt.

Codieren Sie die Instruktion

a) für eine Verzweigung nach A4, wenn das Rechenergebnis größer als 0 ist.

b) für eine Verzweigung nach D5, wenn das Rechenergebnis nicht größer als 0 ist.

c) für eine Verzweigung nach H3 bei jedem beliebigen Wert des Rechenergebnisses.

	Name (1 … 8)	Operation (10 … 14)	Operand und Bemerkungen (20 … 25 … 30)
a)		B,P,	A,4,
b)		B,N,P,	D,5,
c)		B,	H,3,

Entnehmen Sie bitte Ihren Arbeitsmitteln das Merkblatt 408. Sehen Sie sich insbesondere die "Ergänzung" an.



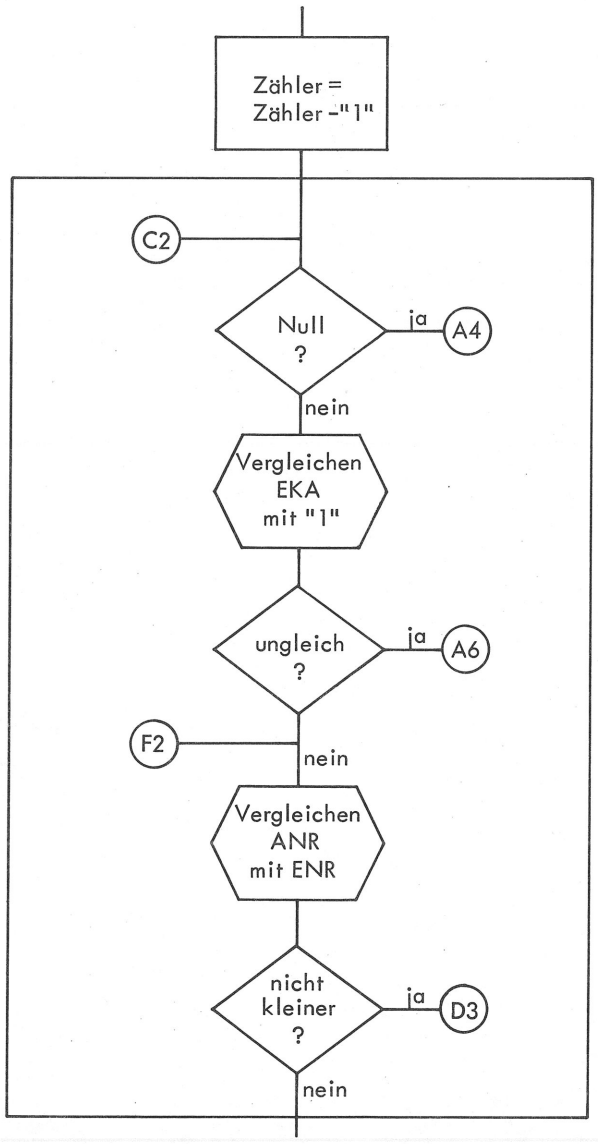
Band 1, Seite 81

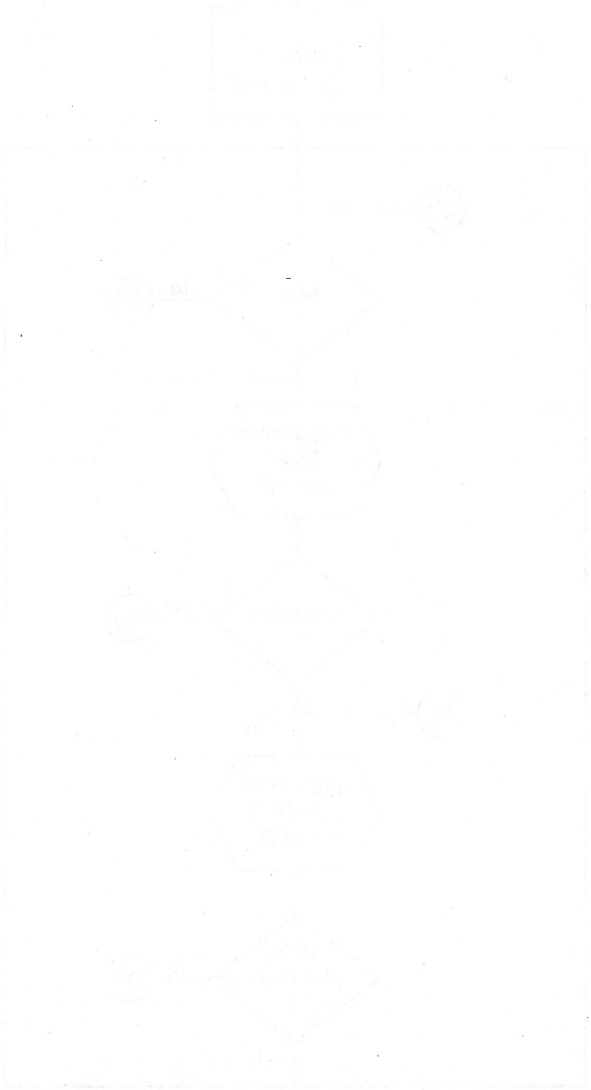
FRAGEN ZUM KAPITEL II

Überprüfen Sie bitte anhand der folgenden Aufgabe Ihre Kenntnisse bzgl. der Vergleichs- und Verzweigungsoperationen.

Nehmen Sie bitte ein Programmschema aus Ihren Arbeitsmitteln!

Codieren Sie den eingerahmten Teil des folgenden Programmablaufplans.

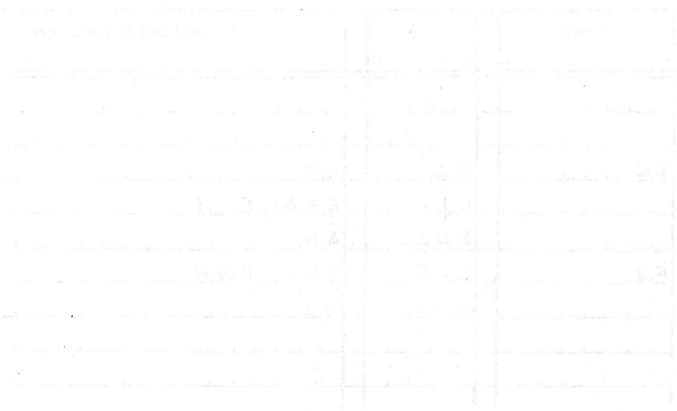




ANTWORTEN ZUM KAPITEL II

Überprüfen Sie nun Ihre Antworten. Beseitigen Sie erst alle Unklarheiten, ehe Sie fortfahren. Wiederholen Sie gegebenenfalls den in Klammern angegebenen Lernstoff.

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)	
C,2,	B,Z,	A,4,	(Seite 127 - 129)
	C,L,I,	E,K,A,,C,',1,'	(Seite 98 ff.)
	B,N,E,	A,6,	(Seite 117 - 125)
F,2,	C,L,C,	A,N,R,,E,N,R,	(Seite 80 ff.)
	B,N,L,	D,3,	(Seite 117 - 125)



KAPITEL III

DIE OPERATIONEN
PACK, UNPK, AP, SP, ZAP

Lernstoff:

PACK	-	Packen
UNPK	-	Entpacken
AP	-	Addieren Dezimal (Gepackt)
SP	-	Subtrahieren Dezimal (Gepackt)
ZAP	-	Löschen und Addieren (Gepackt)

PACK

PACKEN

(PACK)

Diese Operation ändert die Speicherungsform von numerischen Daten von "ungepackt" in "gepackt".

Beispiel einer Instruktion:

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
. . . . .	P,A,C,K,	P,W,E,R,T,,E,W,E,R,T, ...

Wirkungsweise:

Der ungepackte Inhalt des zweiten Operanden (im Beispiel EWERT) wird gepackt und in den ersten Operanden (im Bespiel PWERT) übertragen.

Verarbeitet wird byteweise von rechts nach links in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die Feldlängen müssen nicht übereinstimmen. Wenn der erste Operand zu lang ist, werden links führende Nullen eingeschoben. Wenn der erste Operand zu kurz ist, werden die höchsten (linken) Ziffern aus dem zweiten Operanden nicht übertragen.

Die Feldlänge jedes Operanden darf maximal 16 Bytes betragen.

Durchführung der Operation:

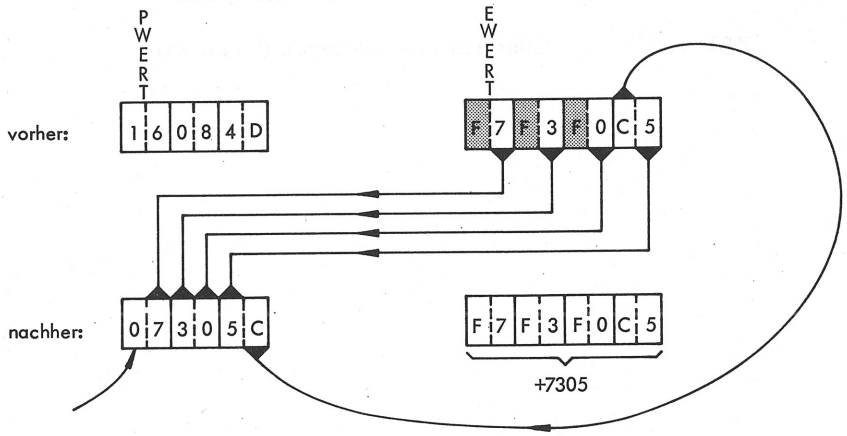


Abb. 1

1 Wir nehmen an, daß es im Hauptspeicher ein 4-stelliges Feld mit dem symbolischen Namen EWERT gibt.

Dieses Feld enthält eine Dezimalzahl in ungepackter Darstellungsform. Die Dezimalzahl soll in ein Summenfeld addiert werden.

Voraussetzung für dezimales Rechnen in der Zentraleinheit ist die _____ Speicherungsform der Rechenwerte.

gepackte

2 Die Operation zur Umwandlung einer ungepackten Dezimalzahl in eine gepackte heißt "_____".

Packen

3 In Abb. 1 wird eine 4-stellige Zahl in ein __(Z)__ Bytes großes Feld gepackt.

Dieses Feld heißt in unserem Beispiel _____.

3

PWERT

4 Das Feld PWERT hat Platz für eine gepackte Dezimalzahl mit __(Z)__ Ziffern und __(WW)__.

Die Zahl im Feld EWERT ist (zu lang, genau passend, zu kurz) _____, um alle Halbbytes in PWERT auszufüllen.

5

einem Vorzeichen

zu kurz

PACK

PACKEN

(PACK)

Diese Operation ändert die Speicherungsform von numerischen Daten von "ungepackt" in "gepackt".

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
. . . . .	P,A,C,K,	P,W,E,R,T,,E,W,E,R,T . . . . .

Wirkungsweise:

Der ungepackte Inhalt des zweiten Operanden (im Beispiel EWERT) wird gepackt und in den ersten Operanden (im Bespiel PWERT) übertragen.

Verarbeitet wird byteweise von rechts nach links in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die Feldlängen müssen nicht übereinstimmen. Wenn der erste Operand zu lang ist, werden links führende Nullen eingeschoben. Wenn der erste Operand zu kurz ist, werden die höchsten (linken) Ziffern aus dem zweiten Operanden nicht übertragen.

Die Feldlänge jedes Operanden darf maximal 16 Bytes betragen.

Durchführung der Operation:

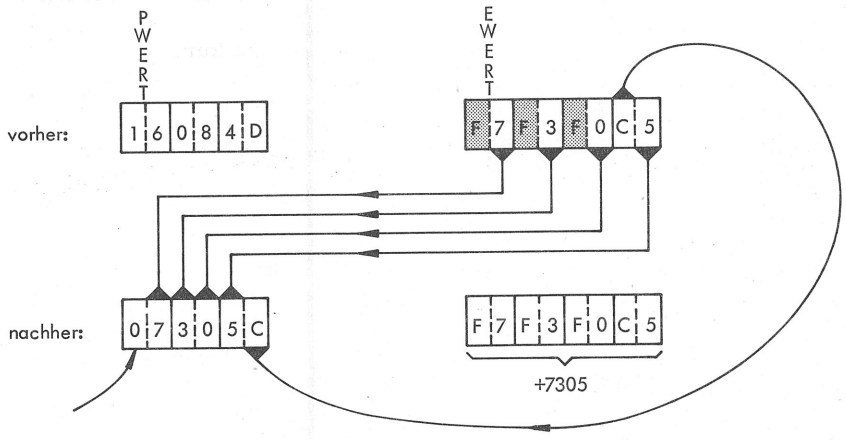


Abb. 1

5 In der Beschreibung der Operation taucht der Begriff "führende Null" auf.

Eine Null heißt "führende Null", wenn links von ihr in der gleichen Zahl keine der Ziffern 1 bis 9 mehr steht.

Geben Sie bitte an, wie viele führende Nullen folgende Zahlen haben:

0012503	(Z)
3600520	(Z)
320	(Z)
01257	(Z)
0000405	(Z)

2

0

0

1

4

6 Das linke Halbbyte in PWERT wird nicht durch eine Ziffer aus EWERT ausgefüllt.

Formulieren Sie bitte in einem Satz, was mit diesem Halbbyte geschieht.

Sinngemäß:

Beim Packen wird in das linke Halbbyte von PWERT eine führende Null eingeschoben.

7 Stellen Sie sich bitte vor, daß der erste Operand PWERT eine Länge von 5 Bytes hat, während EWERT weiterhin 4 Bytes groß ist.

In diesem Fall werden beim Packen in PWERT (Z) führende Nullen eingeschoben.

5

PACK

PACKEN

(PACK)

Diese Operation ändert die Speicherungsform von numerischen Daten von "ungepackt" in "gepackt".

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	PACK	PWERT,EWERT

Wirkungsweise:

Der ungepackte Inhalt des zweiten Operanden (im Beispiel EWERT) wird gepackt und in den ersten Operanden (im Bespiel PWERT) übertragen.

Verarbeitet wird byteweise von rechts nach links in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die Feldlängen müssen nicht übereinstimmen. Wenn der erste Operand zu lang ist, werden links führende Nullen eingeschoben. Wenn der erste Operand zu kurz ist, werden die höchsten (linken) Ziffern aus dem zweiten Operanden nicht übertragen.

Die Feldlänge jedes Operanden darf maximal 16 Bytes betragen.

Durchführung der Operation:

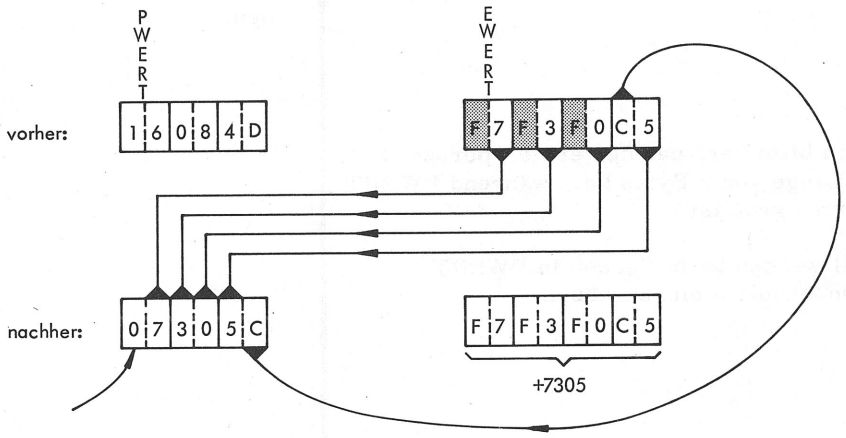


Abb. 1

8 Es kann auch vorkommen, daß PWERT zu kurz ist, um die gepackte Dezimalzahl vollständig aufzunehmen.

Ist z.B. PWERT nur 2 Bytes groß, während EWERT weiterhin eine Länge von 4 Bytes hat, so werden nur die Ziffern __(ZZ)__ und das Vorzeichen nach PWERT gespeichert.

Die (linke, rechte) _____ Ziffer wird nicht mehr übertragen.

3 0 5

linke

9 Wir sehen daraus:

Im Gegensatz zu den logischen Operationen ist für die Durchführung der Operation "Packen" nicht die Länge des ersten Operanden allein entscheidend, sondern auch __(WW)__.

die Länge des zweiten Operanden

10 Bei den logischen Übertragungsoperationen gibt der 1. Operand an, WOHIN Daten übertragen werden. Der 2. Operand bestimmt, WOHER die Daten kommen.

Bei der Operation "Packen" gilt diese Reihenfolge (nicht, ebenso) _____.

ebenso

11 Wir können auch die Verarbeitungsrichtung der logischen Operationen und der Operation "Packen" vergleichen.

Das Packen beginnt bei den _____ Bytes der Felder und wird nach _____ hin fortgesetzt.

Die Verarbeitungsrichtung ist also (genauso, entgegengesetzt) _____ wie bei den logischen Operationen.

rechten

links

entgegengesetzt

PACK

PACKEN

(PACK)

Diese Operation ändert die Speicherungsform von numerischen Daten von "ungepackt" in "gepackt".

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	PACK	PWERT,EWERT

Wirkungsweise:

Der ungepackte Inhalt des zweiten Operanden (im Beispiel EWERT) wird gepackt und in den ersten Operanden (im Bespiel PWERT) übertragen.

Verarbeitet wird byteweise von rechts nach links in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die Feldlängen müssen nicht übereinstimmen. Wenn der erste Operand zu lang ist, werden links führende Nullen eingeschoben. Wenn der erste Operand zu kurz ist, werden die höchsten (linken) Ziffern aus dem zweiten Operanden nicht übertragen.

Die Feldlänge jedes Operanden darf maximal 16 Bytes betragen.

Durchführung der Operation:

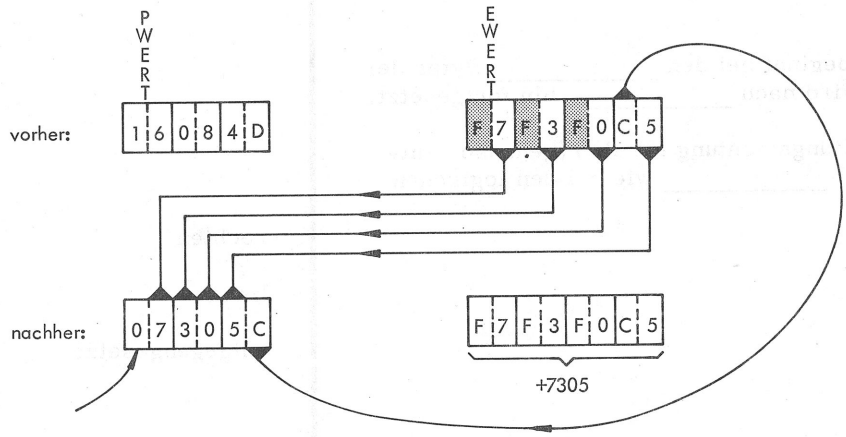


Abb. 1

12 Sind die beiden Operanden verschieden lang, so werden entweder nicht alle Ziffern in den ersten Operanden aufgenommen oder er wird mit Nullen aufgefüllt.

Da die Verarbeitungsrichtung von rechts nach links geht, sind es gegebenfalls die _____ Ziffern, die wegfallen, bzw. werden gegebenfalls _____ Nullen eingesteuert.

linken

links

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	:	
ELOHN	DS	CL6
	:	
	PACK	ELOHN,ELOHN

ELOHN

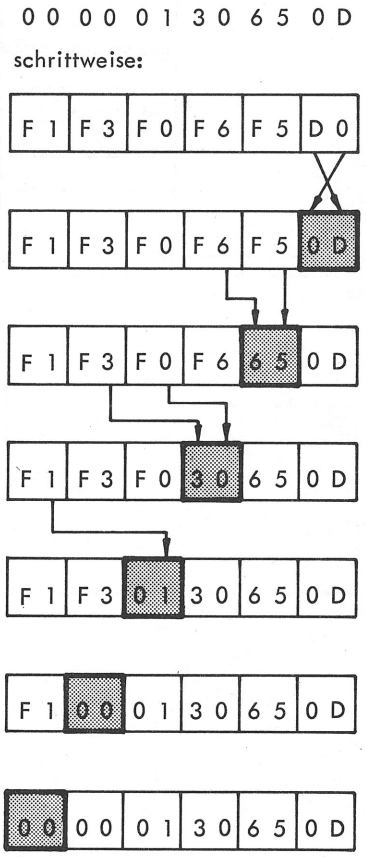
F 1	F 3	F 0	F 6	F 5	D 0

Abb. 2

Das DVS kann keine dezimalen Rechenoperationen mit ungepackten Daten durchführen. Ungepackte Eingabewerte müssen daher vor dem Rechnen gepackt werden.
Wir untersuchen jetzt, ob ein Wert im Eingabefeld selbst gepackt werden darf. Allgemein ausgedrückt: Wir fragen, ob "Packen ins gleiche Feld" erlaubt ist.

13 Betrachten Sie das Beispiel in Abb. 2. Die Zahl in ELOHN wird ins gleiche Feld gepackt, denn der erste und der zweite Operand sind gleich.

Welchen Inhalt hat ELOHN nach dieser Operation?



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	:	
ELOHN	DS	CL6
	:	
	PACK	ELOHN,ELOHN

ELOHN

F 1	F 3	F 0	F 6	F 5	D 0

Abb. 2

Abb. 2

14 Das Feld ELOHN enthält nach dem Packen die gleiche Zahl wie vorher, nämlich "-130650", allerdings jetzt in gepackter Form.

Die führenden Nullen beeinflussen den Wert der Zahl nicht. Die ersten 4 können außerdem bei der weiteren Verarbeitung durch Adreßrechnung und explizite Längenangabe ausgeschlossen werden.

Sie sehen: Eine ungepackte Zahl darf _____ gepackt werden.

Sinngemäß:

ins gleiche Feld

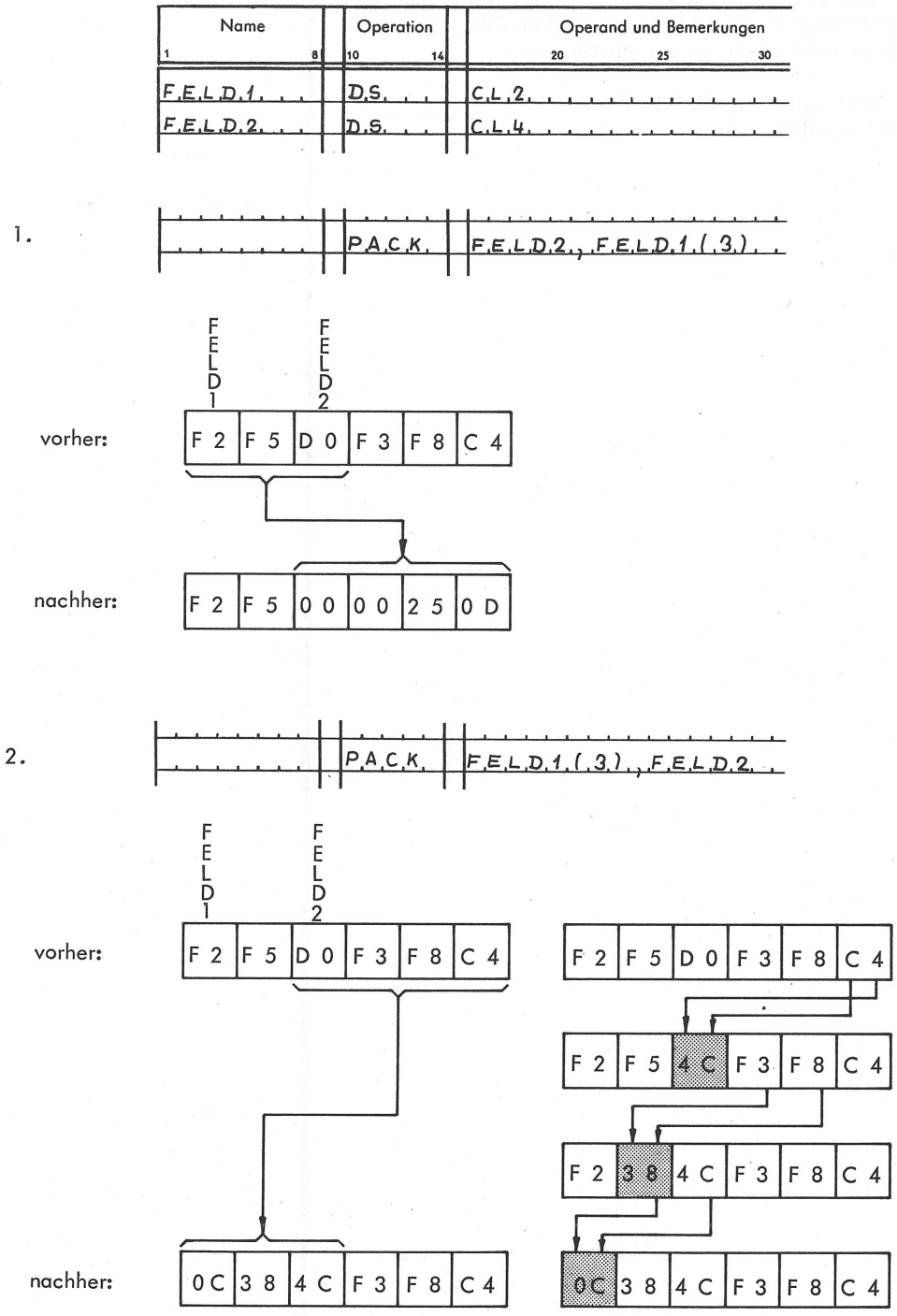

Name
Operation
Operand und Bemerkungen
1 8
10 14
20 25 30
FELD1 DS CL2
FELD2 DS CL4
1.
PACK FELD2,FELD1(3)
FELD1
FELD2
vorher:
F2 F5 D0 F3 F8 C4
nachher:
F2 F5 00 00 25 0D
2.
PACK FELD1(3),FELD2
FELD1
FELD2
vorher:
F2 F5 D0 F3 F8 C4
nachher:
0C 38 4C F3 F8 C4
F2 F5 D0 F3 F8 C4
F2 F5 4C F3 F8 C4
F2 38 4C F3 F8 C4
0C 38 4C F3 F8 C4

Abb. 3

15 Wir können uns auch den Fall vorstellen, daß der erste und der zweite Operand nicht vollständig gleich sind, sondern nur einige Bytes gemeinsam haben. Man sagt in diesem Fall: Die Operanden "überlappen" sich.

Betrachten Sie die beiden Bespiele in Abb. 3.

Wie heißt die symbolische Adresse des Bytes, das die beiden Operanden jeweils gemeinsam haben?

In welchem der beiden Beispiele entsteht durch die PACK-Operation eine gepackte Zahl?

FELD2

im 1. Beispiel

Im 2. Beispiel erhalten einige Bytes einen neuen Inhalt, bevor der alte Inhalt gepackt werden konnte.

Beim Packen sich überlappender Felder ist also Vorsicht geboten. Keine Schwierigkeiten gibt es, wenn sich die Felder vollständig überdecken (Packen ins gleiche Feld) oder wenn der erste Operand nach rechts hin über den zweiten Operanden hinausragt.

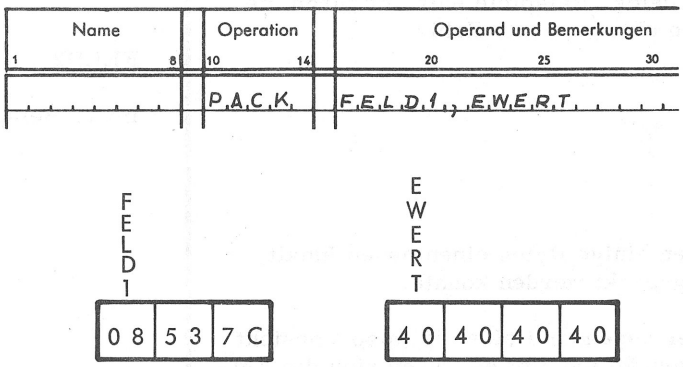

Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
P,A,C,K,
F,E,L,D,1,,E,W,E,R,T,
FELD1
0 8 5 3 7 C
EWERT
4 0 4 0 4 0 4 0

Abb. 4

16 Wir kommen jetzt zu einer anderen Frage.

In Abb. 4 enthält der 2. Operand "EWERT" 4 Blanks.

In der entsprechenden Eingabekarte ist jede der 4 Spalten eine _____-spalte.

Hexadezimal ist die Verschlüsselung eines Blanks in einem Byte __(ZZ)__.

Leer

4 0

17 Welchen Inhalt hat FELD1 in Abb. 4 nach dem Packen, hexadezimal angegeben?

FELD1

0 0	0 0	0 4

Sie sehen: Gepackte Blanks ergeben keine gepackte Null. Im rechten Halbbyte steht nach dem Packen statt eines Vorzeichens die Ziffer 4.

Soll beim Packen eine Null entstehen, so muß das betreffende Feld in der Eingabekarte wenigstens in der rechten Spalte die Ziffernlochung 0 - mit oder ohne Zonenlochung - haben.

Wir fassen zusammen, was beim Packen zu beachten ist:

1. Der zweite Operand muß vor dem Packen eine ungepackte Dezimalzahl enthalten.

2. Packen ins gleiche Feld ist erlaubt.

3. Die beiden Operanden dürfen sich außerdem so überlappen, daß der erste Operand rechts über den zweiten hinausragt.

4. Beim Packen eines Blank-Feldes entsteht keine gepackte Null.

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
	⋮	
* DEFINI	TIONEN	
	⋮	
	⋮	
EWERT	DS	CL4
EBETRAG	DS	CL3
ELOHN	DS	CL6
EBEZ	DS	CL4
	⋮	
PWERT	DS	CL5
PBETRAG	DS	CL2
	⋮	
	⋮	
	⋮	

Abb. 5

Übungen

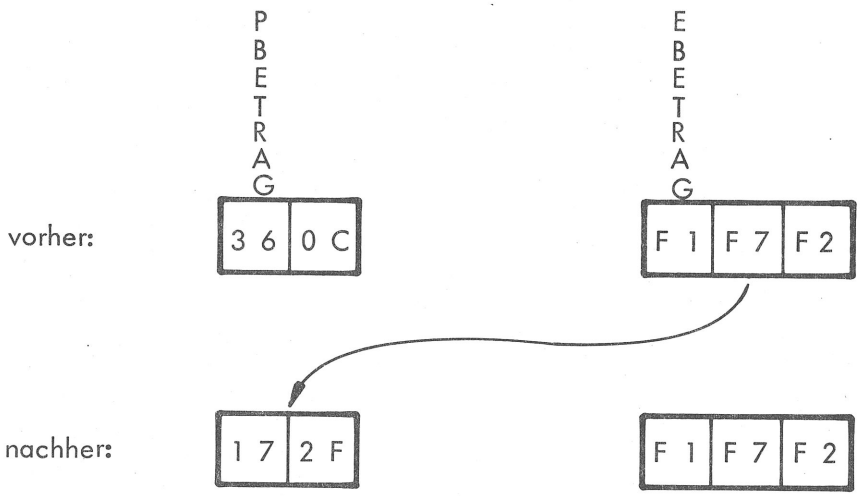
Entnehmen Sie Ihren Arbeitsmitteln Übung 410, Seite 1 und 2.

Tragen Sie Ihre Lösungen ein. Prüfen Sie die Richtigkeit anhand der folgenden Lösungsblätter, sobald Sie eine Seite vollständig bearbeitet haben.

Abb. 5 enthält die Definitionen der Speicherfelder, auf die sich die Operationen beziehen.

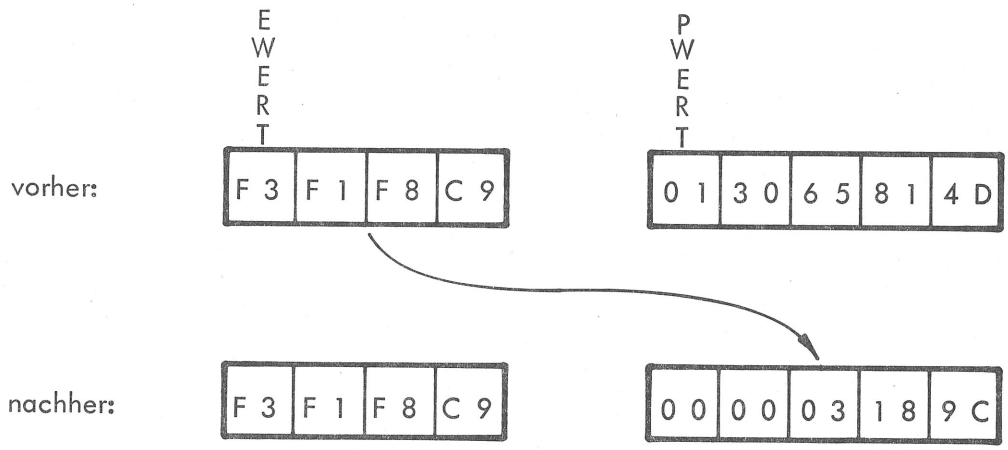
Lösungen zu den Übungen

1.



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	PACK	PBETRAG,EBETRAG

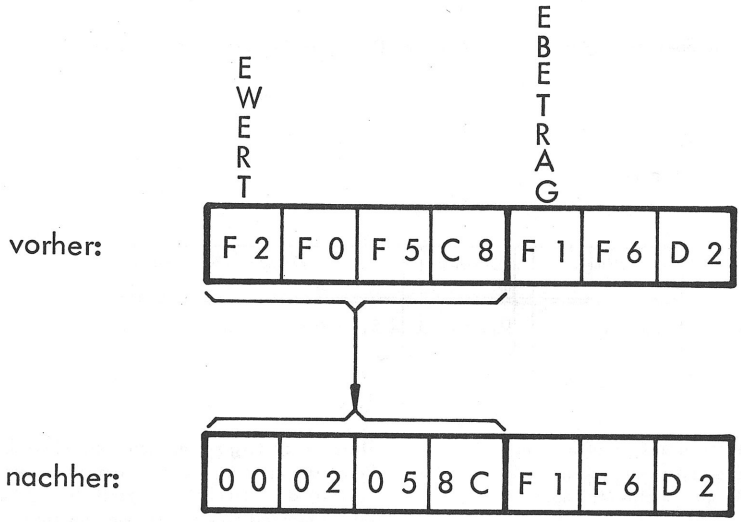
2.



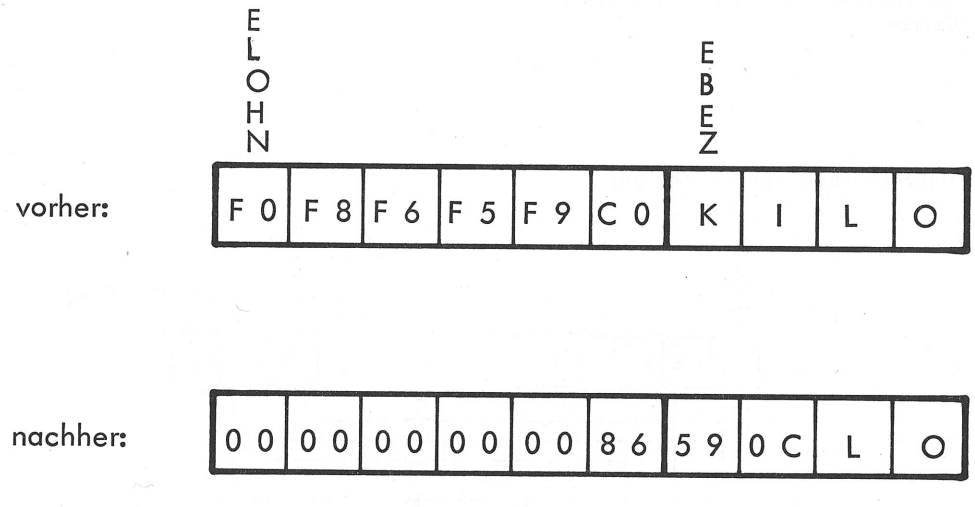
Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	PACK	PWERT,EWERT

3.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	PACK	EWERT,EWERT



4.



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	PACK	ELOHN(8),ELOHN

Merkblatt 409 enthält die Beschreibung der Operation "Packen" (vgl. Abb. 1).

UNPK

ENTPACKEN

(UNPacK)

Diese Operation ändert die Speicherungsform von numerischen Daten von "gepackt" in "ungepackt".

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	U N P K	A S U M M E , R S U M M E

Wirkungsweise:

Der gepackte Inhalt des zweiten Operanden (im Beispiel RSUMME) wird entpackt und in den ersten Operanden (im Beispiel ASUMME) übertragen.

Verarbeitet wird byteweise von rechts nach links in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die Feldlängen müssen nicht übereinstimmen. Wenn der erste Operand zu lang ist, werden links führende Nullen eingeschoben. Wenn der erste Operand zu kurz ist, werden die höchsten (linken) Ziffern aus dem zweiten Operanden nicht übertragen.

Die Feldlänge jedes Operanden darf maximal 16 Bytes betragen.

Durchführung der Operation:

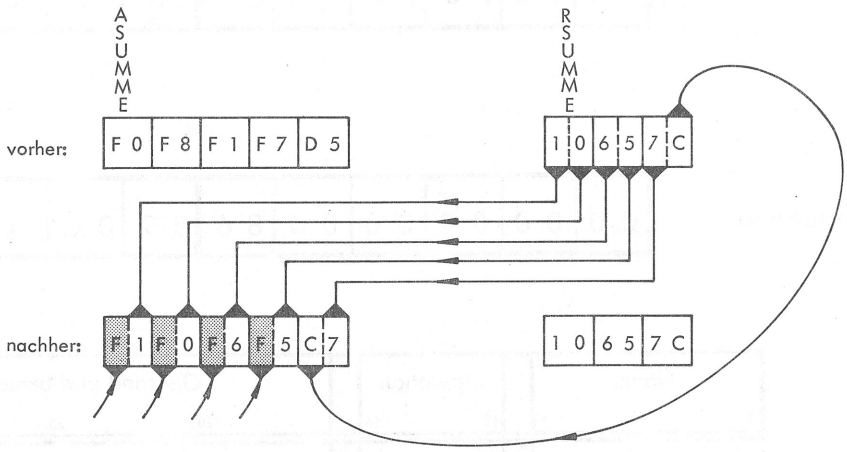


Abb. 1

1 Im Instruktionsbeispiel in Abb. 1 heißt der zweite Operand "RSUMME". Der Name deutet an: Im DVS wurde eine Rechnung durchgeführt, bei der aus mehreren Einzelwerten eine Summe gebildet wurde.

Die Summe steht im Feld RSUMME in _____ Darstellungsform.

gepackter

2 Das Rechenergebnis in RSUMME soll in eine Ausgabekarte gestanzt werden.

Bei der Behandlung gepackter Daten haben wir festgestellt:
Werden gepackte Daten in eine Karte gestanzt, so erhält jede Kartenspalte (eine, mehrere) _____ Lochungen.

mehrere
(bis zu 6 Lochungen je Spalte)

3 Wir können die errechnete Summe ungepackt ausgeben. Dazu muß vor dem Ausgabebefehl die Operation "Entpacken" durchgeführt werden.

Im Operationsbeispiel in Abb. 1 wird eine gepackte Dezimalzahl mit __(Z)__ Ziffern und einem Vorzeichen in ein Feld entpackt, das __(ZW)__ groß ist.

5

5 Bytes

4 Der erste Operand enthält nach dem Entpacken eine ungepackte oder gezonte Dezimalzahl.

In jedem rechten Halbbyte steht eine _____, während jedes linke Halbbyte eine _____ enthält.

Dezimalziffer (0-9)

Zonenverschlüsselung

UNPK

ENTPACKEN

(UNPacK)

Diese Operation ändert die Speicherungsform von numerischen Daten von "gepackt" in "ungepackt".

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	U N P K	A S U M M E , R S U M M E

Wirkungsweise:

Der gepackte Inhalt des zweiten Operanden (im Beispiel RSUMME) wird entpackt und in den ersten Operanden (im Beispiel ASUMME) übertragen.

Verarbeitet wird byteweise von rechts nach links in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die Feldlängen müssen nicht übereinstimmen. Wenn der erste Operand zu lang ist, werden links führende Nullen eingeschoben. Wenn der erste Operand zu kurz ist, werden die höchsten (linken) Ziffern aus dem zweiten Operanden nicht übertragen.

Die Feldlänge jedes Operanden darf maximal 16 Bytes betragen.

Durchführung der Operation:

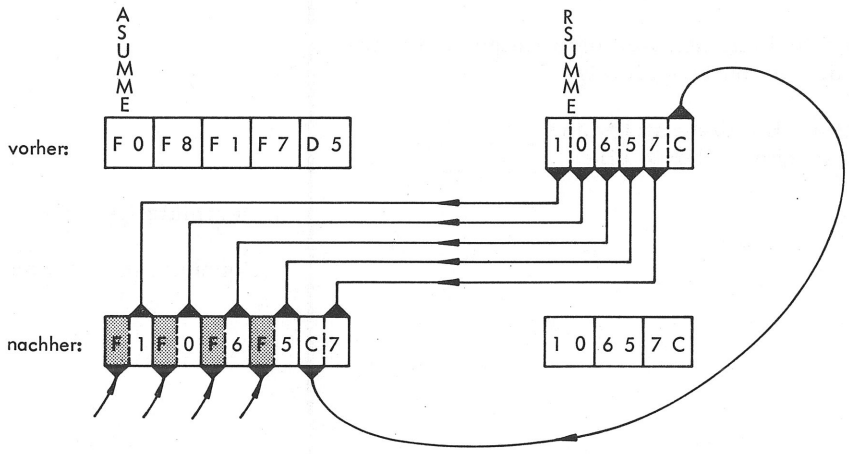


Abb. 1

5 Die Dezimalziffern werden von rechts nach links aus dem zweiten Operanden in die ___(WW)___ des ersten Operanden eingefüllt.

Bleiben links noch rechte Halbbytes übrig, so wird jeweils die Dezimalziffer __(Z)__ ergänzt.

rechten Halbbytes/Ziffernteile

0

6 Betrachten Sie den Inhalt der linken Halbbytes im ersten Operanden nach dem Entpacken.

Aus dem zweiten Operanden stammt nur die Hexadezimalziffer __(Z)__ im _____ Byte. Sie kennzeichnet das Rechenergebnis als _____.

C

rechten

positiv

7 In die übrigen linken Halbbytes des ersten Operanden wird beim Entpacken die Hexadezimalziffer __(Z)__ eingesteuert.

Bei der 5-stelligen Dezimalzahl in unserem Beispiel wird diese Hexadezimalziffer __(Z)__ mal eingesteuert.

F

4

8 Nach dem Entpacken kann das Rechenergebnis aus ASUMME zum Stanzen ausgegeben werden.
Die Einerspalte des Kartenfeldes erhält dann __(Z)__ Lochung(en) und die übrigen Spalten je __(Z)__ Lochung(en).

2

1

UNPK

ENTPACKEN

(UNPacK)

Diese Operation ändert die Speicherungsform von numerischen Daten von "gepackt" in "ungepackt".

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	UNPK	ASUMME,RSUMME

Wirkungsweise:

Der gepackte Inhalt des zweiten Operanden (im Beispiel RSUMME) wird entpackt und in den ersten Operanden (im Beispiel ASUMME) übertragen.

Verarbeitet wird byteweise von rechts nach links in der Länge des ersten Operanden.

Der Inhalt des zweiten Operanden wird nicht verändert.

Die Feldlängen müssen nicht übereinstimmen. Wenn der erste Operand zu lang ist, werden links führende Nullen eingeschoben. Wenn der erste Operand zu kurz ist, werden die höchsten (linken) Ziffern aus dem zweiten Operanden nicht übertragen.

Die Feldlänge jedes Operanden darf maximal 16 Bytes betragen.

Durchführung der Operation:

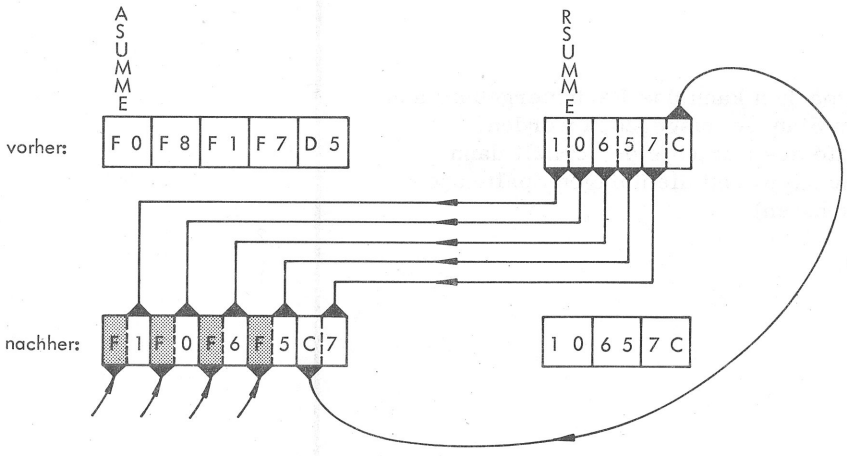


Abb. 1

9 Zur Verarbeitungsrichtung stellen wir fest, daß zuerst die _____ Bytes der beiden Felder verarbeitet werden.

Die Operation geht dann nach _____ hin weiter.

rechten

links

10 Es kann vorkommen, daß beim Entpacken der zweite Operand nicht alle Bytes des ersten Operanden ausfüllt.

Wird eine 5-stellige gepackte Dezimalzahl in ein 7 Bytes großes Feld entpackt, so werden die 2 linken Bytes mit ___(WW)___ aufgefüllt.

Diese Bytes enthalten dann beide hexadezimal __(ZZ)__.

führenden Nullen

F 0

11 Andererseits kann der erste Operand zu kurz sein.

Wir nehmen z.B. an, daß die Operanden ASUMME und RSUMME bei einer UNPK-Operation beide 3 Bytes groß sind.

In diesem Fall kann ASUMME von den 5 Ziffern aus RSUMME nur ___(ZW)___ aufnehmen.

Die restlichen linken Ziffern werden nicht nach ASUMME entpackt.

3 Ziffern

Wie bei PACK, so ergibt sich auch bei UNPK in einigen Fällen kein sinnvolles Operationsergebnis.

Denken Sie bitte an die Beispiele, in denen der erste und der zweite Operand alle oder einige Bytes gemeinsam haben. Beim Entpacken besteht in allen diesen Fällen die Gefahr, daß der alte Inhalt einiger Bytes vor dem Entpacken durch die Operation selbst zerstört wurde.

Außerdem ist "Entpacken" im allgemeinen nur sinnvoll, wenn es sich auf eine gepackte Zahl bezieht.

Wir merken uns als Richtlinie:

1. Der zweite Operand muß vor dem Entpacken eine gepackte Dezimalzahl enthalten.

2. Entpacken ins gleiche Feld oder bei sich überlappenden Feldern ist nicht erlaubt.

Wir wissen aber:
Die PACK- und die UNPK-Operation werden vom DVS o h n e Rücksicht auf die jeweiligen Feldinhalte durchgeführt, sobald ein entsprechender Befehl erteilt ist. Wenn es sich also im Einzelfall als sinnvoll erweist, kann der Programmierer z.B. einen Befehl für "Entpacken ins gleiche Feld" geben. Auch ein solcher Befehl wird vom DVS ausgeführt.

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	:	
*,D,E,F,I,N,I,T	I,O,N,E,N	
	:	
A,B,E,R,	D,S,	C,L,8,
	:	
A,W,E,R,T,	D,S,	C,L,3,
	:	
R,W,E,R,T,	D,S,	C,L,2,
	:	
F,E,L,D,1,	D,S,	C,L,3,
	:	
F,E,L,D,2,	D,S,	C,L,7,
	:	
F,E,L,D,3,	D,S,	C,L,4,

Abb. 2

Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 411, Seite 1, 2 und 3.

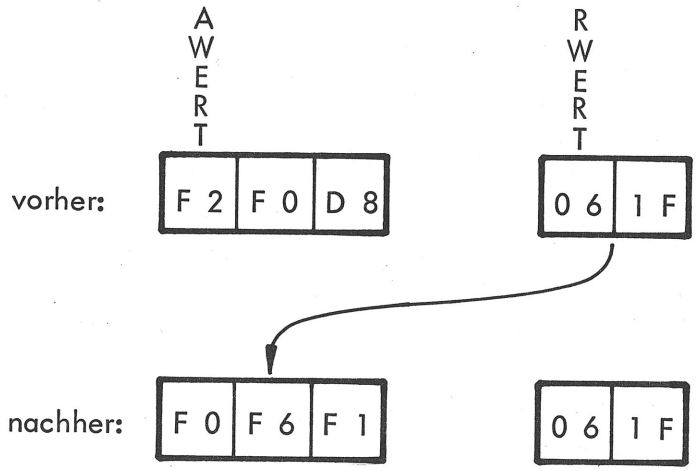
Tragen Sie Ihre Lösungen ein.

Prüfen Sie die Richtigkeit anhand der folgenden Lösungsblätter, sobald Sie eine Seite bearbeitet haben.

Abb. 2 enthält die zugehörigen Definitionen.

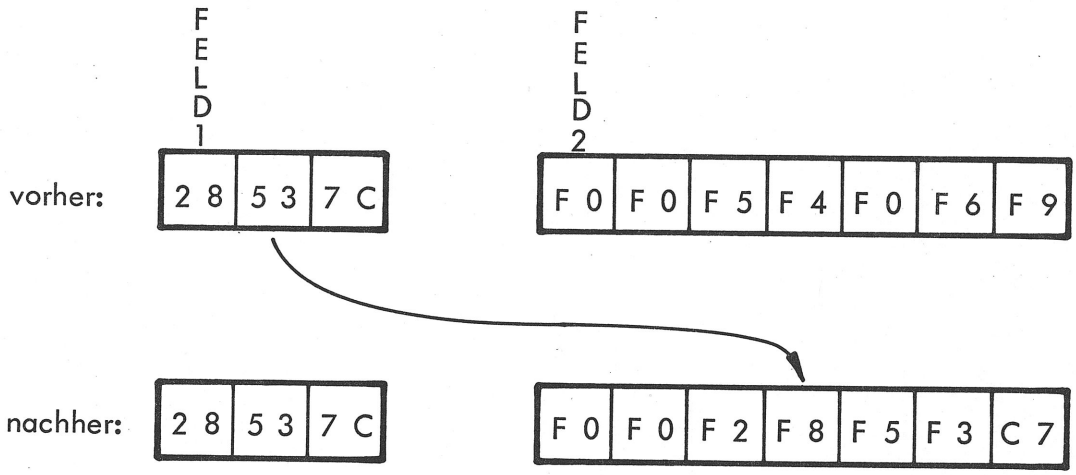
Lösungen zu den Übungen

1.



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	UNPK	AWERT,RWERT

2.



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	UNPK	FELD2,FELD1



3.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	UNPK	FELD3,FELD1

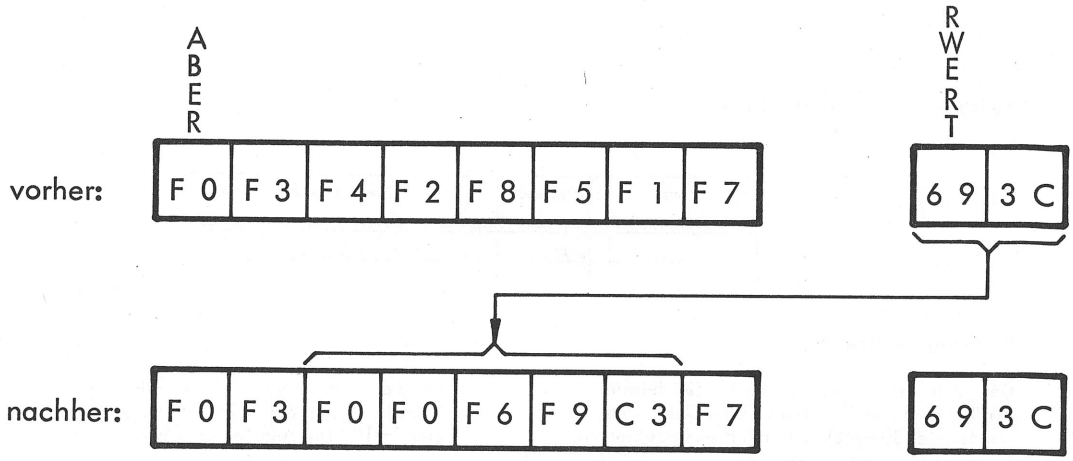
	FELD1	FELD3
vorher:	3 8 5 6 2 F	F 7 F 0 F 4 C 9
nachher:	3 8 5 6 2 F	F 8 F 5 F 6 F 2

4.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	UNPK	ABER+5(3),RWERT

	ABER	RWERT
vorher:	F 4 F 0 F 5 F 6 D 8 F 1 F 4 C 7	3 5 2 D
nachher:	F 4 F 0 F 5 F 6 D 8 F 3 F 5 D 2	3 5 2 D

5.



Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	UNPK	ABER+2(5),RWERT

Merkblatt 410 enthält die Beschreibung der Operation "Entpacken" (vgl. Abb. 1).

AP

ADDIEREN DEZIMAL (GEPACKT)

(Add decimal Packed)

Diese Operation addiert den (gepackten) Inhalt zweier Speicherfelder.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	AP	SUMME,PWERT

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel PWERT) wird zum Inhalt des ersten Operanden (im Beispiel SUMME) addiert und das Ergebnis im ersten Operanden gespeichert.

Vor Durchführung der Addition werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

Das Vorzeichen wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Ein Ergebnis von Null ist stets positiv.

Der Bedingungsschlüssel wird entweder auf "Überlauf" oder auf "positiv", "negativ" oder "null" eingestellt. Die Anzeige des Überlaufs hat Vorrang.

Ein Überlauf liegt vor, wenn der erste Operand zu kurz ist, um alle Ziffern der Summe - außer führenden Nullen - aufzunehmen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler oder Überlauf tritt Programmunterbrechung ein.

Durchführung der Operation:

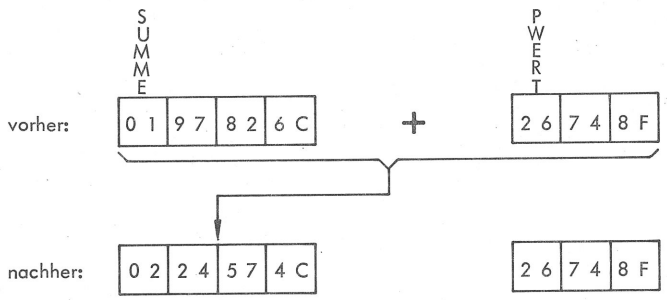


Abb. 1

1 Im Beispiel in Abb. 1 wird ein 5-stelliger Wert zu einer 7-stelligen Summe addiert.

Das Additionsergebnis wird im ___(WW)___ gespeichert.

ersten Operanden

2 Die Operation "Addieren Dezimal" hat den Operationsschlüssel _____.

Das "P" deutet an, daß beide Operanden vor der Addition ___(WW)___ enthalten müssen.

Falls die beiden Summanden nicht schon gepackt im Speicher stehen, müssen sie vor der Addition gepackt werden.

AP

gepackte Daten (Dezimalzahlen)

3 Alle Operationen, die Sie bisher kennen, werden vom DVS ohne vorherige Prüfung der jeweiligen Feldinhalte durchgeführt.

Die Addition und die übrigen dezimalen Rechenoperationen dagegen werden nicht durchgeführt, wenn nicht beide Operanden gepackten Inhalt haben.

In Abb. 1 finden Sie den Fachausdruck: Vor Ausführung der Operation werden beide Operanden auf _____ geprüft.

Gültigkeit

4 Der Inhalt eines Feldes ist gültig für eine Rechenoperation, wenn das rechte Halbbyte ein _____ enthält und alle übrigen Halbbytes _____ enthalten.

Vorzeichen

Dezimalziffern

AP ADDIEREN DEZIMAL (GEPACKT)

(Add decimal Packed)

Diese Operation addiert den (gepackten) Inhalt zweier Speicherfelder.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	AP	SUMME,PWERT

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel PWERT) wird zum Inhalt des ersten Operanden (im Beispiel SUMME) addiert und das Ergebnis im ersten Operanden gespeichert.

Vor Durchführung der Addition werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

Das Vorzeichen wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Ein Ergebnis von Null ist stets positiv.

Der Bedingungsschlüssel wird entweder auf "Überlauf" oder auf "positiv", "negativ" oder "null" eingestellt. Die Anzeige des Überlaufs hat Vorrang.

Ein Überlauf liegt vor, wenn der erste Operand zu kurz ist, um alle Ziffern der Summe - außer führenden Nullen - aufzunehmen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler oder Überlauf tritt Programmunterbrechung ein.

Durchführung der Operation:

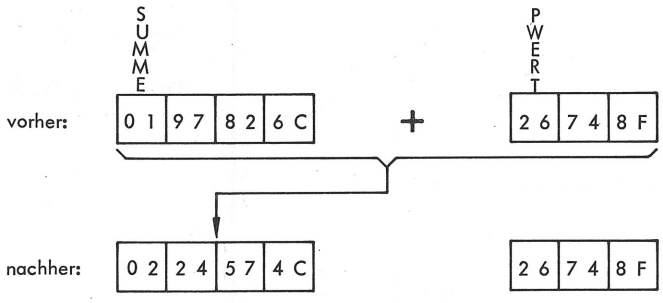


Abb. 1

5 Der Operand PWERT in einer Additions-Instruktion habe folgenden Inhalt:

a)

F 0	F 2	F 5	F 8	C 3

b)

0 0	0 0	0 3	5 6	4 F

c)

S	U	M	M	E

d)

0 3	6 5	4 D	7 3	4 2

In welchem Fall bzw. welchen Fällen wird die Operation nicht durchgeführt?

bei a), c) und d)

6 Wie bei den Operationen Packen und Entpacken, so werden auch bei der Addition die beiden Operanden im _____ Byte adressiert, während die Ausführung der Operation im _____ Byte beginnt.

linken

rechten

7 Die Adresse des rechten Bytes eines Feldes errechnet das DVS aus der Adresse und der Länge des Feldes.

Vom rechten Byte aus geht die Operation nach _____ hin weiter und addiert Byte für Byte.

links

8 Das Vorzeichen des Rechenergebnisses wird nach den Regeln der Algebra gebildet.

Verschlüsselung:

Ein positives Ergebnis hat im rechten Halbbyte den Inhalt __(Z)__ und ein negatives Ergebnis den Inhalt __(Z)__.

Bei dem Ergebnis Null enthält das rechte Halbbyte immer die Hexadezimalziffer __(Z)__.

C

D

C

AP

ADDIEREN DEZIMAL (GEPACKT)

(Add decimal Packed)

Diese Operation addiert den (gepackten) Inhalt zweier Speicherfelder.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	A,P,	S,U,M,M,E,,P,W,E,R,T,

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel PWERT) wird zum Inhalt des ersten Operanden (im Beispiel SUMME) addiert und das Ergebnis im ersten Operanden gespeichert.

Vor Durchführung der Addition werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

Das Vorzeichen wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Ein Ergebnis von Null ist stets positiv.

Der Bedingungsschlüssel wird entweder auf "Überlauf" oder auf "positiv", "negativ" oder "null" eingestellt. Die Anzeige des Überlaufs hat Vorrang.

Ein Überlauf liegt vor, wenn der erste Operand zu kurz ist, um alle Ziffern der Summe - außer führenden Nullen - aufzunehmen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler oder Überlauf tritt Programmunterbrechung ein.

Durchführung der Operation:

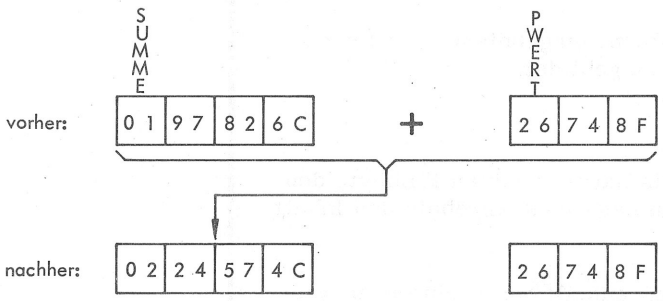


Abb. 1

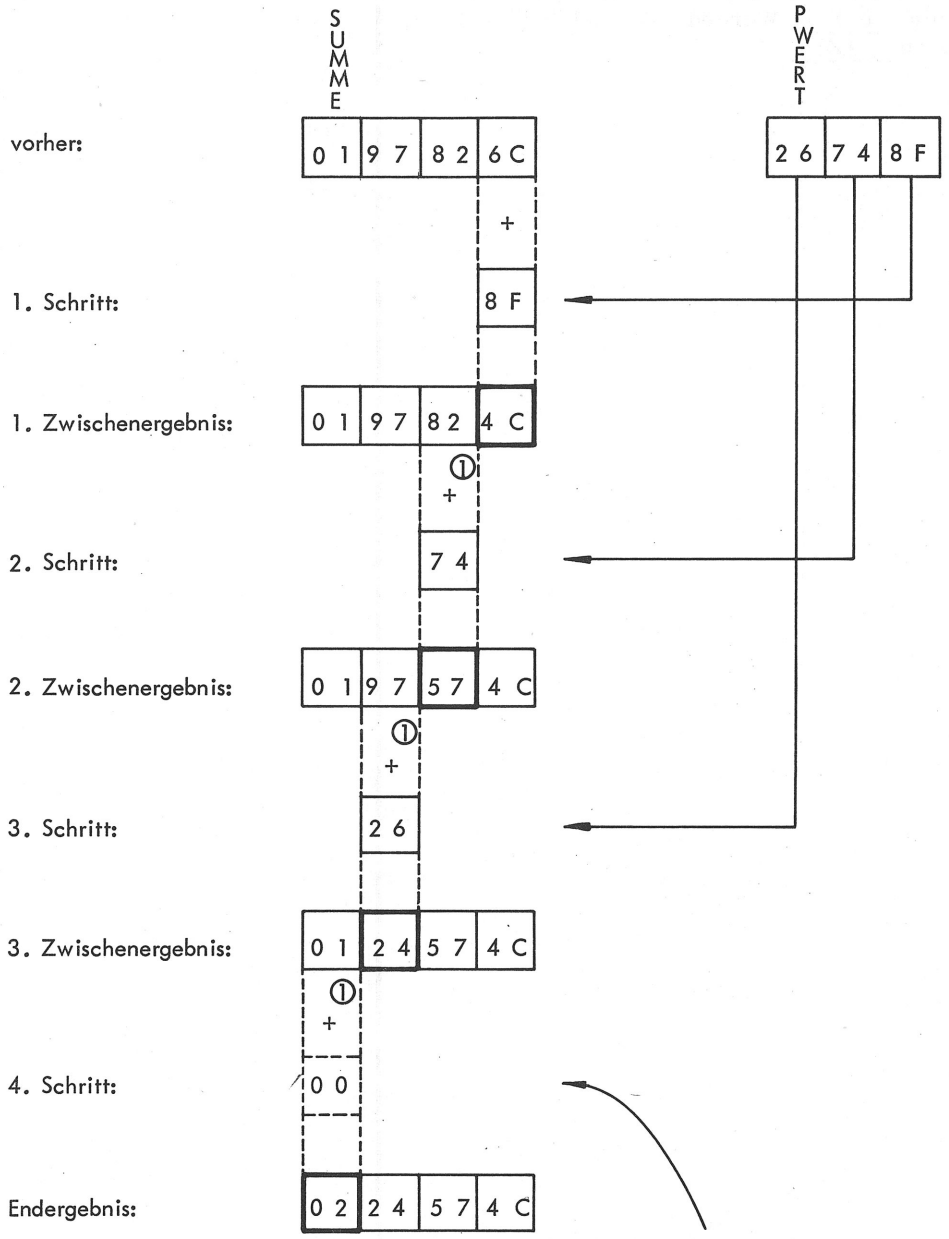
9 Bei einer Addition kann ein negatives Ergebnis entstehen, wenn einer der Operanden oder beide vor der Addition eine _____ Zahl enthalten.

Werden die Zahlen "+7" und "-3" addiert, so heißt das Ergebnis __(Z)__. Werden "-7" und "+3" addiert, so ergibt sich __(Z)__.

negative

+4

-4



"①" = Zehnerübertrag

Abb. 2

10 Schauen Sie sich bitte Abb. 2 genau an und verfolgen Sie den Ablauf der Addition.

Die Operation wird in der Länge des (längeren, kürzeren) _____ Operanden durchgeführt.

längeren

11 Weil PWERT um 1 Byte kürzer ist als SUMME, wird zum linken Byte von SUMME ein Byte mit dem angenommenen Inhalt _(ZZ)_ addiert.

0 0

12 Wir stellen uns die Frage, warum die Operation nicht auf die Länge des kürzeren Feldes beschränkt wird.

Würde in unserem Beispiel das linke Byte von SUMME von der Additions-Operation ausgeschlossen, so wäre das Ergebnis (richtig, falsch) _____.

In diesem Fall würde nämlich der _____ in das linke Byte nicht aufgenommen. Das linke Byte von SUMME hätte nach der Operation den alten Inhalt _(ZZ)_.

falsch

Zehnerübertrag

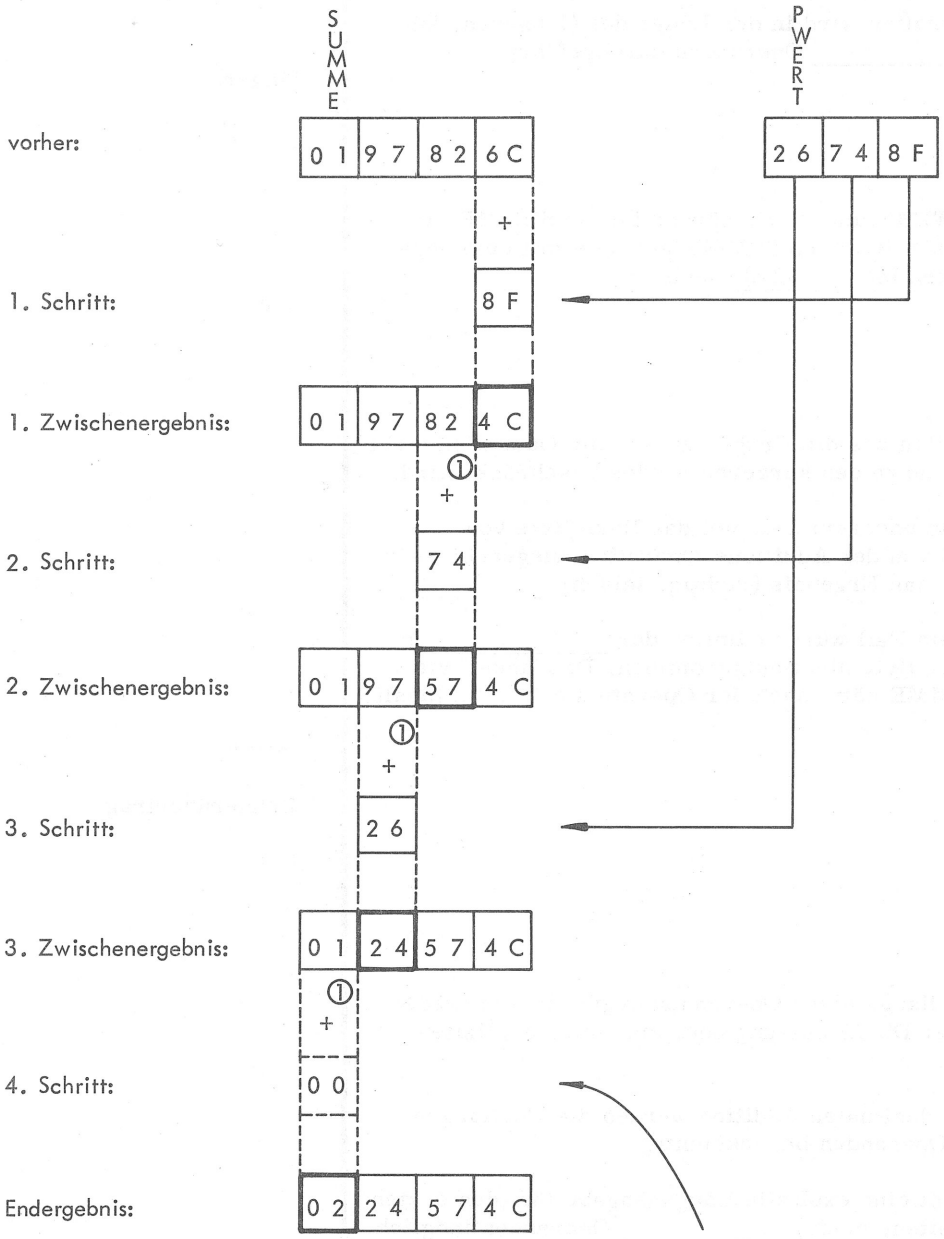
0 1

13 Die Feldlänge eines Operanden ergibt sich entweder aus einer DS-Anweisung oder aus einer expliziten Angabe.

Bei der dezimalen Addition werden die Feldlängen beider Operanden berücksichtigt.

Daher ist eine explizite Längenangabe für (den ersten, den zweiten, beide) _____ Operanden möglich.

beide



"①" = Zehnerübertrag

Abb. 2

14 Außer der Additionsoperation kennen wir schon mehrere andere Operationen.

Ordnen Sie die Operationsschlüssel

MVC, CLC, PACK, UNPK, AP

den folgenden Aussagen zu:

a) Bei der Durchführung der Operation wird nur die Länge des ersten Operanden berücksichtigt. Explizite Längenangabe ist nur für den ersten Operanden möglich.

b) Explizite Längenangabe ist nur für den zweiten Operanden möglich.

c) Explizite Längenangabe ist für beide Operanden möglich.

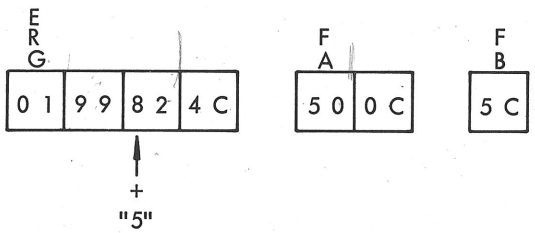
a) MVC, CLC
(- logische Operationen)

b) —

c) PACK, UNPK, AP

15 Wir führen jetzt eine kleine Übung durch.

Es werden folgende Felder benutzt:



Auf die Hunderterstelle von ERG soll eine 5 addiert werden.

Welche der folgenden Instruktionen sind falsch und warum?

	Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
a)		AP	ERG(3),FA(1)
b)		AP	ERG+2(2),FA
c)		AP	ERG(3),FB
d)		AP	ERG,FA
e)		AP	ERG+1(3),FA

Überprüfen Sie Ihre Antworten auf der nächsten Seite!

a) Grund: Die beiden Operanden enthalten im rechten Halbbyte die Ziffern 2 und 0, haben also keinen "gültigen" Inhalt für eine Addition.

b) Grund: Ein evtl. auftretender Zehnerübertrag nach links hin wird von ERG nicht aufgenommen. In unserem Beispiel hat ERG nach dieser Operation den Inhalt 01 99 32 4C anstelle von 02 00 32 4C.

c) Grund: Der erste Operand enthält im rechten Halbbyte keine Vorzeichenverschlüsselung, hat also keinen "gültigen" Inhalt für eine Addition.

e) Grund: Ein evtl. auftretender Zehnerübertrag in das linke Byte von ERG wird nicht aufgenommen. In unserem Beispiel hat ERG nach dieser Operation den Inhalt 01 00 32 4C anstelle von 02 00 32 4C.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30 35
	AP	SUMMAND1,SUMMAND2

a)

	SUMMAND1				SUMMAND2			
vorher:	41	30	0C		07	20	01	4F
nachher:					07	20	01	4F

b)

	SUMMAND1				SUMMAND2			
vorher:	41	30	0C		00	14	20	0F
nachher:					00	14	20	0F

Abb. 3

16 Bei einer Additions-Operation darf auch der erste Operand kürzer sein als der zweite. Der Programmierer muß nur darauf achten, daß er groß genug ist, um das Ergebnis aufzunehmen.

Abb. 3 zeigt 2 Beispiele. Geben Sie an, welchen Inhalt SUMMAND1 nach Ausführung der Addition hat:

a) (ZZ)

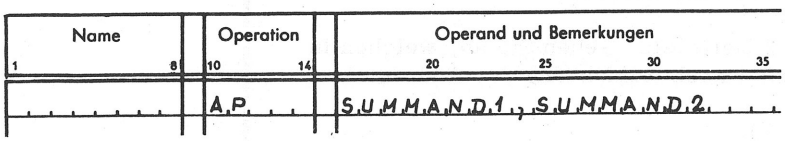
b) (ZZ)

In welchem Fall ist die Additions-Instruktion nicht sinnvoll?

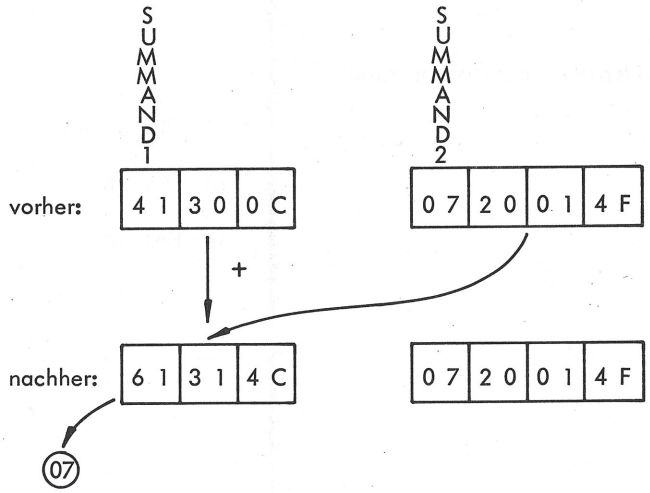
a) 61 31 4C

b) 55 50 0C

im Fall a)


Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
35
A,P
S,U,M,M,A,N,D,1,,S,U,M,M,A,N,D,2,

a)


SUMMAND 1
SUMMAND 2
vorher:
4 1 3 0 0 C
0 7 2 0 0 1 4 F
+
nachher:
6 1 3 1 4 C
0 7 2 0 0 1 4 F
07

b)

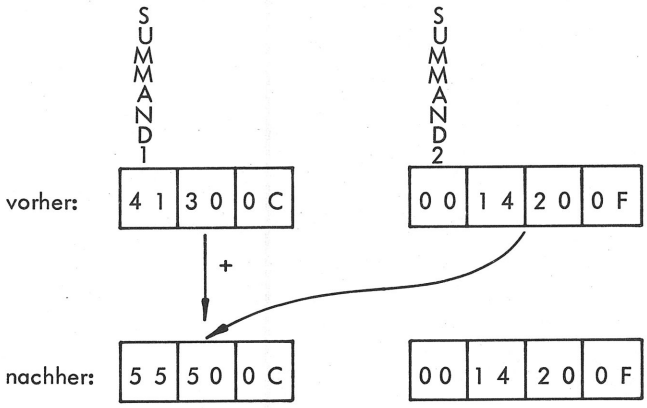

SUMMAND 1
SUMMAND 2
vorher:
4 1 3 0 0 C
0 0 1 4 2 0 0 F
+
nachher:
5 5 5 0 0 C
0 0 1 4 2 0 0 F

Abb. 4

17 Abb. 4 enthält die Ergebnisse.

Im Fall a) ist der erste Summand zu klein für das ganze Ergebnis.
SUMMAND1 läuft über.
Der Überlauf wird im _____ angezeigt.

Bedingungsschlüssel

18 Wir nehmen an, daß der Additions-Instruktion in Abb. 4 zwei Verzweigungsbefehle folgen:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
	A,P,	SUMMAND1,SUMMAND2,
	BO,	A7,
	B,	A3,

Dann wird nach der Addition im Fall a) zum Befehl _____ verzweigt und im Fall b) zum Befehl _____.

A7

A3

19 Der Operationsschlüssel BO taucht selten auf in einem Programm. Der Programmierer sorgt dafür, daß die Ergebnisfelder für jeden möglichen Fall groß genug sind.

Zeigt der Bedingungsschlüssel nach einer Addition keinen "Überlauf" an, so ist er entweder auf "_____" oder auf "_____" oder auf "_____" eingestellt.

positiv

negativ

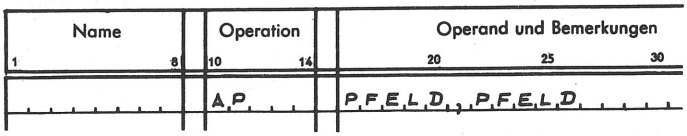
null

20 Im Fall b) in Abb. 4 steht der Bedingungsschlüssel nach der Addition auf _____.

Bei welchem Ergebnis (hexadezimal) in 4 Bytes steht der Bedingungsschlüssel auf null?

positiv

bei dem Ergebnis 00 00 00 0C



PFELD

vorher:	00	25	36	2F
nachher:	00	50	72	4C

Abb. 5

21 Abb. 5 zeigt eine Additionsoperation besonderer Art.

PFELD ist der Name des ___(WW)___ Operanden.

Die Prüfung auf Gültigkeit ergibt:
_____.

ersten und zweiten

gültig

(Beide Operanden enthalten gültige (gepackte) Daten, nämlich rechts ein Vorzeichen und im übrigen Ziffern.)

22 Die Addition des Feldes PFELD zu sich selbst kann durchgeführt werden.

Dabei wird der Inhalt von PFELD _____.

verdoppelt

23 Im Beispiel in Abb. 5 ist der Ausgangswert in PFELD positiv.

Wie würde das Ergebnis der Addition lauten, wenn PFELD vorher den Inhalt 00 25 36 2D hätte?

00 50 72 4D

AP

ADDIEREN DEZIMAL (GEPACKT)

(Add decimal Packed)

Diese Operation addiert den (gepackten) Inhalt zweier Speicherfelder.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	A,P, . .	S,U,M,M,E,,,P,W,E,R,T, ...

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel PWERT) wird zum Inhalt des ersten Operanden (im Beispiel SUMME) addiert und das Ergebnis im ersten Operanden gespeichert.

Vor Durchführung der Addition werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

Das Vorzeichen wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Ein Ergebnis von Null ist stets positiv.

Der Bedingungsschlüssel wird entweder auf "Überlauf" oder auf "positiv", "negativ" oder "null" eingestellt. Die Anzeige des Überlaufs hat Vorrang.

Ein Überlauf liegt vor, wenn der erste Operand zu kurz ist, um alle Ziffern der Summe - außer führenden Nullen - aufzunehmen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler oder Überlauf tritt Programmunterbrechung ein.

Durchführung der Operation:

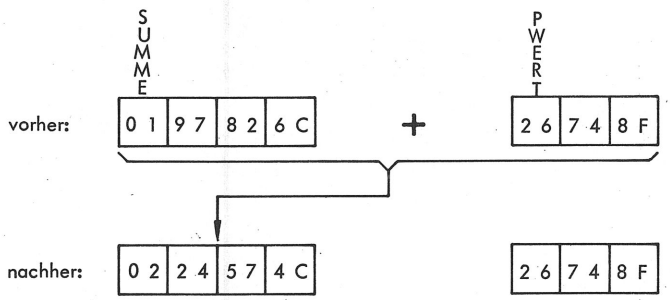


Abb. 1

Tritt bei einer Additionsoperation ein Gültigkeitsfehler oder Überlauf auf, so wird er vom DVS angezeigt. Gleichzeitig findet eine Programmunterbrechung statt. Was damit gemeint ist und wie man diese Programmunterbrechung behebt, darauf können wir in diesem Lehrgang nicht eingehen.

Zu Übung 412, Seite 1:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
ERG	DS	CL3
WERT	DS	CL2
	AP	ERG,WERT

Zu Übung 412, Seite 2:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
FELD1	DS	CL3
FELD2	DS	CL5

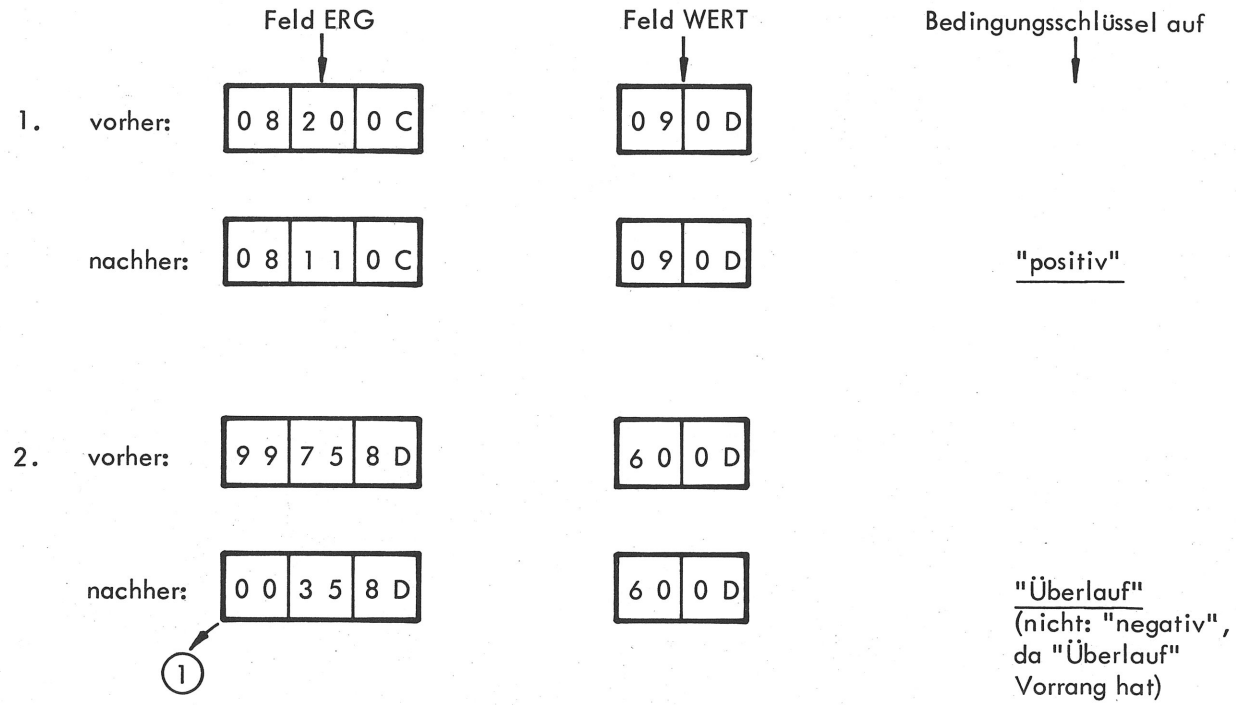
Abb. 6

Übungen

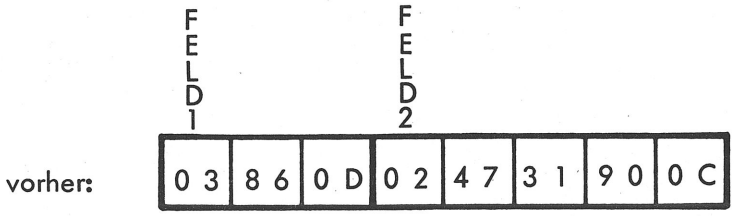
Entnehmen Sie Ihren Arbeitsmitteln Übung 412, Seite 1, 2 und 3.

Tragen Sie Ihre Lösungen ein.
Abb. 6 enthält die Additions-Instruktion zur Seite 1 und die Definitionen zu den Seiten 1 und 2.

Sobald Sie eine Seite bearbeitet haben, prüfen Sie bitte die Richtigkeit anhand der folgenden Lösungsblätter.

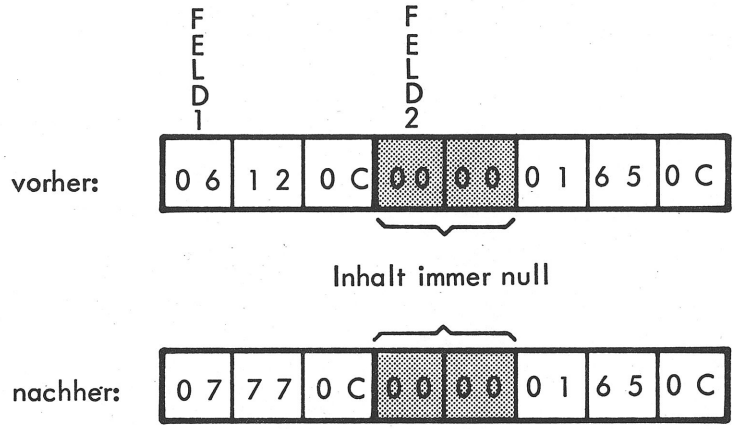

Feld ERG
Feld WERT
Bedingungsschlüssel auf
1.
vorher:
0 8 2 0 0 C
0 9 0 D
nachher:
0 8 1 1 0 C
0 9 0 D
"positiv"
2.
vorher:
9 9 7 5 8 D
6 0 0 D
nachher:
0 0 3 5 8 D
6 0 0 D
"Überlauf"
(nicht: "negativ",
da "Überlauf"
Vorrang hat)
1
3.
vorher:
0 0 0 5 0 D
9 2 0 C
nachher:
0 0 8 7 0 C
9 2 0 C
"positiv"
4.
vorher:
0 0 0 8 7 D
0 8 7 C
nachher:
0 0 0 0 0 C
0 8 7 C
"null"
5.
vorher:
0 0 1 9 3 C
1 9 3 D
nachher:
0 0 0 0 0 C
1 9 3 D
"null"

6.



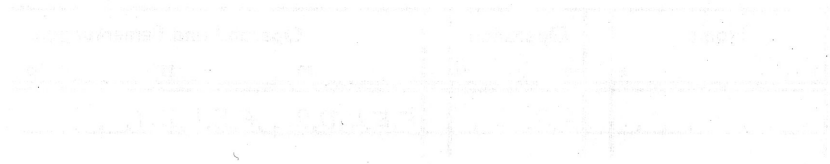
Name	Operation	Operand und Bemerkungen
1 ... 8	10 ... 14	20 ... 25 ... 30
	AP	FELD2,FELD1

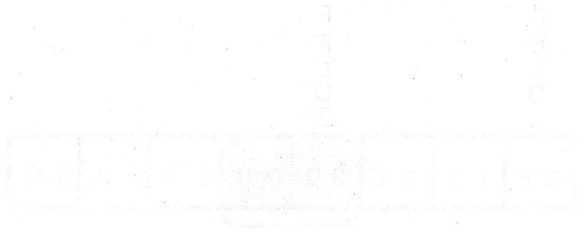
7.

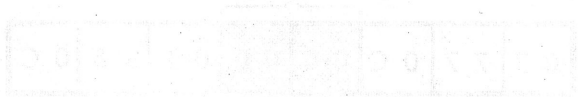


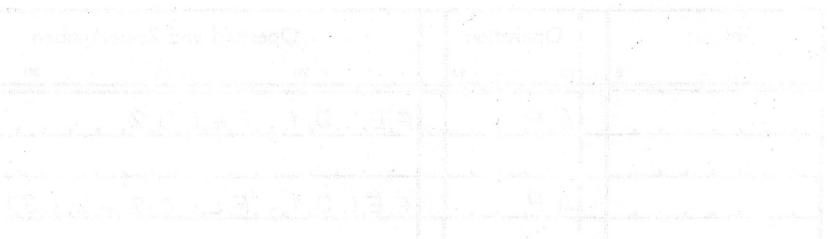
	Name	Operation	Operand und Bemerkungen
	1 ... 8	10 ... 14	20 ... 25 ... 30
1. Lösung:		AP	FELD1,FELD2
2. Lösung:		AP	FELD1,FELD2+2(,3)











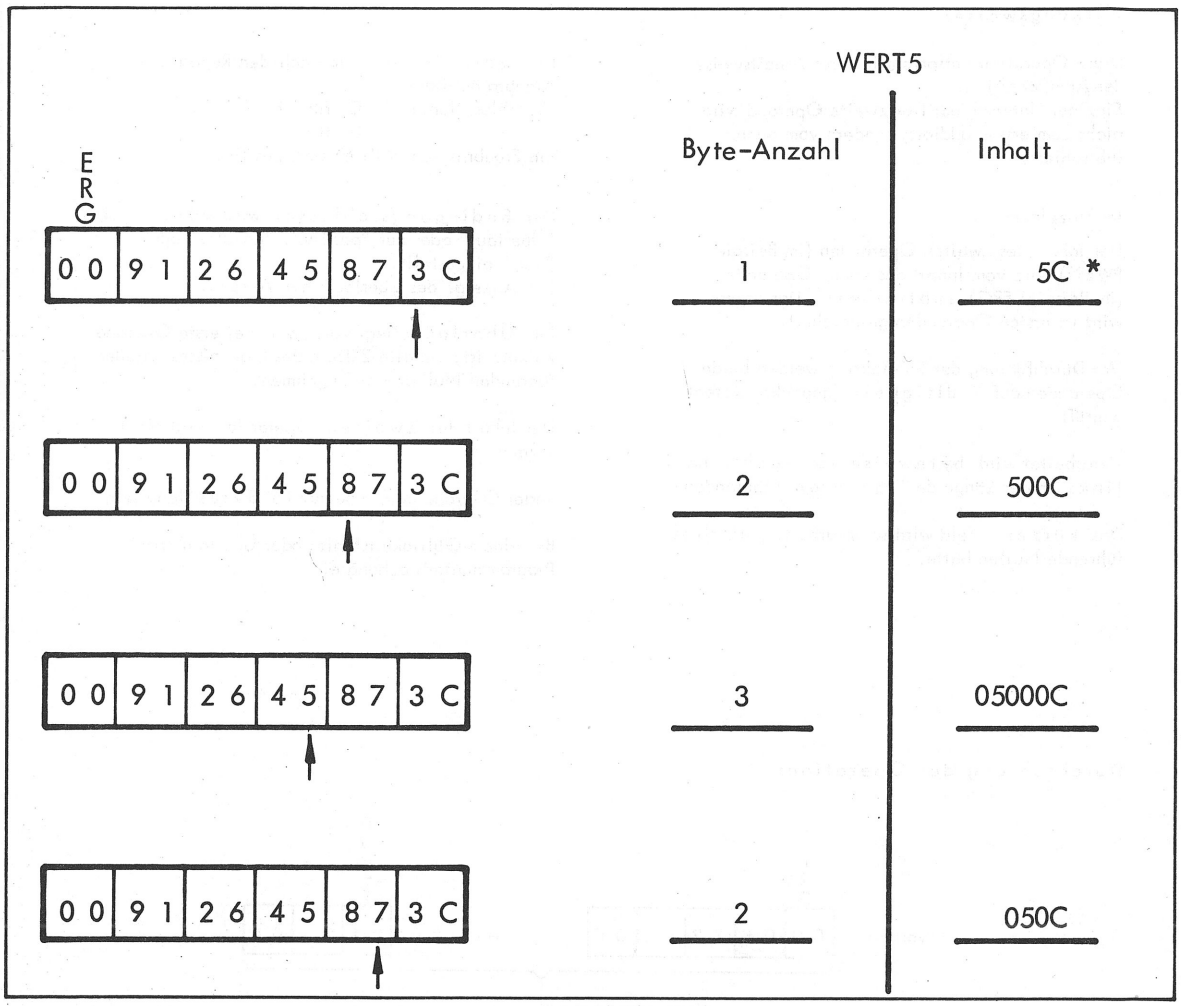
8.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	AP	ERG,WERT5

Diese Übung befaßt sich mit dem Runden von Ergebnissen durch Addition einer Konstanten "5".

Der Pfeil zeigt jeweils auf die zu rundende Stelle.

Geben Sie bitte an, wie der Wert "5" in den einzelnen Fällen gespeichert sein muß. Beachten Sie, daß bei einer Addition beide Operanden auf Gültigkeit geprüft werden.



* Statt C kann natürlich auch F oder eine andere positive Vorzeichenverschlüsselung gewählt werden.

Merkblatt 411 enthält die Beschreibung der Operation "Addieren Dezimal (Gepackt)" (vgl. Abb. 1).

SP

SUBTRAHIEREN DEZIMAL (GEPACKT)

(Subtract decimal Packed)

Diese Operation berechnet die Differenz aus dem (gepackten) Inhalt zweier Speicherfelder.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	SP	ERG,PWERT

Wirkungsweise:

Diese Operation entspricht in ihrer Arbeitsweise der AP-Operation.
Einziger Unterschied: Der zweite Operand wird nicht zum ersten addiert, sondern vom ersten subtrahiert.

Im Einzelnen:

Der Inhalt des zweiten Operanden (im Beispiel PWERT) wird vom Inhalt des ersten Operanden (im Beispiel ERG) subtrahiert. Das Ergebnis wird im ersten Operanden gespeichert.

Vor Durchführung der Subtraktion werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

Das Vorzeichen wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Ein Ergebnis von Null ist stets positiv.

Der Bedingungsschlüssel wird entweder auf "Überlauf" oder auf "positiv", "negativ" oder "null" eingestellt.
Die Anzeige des Überlaufs hat Vorrang.

Ein Überlauf liegt vor, wenn der erste Operand zu kurz ist, um alle Ziffern des Ergebnisses - außer führenden Nullen - aufzunehmen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler oder Überlauf tritt Programmunterbrechung ein.

Durchführung der Operation:

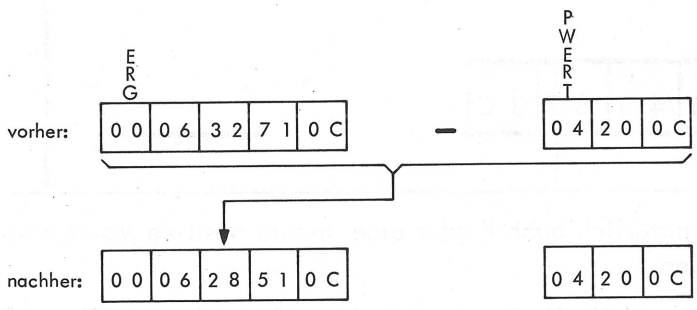


Abb. 1

Wie Sie in Abb. 1 gelesen haben, entspricht die Operation "Subtrahieren Dezimal" in ihrer Arbeitsweise der Operation "Addieren Dezimal".

Wir besprechen daher nur die Einzelheiten, die bei der Subtraktion besonders zu beachten sind.

Erinnern Sie sich bitte anhand der folgenden Beispiele an die Vorzeichenregeln:

+2 - (+3) = +2 - 3 = -1
+2 - (-3) = +2 + 3 = +5
-2 - (+3) = -2 - 3 = -5
-2 - (-3) = -2 + 3 = +1

1 Führen Sie jetzt die folgende Übung durch:

1. Operand	2. Operand	Ergebnis	
minus		Wert	Vorzeichen-verschlüsselung
+350	+15	(Z)	(Z)
-300	+10	(Z)	(Z)
-2650	-100	(Z)	(Z)
-250	+840	(Z)	(Z)
+80	-30	(Z)	(Z)

+335	C
-310	D
-2550	D
-1090	D
+110	C

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	SP	RFELD,RFELD

1.

RFELD

vorher: | 00 | 38 | 05 | 62 | 1C |

nachher: | 00 | 00 | 00 | 00 | 0C |

2.

RFELD

vorher: | 03 | 91 | 4D |

nachher: | 00 | 00 | 0C |

Abb. 2

2 Bei der Addition haben wir den Sonderfall besprochen, daß beide Operanden gleich sind. Eine solche Operation verdoppelt den positiven bzw. negativen Feldinhalt.

Entsprechend kann eine Subtraktionsoperation zwei gleiche Operanden haben.

Betrachten Sie Abb. 2.

Wird ein positiver oder negativer gepackter Feldinhalt von sich selbst subtrahiert, so entsteht als Ergebnis eine ___(WW)___.

positive gepackte Null

3 Ein Datenfeld kann also durch Subtraktion von sich selbst auf null gelöscht werden.

Voraussetzung ist nur, daß das Feld vorher eine ___(WW)___ enthält.

gepackte Dezimalzahl

Zu Übung 413, Seite 1:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	:	
WERT1	DS	CL3
WERT2	DS	CL3
	:	
	SP	WERT1,WERT2
	:	

Zu Übung 413, Seite 2:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	:	
WERT1	DS	CL2
WERT2	DS	CL5
WERT3	DS	CL4
WERT4	DS	CL3
	:	

Abb. 3

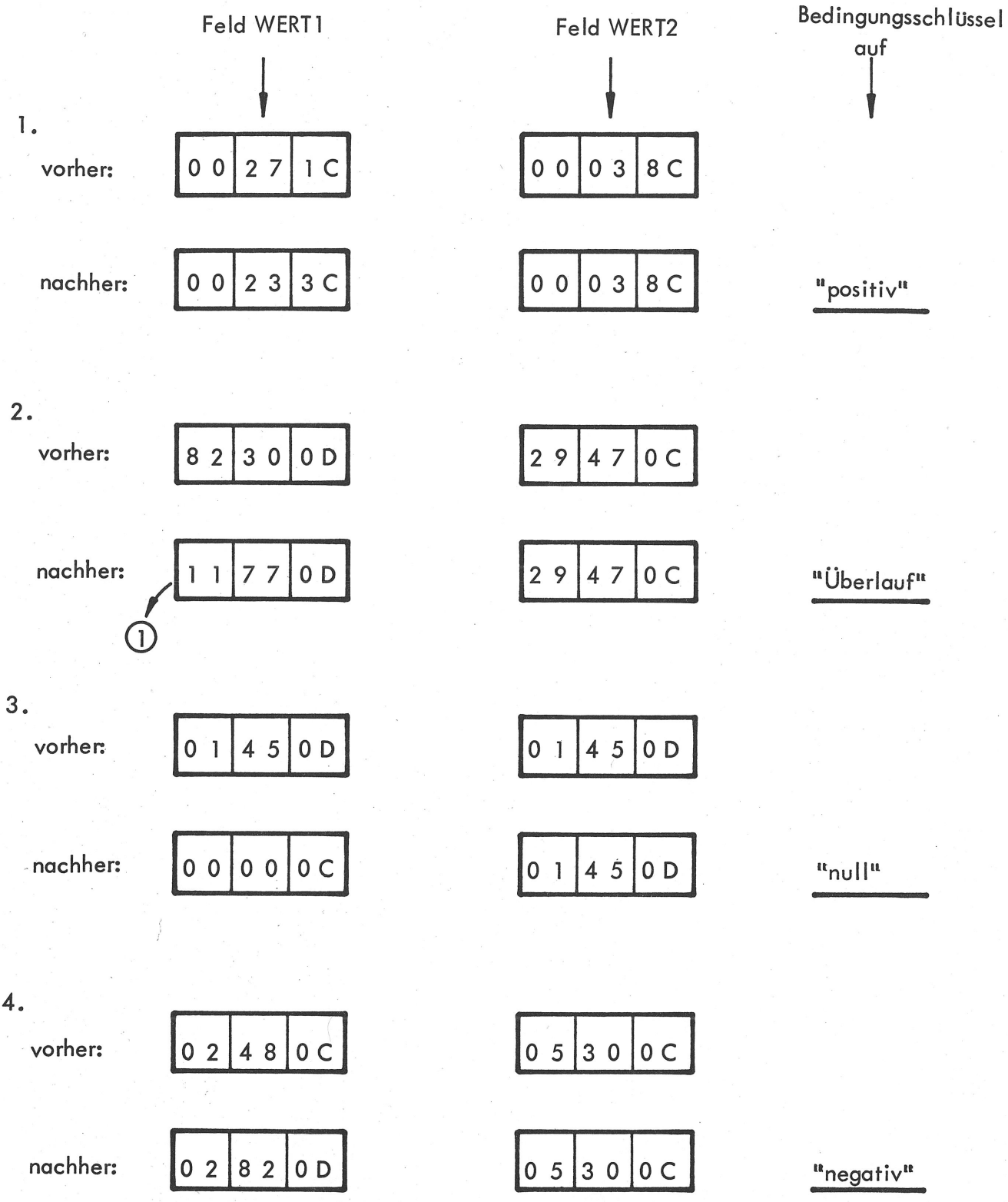
Übungen

Entnehmen Sie bitte Ihren Arbeitsmitteln Übung 413,
Seite 1 und 2.

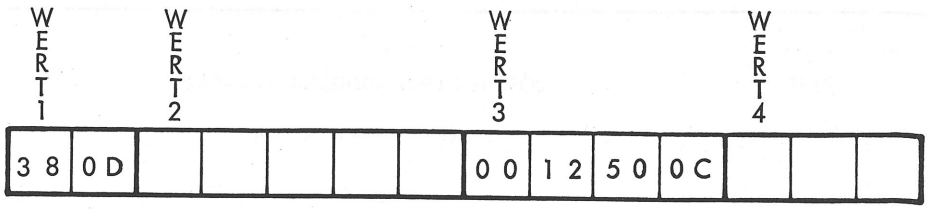
Tragen Sie Ihre Lösungen ein.
Abb. 3 enthält die Subtraktions-Instruktion zur Seite 1
und die Definitionen zu den Seiten 1 und 2.

Sobald Sie eine Seite bearbeitet haben, prüfen Sie bitte die
Richtigkeit anhand der folgenden Lösungsblätter.

Lösungen zu den Übungen

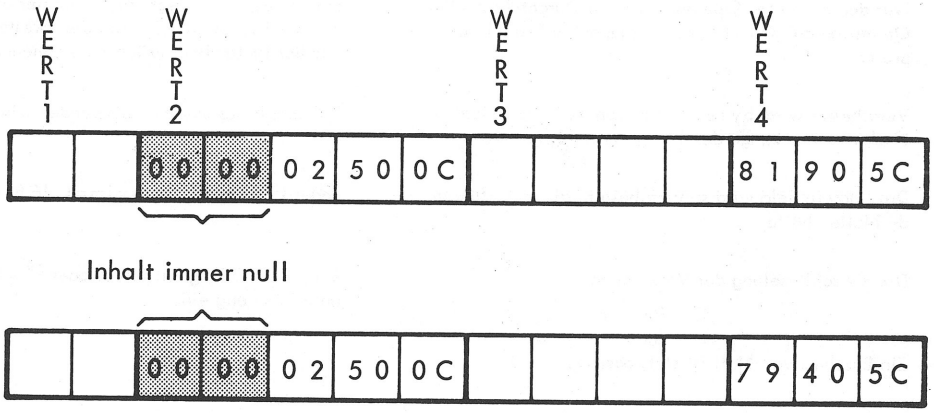


5.



Name	Operation	Operand und Bemerkungen
	S,P,	W,E,R,T,3,,W,E,R,T,1,

6.



	Name	Operation	Operand und Bemerkungen
1. Lösung:		S,P,	W,E,R,T,4,,W,E,R,T,2,
2. Lösung:		S,P,	W,E,R,T,4,,W,E,R,T,2,+,2,(,3,)

Merkblatt 412 enthält die Beschreibung der Operation "Subtrahieren Dezimal (Gepackt)" (vgl. Abb. 1).

Band 1, Seite 119

ZAP — LÖSCHEN UND ADDIEREN (GEPACKT)

(Zero and Add Packed)

Diese Operation ist gleichbedeutend mit der Addition eines Speicherfeldes in ein anderes Feld, das den Wert Null enthält.

Beispiel einer Instruktion:

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	ZAP	RFELD,PWERT

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel PWERT) wird so in den ersten Operanden (im Beispiel RFELD) gebracht, als ob er zum Inhalt Null im ersten Operanden addiert würde.

Nur der zweite Operand wird vor Durchführung der Operation auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

Die Verschlüsselung der Vorzeichen:

C für +
D für -

Ein Ergebnis von Null ist stets positiv.

Der Bedingungsschlüssel wird entweder auf "Überlauf" oder auf "positiv", "negativ" oder "null" eingestellt.
Die Anzeige des Überlaufs hat Vorrang.

Ein Überlauf liegt vor, wenn der erste Operand zu kurz ist, um alle Ziffern des zweiten Operanden - außer führenden Nullen - aufzunehmen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler oder Überlauf tritt Programmunterbrechung ein.

Durchführung der Operation:

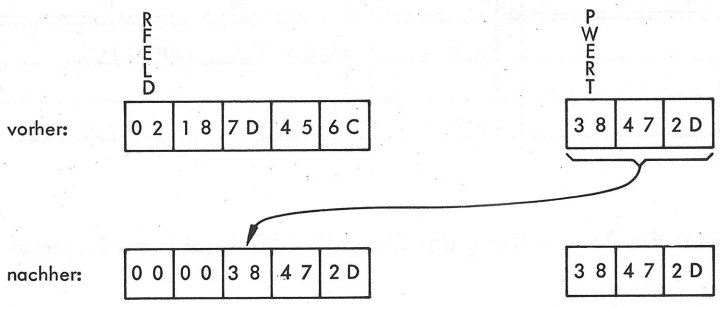


Abb. 1

1 Sie kennen bereits die Operation "Addieren Dezimal (Gepackt)".

Die in Abb. 1 beschriebene Operation "___(WW)___" ist mit ihr verwandt.

Der Operationsschlüssel AP ist erweitert um den Buchstaben _____.

Löschen und Addieren (Gepackt)

Z

2 "Z" ist die Abkürzung des englischen Wortes "Zero" (Null).

Die Operation entspricht einer Addition des zweiten Operanden zu _____.

Null

3 In unserem Beispiel hat der erste Operand RFELD eine Länge von ___(ZW)___ und der zweite Operand PWERT eine Länge von ___(ZW)___.

Infolgedessen werden __(Z)__ Bytes verarbeitet.

5 Bytes

3 Bytes

5

4 Bei der Operation "Löschen und Addieren (Gepackt)" wird nur der ___(WW)___ auf Gültigkeit geprüft.

Der erste Operand darf vor der Operation jeden beliebigen Inhalt haben.

zweite Operand

ZAP — LÖSCHEN UND ADDIEREN (GEPACKT)

(Zero and Add Packed)

Diese Operation ist gleichbedeutend mit der Addition eines Speicherfeldes in ein anderes Feld, das den Wert Null enthält.

Beispiel einer Instruktion:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
	ZAP	RFELD,PWERT

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel PWERT) wird so in den ersten Operanden (im Beispiel RFELD) gebracht, als ob er zum Inhalt Null im ersten Operanden addiert würde.

Nur der zweite Operand wird vor Durchführung der Operation auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

Die Verschlüsselung der Vorzeichen:

C für +
D für -

Ein Ergebnis von Null ist stets positiv.

Der Bedingungsschlüssel wird entweder auf "Überlauf" oder auf "positiv", "negativ" oder "null" eingestellt.
Die Anzeige des Überlaufs hat Vorrang.

Ein Überlauf liegt vor, wenn der erste Operand zu kurz ist, um alle Ziffern des zweiten Operanden - außer führenden Nullen - aufzunehmen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler oder Überlauf tritt Programmunterbrechung ein.

Durchführung der Operation:

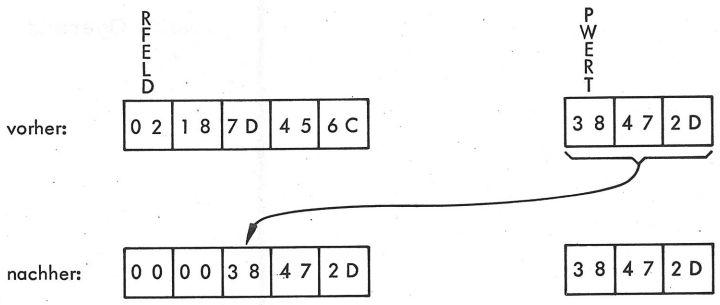


Abb. 1

5 Im Beispiel in Abb. 1 wird ein 3 Bytes großes Feld mit der Operation "Löschen und Addieren" in ein 5 Bytes großes Feld gebracht.

Danach hat das Ergebnis im ersten Operanden _(Z)_ führende Nullen.

Nach der Operation enthalten immer mindestens so viele Bytes führende Nullen, wie der erste Operand _____ ist als der zweite Operand.

4

länger

6 Ist der erste Operand zu kurz, um den zweiten Operanden ohne seine führenden Nullen aufzunehmen, so zeigt der Bedingungsschlüssel "_____" an.

Überlauf

7 Entsteht kein Überlauf im ersten Operanden, so zeigt der Bedingungsschlüssel entsprechend dem Ergebnis eine der drei Möglichkeiten _(WW)_ an.

"positiv", "negativ" oder "null"

8 Das Ergebnis einer ZAP-Operation erhält immer dann die Hexadezimalziffer C als Vorzeichenverschlüsselung, wenn im zweiten Operanden ein Wert steht, der größer oder _____ null ist.

gleich

9 Die Operation ZAP bietet neben SP eine zweite Möglichkeit, ein Datenfeld auf null zu löschen.

SP kann nur dann benutzt werden, wenn das zu löschende Feld eine Dezimalzahl in _____ Form enthält.

Der Feldinhalt wird gelöscht, indem er _(WW)_ wird.

gepackter

Sinngemäß:

von sich selbst subtrahiert

ZAP — LÖSCHEN UND ADDIEREN (GEPACKT)

(Zero and Add Packed)

Diese Operation ist gleichbedeutend mit der Addition eines Speicherfeldes in ein anderes Feld, das den Wert Null enthält.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	Z A P . .	R F E L D , , P W E R T . . . . .

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel PWERT) wird so in den ersten Operanden (im Beispiel RFELD) gebracht, als ob er zum Inhalt Null im ersten Operanden addiert würde.

Nur der zweite Operand wird vor Durchführung der Operation auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

Die Verschlüsselung der Vorzeichen:

C für +
D für -

Ein Ergebnis von Null ist stets positiv.

Der Bedingungsschlüssel wird entweder auf "Überlauf" oder auf "positiv", "negativ" oder "null" eingestellt.
Die Anzeige des Überlaufs hat Vorrang.

Ein Überlauf liegt vor, wenn der erste Operand zu kurz ist, um alle Ziffern des zweiten Operanden - außer führenden Nullen - aufzunehmen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler oder Überlauf tritt Programmunterbrechung ein.

Durchführung der Operation:

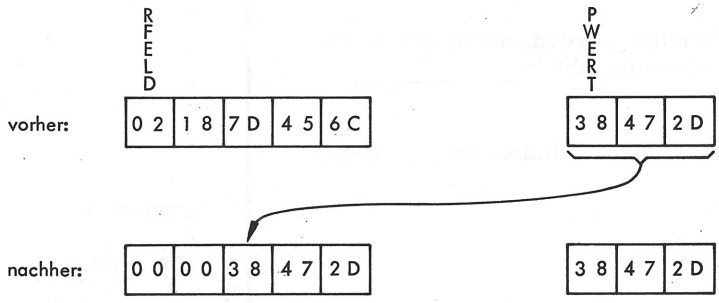


Abb. 1

10 Bei ZAP wird nur der zweite Operand auf Gültigkeit geprüft. Eine Bedingung für den Inhalt des zu löschenden Feldes gibt es hier also nicht.

Bei ZAP muß aber eine andere Voraussetzung erfüllt sein: Der zweite Operand muß die Zahl "0" enthalten, wenn das Feld, daß durch den 1. Operanden bezeichnet wird, auf null gelöscht werden soll.

Diese Null muß in _____ Form gespeichert sein.

gepackter

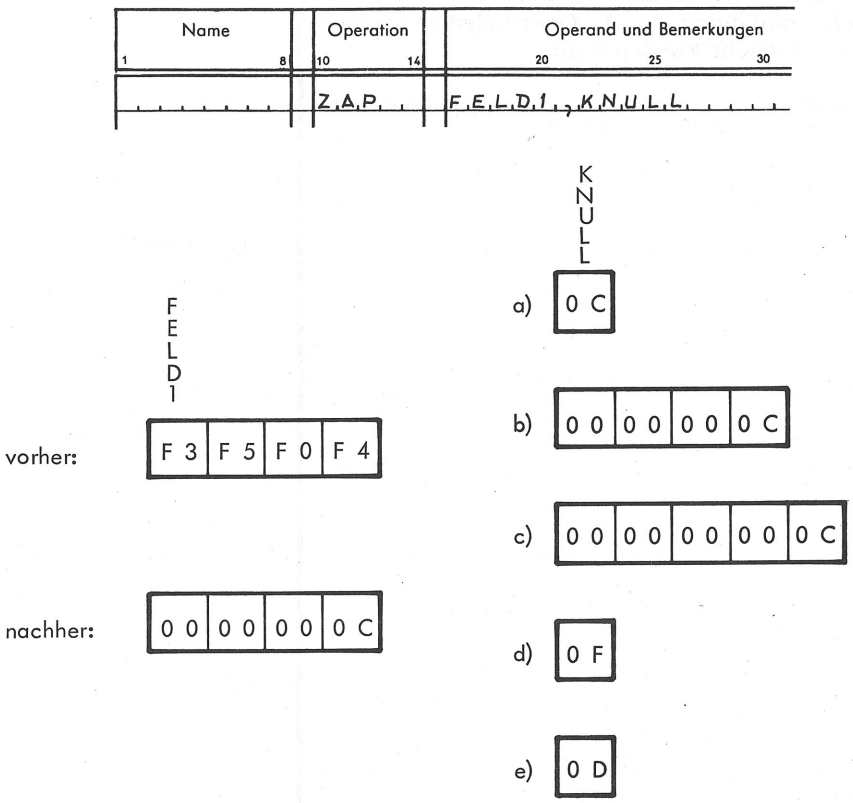

Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
Z,A,P,
F,E,L,D,1,,K,N,U,L,L,
KNULL
FELD1
vorher:
F 3 F 5 F 0 F 4
nachher:
0 0 0 0 0 0 0 C
a)
0 C
b)
0 0 0 0 0 0 0 C
c)
0 0 0 0 0 0 0 0 0 C
d)
0 F
e)
0 D

Abb. 2

11 Betrachten Sie Abb. 2.

Bei welchem der Beispiele a) bis e) wird der erste Operand wie gewünscht gelöscht?

bei jedem der Beispiele a) bis e)

Wir merken uns:
Die Operation ZAP wird benutzt, wenn ein Feld, das keinen gepackten Inhalt hat, auf den gepackten Inhalt null gebracht werden soll. Als zweiter Operand dient eine gepackte Null, die ein oder mehrere Bytes lang ist.

ZAP — LÖSCHEN UND ADDIEREN (GEPACKT)

(Zero and Add Packed)

Diese Operation ist gleichbedeutend mit der Addition eines Speicherfeldes in ein anderes Feld, das den Wert Null enthält.

Beispiel einer Instruktion:

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	ZAP	RFELD,PWERT

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel PWERT) wird so in den ersten Operanden (im Beispiel RFELD) gebracht, als ob er zum Inhalt Null im ersten Operanden addiert würde.

Nur der zweite Operand wird vor Durchführung der Operation auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

Die Verschlüsselung der Vorzeichen:

C für +
D für –

Ein Ergebnis von Null ist stets positiv.

Der Bedingungsschlüssel wird entweder auf "Überlauf" oder auf "positiv", "negativ" oder "null" eingestellt.
Die Anzeige des Überlaufs hat Vorrang.

Ein Überlauf liegt vor, wenn der erste Operand zu kurz ist, um alle Ziffern des zweiten Operanden – außer führenden Nullen – aufzunehmen.

Der Inhalt des zweiten Operanden wird nicht verändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler oder Überlauf tritt Programmunterbrechung ein.

Durchführung der Operation:

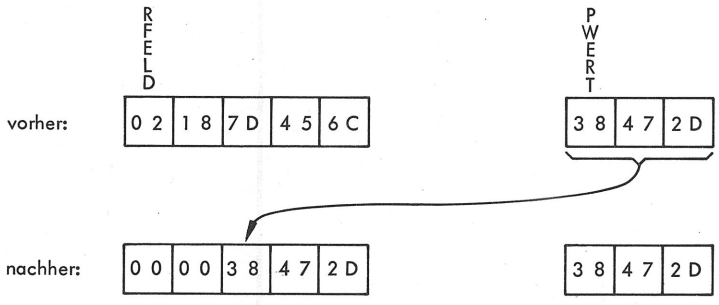


Abb. 1

Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 414, Seite 1 und 2.

Tragen Sie Ihre Lösungen ein.
Prüfen Sie Ihre Eintragungen Blatt für Blatt anhand der folgenden Lösungsblätter.

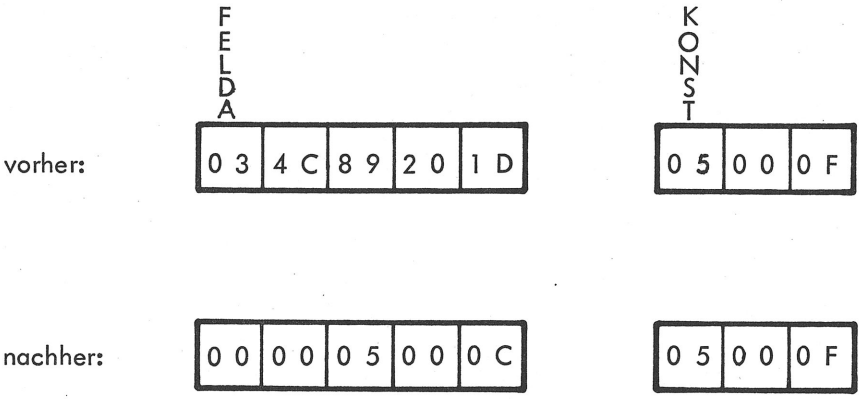
Benutzen Sie, wenn nötig, Abb. 1 als Hilfsmittel.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30 35
	ZAP	FELD1,FELD2

	FELD1 (vorher)	FELD2 (vorher)	Falls die Operation durchführbar ist: FELD1 (nachher)	Falls die Operation durchführbar ist: Bedingungsschlüssel (nachher)
1.	03658C3D	1C	0000001C	"positiv"
2.	F3C7	00369 15C	915C	"Überlauf"
3.	F4C7D9	F0F0F0C0		
4.	04156 84C	035D	0000035D	"negativ"
5.	389D	00002 57C	257C	"positiv"
6.	00043D578C	C0		
7.	0C6EF5	170F	00170C	"positiv"



8.



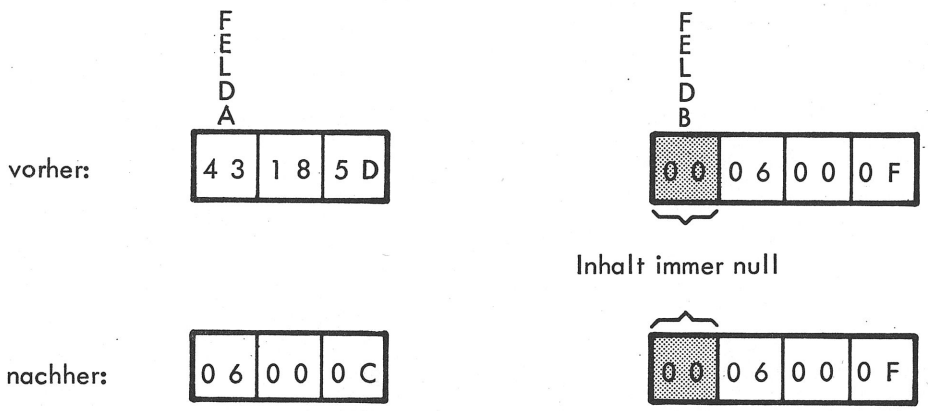
Codieren Sie:

a) die Definition der Konstanten KONST,

b) die erforderliche Instruktion.

	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20, 25, 30)
a)	K,O,N,S,T,	D,C,	X,',5,Ø,Ø,Ø,F,'
b)		Z,A,P,	F,E,L,D,A,,K,O,N,S,T,

9.



Codieren Sie für dieses Beispiel zwei verschiedene ZAP-Instruktionen.

	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20, 25, 30)
a)		Z,A,P,	F,E,L,D,A,,F,E,L,D,B,
b)		Z,A,P,	F,E,L,D,A,,F,E,L,D,B,+,1,(,3,)

Merkblatt 413 enthält die Beschreibung der Operation "Löschen und Addieren (Gepackt)" (vgl. Abb. 1).

FRAGEN ZUM KAPITEL III

Überprüfen Sie bitte anhand der folgenden Fragen Ihre Kenntnisse bzgl. der Additions- und Subtraktionsoperationen für Dezimalzahlen sowie der Operationen "Packen" und "Entpacken".

1. Die folgenden Fragen beziehen sich auf die Operationen PACK, UNPK, AP, SP und ZAP.

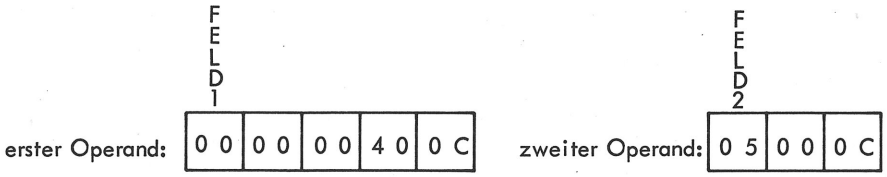
   Ein Operandenfeld im Speicher hat den Inhalt:

F 1	F 2	F 0	F 5

   a) Welche der obigen Operationen wird nicht durchgeführt, falls dieses Feld erster Operand ist?

   b) Welche der obigen Operationen wird nicht durchgeführt, falls dieses Feld zweiter Operand ist?

2. Vor der Ausführung der folgenden Instruktionen haben die Operanden folgenden Inhalt:



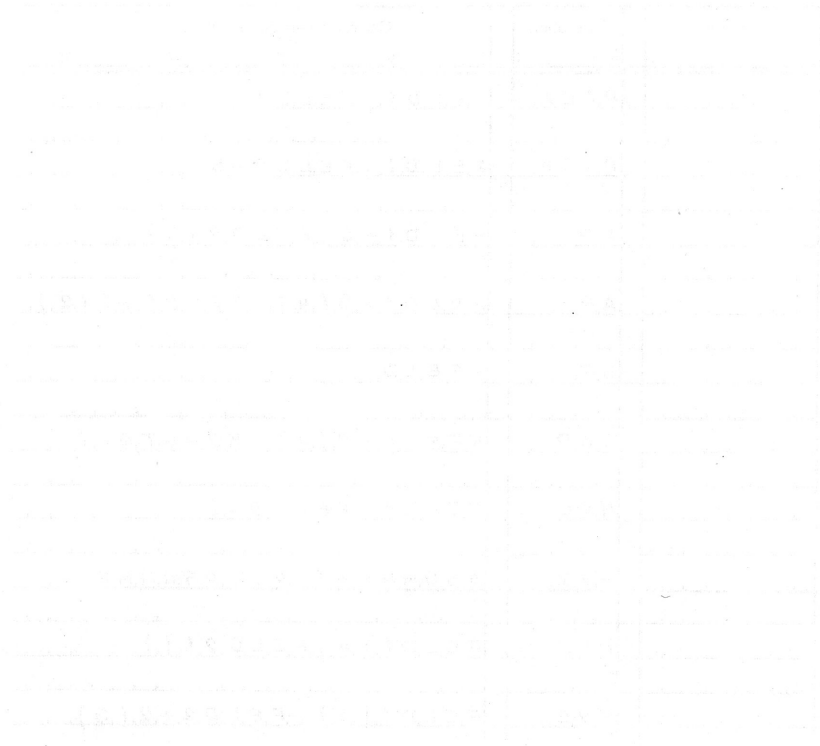
Geben Sie den Inhalt des ersten Operanden nach der Ausführung der folgenden Instruktionen an:

	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
a)		AP	FELD1,FELD2
b)		SP	FELD1,FELD2
c)		ZAP	FELD1,FELD2
d)		UNPK	FELD1,FELD2
e)		AP	FELD1,FELD1
f)		SP	FELD1,FELD1
g)		ZAP	FELD1,FELD1



3. Welche der folgenden Instruktionen sind schon von ihrer äußeren Form her nicht richtig?

	Name (1 … 8)	Operation (10 … 14)	Operand und Bemerkungen (20 … 25 … 30 … 35)
a)		P,A,C,K,	F,E,L,D,1,,F,E,L,D,1,
b)		P,A,C,K,	F,E,L,D,1,,F,E,L,D,2,+,5,
c)		A,P,	F,E,L,D,1,+,5,,F,E,L,D,2,(,4,)
d)		A,P,	F,E,L,D,1,+,3,(,4,),,F,E,L,D,2,+,2,(,2,)
e)		S,P,	R,F,E,L,D,
f)		Z,A,P,	R,F,E,L,D,+,2,(,4,),,K,O,N,S,T,ϕ,+,1,
g)		M,V,C,	F,E,L,D,1,,F,E,L,D,2,+,6,
h)		M,V,Z,	A,S,U,M,M,E,+,1,(,4,),,A,S,U,M,M,E,
i)		M,V,N,	F,E,L,D,1,+,4,,F,E,L,D,2,(,1,)
k)		M,V,C,	F,E,L,D,1,(,5,),,F,E,L,D,2,+,2,(,3,)



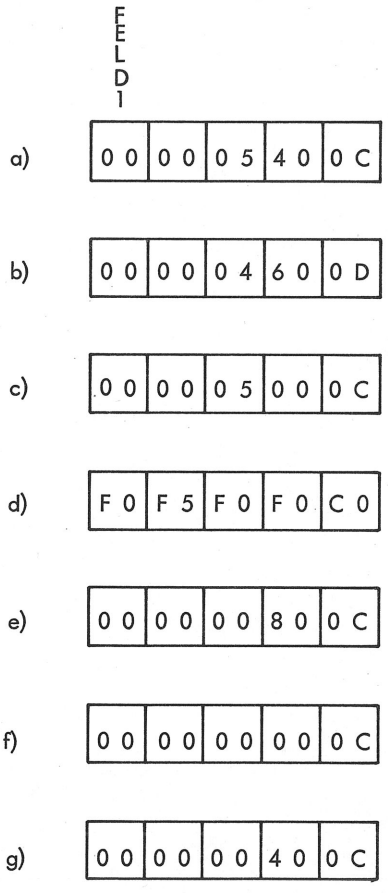
ANTWORTEN ZUM KAPITEL III

Wenn bei der Überprüfung Ihrer Lösungen Unklarheiten auftauchen, dann beseitigen Sie diese. Wiederholen Sie gegebenenfalls den in Klammern angegebenen Lernstoff zu diesem Kapitel.

1. a) AP und SP
   b) AP, SP und ZAP

   } (Seiten 146, 168, 184, 210 und 222)

2.



(Seite 184 ff.)

(Seite 210 ff.)

(Seite 222 ff.)

(Seite 168 ff.)

(Seite 184 ff.)

(Seite 210 ff.)

(Seite 222 ff.)

3. e) (Der zweite Operand fehlt) (Seite 210)

i) (Eine explizite Längenangabe ist bei logischen Operationen nur für den ersten Operanden möglich.) (Seite 42 ff.)

k) (Begründung wie zu i)) (Seite 8 ff.)

Sie haben jetzt fünf Operationen kennengelernt, die sich auf dezimales Rechnen beziehen.
Die Beschreibung der einzelnen Operationen ist in den Merkblättern 409 bis 413 enthalten.

Darüberhinaus sind die Hauptmerkmale dieser fünf Operationen im Merkblatt 414 zusammengestellt. Sehen Sie sich bitte dieses Merkblatt genau an.

Beantworten Sie mit Hilfe dieses Merkblatts die folgenden Fragen.

1 Bei allen fünf Operationen ist die Verarbeitungsrichtung ___(WW)___.

Eine Gültigkeitsprüfung vor Durchführung der Operation findet statt:

a) für den ersten Operanden bei den Operationen _____,

b) für den zweiten Operanden bei den Operationen _____.

von rechts nach links (◄——)

AP, SP

AP, SP, ZAP

2 Ist bei AP ein Operand länger als der andere, so wird die Addition in der Länge des _____ Operanden durchgeführt.

Ist dabei der erste Operand zu kurz, um alle Ziffern des Ergebnisses - außer führenden Nullen - aufzunehmen, so entfallen die _____ Ziffern. Außerdem wird ein _____ angezeigt.

längeren

linken

Fehler/Überlauf

3 Bei welchen der fünf Operationen wird kein Fehler angezeigt, wenn der erste Operand nicht alle Ziffern aufnehmen kann?

PACK und UNPK

4 Bei AP, SP und ZAP wird der Bedingungsschlüssel neu eingestellt.

Das Ergebnis einer vorhergehenden Vergleichsoperation (z.B. CLC) kann nach einer solchen Rechenoperation (noch, nicht mehr) _____ abgefragt werden.

nicht mehr



Seite 252 (MP)

Nur für Lernende, die Band 2, 3 und 4 hintereinander durchgearbeitet haben (ohne Benutzung des Aufgabenbandes):

Sie erhalten jetzt Gelegenheit, Ihre Kenntnisse bei der Lösung einer vollständigen kleinen Aufgabe anzuwenden.
Nehmen Sie bitte den Band 1 vor und bearbeiten Sie dort die Seiten 119 bis 151 (Aufgabe 5). Lassen Sie den Verweis auf Seite 137 zu den Instruktionen MVZ und MVN unbeachtet.

Nach Abschluß der Aufgabe 5 (Seite 151 oben) arbeiten Sie bitte an dieser Stelle im Band 4 weiter.

KAPITEL IV

DIE OPERATIONEN MP, DP, MVO, CP

Lernstoff:

MP	-	Multiplizieren Dezimal (Gepackt)
DP	-	Dividieren Dezimal (Gepackt)
MVO	-	Umpacken
CP	-	Vergleichen Dezimal (Gepackt)

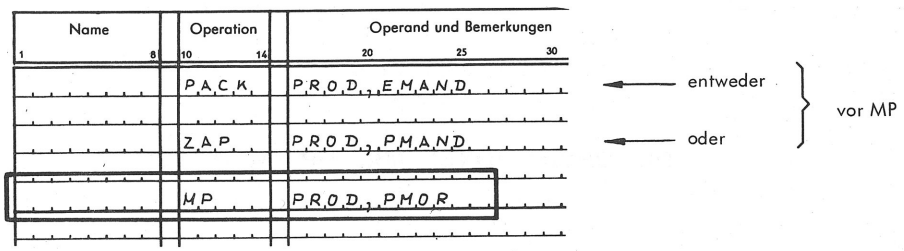
MP

MULTIPLIZIEREN DEZIMAL (GEPACKT)

(Multiply decimal Packed)

Diese Operation multipliziert den (gepackten) Inhalt zweier Speicherfelder miteinander.

Beispiel einer Instruktion:



Wirkungsweise:

Der Multiplikand (MAND) muß vor der Ausführung der Multiplikation rechtsbündig mit führenden Nullen im Produktfeld (im Beispiel PROD) stehen.

Das Produkt aus dem ersten Operanden (im Beispiel PROD) und dem zweiten Operanden (im Beispiel PMOR) wird errechnet und im Feld des ersten Operanden gespeichert.

Vor Durchführung der Multiplikation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen wird nach den Regeln der Algebra gebildet, auch wenn einer oder beide Operanden null sind.
Verschlüsselung: C für +
D für -

Der Bedingungsschlüssel wird nicht verändert.

Der Inhalt des zweiten Operanden bleibt erhalten.

Für die Länge des Produktfeldes gilt folgende Regel:

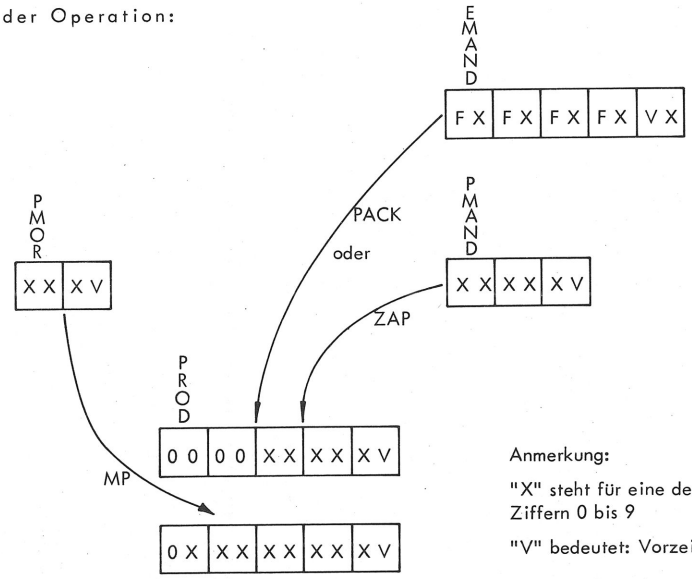
Länge PROD = Länge MAND + Länge MOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
PROD = Produkt, MAND = Multiplikand,
MOR = Multiplikator.

Längenbegrenzung:
Erster Operand (PROD) maximal 16 Bytes, zweiter Operand (PMOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:



Anmerkung:

"X" steht für eine der Ziffern 0 bis 9

"V" bedeutet: Vorzeichen

Abb. 1

1 Sie haben gerade in Abb. 1 die Beschreibung der dezimalen Multiplikationsoperation gelesen. Sie heißt ___(WW)___ und hat den Operationsschlüssel _____.

Multiplizieren Dezimal (Gepackt)

MP

2 Wir nehmen an, daß die beiden Faktoren aus einer Lochkarte in den Speicher eingelesen worden sind.

Es soll die Rechnung

Multiplikand x Multiplikator = Produkt
(kürzer: Mand x Mor = Prod)

durchgeführt werden.

Im Beispiel in Abb. 1 ist der Mand ein ___(Z)___-stelliger Wert und der Mor ein ___(Z)___-stelliger Wert. Das Produkt aus diesen beiden Werten wird ___(Z)___ Stellen groß sein.

5

3

8 (3+5 = 8)

3 Mand und Mor müssen vor der Multiplikationsoperation in _____ Darstellungsform zur Verfügung stehen.

Der erste Operand enthält nach der Operation das Produkt in _____ Darstellungsform.

gepackter (sonst liegt ein Fehler vor und der Programmablauf wird unterbrochen)

gepackter

4 In gepackter Form benötigt der 5-stellige Mand ___(Z)___ Bytes und der 3-stellige Mor ___(ZW)___.

Das Produkt wird maximal 8-stellig und benötigt daher gepackt ___(ZW)___.

3

2 Bytes

5 Bytes

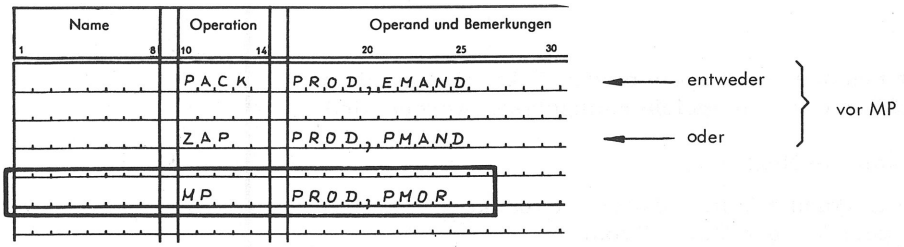
MP

MULTIPLIZIEREN DEZIMAL (GEPACKT)

(Multiply decimal Packed)

Diese Operation multipliziert den (gepackten) Inhalt zweier Speicherfelder miteinander.

Beispiel einer Instruktion:



Wirkungsweise:

Der Multiplikand (MAND) muß vor der Ausführung der Multiplikation rechtsbündig mit führenden Nullen im Produktfeld (im Beispiel PROD) stehen.

Das Produkt aus dem ersten Operanden (im Beispiel PROD) und dem zweiten Operanden (im Beispiel PMOR) wird errechnet und im Feld des ersten Operanden gespeichert.

Vor Durchführung der Multiplikation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen wird nach den Regeln der Algebra gebildet, auch wenn einer oder beide Operanden null sind.
Verschlüsselung: C für +
D für -

Der Bedingungsschlüssel wird nicht verändert.

Der Inhalt des zweiten Operanden bleibt erhalten.

Für die Länge des Produktfeldes gilt folgende Regel:

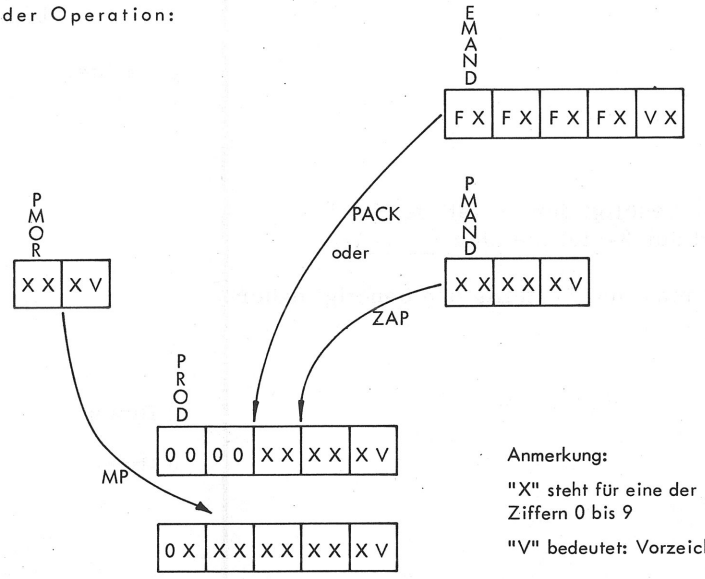
Länge PROD = Länge MAND + Länge MOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
PROD = Produkt, MAND = Multiplikand,
MOR = Multiplikator.

Längenbegrenzung:
Erster Operand (PROD) maximal 16 Bytes, zweiter Operand (PMOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:



Anmerkung:

"X" steht für eine der Ziffern 0 bis 9

"V" bedeutet: Vorzeichen

Abb. 1

5 Das Produktfeld muß mindestens 5 Bytes groß sein. Es darf natürlich größer sein.

Abb. 1 enthält eine einfache Regel zur Berechnung der Länge des Produktfeldes. Sie heißt

_____.

Für das Beispiel in Abb. 1 ergibt sich die Rechnung:
(ZZ) .

Länge PROD = Länge MAND + Länge MOR

3 + 2 = 5

6 Geben Sie die nötigen Feldlängen der Faktoren und des Produkts in den folgenden Beispielen an:

	Byteanzahl ungepackt (= Lochspaltenanzahl)		Byteanzahl gepackt		Byteanz. gepackt
	Mand	Mor	Mand	Mor	Prod
a)	6	2			
b)	5	4			
c)	8	3			
d)	3	5			

a)	4	2	6
b)	3	3	6
c)	5	2	7
d)	2	3	5

MP

MULTIPLIZIEREN DEZIMAL (GEPACKT)

(Multiply decimal Packed)

Diese Operation multipliziert den (gepackten) Inhalt zweier Speicherfelder miteinander.

Beispiel einer Instruktion:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)	
	PACK	PROD,,EMAND	← entweder
	ZAP	PROD,,PMAND	← oder
	MP	PROD,,PMOR	vor MP

Wirkungsweise:

Der Multiplikand (MAND) muß vor der Ausführung der Multiplikation rechtsbündig mit führenden Nullen im Produktfeld (im Beispiel PROD) stehen.

Das Produkt aus dem ersten Operanden (im Beispiel PROD) und dem zweiten Operanden (im Beispiel PMOR) wird errechnet und im Feld des ersten Operanden gespeichert.

Vor Durchführung der Multiplikation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen wird nach den Regeln der Algebra gebildet, auch wenn einer oder beide Operanden null sind.
Verschlüsselung: C für +
D für -

Der Bedingungsschlüssel wird nicht verändert.

Der Inhalt des zweiten Operanden bleibt erhalten.

Für die Länge des Produktfeldes gilt folgende Regel:

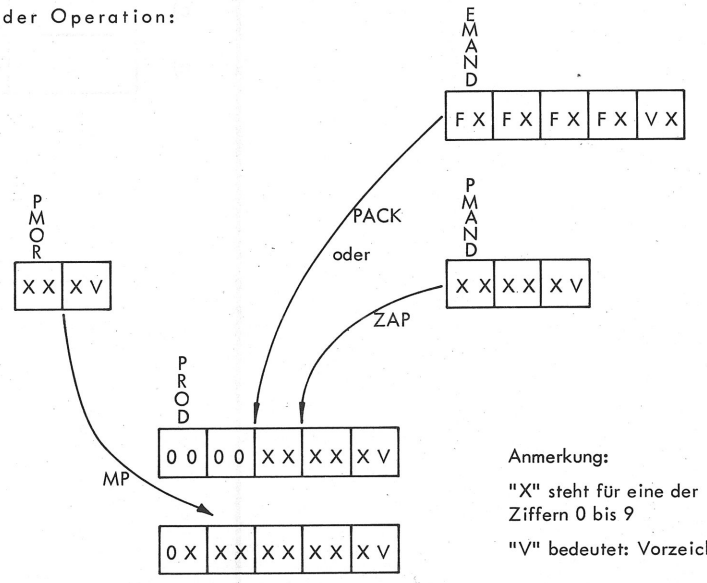
Länge PROD = Länge MAND + Länge MOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
PROD = Produkt, MAND = Multiplikand,
MOR = Multiplikator.

Längenbegrenzung:
Erster Operand (PROD) maximal 16 Bytes, zweiter Operand (PMOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:



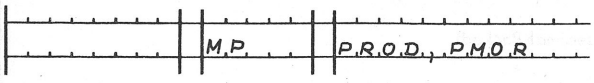
Anmerkung:

"X" steht für eine der Ziffern 0 bis 9

"V" bedeutet: Vorzeichen

Abb. 1

7 Die MP-Instruktion in Abb. 1 heißt



Dadurch wird der Inhalt des Feldes PROD mit dem Inhalt des Feldes PMOR _____.

Natürlich muß vorher der Mand im Feld _____ stehen.

multipliziert

PROD

8 Der Multiplikand im Produktfeld muß bestimmten Bedingungen genügen.

Er muß dort vor der Multiplikation rechtsbündig mit ___(WW)___ stehen.

Im Beispiel in Abb. 1 hat der Mand __(Z)__ führende Nullen im Produktfeld.

führenden Nullen

4

9 Es stellt sich die Frage, mit welcher Operation wir den Multiplikanden richtig ins Produktfeld bringen können.

Abb. 1 zeigt zwei Lösungen.
Ist der Mand ungepackt, so wird er mit der Operation _____ ins Produktfeld gebracht. Bei dieser Operation wird (kein, ein, beide) _____ Operand(en) auf Gültigkeit geprüft.

Steht der Mand dagegen gepackt zur Verfügung, so müssen wir die Operation ___(WW)___ benutzen.

Packen/PACK

kein

Löschen und Addieren/ZAP

MP

MULTIPLIZIEREN DEZIMAL (GEPACKT)

(Multiply decimal Packed)

Diese Operation multipliziert den (gepackten) Inhalt zweier Speicherfelder miteinander.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen	
1 8	10 14	20 25 30	
	PACK	PROD,,EMAND	← entweder
	ZAP	PROD,,PMAND	← oder
	MP	PROD,,PMOR	vor MP

Wirkungsweise:

Der Multiplikand (MAND) muß vor der Ausführung der Multiplikation rechtsbündig mit führenden Nullen im Produktfeld (im Beispiel PROD) stehen.

Das Produkt aus dem ersten Operanden (im Beispiel PROD) und dem zweiten Operanden (im Beispiel PMOR) wird errechnet und im Feld des ersten Operanden gespeichert.

Vor Durchführung der Multiplikation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen wird nach den Regeln der Algebra gebildet, auch wenn einer oder beide Operanden null sind.
Verschlüsselung: C für +
D für -

Der Bedingungsschlüssel wird nicht verändert.

Der Inhalt des zweiten Operanden bleibt erhalten.

Für die Länge des Produktfeldes gilt folgende Regel:

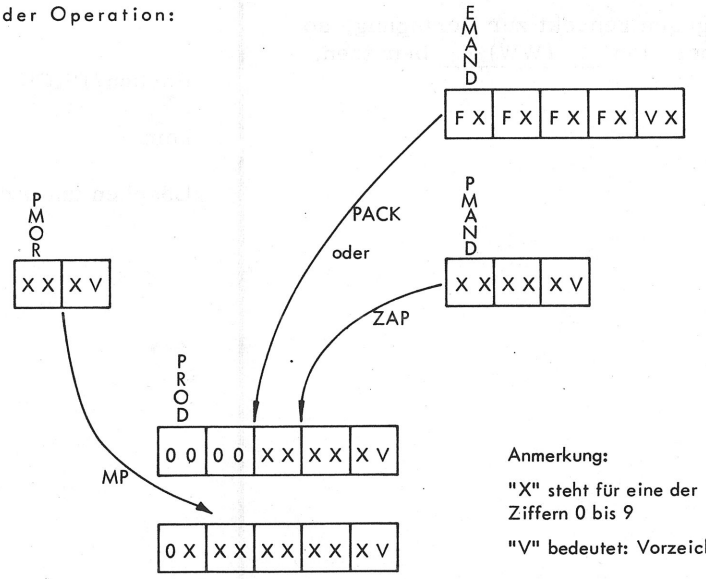
Länge PROD = Länge MAND + Länge MOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
PROD = Produkt, MAND = Multiplikand,
MOR = Multiplikator.

Längenbegrenzung:
Erster Operand (PROD) maximal 16 Bytes, zweiter Operand (PMOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:



Anmerkung:

"X" steht für eine der Ziffern 0 bis 9

"V" bedeutet: Vorzeichen

Abb. 1

10 Sehen Sie sich den Inhalt des Feldes PROD nach der Multiplikation an.

PROD enthält außer dem 8-stelligen Produkt ein
_____ im rechten Halbbyte und eine
(WW) im linken Halbbyte.

Vorzeichen

führende Null

11 Aus technischen Gründen muß jedes Multiplikationsergebnis mindestens eine führende Null haben.

Diese Bedingung erfüllen wir automatisch bei Beachtung der Längenregel in Abb. 1. Die folgenden Beispiele machen dies deutlich.

Entnehmen Sie Ihren Arbeitsmitteln Übung 415.

Tragen Sie die fehlenden Felder mit Inhalt ein.

Vergleichen Sie dann Ihre Eintragungen mit der Lösung in Abb. 2 (siehe nächste Seite).

	Multiplikand und Multiplikator ungepackt	Multiplikand und Multiplikator gepackt	Errechnetes Produkt gepackt
1.	EMAND1 | F X | F X | F X | F X | F X | V X | EMOR1 | F X | F X | V X |	PMAND1 | 0 X | X X | X X | X V | PMOR1 | X X | X V |	PROD1 | 0 0 | X X | X X | X X | X X | X V |
2.	EMAND2 | F X | F X | F X | V X | EMOR2 | F X | F X | F X | V X |	PMAND2 | 0 X | X X | X V | PMOR2 | 0 X | X X | X V |	PROD2 | 0 0 | 0 X | X X | X X | X X | X V |
3.	EMAND3 | F X | F X | F X | F X | V X | EMOR3 | F X | F X | V X |	PMAND3 | X X | X X | X V | PMOR3 | X X | X V |	PROD3 | 0 X | X X | X X | X X | X V |
4.	EMAND4 | F X | F X | F X | F X | F X | F X | V X | EMOR4 | F X | V X |	PMAND4 | X X | X X | X X | X V | PMOR4 | 0 X | X V |	PROD4 | 0 0 | X X | X X | X X | X X | X V |

Abb. 2

12 Wir stellen fest:

Die Produktfelder in Abb. 2 sind auf jeden Fall lang genug. Die Ergebnisse haben eine, zwei oder drei führende Nullen. Eine führende Null ergibt sich allein dadurch, daß die beiden Vorzeichen der Operanden zu einem Vorzeichen des Produkts zusammengefaßt werden.

In einem Fall könnte das Produktfeld sogar um ein Byte kürzer sein. Es handelt sich um das (Z) Beispiel.

2.

In manchen Fällen (vgl. 2. Beispiel in Abb. 2) darf das Produktfeld um ein oder mehrere Bytes kürzer sein als die Regel verlangt.
Wir werden uns hier jedoch immer an die Längenregel in Abb. 1 halten und dadurch den Lehrgang von zusätzlichen Rechnungen entlasten.
Die Möglichkeit, das Produktfeld evtl. kürzer zu wählen, ist im Merkblatt zur Multiplikation vermerkt.

MP

MULTIPLIZIEREN DEZIMAL (GEPACKT)

(Multiply decimal Packed)

Diese Operation multipliziert den (gepackten) Inhalt zweier Speicherfelder miteinander.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen	
1 8	10 14	20 25 30	
	PACK	PROD,,EMAND	← entweder
	ZAP	PROD,,PMAND	← oder
	MP	PROD,,PMOR	vor MP

Wirkungsweise:

Der Multiplikand (MAND) muß vor der Ausführung der Multiplikation rechtsbündig mit führenden Nullen im Produktfeld (im Beispiel PROD) stehen.

Das Produkt aus dem ersten Operanden (im Beispiel PROD) und dem zweiten Operanden (im Beispiel PMOR) wird errechnet und im Feld des ersten Operanden gespeichert.

Vor Durchführung der Multiplikation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen wird nach den Regeln der Algebra gebildet, auch wenn einer oder beide Operanden null sind.
Verschlüsselung: C für +
D für -

Der Bedingungsschlüssel wird nicht verändert.

Der Inhalt des zweiten Operanden bleibt erhalten.

Für die Länge des Produktfeldes gilt folgende Regel:

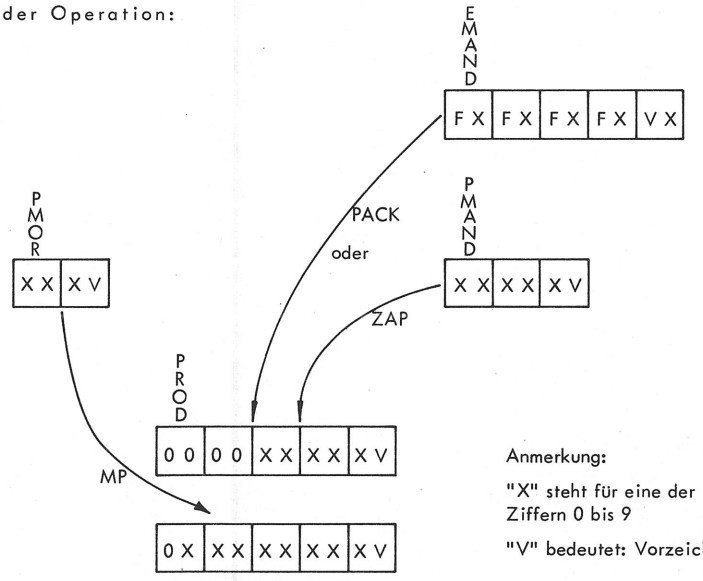
Länge PROD = Länge MAND + Länge MOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
PROD = Produkt, MAND = Multiplikand,
MOR = Multiplikator.

Längenbegrenzung:
Erster Operand (PROD) maximal 16 Bytes, zweiter Operand (PMOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:



Anmerkung:

"X" steht für eine der Ziffern 0 bis 9

"V" bedeutet: Vorzeichen

Abb. 1

13 Die Länge der beiden Operanden einer MP-Operation ist begrenzt.
Lesen Sie hierüber die Regeln in Abb. 1.

Prüfen Sie, welche(s) der folgenden Beispiele nicht erlaubt sind (ist) und warum:

	Byte-Anzahl		
	PMAND	PMOR	PROD
a)	2	9	11
b)	10	5	15
c)	14	4	18

a) Der 2. Operand (PMOR) ist länger als 8 Bytes.

c) Der 1. Operand (PROD) ist länger als 16 Bytes.

14 Der gepackte Multiplikator darf länger sein als der gepackte Multiplikand. Er muß aber kürzer sein als das Produktfeld (= 1. Operand).

Wird die Länge des Produktfeldes nach der Regel in Abb. 1 berechnet, so ist dieser Teil der Längenbegrenzung (immer, manchmal, nie) _____ erfüllt.

immer

15 Das Ergebnisfeld kann maximal 16 Bytes lang sein.

Bei einem Vorzeichen und mindestens einer führenden Null kann das Produkt in diesem Feld maximal
__(Z)__ Ziffernstellen haben.

30

MP

MULTIPLIZIEREN DEZIMAL (GEPACKT)

(Multiply decimal Packed)

Diese Operation multipliziert den (gepackten) Inhalt zweier Speicherfelder miteinander.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	PACK	PROD,,EMAND
	ZAP	PROD,,PMAND
	MP	PROD,,PMOR

entweder / oder } vor MP

Wirkungsweise:

Der Multiplikand (MAND) muß vor der Ausführung der Multiplikation rechtsbündig mit führenden Nullen im Produktfeld (im Beispiel PROD) stehen.

Das Produkt aus dem ersten Operanden (im Beispiel PROD) und dem zweiten Operanden (im Beispiel PMOR) wird errechnet und im Feld des ersten Operanden gespeichert.

Vor Durchführung der Multiplikation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen wird nach den Regeln der Algebra gebildet, auch wenn einer oder beide Operanden null sind.
Verschlüsselung: C für +
D für -

Der Bedingungsschlüssel wird nicht verändert.

Der Inhalt des zweiten Operanden bleibt erhalten.

Für die Länge des Produktfeldes gilt folgende Regel:

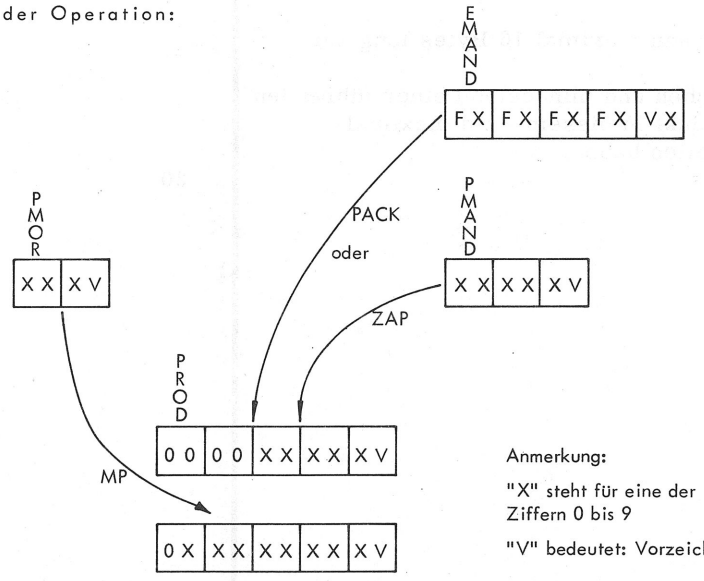
Länge PROD = Länge MAND + Länge MOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
PROD = Produkt, MAND = Multiplikand,
MOR = Multiplikator.

Längenbegrenzung:
Erster Operand (PROD) maximal 16 Bytes, zweiter Operand (PMOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:



Anmerkung:

"X" steht für eine der Ziffern 0 bis 9

"V" bedeutet: Vorzeichen

Abb. 1

16 Bezüglich Adreßrechnung und expliziter Längenangabe gilt für MP das gleiche wie für AP, SP und ZAP.

Bei welchem(n) Operanden einer MP-Instruktion dürfen Adreßrechnung und explizite Längenangabe benutzt werden?

Bei beiden Operanden

17 Das Vorzeichen des Produkts wird nach den algebraischen Regeln gesetzt.

Sind also Multiplikand (im Produktfeld) und Multiplikator beide positiv oder beide negativ, so ist das Ergebnis _____. Die Vorzeichenverschlüsselung ist dann _____.

positiv

Haben die Faktoren verschiedene Vorzeichen (+ und -), so steht im rechten Halbbyte des Produktes die Hexadezimalziffer _____.

C

D

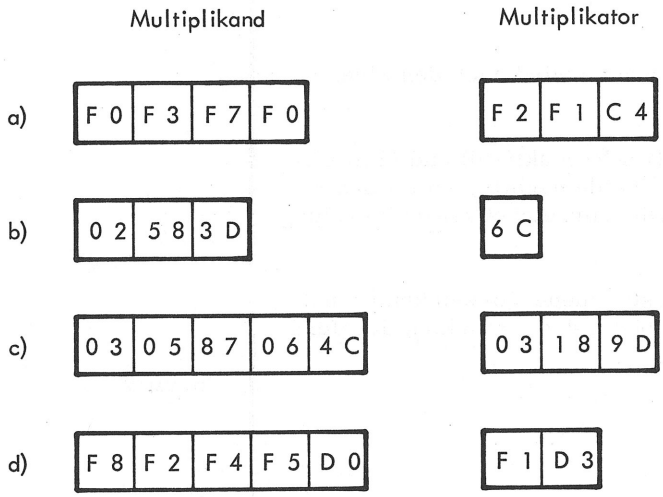

Multiplikand
Multiplikator
a)
F 0 F 3 F 7 F 0
F 2 F 1 C 4
b)
0 2 5 8 3 D
6 C
c)
0 3 0 5 8 7 0 6 4 C
0 3 1 8 9 D
d)
F 8 F 2 F 4 F 5 D 0
F 1 D 3

Abb. 3

18 In Abb. 3 sind einige Beispiele zu den Faktoren einer Multiplikation gezeigt.

Welche Vorzeichen und welche Vorzeichenverschlüsselungen erhalten die Produkte aus diesen Faktoren?

a) positiv, C

b) negativ, D

c) negativ, D

d) positiv, C

MP

MULTIPLIZIEREN DEZIMAL (GEPACKT)

(Multiply decimal Packed)

Diese Operation multipliziert den (gepackten) Inhalt zweier Speicherfelder miteinander.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen	
1 8	10 14	20 25 30	
	PACK	PROD,,EMAND	← entweder
	ZAP	PROD,,PMAND	← oder
	MP	PROD,,PMOR	vor MP

Wirkungsweise:

Der Multiplikand (MAND) muß vor der Ausführung der Multiplikation rechtsbündig mit führenden Nullen im Produktfeld (im Beispiel PROD) stehen.

Das Produkt aus dem ersten Operanden (im Beispiel PROD) und dem zweiten Operanden (im Beispiel PMOR) wird errechnet und im Feld des ersten Operanden gespeichert.

Vor Durchführung der Multiplikation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen wird nach den Regeln der Algebra gebildet, auch wenn einer oder beide Operanden null sind.
Verschlüsselung: C für +
D für -

Der Bedingungsschlüssel wird nicht verändert.

Der Inhalt des zweiten Operanden bleibt erhalten.

Für die Länge des Produktfeldes gilt folgende Regel:

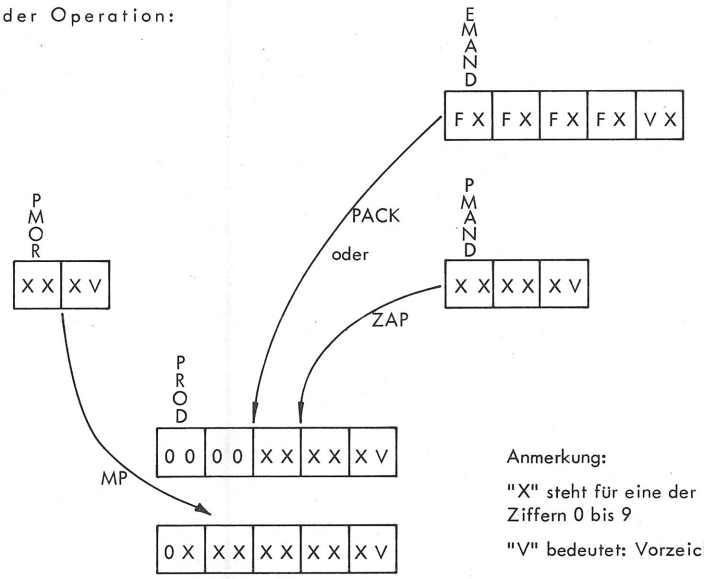
Länge PROD = Länge MAND + Länge MOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
PROD = Produkt, MAND = Multiplikand,
MOR = Multiplikator.

Längenbegrenzung:
Erster Operand (PROD) maximal 16 Bytes, zweiter Operand (PMOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:



Anmerkung:

"X" steht für eine der Ziffern 0 bis 9

"V" bedeutet: Vorzeichen

Abb. 1

19 Wir nehmen an, daß ein Produkt vom Wert "-31670" errechnet wird.

Der Bedingungsschlüssel wird bei dieser Berechnung _____ gesetzt.

nicht

20 Bei einer Multiplikation wird der Bedingungsschlüssel nicht neu eingestellt.

Das Ergebnis kann aber durch einen nachfolgenden Vergleich mit null oder durch eine ZAP-Operation ins gleiche Feld geprüft werden.

Danach zeigt der Bedingungsschlüssel an, ob das Produkt ___(WW)___, ___(WW)___ oder ___(WW)___ ist.

größer als null

kleiner als null

gleich null

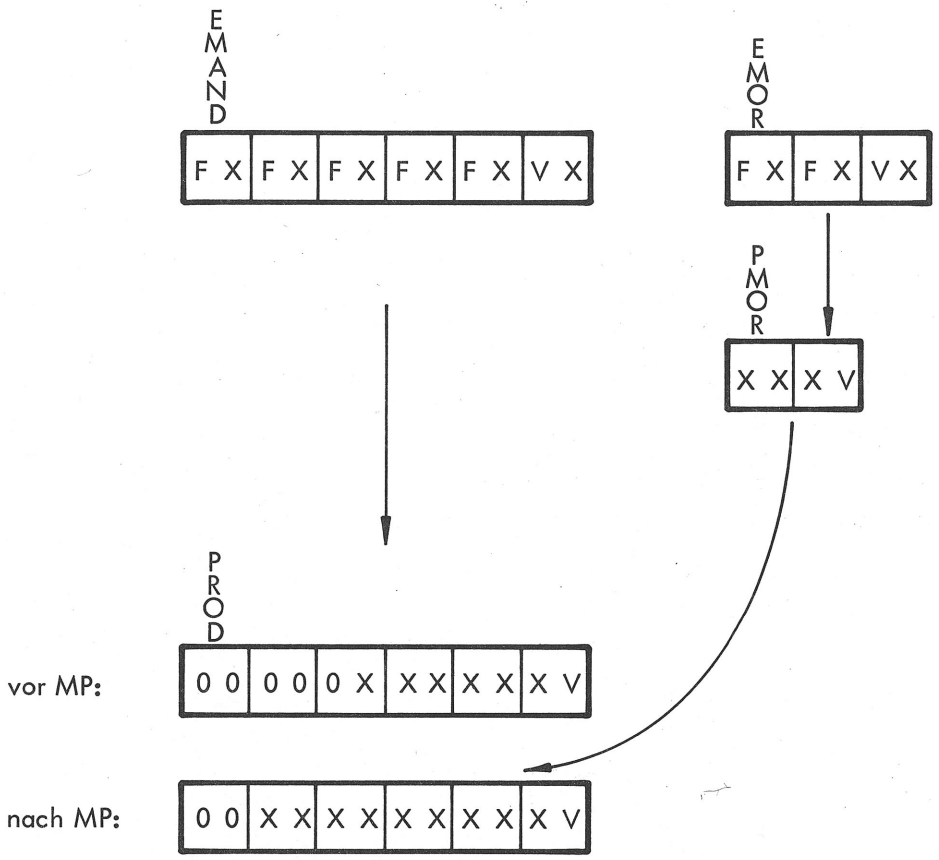
Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 416, Seite 1, 2 und 3.

Tragen Sie Ihre Lösungen ein.

Prüfen Sie Ihre Eintragungen anhand der folgenden Lösungsblätter, sobald Sie eine Seite vollständig bearbeitet haben.

1. Zwei Faktoren stehen in den Feldern EMAND und EMOR im Eingabebereich zur Verfügung. Die Multiplikation soll durchgeführt werden unter Benutzung der Felder PMOR und PROD.
Geben Sie Länge und Inhalt dieser zwei Felder an und codieren Sie die nötigen drei Instruktionen.



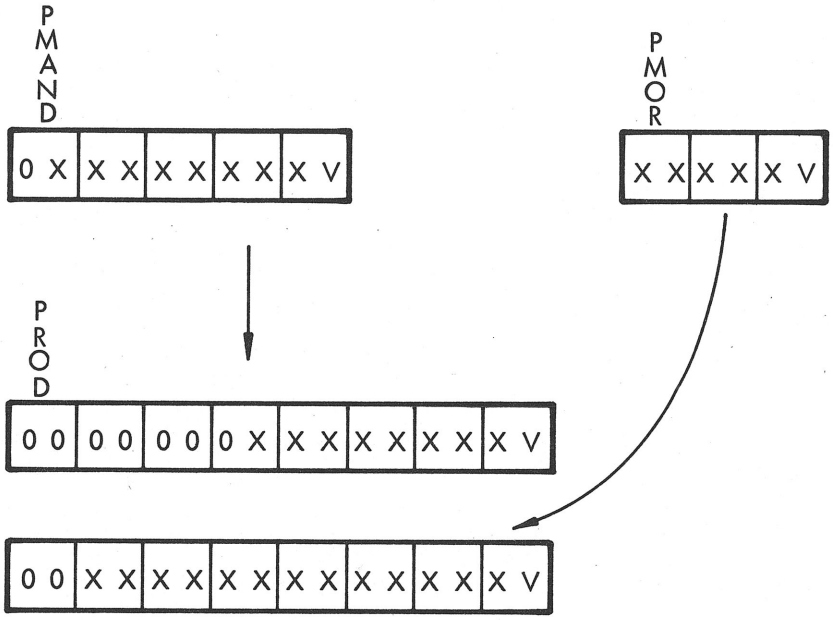
	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–25–30)
*		PACK	PROD,EMAND
*		PACK	PMOR,EMOR
		MP	PROD,PMOR

* Die Reihenfolge der beiden PACK-Instruktionen ist beliebig.



2. In diesem Beispiel stehen die Faktoren für eine Multiplikation schon in gepackter Form zur Verfügung.

Geben Sie die erforderlichen Instruktionen und den dabei im Produktfeld erzeugten Inhalt an.

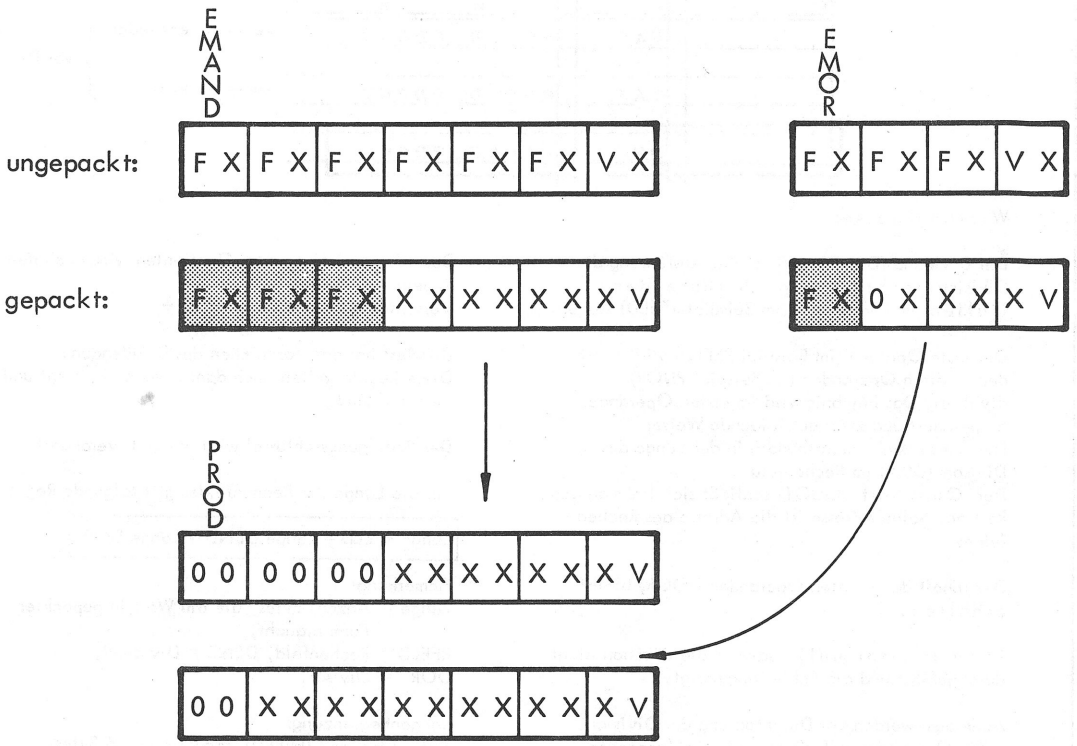


Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	ZAP	PROD,PMAND
	MP	PROD,PMOR



3. In diesem Beispiel sind Mand und Mor ins eigene Feld gepackt worden.

Bestimmen Sie Länge und Inhalt des Produktfeldes und codieren Sie die erforderlichen Instruktionen für die Multiplikation.



vorgegeben:

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
. . . . .	P.A.C.K.	E.M.A.N.D.+.3.(.4.)., .E.M.A.N.D. . .
. . . . .	P.A.C.K.	E.M.O.R.+.1.(.3.)., .E.M.O.R. . . . .
. . . . .	Z.A.P. .	P.R.O.D., .E.M.A.N.D.+.3.(.4.) . . . .
. . . . .	M.P. . .	P.R.O.D., .E.M.O.R.+.1.(.3.) . . . . .

Merkblatt 415 enthält die Beschreibung der Operation "Multiplizieren Dezimal (Gepackt)" (vgl. Abb. 1).

Band 1, Seite 185

DP DIVIDIEREN DEZIMAL (GEPACKT)

(Divide decimal Packed)

Diese Operation dividiert den (gepackten) Inhalt eines Feldes durch den (gepackten) Inhalt eines anderen Feldes.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen	
	PACK	RFELD,EDEND	← entweder
	ZAP	RFELD,PDEND	← oder
	DP	RFELD,PDOR	

} vor DP

Wirkungsweise:

Der Dividend (DEND) muß vor der Ausführung der Division rechtsbündig mit führenden Nullen im Rechenfeld (im Beispiel RFELD) stehen.

Der erste Operand (im Beispiel RFELD) wird durch den zweiten Operanden (im Beispiel PDOR) dividiert. Das Ergebnis wird im ersten Operanden gespeichert und zwar auf folgende Weise:
Der Rest steht rechtsbündig in der Länge des Divisors (DOR) im Rechenfeld.
Der Quotient (QUOT) schließt sich links an den Rest an. Seine Adresse ist die Adresse des Rechenfeldes.

Der Inhalt des zweiten Operanden (PDOR) bleibt erhalten.

Ist der Divisor null, so wird die Division nicht durchgeführt und ein Fehler angezeigt.

Außerdem werden vor Durchführung der Division beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen des Quotienten wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Der Rest hat das Vorzeichen des Dividenden.
Diese Regeln gelten auch dann, wenn Quotient und Rest null sind.

Der Bedingungsschlüssel wird nicht verändert.

Für die Länge des Rechenfeldes gilt folgende Regel:

Länge RFELD = Länge DEND + Länge DOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
RFELD = Rechenfeld, DEND = Dividend,
DOR = Divisor.

Längenbegrenzung:
Erster Operand (RFELD) maximal 16 Bytes, zweiter Operand (PDOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler, Divisor gleich null oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:

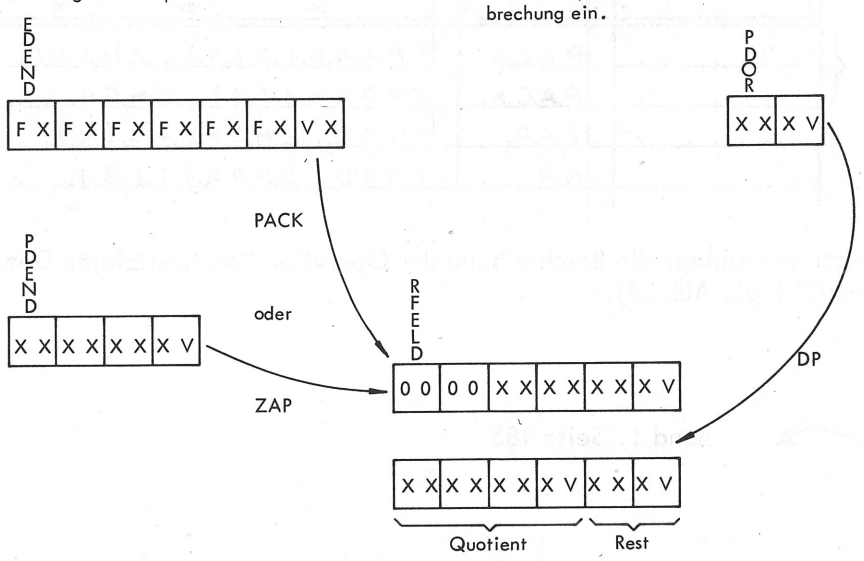


Abb. 1

1 Abb. 1 enthält die Beschreibung der dezimalen Divisionsoperation.

Sie heißt ____(WW)____ und hat den symbolischen Operationsschlüssel _____.

Dividieren Dezimal (Gepackt)

DP

2 Wir erinnern uns kurz an die Bedeutung der Bezeichnungen Dividend, Divisor, Quotient und Rest:

145 000	:	120	=	1 208	40
Dividend		Divisor		Quotient	Rest

Geben Sie für dieses Zahlenbeispiel den Inhalt der folgenden Speicherfelder an (soweit nicht schon vorgegeben):

EDEND: | F 0 | F 1 | F 4 | F 5 | F 0 | F 0 | C 0 |

EDOR: | F 1 | F 2 | C 0 |

PDEND: | | | | |

PDOR: | | |

RFELD: (vor DP, Inhalt Dend) | | | | | | |

RFELD: (nach DP) | | | | | | |

Überprüfen Sie Ihre Ergebnisse auf dem folgenden Blatt.

DP

DIVIDIEREN DEZIMAL (GEPACKT)

(Divide decimal Packed)

Diese Operation dividiert den (gepackten) Inhalt eines Feldes durch den (gepackten) Inhalt eines anderen Feldes.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen	
	PACK	RFELD,EDEND	← entweder
	ZAP	RFELD,PDEND	← oder
	DP	RFELD,PDOR	} vor DP

Wirkungsweise:

Der Dividend (DEND) muß vor der Ausführung der Division rechtsbündig mit führenden Nullen im Rechenfeld (im Beispiel RFELD) stehen.

Der erste Operand (im Beispiel RFELD) wird durch den zweiten Operanden (im Beispiel PDOR) dividiert. Das Ergebnis wird im ersten Operanden gespeichert und zwar auf folgende Weise:
Der Rest steht rechtsbündig in der Länge des Divisors (DOR) im Rechenfeld.
Der Quotient (QUOT) schließt sich links an den Rest an. Seine Adresse ist die Adresse des Rechenfeldes.

Der Inhalt des zweiten Operanden (PDOR) bleibt erhalten.

Ist der Divisor null, so wird die Division nicht durchgeführt und ein Fehler angezeigt.

Außerdem werden vor Durchführung der Division beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen des Quotienten wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Der Rest hat das Vorzeichen des Dividenden.
Diese Regeln gelten auch dann, wenn Quotient und Rest null sind.

Der Bedingungsschlüssel wird nicht verändert.

Für die Länge des Rechenfeldes gilt folgende Regel:

Länge RFELD = Länge DEND + Länge DOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
RFELD = Rechenfeld, DEND = Dividend,
DOR = Divisor.

Längenbegrenzung:
Erster Operand (RFELD) maximal 16 Bytes, zweiter Operand (PDOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler, Divisor gleich null oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:

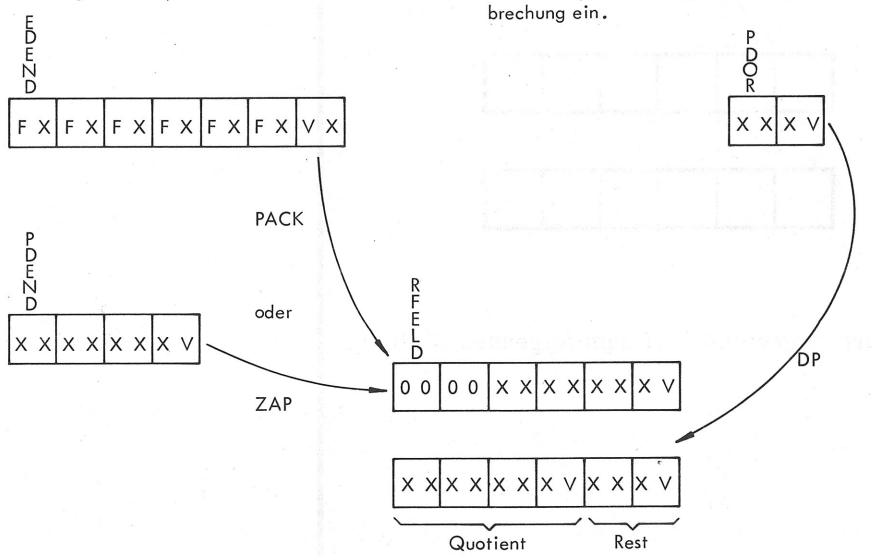


Abb. 1

PDEND: | 0 1 | 4 5 | 0 0 | 0 C |

PDOR: | 1 2 | 0 C |

RFELD: (vor DP) | 0 0 | 0 0 | 0 1 | 4 5 | 0 0 | 0 C |

RFELD: (nach DP) | 0 0 | 0 1 | 2 0 | 8 C | 0 4 | 0 C |

Quotient Rest

3 Quotient und Rest haben beide ihren festen Platz im Ergebnisfeld.

Links steht der _____ und rechts der _____.

Der Rest hat immer die gleiche Länge wie der _____ in gepackter Form.

Quotient

Rest

Divisor

4 Bei der Division wird der Quotient umso größer, je kleiner der Divisor ist.

Bei der dezimalen Division mit DP ist der kleinste Divisor "1". In diesem Fall wird der Quotient genauso groß wie der _____.

Dividend

DP

DIVIDIEREN DEZIMAL (GEPACKT)

(Divide decimal Packed)

Diese Operation dividiert den (gepackten) Inhalt eines Feldes durch den (gepackten) Inhalt eines anderen Feldes.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen	
	PACK	RFELD,EDEND	← entweder
	ZAP	RFELD,PDEND	← oder
	DP	RFELD,PDOR	} vor DP

Wirkungsweise:

Der Dividend (DEND) muß vor der Ausführung der Division rechtsbündig mit führenden Nullen im Rechenfeld (im Beispiel RFELD) stehen.

Der erste Operand (im Beispiel RFELD) wird durch den zweiten Operanden (im Beispiel PDOR) dividiert. Das Ergebnis wird im ersten Operanden gespeichert und zwar auf folgende Weise:
Der Rest steht rechtsbündig in der Länge des Divisors (DOR) im Rechenfeld.
Der Quotient (QUOT) schließt sich links an den Rest an. Seine Adresse ist die Adresse des Rechenfeldes.

Der Inhalt des zweiten Operanden (PDOR) bleibt erhalten.

Ist der Divisor null, so wird die Division nicht durchgeführt und ein Fehler angezeigt.

Außerdem werden vor Durchführung der Division beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen des Quotienten wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Der Rest hat das Vorzeichen des Dividenden.
Diese Regeln gelten auch dann, wenn Quotient und Rest null sind.

Der Bedingungsschlüssel wird nicht verändert.

Für die Länge des Rechenfeldes gilt folgende Regel:

Länge RFELD = Länge DEND + Länge DOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
RFELD = Rechenfeld, DEND = Dividend,
DOR = Divisor.

Längenbegrenzung:
Erster Operand (RFELD) maximal 16 Bytes,
zweiter Operand (PDOR) maximal 8 Bytes,
dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler, Divisor gleich null oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:

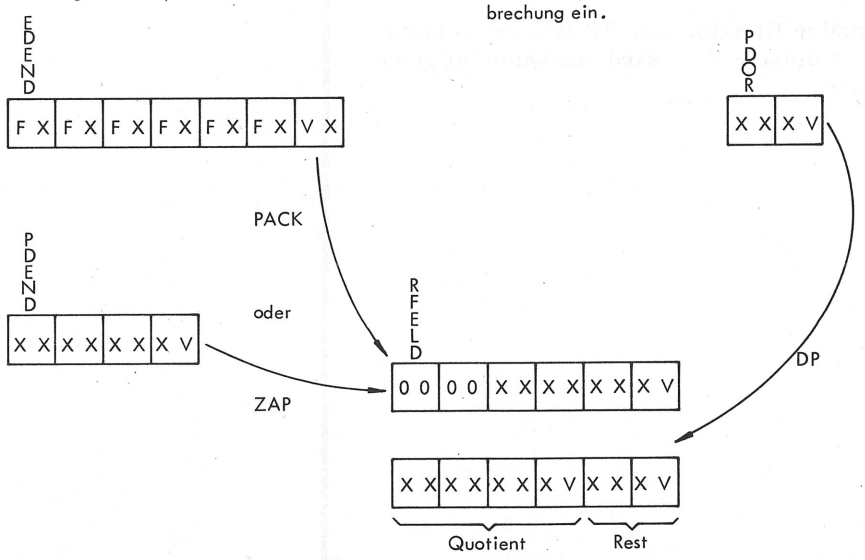


Abb. 1

5 Bei der dezimalen Division wird der Quotient höchstens so groß wie der Dividend.

Darum berechnet sich die Länge des Rechenfeldes nach der einfachen Regel in Abb. 1:
Sind Dividend und Divisor in gepackter Form 5 und 3 Bytes lang, so muß das Rechenfeld ___(ZW)___ lang sein.

Nach der Division steht der Quotient in den linken ___(ZW)___ und der Rest in den ___(WZ)___ Bytes.

8 Bytes (5+3)

5 Bytes

rechten 3

6 Die beiden Operanden einer DP-Operation unterliegen einer ähnlichen Längenbegrenzung wie die Operanden einer MP-Operation.

Das Divisionsfeld darf nicht länger als ___(ZW)___ sein. Der gepackte Divisor darf nicht länger als ___(ZW)___ sein.

Dabei muß der Divisor unbedingt _____ als das Rechenfeld sein.

Wird die Länge des Rechenfeldes nach der Regel in Abb. 1 berechnet (Länge RFELD = Länge DEND + Länge DOR), so ist der Divisor immer kürzer als das Rechenfeld. Das Rechenfeld ist dann immer um soviel länger als der Divisor, wie der _____ lang ist.

16 Bytes

8 Bytes

kürzer

Dividend

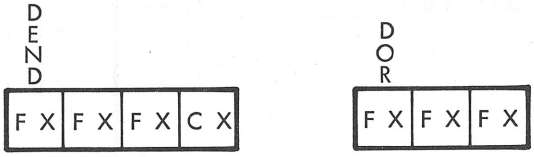


Abb. 2

7 In den folgenden Lernelementen gehen wir von den beiden Feldern in Abb. 2 aus. Dividend und Divisor sind ungepackt.

Für dieses Beispiel muß das Rechenfeld eine Länge von ___(ZW)___ haben.

5 Bytes (3 Bytes + 2 Bytes)

8 Bevor die Divisionsoperation ausgeführt werden kann, müssen Dividend und Divisor _____ werden.

gepackt

9 Im Beispiel in Abb. 2 soll der Divisor ins gleiche Feld gepackt werden und der Dividend ins Rechenfeld mit dem Namen RFELD.

a) Wie lauten die beiden PACK-Instruktionen?

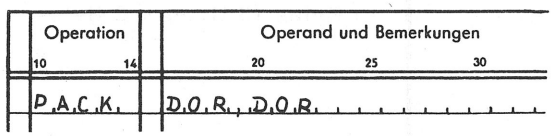
b) Welchen Inhalt haben die Felder DOR und RFELD "nachher"?

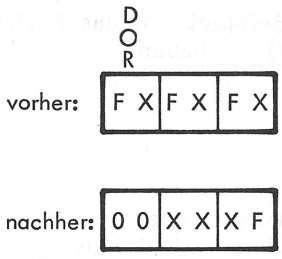
a)

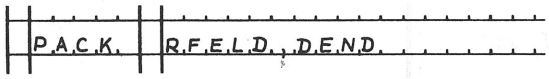
	Operation	Operand und Ber
	10 14	20 25
	P,A,C,K,	D,O,R,,D,O,R, . . .
	P,A,C,K,	R,F,E,L,D,,D,E,N,D,

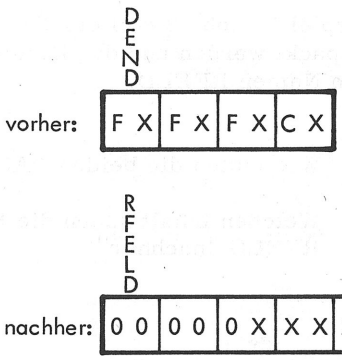
b)

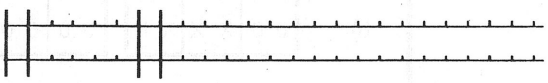
DOR				RFELD					
0 0	X X	X F		0 0	0 0	0 X	X X	X C	


Operation
Operand und Bemerkungen
10
14
20
25
30
P,A,C,K,
D,O,R,,D,O,R,


DOR
vorher:
F X F X F X
nachher:
0 0 X X X F


P,A,C,K,
R,F,E,L,D,,D,E,N,D,


DEND
vorher:
F X F X F X C X
RFELD
nachher:
0 0 0 0 0 X X X C



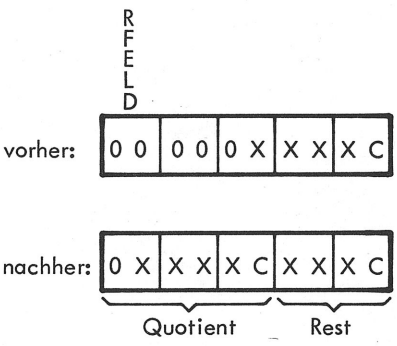

RFELD
vorher:
0 0 0 0 0 X X X C
nachher:
0 X X X C X X X C
Quotient
Rest

Abb. 3

10 In Abb. 3 ist das Beispiel aus Abb. 2 fortgeführt.

Es fehlt noch die Divisionsinstruktion selbst. Wie heißt sie?

| | D.P. . . | | R.F.E.L.D., D.O.R.+.1.(.2.). .

(Sie könnten die Division auch mit der Instruktion

| | D.P. . . | | R.F.E.L.D., D.O.R. . . . . .

durchführen. Dann müßte aber das Rechenfeld 1 Byte länger sein, weil der Rest in diesem Fall 3 Bytes belegen würde.)

Wir fassen zusammen:

Die dezimale Division verlangt ebenso wie Addition, Subtraktion und Multiplikation gepackte Daten.

Durchführung der Division:
1. Schritt: Der Dividend wird rechtsbündig mit führenden Nullen ins Rechenfeld gebracht. Dazu dient eine PACK-Instruktion - wenn der Dividend ungepackt vorliegt - bzw. eine ZAP-Instruktion - wenn der Dividend gepackt vorliegt.

2. Schritt: Der Dividend im Rechenfeld wird durch den Divisor dividiert. Dazu dient eine DP-Instruktion.

Anmerkung: Der 1. Schritt entfällt, wenn der Dividend schon vorher richtig im Rechenfeld steht.

DP

DIVIDIEREN DEZIMAL (GEPACKT)

(Divide decimal Packed)

Diese Operation dividiert den (gepackten) Inhalt eines Feldes durch den (gepackten) Inhalt eines anderen Feldes.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen	
1 8	10 14	20 25 30	
	PACK	RFELD, EDEND	← entweder
	ZAP	RFELD, PDEND	← oder
	DP	RFELD, PDOR	vor DP

Wirkungsweise:

Der Dividend (DEND) muß vor der Ausführung der Division rechtsbündig mit führenden Nullen im Rechenfeld (im Beispiel RFELD) stehen.

Der erste Operand (im Beispiel RFELD) wird durch den zweiten Operanden (im Beispiel PDOR) dividiert. Das Ergebnis wird im ersten Operanden gespeichert und zwar auf folgende Weise:
Der Rest steht rechtsbündig in der Länge des Divisors (DOR) im Rechenfeld.
Der Quotient (QUOT) schließt sich links an den Rest an. Seine Adresse ist die Adresse des Rechenfeldes.

Der Inhalt des zweiten Operanden (PDOR) bleibt erhalten.

Ist der Divisor null, so wird die Division nicht durchgeführt und ein Fehler angezeigt.

Außerdem werden vor Durchführung der Division beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen des Quotienten wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Der Rest hat das Vorzeichen des Dividenden.
Diese Regeln gelten auch dann, wenn Quotient und Rest null sind.

Der Bedingungsschlüssel wird nicht verändert.

Für die Länge des Rechenfeldes gilt folgende Regel:

Länge RFELD = Länge DEND + Länge DOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
RFELD = Rechenfeld, DEND = Dividend,
DOR = Divisor.

Längenbegrenzung:
Erster Operand (RFELD) maximal 16 Bytes, zweiter Operand (PDOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler, Divisor gleich null oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:

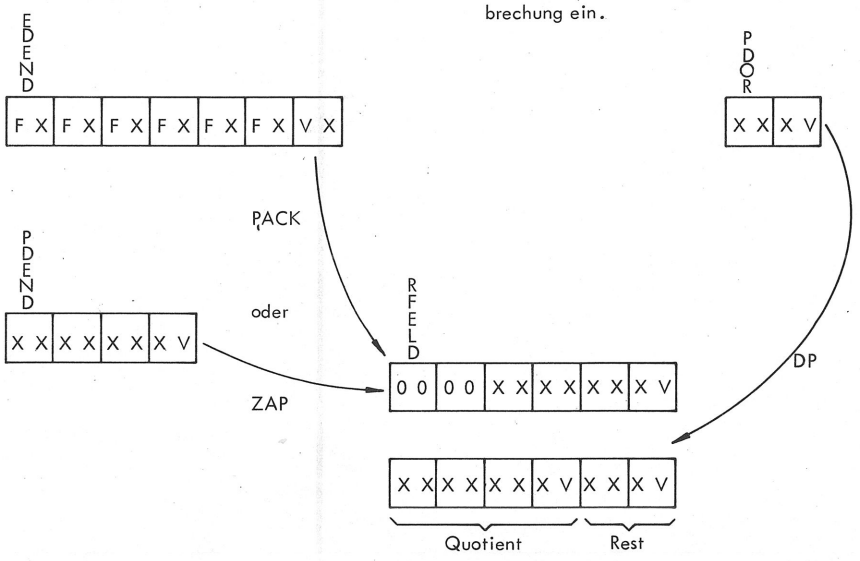


Abb. 1

Zum Vorzeichen von Quotient und Rest:

Das Vorzeichen des Quotienten wird nach den algebraischen Regeln gesetzt.

Haben Dividend und Divisor die gleichen Vorzeichen (+ oder -), so ist der Quotient positiv.

Haben Dividend und Divisor verschiedene Vorzeichen (+ und -), so ist der Quotient negativ.

Der Rest hat immer das Vorzeichen des Dividenden.

11 In der folgenden Übung sind die Vorzeichen für den Dividenden und den Divisor gegeben. Bestimmen Sie die Vorzeichen für den Quotienten und den Rest.

	Dividend	Divisor	Quotient	Rest
a)	+	+		
b)	+	-		
c)	-	+		
d)	-	-		

	Quotient	Rest
a)	+	+
b)	-	+
c)	-	-
d)	+	-

DP DIVIDIEREN DEZIMAL (GEPACKT)

(Divide decimal Packed)

Diese Operation dividiert den (gepackten) Inhalt eines Feldes durch den (gepackten) Inhalt eines anderen Feldes.

Beispiel einer Instruktion:

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)	
	PACK	RFELD,EDEND	← entweder
	ZAP	RFELD,PDEND	← oder
	DP	RFELD,PDOR	} vor DP

Wirkungsweise:

Der Dividend (DEND) muß vor der Ausführung der Division rechtsbündig mit führenden Nullen im Rechenfeld (im Beispiel RFELD) stehen.

Der erste Operand (im Beispiel RFELD) wird durch den zweiten Operanden (im Beispiel PDOR) dividiert. Das Ergebnis wird im ersten Operanden gespeichert und zwar auf folgende Weise:
Der Rest steht rechtsbündig in der Länge des Divisors (DOR) im Rechenfeld.
Der Quotient (QUOT) schließt sich links an den Rest an. Seine Adresse ist die Adresse des Rechenfeldes.

Der Inhalt des zweiten Operanden (PDOR) bleibt erhalten.

Ist der Divisor null, so wird die Division nicht durchgeführt und ein Fehler angezeigt.

Außerdem werden vor Durchführung der Division beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Das Vorzeichen des Quotienten wird nach den Regeln der Algebra gebildet.
Verschlüsselung: C für +
D für -
Der Rest hat das Vorzeichen des Dividenden.
Diese Regeln gelten auch dann, wenn Quotient und Rest null sind.

Der Bedingungsschlüssel wird nicht verändert.

Für die Länge des Rechenfeldes gilt folgende Regel:

Länge RFELD = Länge DEND + Länge DOR

Anmerkung:
Länge = Anzahl Bytes, die ein Wert in gepackter Form braucht,
RFELD = Rechenfeld, DEND = Dividend,
DOR = Divisor.

Längenbegrenzung:
Erster Operand (RFELD) maximal 16 Bytes, zweiter Operand (PDOR) maximal 8 Bytes, dabei unbedingt zweiter Operand kürzer als erster Operand.

Bei Gültigkeitsfehler, Divisor gleich null oder falscher Operandenlänge tritt Programmunterbrechung ein.

Durchführung der Operation:

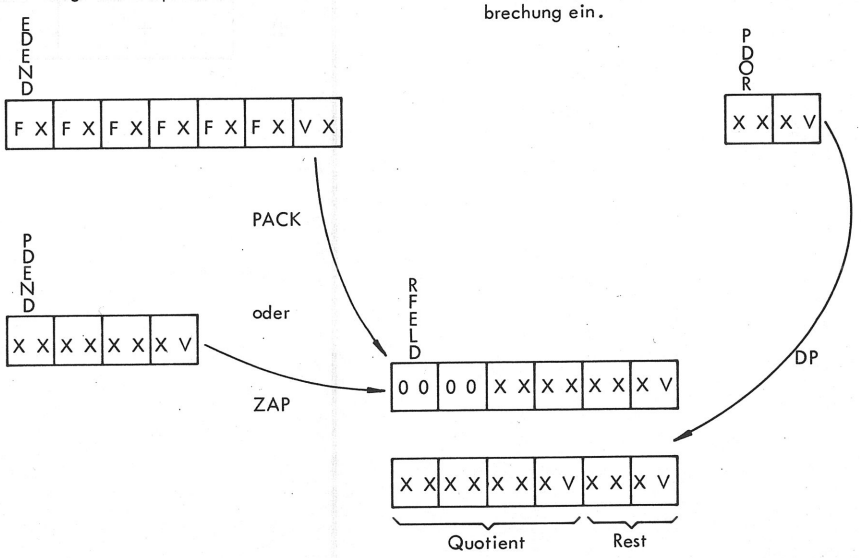


Abb. 1

12 Sie kennen verschiedene Operationen, die aufgrund des Operationsergebnisses den Bedingungsschlüssel neu einstellen.

Wie bei MP, so wird auch bei DP der Bedingungsschlüssel ____(WW)____.

Sinngemäß:

nicht neu eingestellt

13 Abb. 1 enthält eine Regel zur Berechnung der Länge des Rechenfeldes. Wird diese Regel beachtet, so entspricht der Platz für den Quotienten der Länge des gepackten Dividenden.

Da der Quotient nicht größer werden kann als der Dividend, kann ein Überlauf in diesem Fall ____(WW)____.

nicht auftreten

14 Selbstverständlich darf das Rechenfeld für eine Division auch größer sein, als es der Regel in Abb. 1 entspricht.

So könnte z.B. das Rechenfeld in Abb. 1 8 Bytes groß sein statt 6 Bytes. (Etwa wegen einer anderen Rechnung, die im gleichen Feld durchgeführt wird.)

Nach der Durchführung der Division enthalten dann links die überflüssigen Bytes des Rechenfeldes ____(WW)____.

führende Nullen

Ebenso darf das Rechenfeld kürzer sein, als die Regel in Abb. 1 verlangt, wenn die Werte von Dividend und Divisor bestimmte Bedingungen erfüllen.

In unserem Lehrgang werden wir aber die Länge des Rechenfeldes immer nach der Regel
"Länge RFELD = Länge DEND + Länge DOR"
berechnen.

15 Die Operation "Dividieren Dezimal" ist eine Festkomma-Rechnung. Ein eventuell vorhandenes Komma wird weder gespeichert noch beim Rechnen berücksichtigt.

Die Aufgabe, dieses Komma für die Weiterrechnung bzw. für die Ausgabe der Daten zu berücksichtigen, fällt dem _____ zu.

Programmierer

Übungen

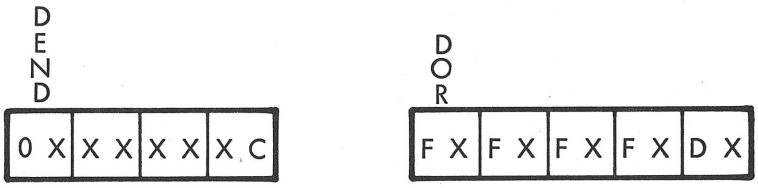
Entnehmen Sie Ihren Arbeitsmitteln Übung 417, Seite 1, 2 und 3.

Tragen Sie Ihre Lösungen ein.

Prüfen Sie Ihre Eintragungen anhand der folgenden Lösungsblätter, sobald Sie eine Seite vollständig bearbeitet haben.

Lösungen zu den Übungen

1. Der Inhalt des Feldes DEND soll durch den Inhalt des Feldes DOR dividiert werden. Das Rechenfeld heiße RFELD.

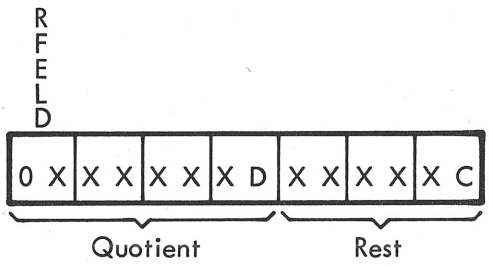


a) Codieren Sie die nötigen Instruktionen:

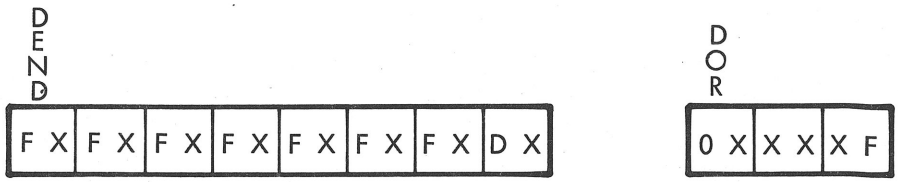
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
	ZAP	RFELD,DEND
	PACK	DOR,DOR
	DP	RFELD,DOR+2(3)

b) Wie lang muß das Rechenfeld sein?
Antwort: 7 Bytes lang

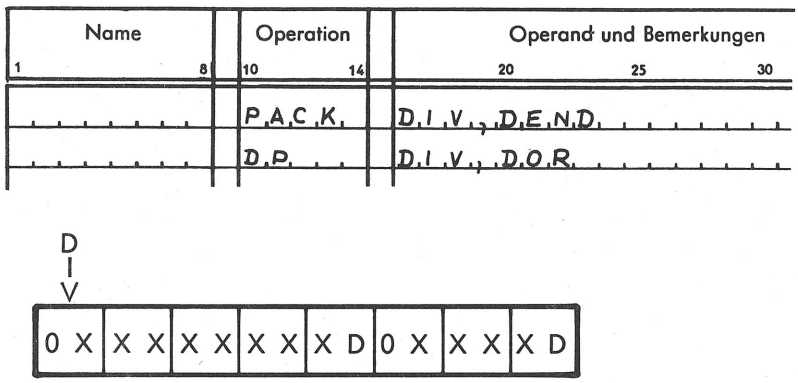
c) Geben Sie das Rechenfeld mit seinem Inhalt nach der Division an.



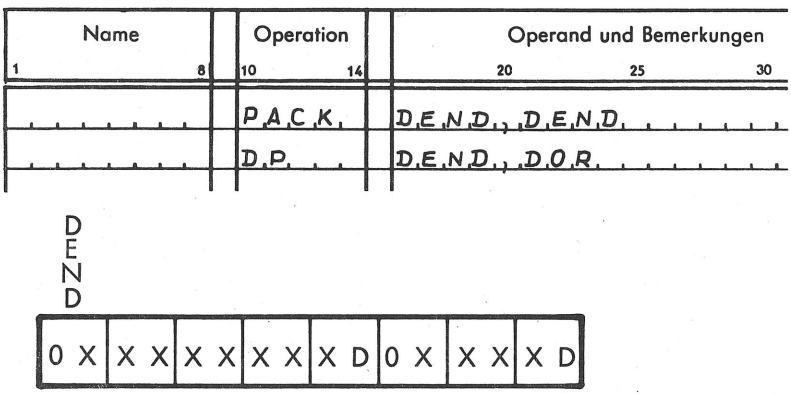
2. Der Inhalt des Feldes DEND soll durch den Inhalt des Feldes DOR dividiert werden.



Fall 1: Als Rechenfeld soll ein Feld mit dem Namen DIV dienen. Geben Sie die nötigen Befehle und den Inhalt des Feldes DIV nach der Division an.



Fall 2: Als Rechenfeld soll das Feld DEND selbst dienen. Geben Sie die nötigen Befehle und den Inhalt des Feldes DEND nach der Division an.



3.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	D.P.	R.F.E.L.D.,D.O.R.

Warum wird in den folgenden Beispielen die Divisionsoperation nicht durchgeführt und ein Fehler angezeigt? Geben Sie bitte in der rechten Spalte in Stichworten die jeweilige Fehlerursache an.

Divisor (DOR)	Rechenfeld (RFELD)	Fehlerursache (sinngemäß)
F8 F3 F0 C5	00 00 00 00 78 12 05 0C Dividend	DOR ungepackt
01 2F	24 01 87 30 25 0F Dividend	RFELD um 2 Bytes zu kurz
00 00 0C	00 00 00 03 07 84 00 0F Dividend	Divisor null
27 8D	05 23 00 6C 50 0C Dividend	DEND nicht rechtsbündig in RFELD
03 58 7D	F0 F1 F8 F5 F6 F0 C0 Dividend	DEND ungepackt

Merkblatt 416 enthält die Beschreibung der Operation "Dividieren Dezimal (Gepackt)" (vgl. Abb. 1).

Nur für Lernende, die vom Aufgabenband ausgehen!

Bevor Sie zum Aufgabenband zurückgehen, entnehmen Sie bitte Ihren Arbeitsmitteln Merkblatt 419.

Dieses Merkblatt enthält eine Zusammenfassung der wesentlichen Eigenschaften der Operationen MP, DP, MVO und CP.

Band 1, Seite 267

MVO

UMPACKEN

(MoVe with Offset)

Diese Operation überträgt den Inhalt eines Feldes um ein halbes Byte nach links versetzt in ein anderes Feld.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	M.V.O. .	F.E.L.D.A.,.F.E.L.D.B. . . . .

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel FELDB) wird in das Feld des ersten Operanden (im Beispiel FELDA) übertragen und zwar um ein halbes Byte nach links versetzt.
Das rechte Halbbyte im rechten Byte des ersten Operanden wird nicht verändert.

Keiner der Operanden wird auf Gültigkeit geprüft.

Verarbeitet wird byteweise von rechts nach links.

Ist der erste Operand zu lang, werden führende Nullen eingeschoben. Ist der erste Operand zu kurz, so werden die linken Daten aus dem zweiten Operanden nicht übertragen.

Der Bedingungsschlüssel wird nicht verändert.

Der Inhalt des zweiten Operanden bleibt erhalten.

Die Feldlänge jedes Operanden beträgt maximal 16 Bytes.

Durchführung der Operation:

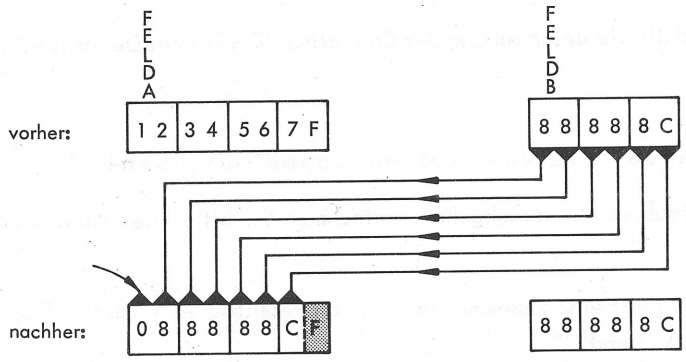


Abb. 1

1 Die Operation, die Sie hier kennenlernen, hat den Namen _____.

Der symbolische Operationsschlüssel heißt _____.

Umpacken

MVO

2 Die Wirkungsweise der Operation Umpacken ist in Abb. 1 beschrieben und an einem Beispiel vorgeführt. Achten Sie bitte nicht darauf, ob das Ergebnis sinnvoll ist, sondern nur, wie es entsteht.

Es wird "umgepackt" aus dem _____ Operanden in den _____ Operanden.

zweiten

ersten

3 Von den 3 Bytes des zweiten Operanden wird im ersten Schritt der Inhalt des (rechten, linken) _____ Bytes übertragen.

Dieses Byte hat in Abb. 1 den Inhalt _(ZZ)_.

rechten

8C

4 Der Inhalt "8C" aus dem rechten Byte von FELDB wird nach FELDA übertragen. Dabei gelangt die Hexadezimalziffer C aus dem rechten Halbbyte in ein _(WW)_ und die Ziffer 8 aus dem linken Halbbyte in ein _(WW)_.

linkes Halbbyte

rechtes Halbbyte

5 Der Inhalt der links anschließenden Bytes aus dem zweiten Operanden wird links anschließend im ersten Operanden gespeichert.

Da FELDA 4 Bytes lang ist, bleibt das linke Halbbyte frei. In dieses Halbbyte wird _(WW)_ eingeschoben.

eine führende Null

MVO

UMPACKEN

(MoVe with Offset)

Diese Operation überträgt den Inhalt eines Feldes um ein halbes Byte nach links versetzt in ein anderes Feld.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	M,V,O, .	F,E,L,D,A,,,F,E,L,D,B, ...

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel FELDB) wird in das Feld des ersten Operanden (im Beispiel FELDA) übertragen und zwar um ein halbes Byte nach links versetzt. Das rechte Halbbyte im rechten Byte des ersten Operanden wird nicht verändert.

Keiner der Operanden wird auf Gültigkeit geprüft.

Verarbeitet wird byteweise von rechts nach links.

Ist der erste Operand zu lang, werden führende Nullen eingeschoben. Ist der erste Operand zu kurz, so werden die linken Daten aus dem zweiten Operanden nicht übertragen.

Der Bedingungsschlüssel wird nicht verändert.

Der Inhalt des zweiten Operanden bleibt erhalten.

Die Feldlänge jedes Operanden beträgt maximal 16 Bytes.

Durchführung der Operation:

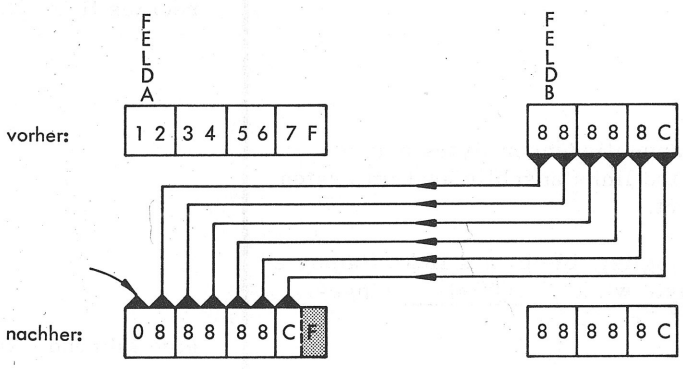


Abb. 1

6 Ein Halbbyte des ersten Operanden bleibt bei der Operation "Umpacken" immer unverändert.

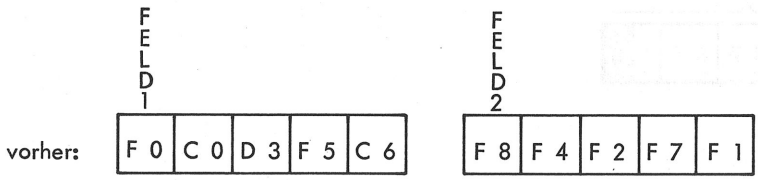
Es ist das _____ Halbbyte des Feldes. In unserem Beispiel enthält dieses Halbbyte die Hexadezimalziffer _(Z)_.

rechte

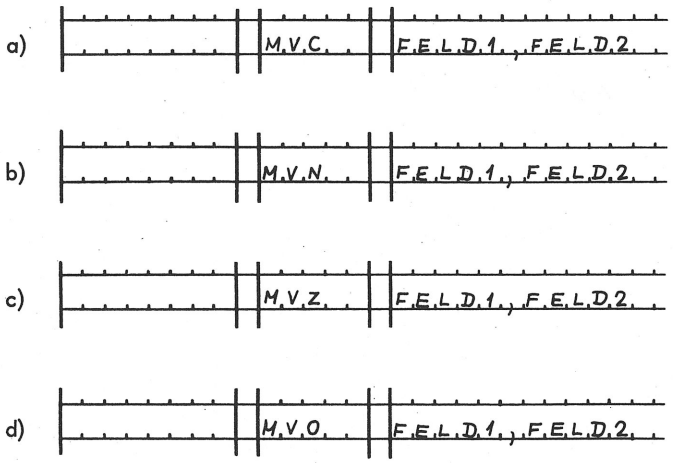
F

7 Die Eigenschaften der Operation "Umpacken" werden besonders deutlich, wenn wir sie mit anderen Operationen vergleichen.

Wir erinnern uns zunächst an die Übertragungsoperationen. Sie übertragen entweder den Inhalt von ganzen Bytes - MVC - oder den Inhalt von halben Bytes - MVN (rechte Halbbytes), MVZ (linke Halbbytes) -.



Schreiben Sie bitte auf ein Extrablatt den jeweiligen Inhalt von FELD1 nach Ausführung der folgenden Instruktionen:



Überprüfen Sie Ihre Ergebnisse auf dem folgenden Blatt.

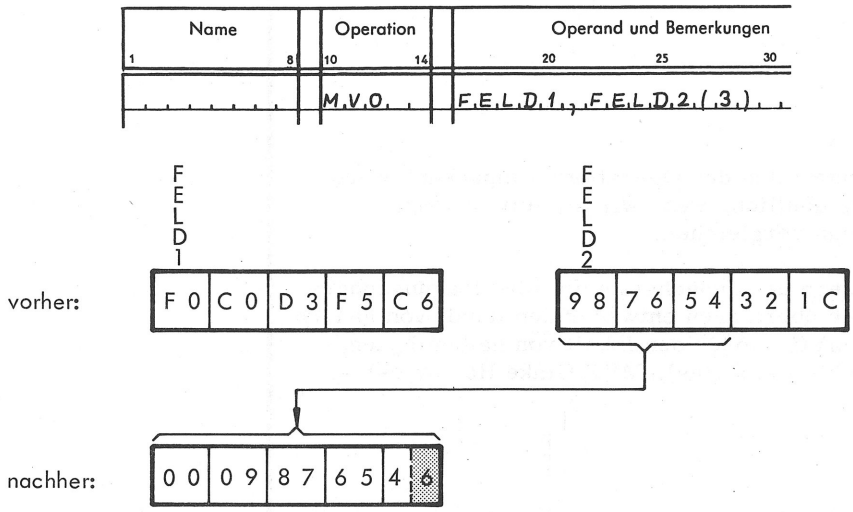

Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
M,V,O,
F,E,L,D,1,,F,E,L,D,2,(,3,)
FELD1
FELD2
vorher:
F 0 C 0 D 3 F 5 C 6
9 8 7 6 5 4 3 2 1 C
nachher:
0 0 0 9 8 7 6 5 4 6

Abb. 2

a) | F 8 | F 4 | F 2 | F 7 | F 1 |

b) | F 8 | C 4 | D 2 | F 7 | C 1 |

c) | F 0 | F 0 | F 3 | F 5 | F 6 |

d) | 8 F | 4 F | 2 F | 7 F | 1 6 |

(F)

Halbbytes, deren Inhalt durch die Operation nicht verändert wurde, sind in der Zeichnung hervorgehoben.

8 Sie haben gesehen:
Bei MVC und MVO ist das Ergebnis der Operation gleich - bis auf einen wesentlichen Unterschied:

Bei der Operation "Umpacken" steht der übertragene Inhalt nachher um ein halbes Byte nach _____ versetzt im ersten Operanden.

links

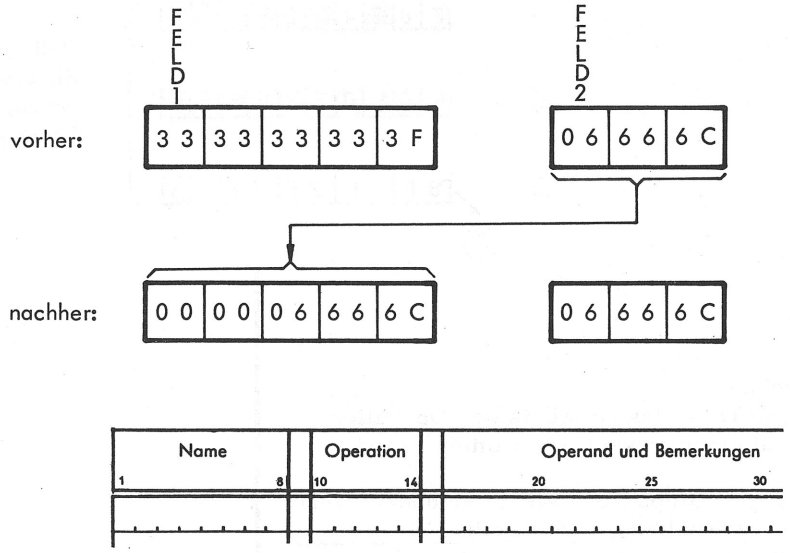
9 Betrachten Sie bitte das Beispiel in Abb. 2.

Die Zahl in FELD2 wird - rechts um 3 Stellen verkürzt - nach FELD1 gebracht. (Anwendung: Abschneiden von Dezimalstellen bei Rechenergebnissen, gegebenenfalls nach dem Runden.)

Obwohl beide Operanden keine gepackten Daten enthalten, wird die Operation durchgeführt. Bei MVO wird nämlich keiner der Operanden ___(WW)___ geprüft.

auf Gültigkeit

1.


FELD 1
FELD 2
vorher:
3 3 3 3 3 3 3 3 3 F
0 6 6 6 6 C
nachher:
0 0 0 0 0 6 6 6 C
0 6 6 6 6 C
Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30

2.

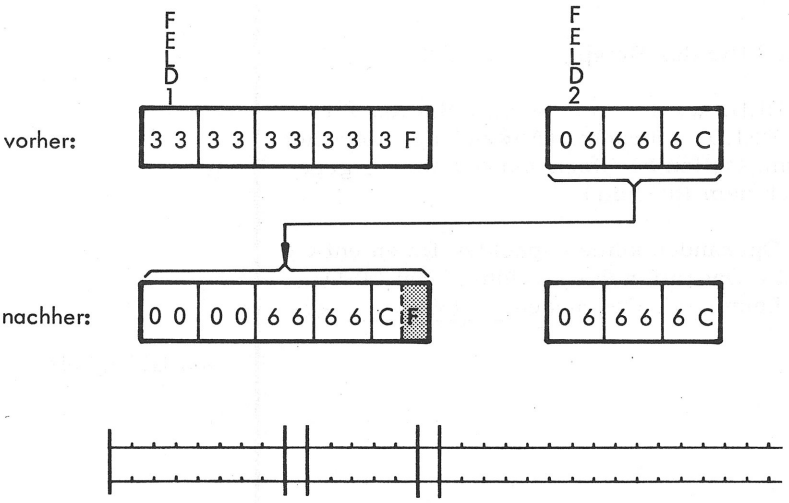
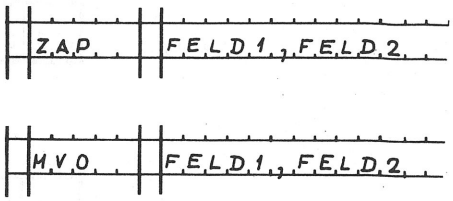

FELD 1
FELD 2
vorher:
3 3 3 3 3 3 3 3 3 F
0 6 6 6 6 C
nachher:
0 0 0 0 6 6 6 6 C F
0 6 6 6 6 C

Abb. 3

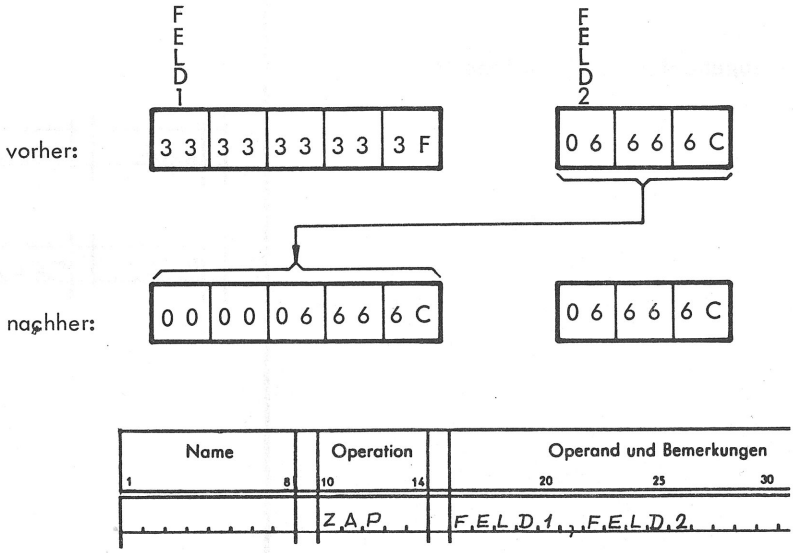
10 Wir vergleichen das "Umpacken" mit einer weiteren Operation.

In Abb. 3 werden zwei Beispiele gezeigt, in denen gepackte Daten aus FELD2 nach FELD1 gebracht werden. Das Ergebnis in FELD1 ist nicht genau gleich.

Wie heißen die zugehörigen Instruktionen?



1.


FELD 1
FELD 2
vorher:
3 3 3 3 3 3 3 3 3 F
0 6 6 6 C
nachher:
0 0 0 0 0 6 6 6 C
0 6 6 6 C
Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
Z A P
F E L D 1 , F E L D 2

2.

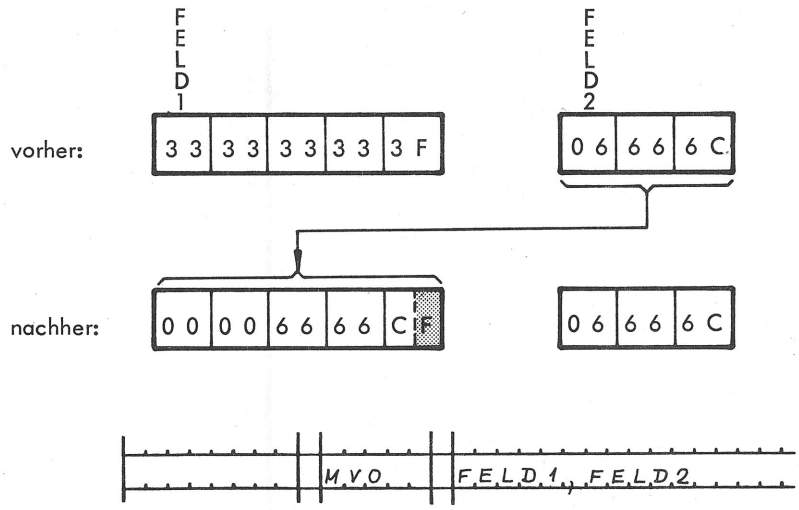

FELD 1
FELD 2
vorher:
3 3 3 3 3 3 3 3 3 F
0 6 6 6 C
nachher:
0 0 0 0 6 6 6 C F
0 6 6 6 C
M V O
F E L D 1 , F E L D 2

Abb. 4

11 Abb. 4 enthält die erforderlichen Instruktionen. Der Unterschied in der Wirkung der beiden Operationen besteht im folgenden:

Mit der Operation Umpacken wird der Inhalt um ein halbes Byte nach links verschoben eingefüllt.

Dadurch bleibt das "F" im ___(WW)___ erhalten.

Links entsteht dadurch eine ___(WW)___ weniger.

rechten Halbbyte

führende Null

12 Sowohl bei ZAP als auch bei MVO werden bei längerem erstem Operanden links führende Nullen eingesteuert.

Diese Eigenschaft ist es unter anderem, die MVO in die Rechenoperationen einreiht.

Bei allen diesen arithmetischen Operationen wie PACK, UNPK, AP, MVO usw. wird die Länge _____ Operanden berücksichtigt.

beider

13 Im Gegensatz zu AP, SP und ZAP wird bei MVO der Bedingungsschlüssel ___(WW)___.

Ist der erste Operand zu kurz, um den ganzen Inhalt des zweiten Operanden aufzunehmen, so wird bei MVO kein _____.____ angezeigt.

Sinngemäß:

nicht neu eingestellt

Überlauf

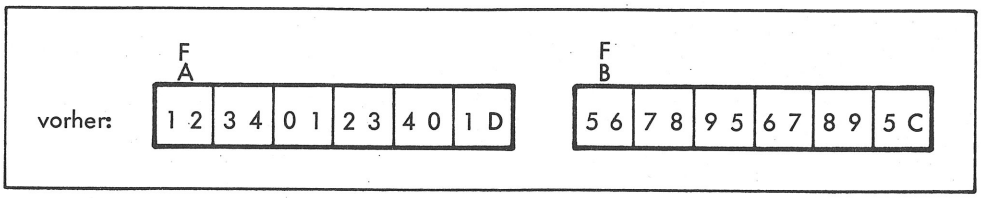
Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 418, Seite 1, 2 3 und 4.

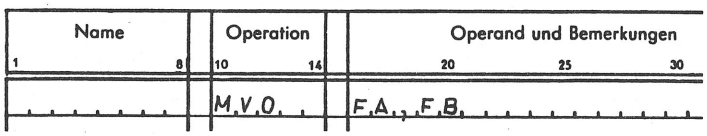
Tragen Sie Ihre Lösungen ein.

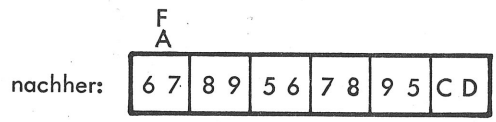
Prüfen Sie Ihre Eintragungen anhand der folgenden Lösungsblätter, sobald Sie eine Seite vollständig bearbeitet haben.

1.


vorher:
FA
1 2 3 4 0 1 2 3 4 0 1 D
FB
5 6 7 8 9 5 6 7 8 9 5 C

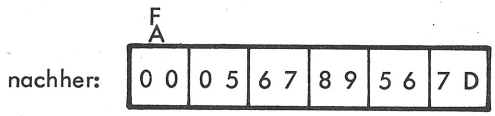
a)


Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
M,V,O,
F,A,,F,B,


nachher:
FA
6 7 8 9 5 6 7 8 9 5 C D

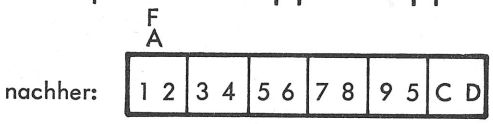
b)


M,V,O,
F,A,,F,B,(,4,)


nachher:
FA
0 0 0 5 6 7 8 9 5 6 7 D

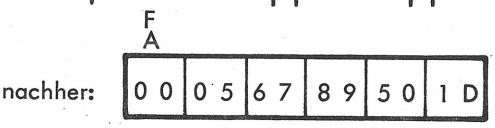
c)

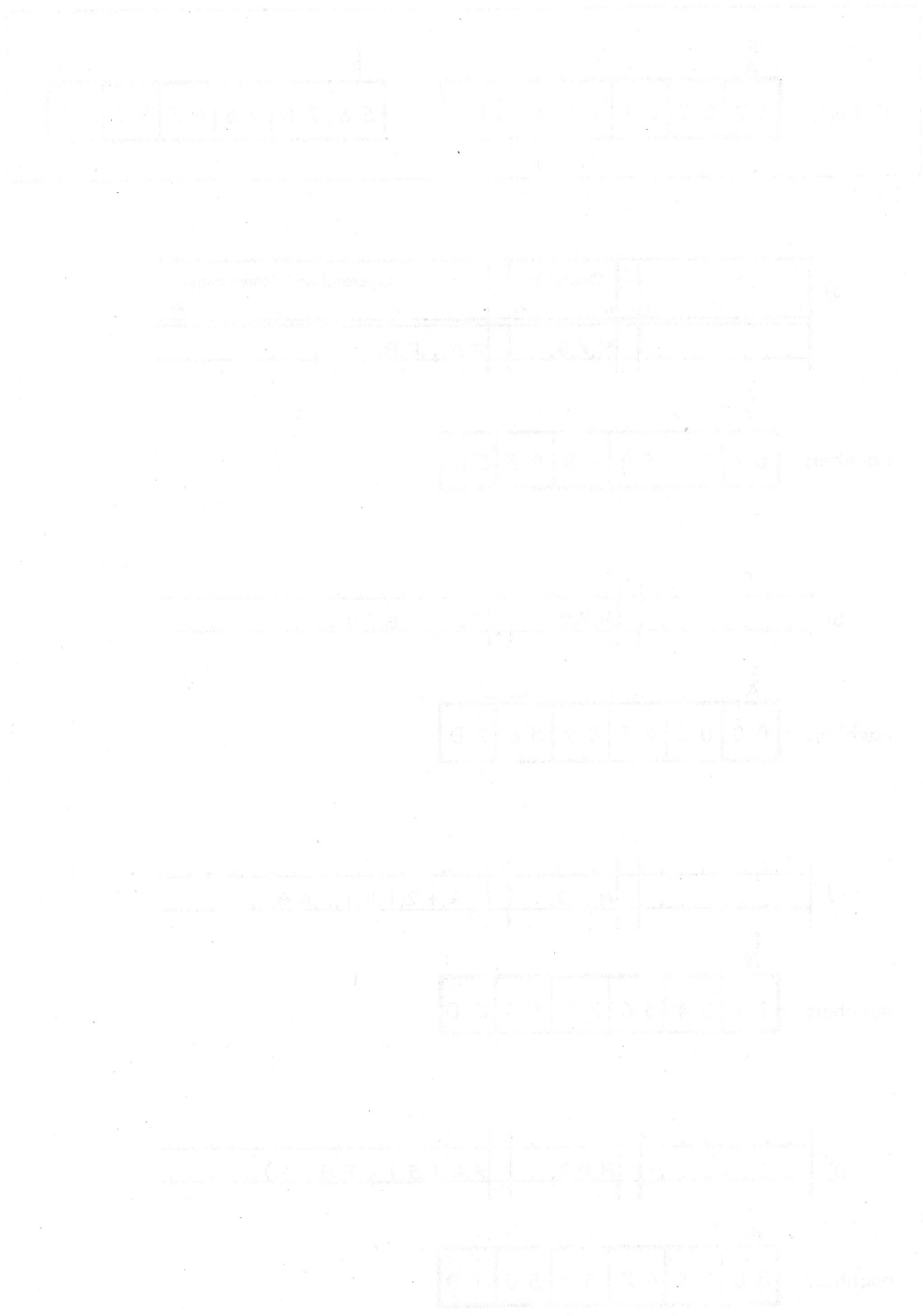

M,V,O,
F,A,+,2,(,4,),,F,B,


nachher:
FA
1 2 3 4 5 6 7 8 9 5 C D

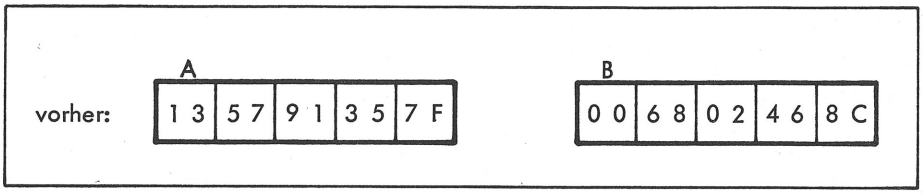
d)


M,V,O,
F,A,(,5,),,F,B,(,3,)

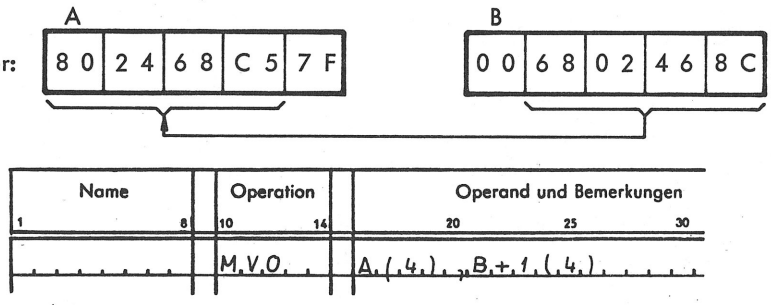

nachher:
FA
0 0 0 5 6 7 8 9 5 0 1 D



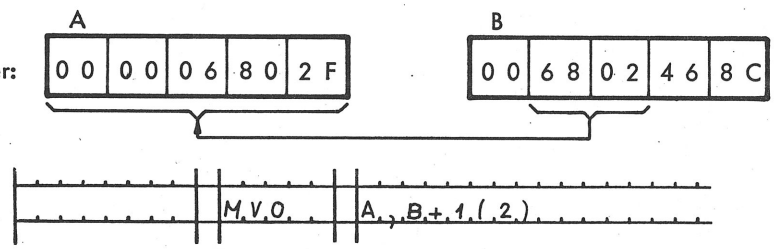
2.



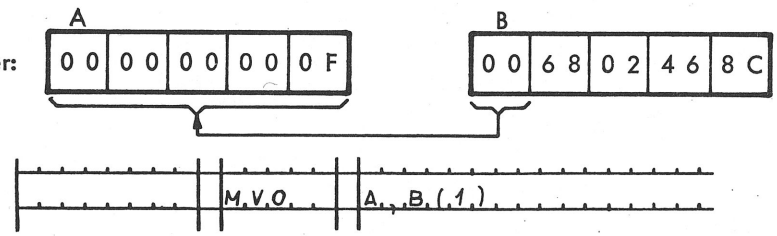
a) nachher:



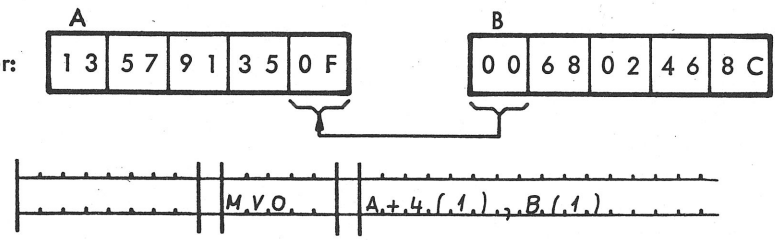
b) nachher:



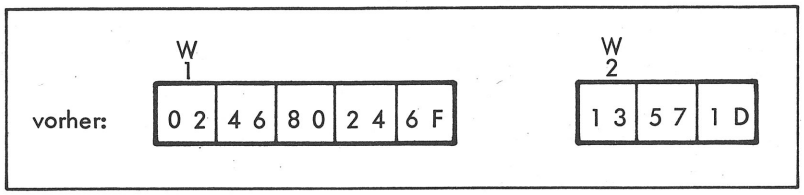
c) nachher:



d) nachher:



3.



a)

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,O,	W,1,,W,2,

W 1
nachher: 0 0 0 1 3 5 7 1 D F

b)

	M,V,O,	W,1,,W,2,(,1,)

W 1
nachher: 0 0 0 0 0 0 1 3 F

c)

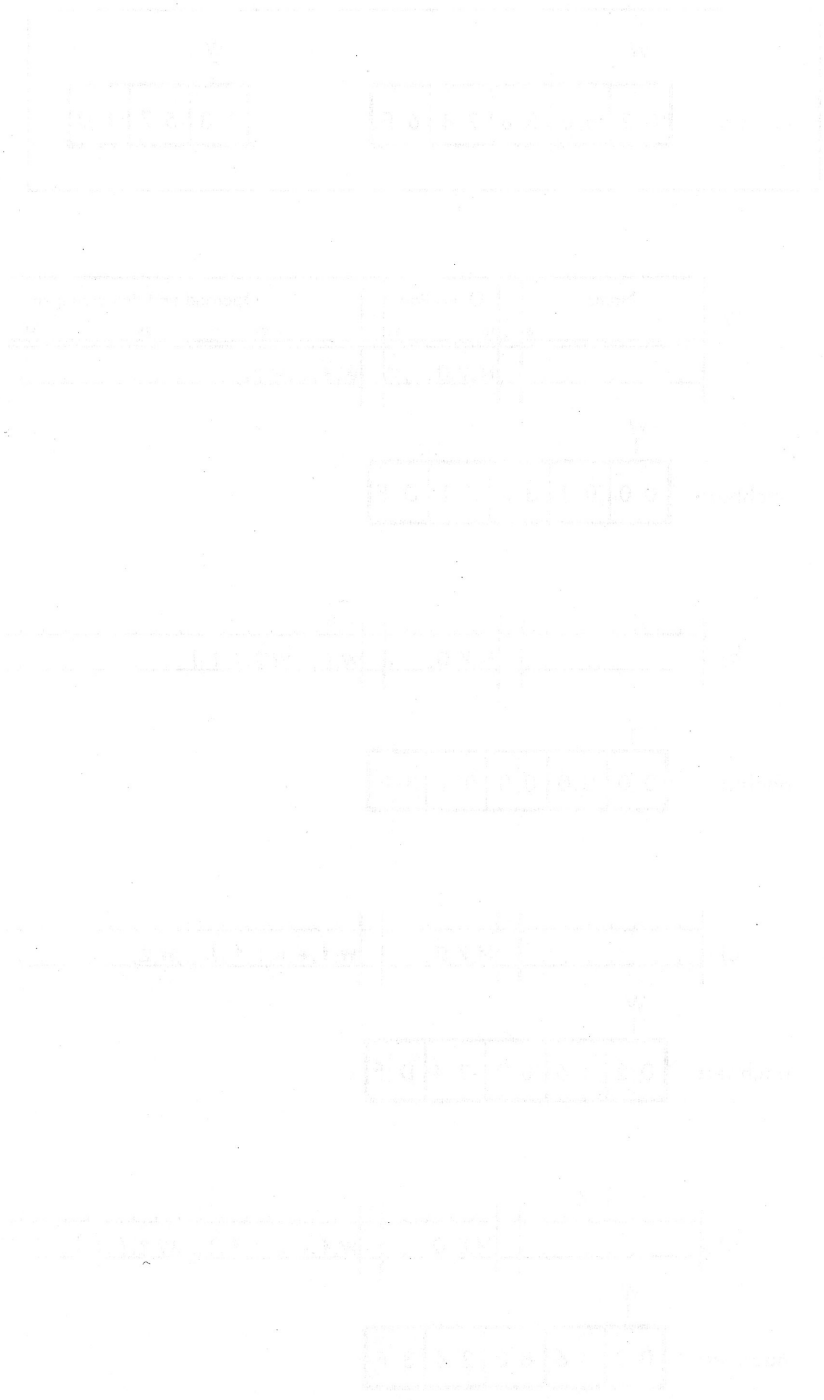
	M,V,O,	W,1,+,4,(,1,),,W,2,

W 1
nachher: 0 2 4 6 8 0 2 4 D F

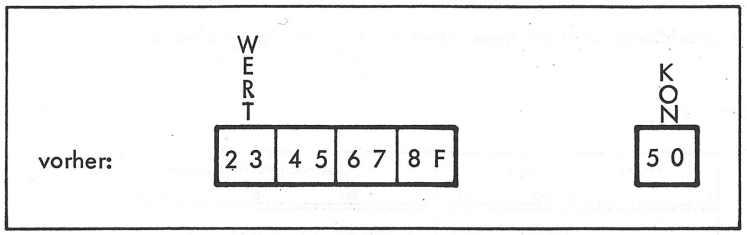
d)

	M,V,O,	W,1,+,4,(,1,),,W,2,(,1,)

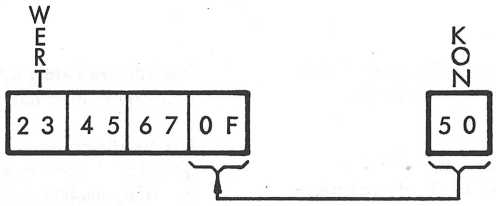
W 1
nachher: 0 2 4 6 8 0 2 4 3 F

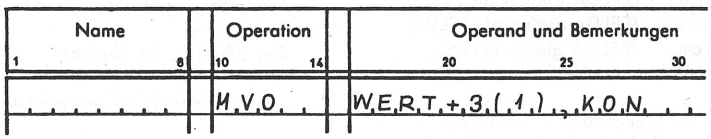


4.

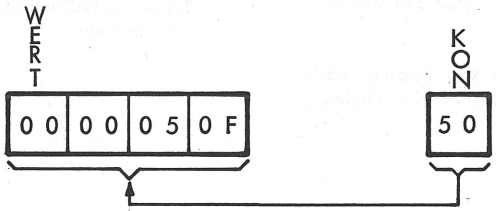


a) nachher:



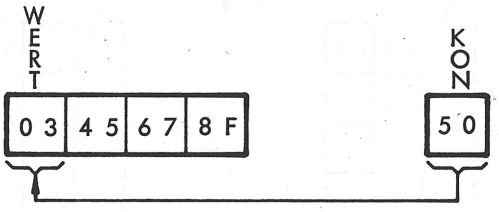


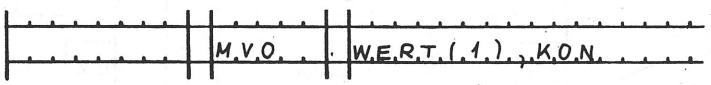
b) nachher:





c) nachher:





Merkblatt 417 enthält die Beschreibung der Operation "Umpacken" (vgl. Abb. 1).

Band 1, Seite 241

CP

VERGLEICHEN DEZIMAL (GEPACKT)

(Compare decimal Packed)

Diese Operation vergleicht vorzeichengerecht den (gepackten) Inhalt zweier Speicherfelder.

Beispiel einer Instruktion:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
	C,P,	F,E,L,D,A,,F,E,L,D,B,

Wirkungsweise:

Der Inhalt des ersten Operanden (im Beispiel FELDA) wird mit dem Inhalt des zweiten Operanden (im Beispiel FELDB) verglichen.

Das Vergleichsergebnis, das sich immer auf den ersten Operanden bezieht, wird im Bedingungsschlüssel angezeigt.
Es lautet kleiner (FELDA kleiner als FELDB)
oder größer (FELDA größer als FELDB)
oder gleich (FELDA gleich FELDB).
Ein Überlauf kann nicht auftreten.

Vor Durchführung der Operation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

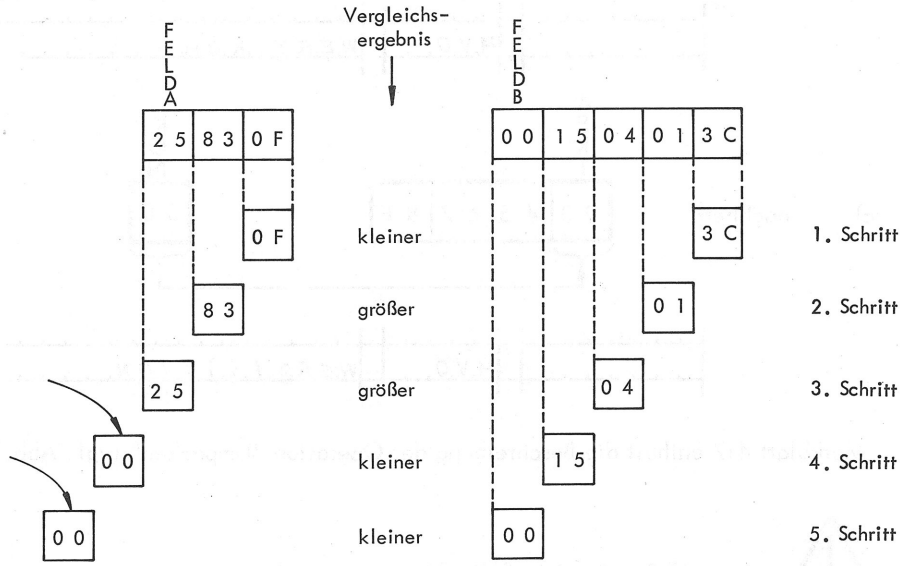
Der Vergleich erfolgt algebraisch (vorzeichengerecht). Negative Zahlen sind immer kleiner als positive, unabhängig von der Verschlüsselung der Vorzeichen.
Sonderfall: "plus null" ist gleich "minus null".

Der Inhalt beider Operanden bleibt unverändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler tritt Programmunterbrechung ein.

Durchführung der Operation:



Endergebnis: FELDA kleiner als FELDB

Abb. 1

Wir werden die Operation "Vergleichen Dezimal (Gepackt)" in zwei Teilen besprechen.

Zunächst beschäftigen wir uns mit den Eigenschaften von CP, wie sie in Abb. 1 beschrieben sind.

Anschließend werden wir CP der logischen Vergleichsoperation CLC gegenüberstellen. Wir werden die Unterschiede herausarbeiten und dadurch die Eigenart beider Operationen deutlicher erkennen.

1 Die dezimale Vergleichsoperation hat den Operationsschlüssel _____.

Das Vergleichsergebnis bezieht sich immer auf den _____ Operanden.

CP

ersten

2 Vor Beginn der Operation wird der Inhalt der Operanden auf _____ Darstellung geprüft.

Bei Gültigkeitsfehler tritt eine Programmunterbrechung ein.

gepackte

3 Im Beispiel in Abb. 1 haben die beiden Operanden verschiedene Länge.

Die Verarbeitung wird durchgeführt in der Länge des __(Z)__ Operanden.

Bei verschiedener Länge wird der dezimale Vergleich immer in der Länge des (kürzeren, längeren) _____ Operanden durchgeführt.

2.

längeren

CP

VERGLEICHEN DEZIMAL (GEPACKT)

(Compare decimal Packed)

Diese Operation vergleicht vorzeichengerecht den (gepackten) Inhalt zweier Speicherfelder.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	C,P,	F,E,L,D,A,,F,E,L,D,B,

Wirkungsweise:

Der Inhalt des ersten Operanden (im Beispiel FELDA) wird mit dem Inhalt des zweiten Operanden (im Beispiel FELDB) verglichen.

Das Vergleichsergebnis, das sich immer auf den ersten Operanden bezieht, wird im Bedingungsschlüssel angezeigt.
Es lautet kleiner (FELDA kleiner als FELDB)
oder größer (FELDA größer als FELDB)
oder gleich (FELDA gleich FELDB).
Ein Überlauf kann nicht auftreten.

Vor Durchführung der Operation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

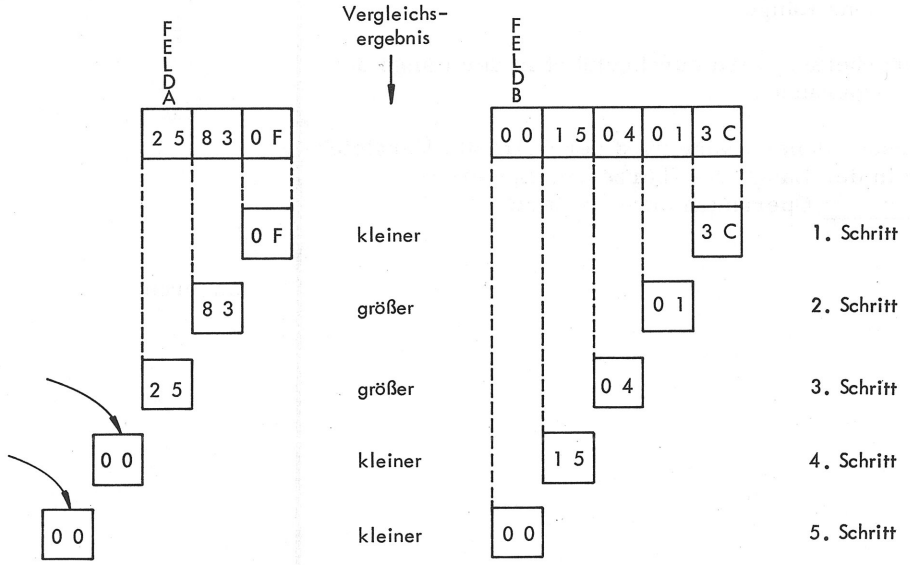
Der Vergleich erfolgt algebraisch (vorzeichengerecht). Negative Zahlen sind immer kleiner als positive, unabhängig von der Verschlüsselung der Vorzeichen.
Sonderfall: "plus null" ist gleich "minus null".

Der Inhalt beider Operanden bleibt unverändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler tritt Programmunterbrechung ein.

Durchführung der Operation:



Endergebnis: FELDA kleiner als FELDB

Abb. 1

4 In Abb. 1 ist der erste Operand um zwei Bytes kürzer als der zweite.

Darum wird so verglichen, als ob der erste Operand zusätzlich zu seinen drei Bytes noch zwei Bytes mit ____(WW)___ hätte.

Das links an FELDA anschließende Speicherfeld wird dadurch in seinem Inhalt (verändert, nicht verändert) _____

führenden Nullen

nicht verändert

5 FELDA und FELDB werden Byte für Byte in 5 Schritten verglichen.

Der Vergleich beginnt in den beiden _____ Bytes und geht nach _____ hin weiter.

rechten

links

6 Bei jedem Schritt kann das vorherige Vergleichsergebnis umgeworfen werden. Das endgültige Ergebnis wird erst gebildet, wenn alle Bytes - in der Länge des längeren Operanden - verglichen sind.

Im Beispiel in Abb. 1 wird das beim 1. Schritt ermittelte vorläufige Ergebnis zweimal umgeworfen und zwar beim ___(Z)____ und ___(Z)____ Vergleichsschritt.

2.

4.

(Beachten Sie bitte, daß bei führenden Nullen das zuletzt ermittelte Vergleichsergebnis erhalten bleibt. 5. Schritt Abb. 1).

7 Der dezimale Vergleich endet erst, wenn alle Bytes verglichen sind. Jedes weiter links stehende Byte kann das bisherige Ergebnis umwerfen.

Dadurch wird die Tatsache berücksichtigt, daß der Stellenwert der Ziffern bei einer Dezimalzahl nach links hin immer (kleiner, größer) _____ wird.

größer
(Weil dagegen CLC von links nach rechts vergleicht, steht bei CLC das Ergebnis bei der ersten Ungleichheit schon fest.)

CP

VERGLEICHEN DEZIMAL (GEPACKT)

(Compare decimal Packed)

Diese Operation vergleicht vorzeichengerecht den (gepackten) Inhalt zweier Speicherfelder.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	C,P,	F,E,L,D,A,,F,E,L,D,B,

Wirkungsweise:

Der Inhalt des ersten Operanden (im Beispiel FELDA) wird mit dem Inhalt des zweiten Operanden (im Beispiel FELDB) verglichen.

Das Vergleichsergebnis, das sich immer auf den ersten Operanden bezieht, wird im Bedingungsschlüssel angezeigt.
Es lautet kleiner (FELDA kleiner als FELDB)
oder größer (FELDA größer als FELDB)
oder gleich (FELDA gleich FELDB).
Ein Überlauf kann nicht auftreten.

Vor Durchführung der Operation werden beide Operanden auf Gültigkeit (gepackte Daten) geprüft.

Verarbeitet wird byteweise von rechts nach links in der Länge des längeren Operanden.

Das kürzere Feld wird so verarbeitet, als ob es führende Nullen hätte.

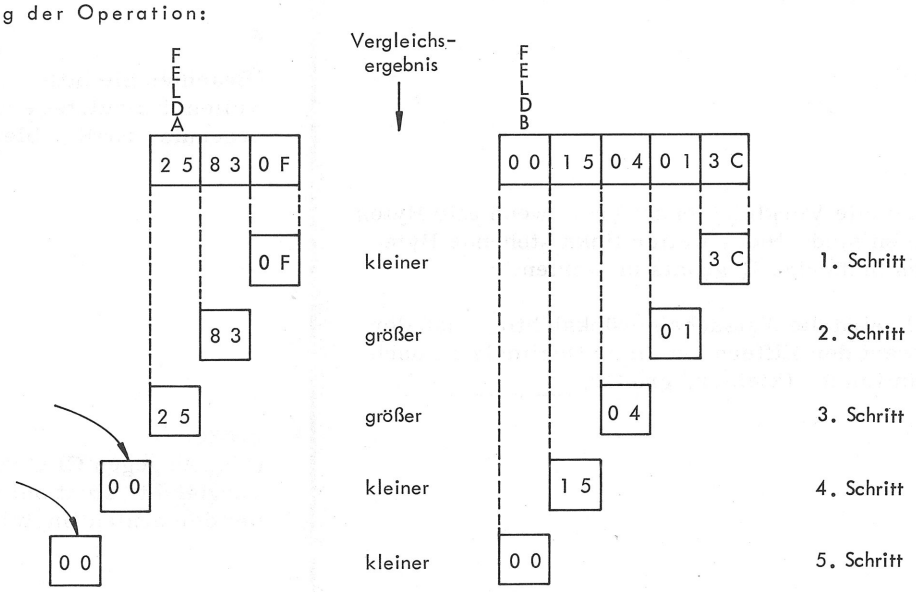
Der Vergleich erfolgt algebraisch (vorzeichengerecht). Negative Zahlen sind immer kleiner als positive, unabhängig von der Verschlüsselung der Vorzeichen.
Sonderfall: "plus null" ist gleich "minus null".

Der Inhalt beider Operanden bleibt unverändert.

Jeder Operand darf maximal 16 Bytes lang sein.

Bei einem Gültigkeitsfehler tritt Programmunterbrechung ein.

Durchführung der Operation:



Endergebnis: FELDA kleiner als FELDB

Abb. 1

8 Nachfolgende bedingte Verzweigoperationen fragen das Vergleichsergebnis ab. Sie fragen die Stellung des _____ ab.

Ein Überlauf kann nicht auftreten. Daher heißt das Vergleichsergebnis entweder _____ oder _____ oder _____.

Bedingungsschlüssels

kleiner (oder) größer (oder) gleich

9 Der dezimale Vergleich erfolgt vorzeichengerecht.

Als Vorzeichenverschlüsselung treten im allgemeinen die drei Hexadezimalziffern __(ZZ)__ auf.
Das Vorzeichen " + " wird dargestellt durch __(Z)__ oder __(Z)__ und das Vorzeichen " - " durch __(Z)__.

C, D (und) F

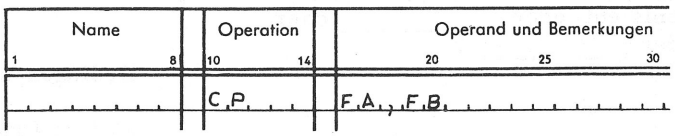
C (oder) F

D

Entnehmen Sie Ihren Arbeitsmitteln Übung 419.

Tragen Sie aufgrund der Angaben in Abb. 1 die Vergleichsergebnisse ein.

Abb. 2 (nächste Seite) enthält die richtigen Ergebnisse.



	FA	Vergleichsergebnis	FB
1.	08045C	gleich	08045F
2.	08045D	kleiner	08045C
3.	08045D	kleiner	08045F
4.	00000C	gleich	00000F
5.	00000D	gleich	00000C
6.	00000D	gleich	00000F

Abb. 2

10 Sie sehen: Bei CP erfolgt der Vergleich vorzeichengerecht.

Negative Beträge sind immer _____ als positive.

Die einzige Ausnahme stellt der Wert _____ dar.

Sinnvollerweise ist
"plus null" _____ "minus null".

kleiner

null

gleich

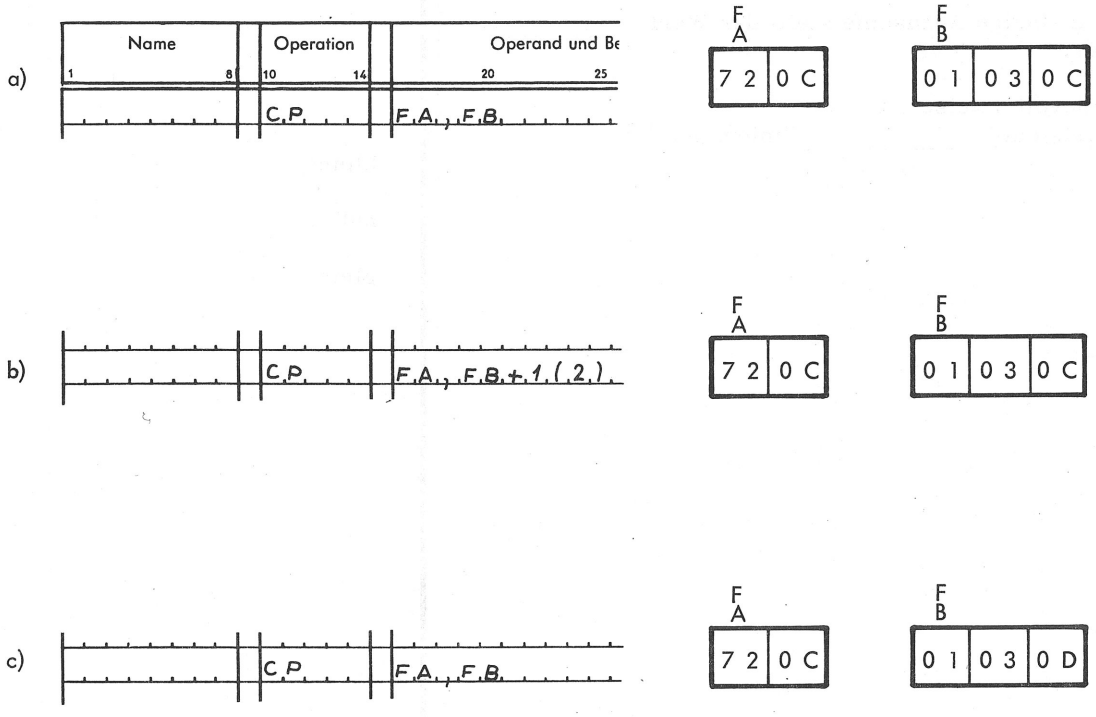

a)
Name
Operation
Operand und Be
1
8
10
14
20
25
C,P,
F,A,,F,B,
F A
7 2 0 C
F B
0 1 0 3 0 C
b)
C,P,
F,A,,F,B,+1,(,2,)
F A
7 2 0 C
F B
0 1 0 3 0 C
c)
C,P,
F,A,,F,B,
F A
7 2 0 C
F B
0 1 0 3 0 D

Abb. 3

11 Die Vorzeichenregeln gelten natürlich nicht nur bei Feldern gleicher Länge.

Betrachten Sie die Beispiele a), b) und c) in Abb. 3.

Geben Sie jeweils an, welche Ziffern - einschließlich ergänzter führender Nullen - verglichen werden und wie die Vergleichsergebnisse lauten.

a) 00 72 0C und 01 03 0C, "kleiner"

b) 72 0C und 03 0C, "größer"

c) 00 72 0C und 01 03 0D, "größer"

12 Im Beispiel b) in Abb. 3 hat der 2. Operand eine explizite Längenangabe.

Bei der Operation CP wird die Länge beider Operanden berücksichtigt. Darum dürfen beide Operanden explizite Längenangaben haben.

Diese Eigenschaft ordnet CP den (logischen, arithmetischen) _____ Operationen zu.

arithmetischen

Operation		Verarbeit. Richtung	Verarbeitet werden	Kürzerer Operand mit führenden Nullen verarbeitet		Gültigkeitsprüfung (gepackte Daten)		Bed.-Schl. neu gesetzt auf "gleich" od. "kleiner" od. "größer"
Schlüssel	Bezeichnung			1. Operand	2. Operand	1. Operand	2. Operand	
CLC	Vergleichen Logisch Zeichen	→	Bits					X
CP								

Abb. 4

Wir führen jetzt die Gegenüberstellung zwischen dezimalem und logischem Vergleich durch.

13 In Abb. 4 sehen Sie eine Tabelle, in der die wichtigsten Eigenschaften der Operation CLC und CP kurz und übersichtlich angegeben werden können. Für den logischen Vergleich sind diese Eigenschaften bereits eingetragen.

Betrachten Sie bitte den Aufbau der Tabelle.

Für CP ist zunächst die Operationsbezeichnung
___(WW)___ zu ergänzen.

Notieren Sie sich auf einem Extrablatt, wie die restlichen Felder für CP auszufüllen sind.
Die Lösung finden Sie in Abb. 5 auf der nächsten Seite.

Vergleichen Dezimal (Gepackt)

Operation		Verarbeit.-Richtung	Verarbeitet werden	Kürzerer Operand mit führenden Nullen verarbeitet		Gültigkeitsprüfung (gepackte Daten)		Bed.-Schl. neu gesetzt auf "gleich" od. "kleiner" od. "größer"
Schlüssel	Bezeichnung			1. Operand	2. Operand	1. Operand	2. Operand	
CLC	Vergleichen Logisch Zeichen	→	Bits					X
CP	Vergleichen Dezimal (Gepackt)	←	Bytes	X	X	X	X	X

Für CLC gilt außerdem:

1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Länge des Vergleichs:
   bis zur 1. Ungleichheit, höchstens jedoch in der Länge des 1. Operanden.
3. Max. Länge des 1. Operanden: 256 Bytes.

Abb. 5

14 Zur vollständigen Charakterisierung von CLC und CP gehören noch einige weitere Eigenschaften. Für CLC sind diese unterhalb der Tabelle in Abb. 5 aufgeführt.

Entsprechend sind für CP folgende Aussagen zu ergänzen.

1. Das Vergleichsergebnis bezieht sich immer auf den ___(ZW)___.
2. Die Länge des Vergleichs richtet sich nach der Länge des ___(WW)___.
3. Der Vergleich erfolgt algebraisch, d.h. unter Berücksichtigung des _____.
4. Die maximale Länge für beide Operanden ist ___(ZW)___.

1. Operanden

längeren Operanden

Vorzeichens

16 Bytes

Operation		Verarbeit.-Richtung	Verarbeitet werden	Kürzerer Operand mit führenden Nullen verarbeitet		Gültigkeitsprüfung (gepackte Daten)		Bed.-Schl. neu gesetzt auf "gleich" od. "kleiner" od. "größer"
Schlüssel	Bezeichnung			1. Operand	2. Operand	1. Operand	2. Operand	
CLC	Vergleichen Logisch Zeichen	→	Bits					X
CP	Vergleichen Dezimal (Gepackt)	←	Bytes	X	X	X	X	X

Für CLC gilt außerdem:

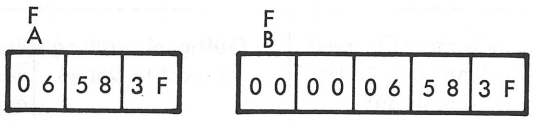
1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Länge des Vergleichs:
   bis zur 1. Ungleichheit, höchstens jedoch in der Länge des 1. Operanden.
3. Max. Länge des 1. Operanden: 256 Bytes.

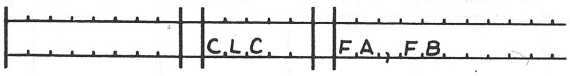
Für CP gilt außerdem:

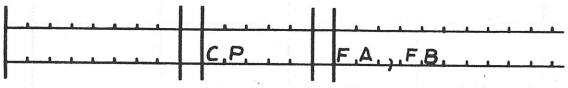
1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Die Länge des Vergleichs richtet sich nach der Länge des längeren Operanden.
3. Der Vergleich erfolgt algebraisch, d.h. vorzeichengerecht.
4. Die maximale Länge für beide Operanden ist 16 Bytes.

Abb. 6

15 Betrachten Sie die folgenden Operandenfelder und Instruktionen:







Beide Felder enthalten den gleichen Wert. Der 2. Operand enthält allerdings zusätzlich ___(ZW)___.

Überlegen Sie sich, wie die beiden Vergleiche durchgeführt werden.
Nehmen Sie evtl. Abb. 6 zu Hilfe.

Das Vergleichsergebnis heißt

bei CLC: _____

bei CP: _____

4 führende Nullen

größer/FA ist größer als FB
(da 6 größer als 0)

gleich/FA ist gleich FB

Operation		Verarbeit.-Richtung	Verarbeitet werden	Kürzerer Operand mit führenden Nullen verarbeitet		Gültigkeitsprüfung (gepackte Daten)		Bed.-Schl. neu gesetzt auf "gleich" od. "kleiner" od. "größer"
Schlüssel	Bezeichnung			1. Operand	2. Operand	1. Operand	2. Operand	
CLC	Vergleichen Logisch Zeichen	→	Bits					X
CP	Vergleichen Dezimal (Gepackt)	←	Bytes	X	X	X	X	X

Für CLC gilt außerdem:

1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Länge des Vergleichs: bis zur 1. Ungleichheit, höchstens jedoch in der Länge des 1. Operanden.
3. Max. Länge des 1. Operanden: 256 Bytes.

Für CP gilt außerdem:

1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Die Länge des Vergleichs richtet sich nach der Länge des längeren Operanden.
3. Der Vergleich erfolgt algebraisch, d.h. vorzeichengerecht.
4. Die maximale Länge für beide Operanden ist 16 Bytes.

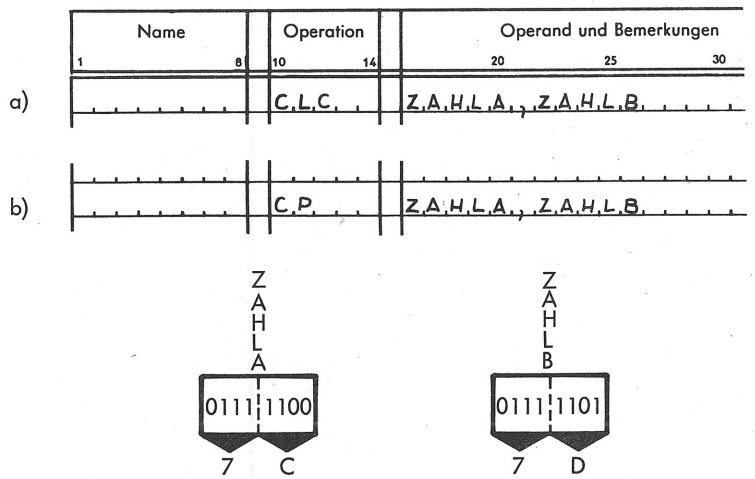


Abb. 7

16 Ein wichtiger Unterschied zwischen CP und CLC wird deutlich, wenn wir Zahlen mit unterschiedlichen Vorzeichen oder nur unterschiedlicher Vorzeichenverschlüsselung vergleichen.

In Abb. 7 unten werden zwei einstellige gepackte Dezimalzahlen verglichen.

ZAHLA enthält eine (positive, negative) _____ "7" und ZAHLB enthält eine _____ "7".

positive

negative

17 Die Zahlen in ZAHLA und ZAHLB werden a) durch CLC und b) durch CP verglichen.

Wie heißen die beiden Vergleichsergebnisse? (Nehmen Sie die Tabelle oben in Abb. 7 zu Hilfe.)

a) "kleiner"
(Bitposition 7 bringt die Entscheidung)

b) "größer"

Das Vergleichsergebnis "+7 ist größer als -7" bringt nur die Operation CP.

Beim logischen Vergleich wird Bit für Bit von links nach rechts verglichen - ohne Rücksicht auf die Vorzeichenbedeutung von C und D im rechten Halbbyte.

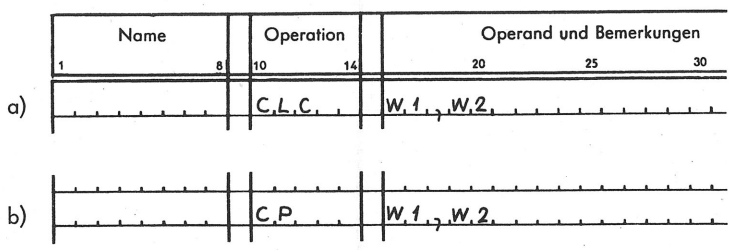
Operation		Verarbeit.-Richtung	Verarbeitet werden	Kürzerer Operand mit führenden Nullen verarbeitet		Gültigkeitsprüfung (gepackte Daten)		Bed.-Schl. neu gesetzt auf "gleich" od. "kleiner" od. "größer"
Schlüssel	Bezeichnung			1. Operand	2. Operand	1. Operand	2. Operand	
CLC	Vergleichen Logisch Zeichen	→	Bits					X
CP	Vergleichen Dezimal (Gepackt)	←	Bytes	X	X	X	X	X

Für CLC gilt außerdem:

1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Länge des Vergleichs: bis zur 1. Ungleichheit, höchstens jedoch in der Länge des 1. Operanden.
3. Max. Länge des 1. Operanden: 256 Bytes.

Für CP gilt außerdem:

1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Die Länge des Vergleichs richtet sich nach der Länge des längeren Operanden.
3. Der Vergleich erfolgt algebraisch, d.h. vorzeichengerecht.
4. Die maximale Länge für beide Operanden ist 16 Bytes.





Binäre Verschlüsselung:
C - 1100
F - 1111

Abb. 8

18 In den Beispielen in Abb. 8 werden zwei positive Zahlen von gleichem Wert verglichen. Nur die Verschlüsselung des positiven Vorzeichens ist verschieden.

Die beiden Vergleichsergebnisse lauten:

a) "_____"

b) "_____"

a) kleiner

b) gleich

Auch in diesem Fall bringt nur die Operation CP das erwartete Ergebnis "gleich".

Für den logischen Vergleich dagegen ist die Hexadezimalziffer "C" kleiner als "F", und dadurch heißt das Vergleichsergebnis "kleiner".

Operation		Verarbeit.-Richtung	Verarbeitet werden	Kürzerer Operand mit führenden Nullen verarbeitet		Gültigkeitsprüfung (gepackte Daten)		Bed.-Schl. neu gesetzt auf "gleich" od. "kleiner" od. "größer"
Schlüssel	Bezeichnung			1. Operand	2. Operand	1. Operand	2. Operand	
CLC	Vergleichen Logisch Zeichen	→	Bits					X
CP	Vergleichen Dezimal (Gepackt)	←	Bytes	X	X	X	X	X

Für CLC gilt außerdem:

1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Länge des Vergleichs:
   bis zur 1. Ungleichheit, höchstens jedoch in der Länge des 1. Operanden.
3. Max. Länge des 1. Operanden: 256 Bytes.

Für CP gilt außerdem:

1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Die Länge des Vergleichs richtet sich nach der Länge des längeren Operanden.
3. Der Vergleich erfolgt algebraisch, d.h. vorzeichengerecht.
4. Die maximale Länge für beide Operanden ist 16 Bytes.

Abb. 6

Wir fassen zu CP und CLC zusammen:

19 Die Operation CLC wird immer ausgeführt, bringt aber beim Vergleich von Zahlen nicht immer das erwartete Ergebnis. Das kommt daher, daß der logische Vergleich von links nach rechts _____ für _____ vergleicht.

Bit (für) Bit

20 Für Zahlen eignet sich CLC nur dann, wenn die Vorzeichen bei beiden Operanden gleich verschlüsselt sind (z.B. bei zwei Konto-Nummern).

Ist diese Bedingung nicht erfüllt, so muß CP verwendet werden. CP vergleicht _____-gerecht.

vorzeichen

21 Es ist aber zu beachten, daß bei CP beide Operanden auf _____ geprüft werden.

CP eignet sich also nur für Zahlen und zwar Zahlen in _____ Darstellungsform.

Gültigkeit

gepackter

22 Bei unterschiedlicher Länge der Operanden ist noch zu beachten, daß bei CLC maximal in der Länge des _____ Operanden verglichen wird. Dann bricht der Vergleich ab ohne Rücksicht darauf, ob der zweite Operand kürzer oder länger ist.

Bei CP dagegen wird solange verglichen, bis auch das linke Byte des längeren Operanden verglichen ist. Beim kürzeren Operanden werden links ___(WW)___ verglichen.

ersten

führende Nullen

Operation		Verarbeit.-Richtung	Verarbeitet werden	Kürzerer Operand mit führenden Nullen verarbeitet		Gültigkeitsprüfung (gepackte Daten)		Bed.-Schl. neu gesetzt auf "gleich" od. "kleiner" od. "größer"
Schlüssel	Bezeichnung			1. Operand	2. Operand	1. Operand	2. Operand	
CLC	Vergleichen Logisch Zeichen	→	Bits					X
CP	Vergleichen Dezimal (Gepackt)	←	Bytes	X	X	X	X	X

Für CLC gilt außerdem:

1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Länge des Vergleichs: bis zur 1. Ungleichheit, höchstens jedoch in der Länge des 1. Operanden.
3. Max. Länge des 1. Operanden: 256 Bytes.

Für CP gilt außerdem:

1. Das Vergleichsergebnis bezieht sich immer auf den 1. Operanden.
2. Die Länge des Vergleichs richtet sich nach der Länge des längeren Operanden.
3. Der Vergleich erfolgt algebraisch, d.h. vorzeichengerecht.
4. Die maximale Länge für beide Operanden ist 16 Bytes.

Abb. 6

Übungen

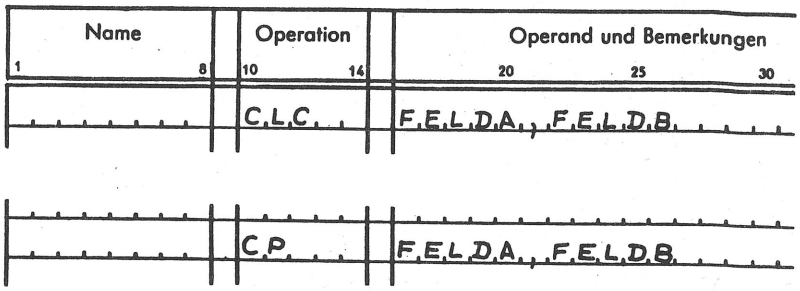
Entnehmen Sie Ihren Arbeitsmitteln Übung 420, Seite 1 und 2.

Tragen Sie Ihre Lösungen ein. (Benutzen Sie Abb. 6 als Hilfsmittel.)

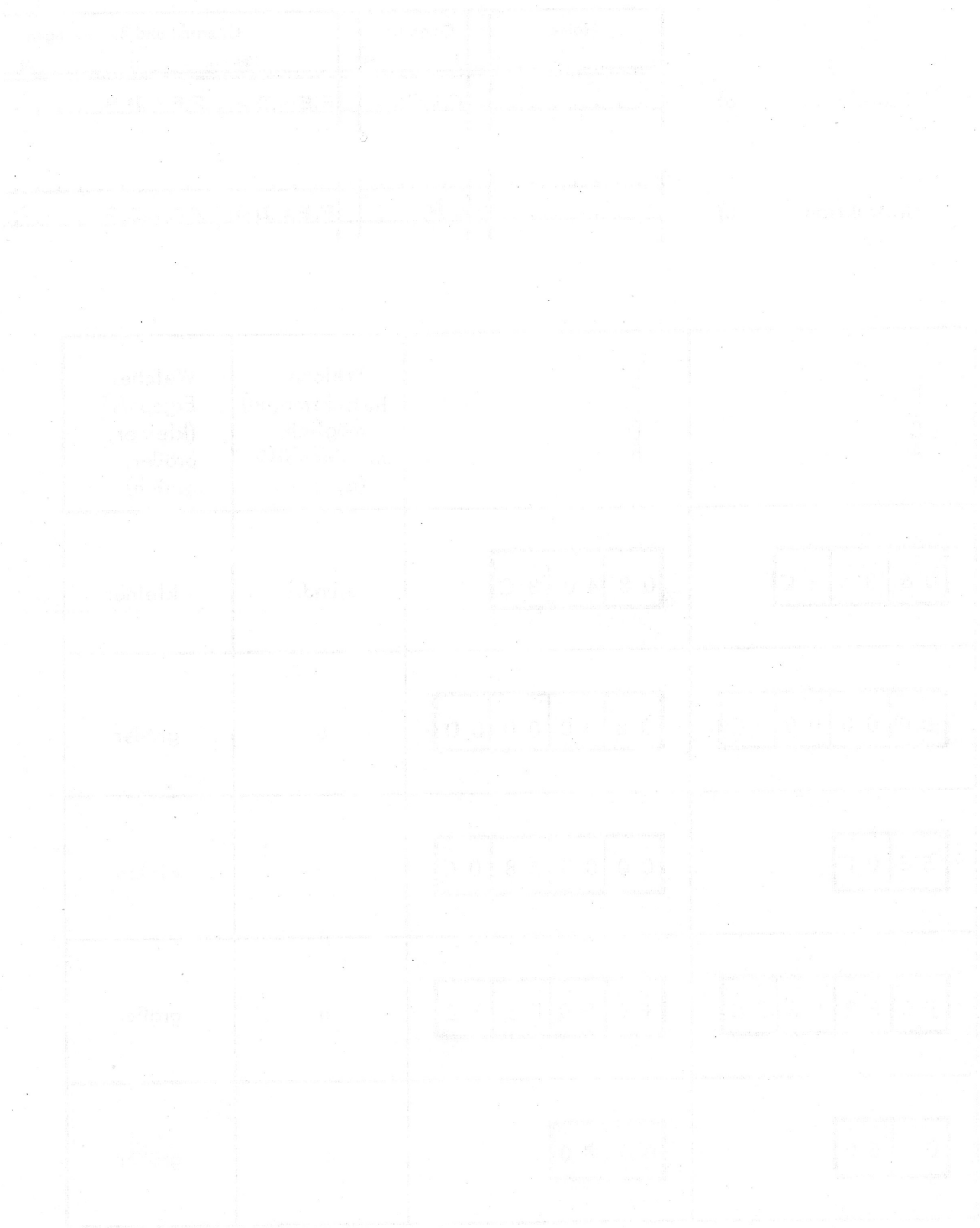
Prüfen Sie Ihre Eintragungen anhand der folgenden Lösungsblätter, sobald Sie eine Seite vollständig bearbeitet haben.

Lösungen zu den Übungen

1.



FELDA	FELDB	Welche Instruktion(en) möglich und sinnvoll? (a, b)	Welches Ergebnis? (kleiner, größer, gleich)
06351C	08403C	a und b	kleiner
0000005C	0800000D	b	größer
580F	00005800C	b	gleich
F0F3F8F0	F0F0F5F2	a	größer
0350	0080	a	größer



Merkblatt 418 enthält die Beschreibung der Operation "Vergleichen Dezimal (Gepackt)" (vgl. Abb. 1).

Band 1, Seite 153

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	:	
P,W,E,R,T,	D,S,	C,L,4,
P,K,O,N,1,	D,C,	P,',-,2,1,5,'
P,K,O,N,2,	D,C,	P,',8,'
R,E,F,E,	D,S,	C,L,7,
A,E,R,G,	D,S,	C,L,1,0,
A,R,E,S,T,	D,S,	C,L,1,
	:	

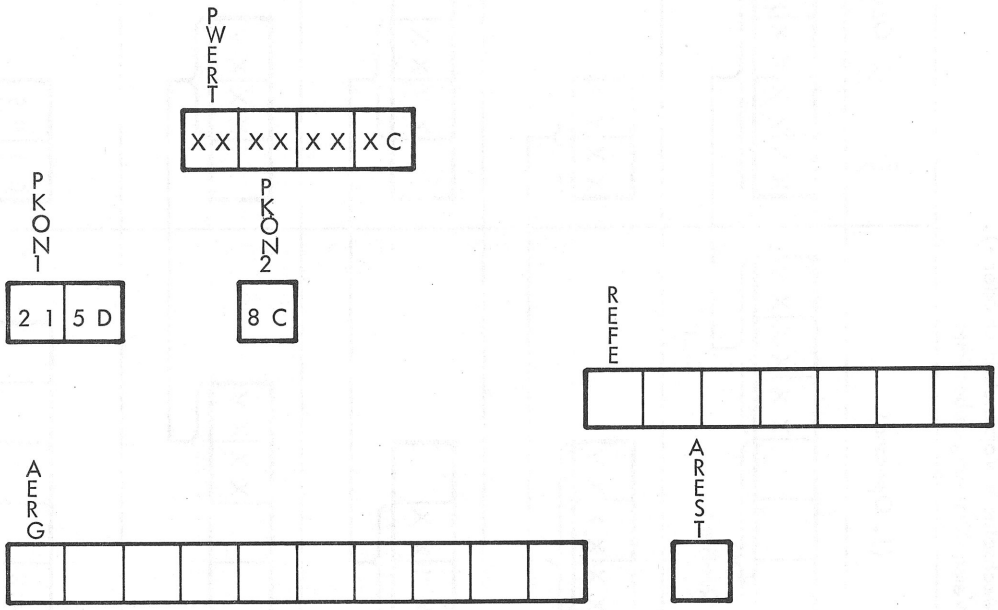


Abb. 1

FRAGEN ZUM KAPITEL IV

Führen Sie bitte die folgenden Übungen durch. Vergleichen Sie danach Ihre Lösungen mit den Eintragungen auf den anschließenden Lösungsblättern.

Die Punkte 1 bis 3 beziehen sich auf die Felder in Abb. 1.. Betrachten Sie zunächst die Abbildung.

1. Codieren Sie folgende Instruktionen und geben Sie jeweils den Inhalt des veränderten Feldes an ("0" für führende Null, "X" für wechselnde Ziffer des Ergebnisses, "C" bzw. "D" für Vorzeichen):

   a) für die Multiplikation der gepackten Zahl in PWERT mit der Konstanten "-215" (Rechenfeld: REFE)

   b) für die anschließende Division des Ergebnisses durch die Konstante "+8" (Rechenfeld: REFE)

   c) für das Entpacken des Quotienten und des Restes nach AERG und AREST.

2. Schreiben Sie die Instruktionen zum Vergleich

   a) von PWERT mit PKON1
   (Das Vergleichsergebnis soll sich auf PWERT beziehen.)

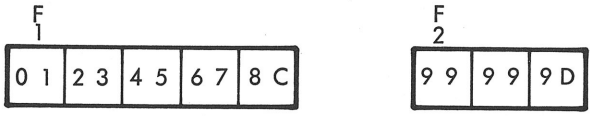
   b) von PWERT mit PKON2
   (Das Vergleichsergebnis soll sich auf PKON2 beziehen.)

3. PWERT soll die Zahl "+38" enthalten. Wie heißen die Vergleichsergebnisse

   a) vom Vergleich in 2 a,

   b) vom Vergleich in 2 b?

4. Die folgenden Befehle beziehen sich auf die Felder F1 und F2 mit dem eingetragenen Inhalt:



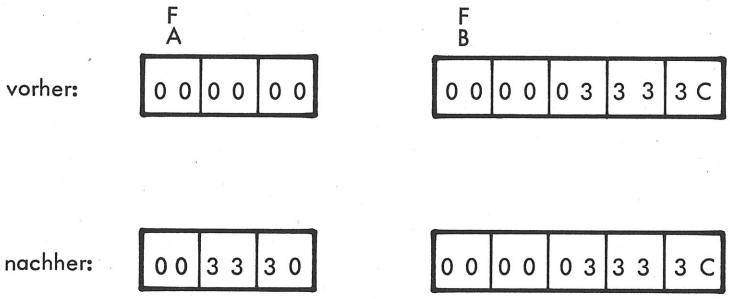
a) M.V.O. F.1.,F.2.

b) M.V.O. F.2.+1.(2.),F.1.(2.)

c) M.V.O. F.2.,F.1.(1.)

Geben Sie jeweils das durch die MVO-Instruktion veränderte Feld mit seinem neuen Inhalt an.

5. Codieren Sie die Instruktion für das folgende Beispiel:



6. Bei welchen der vier Operationen dieses Kapitels (MP, DP, MVO, CP) werden beide Operanden auf gepackten Inhalt geprüft ?

7. Wir haben einfache Regeln zur Berechnung der Länge von Multiplikations- und Divisionsfeldern aufgestellt.
Wie lang müssen nach diesen Regeln die Rechenfelder sein

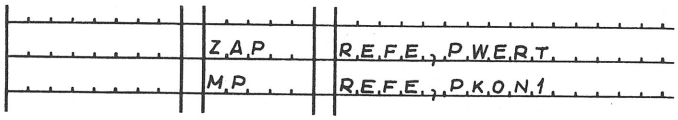
a) für die Multiplikation einer 4-stelligen und einer 7-stelligen Zahl,

b) für die Division einer 8-stelligen Zahl durch eine 5-stellige Zahl ?

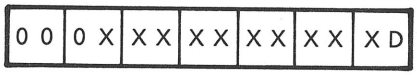
ANTWORTEN ZUM KAPITEL IV

Wenn Ihnen Fehler unterlaufen sind, dann suchen Sie die Fehlerursache. Wiederholen Sie gegebenenfalls den in Klammern angegebenen Lernstoff.

1. a)



REFE:

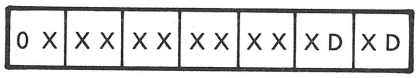


(Seite 252)

b)



REFE:

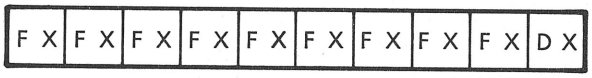


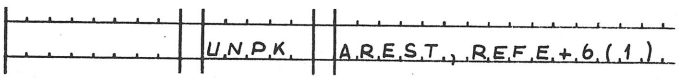
(Seite 278)

c)



AERG:



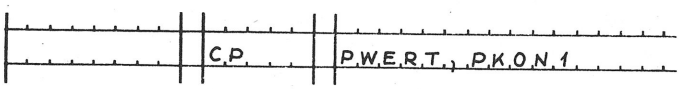


AREST:

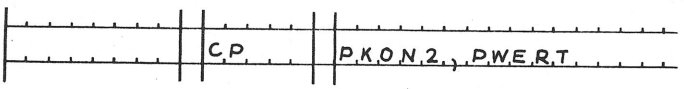


(Seite 279, LE 2 und 3)

2. a)



b)



(Seite 320 - 329)

3. a) PWERT ist größer als PKON1

b) PKON2 ist kleiner als PWERT

(Seite 320 - 329)

4. a)

F1

00	09	99	99	DC

b)

F2

99	12	3D

c)

F2

00	00	1D

(Seite 300 - 309)

5.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,O,	F,A,,F,B,(,4,)

(Seite 300 - 309)

6. Bei MP, DP und CP

(Seiten 252, 278, 300 und 320)

7. a) 7 Bytes (3 Bytes + 4 Bytes)

(Seite 255)

b) 8 Bytes (5 Bytes + 3 Bytes)

(Seite 291, LE 13 und 14)

Entnehmen Sie Ihren Arbeitsmitteln Merkblatt 419.

Dieses Merkblatt enthält eine Zusammenfassung der wesentlichen Eigenschaften der in diesem Kapitel besprochenen Instruktionen.

KAPITEL V

DIE OPERATION ED

Lernstoff:

ED - Aufbereiten zum Drucken

Die folgende Operation "Aufbereiten zum Drucken" ist eine der wirkungsvollsten im Instruktionsvorrat des Systems /360.
Mit Hilfe dieser Operation können wir bei numerischen Daten führende Nullen unterdrücken, Kommas und Dezimalpunkte einfügen, negative Beträge beliebig kennzeichnen, die Unterdrückung führender Nullen wieder aufheben, konstanten Text einfügen.

Wir werden zunächst die Eigenschaften der ED-Operation wie z. B. Operationslänge und -richtung besprechen. Diese Eigenschaften sind in Abb. 1 beschrieben und an einem Beispiel veranschaulicht. (Teil I)

Daran anschließend werden wir anhand von weiteren Beispielen die verschiedenen Möglichkeiten der Druckaufbereitung untersuchen und feststellen, welche Vorschriften dabei beachtet werden müssen. (Teil II)

ED

AUFBEREITEN ZUM DRUCKEN

(EDit)

Diese Operation bewirkt, daß eine gepackte Zahl mit Hilfe einer Schablone ein durch Punkt, Komma, Minuskennzeichnung usw. aufbereitetes Druckbild erhält.

Beispiel einer Instruktion:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
SCHAB	DC	C' ∇∇.∇[∇,∇∇-'
	MVC	ABETRAG,SCHAB
	ED	ABETRAG,PBETRAG

Wirkungsweise:

Diese Operation setzt voraus, daß vor ihrer Durchführung eine entsprechende Schablone (im Beispiel SCHAB) im Ergebnisfeld (im Beispiel ABETRAG) steht.

Die Schablone wird als Konstante definiert. Zur Übertragung dient die MVC-Instruktion.

Der Inhalt des zweiten Operanden (im Beispiel PBETRAG) wird durch ED vom gepackten in das ungepackte Format überführt und unter Kontrolle der Schablone im ersten Operanden aufbereitet.

Die Schablone im ersten Operanden wird nicht auf Gültigkeit geprüft. Sie kann beliebige Zeichen enthalten. Der zweite Operand muß gepacktes Format haben. Genauer gesagt: Jedes linke Halbbyte muß eine der Ziffern 0 bis 9 enthalten. In jedem rechten Halbbyte darf eine der Ziffern 0 bis 9 oder ein Vorzeichen stehen.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Der Bedingungsschlüssel wird neu gesetzt - abhängig vom Inhalt des zweiten Operanden - auf positiv, negativ oder null.

Ein Überlauf wird nicht angezeigt. Ist der erste Operand zu kurz, so wird der rechte Teil des zweiten Operanden nicht aufbereitet.

Der Inhalt des zweiten Operanden bleibt erhalten.

Der erste Operand darf maximal 256 Bytes lang sein.

Bei Gültigkeitsfehler tritt Programmunterbrechung ein.

Durchführung der Operation:

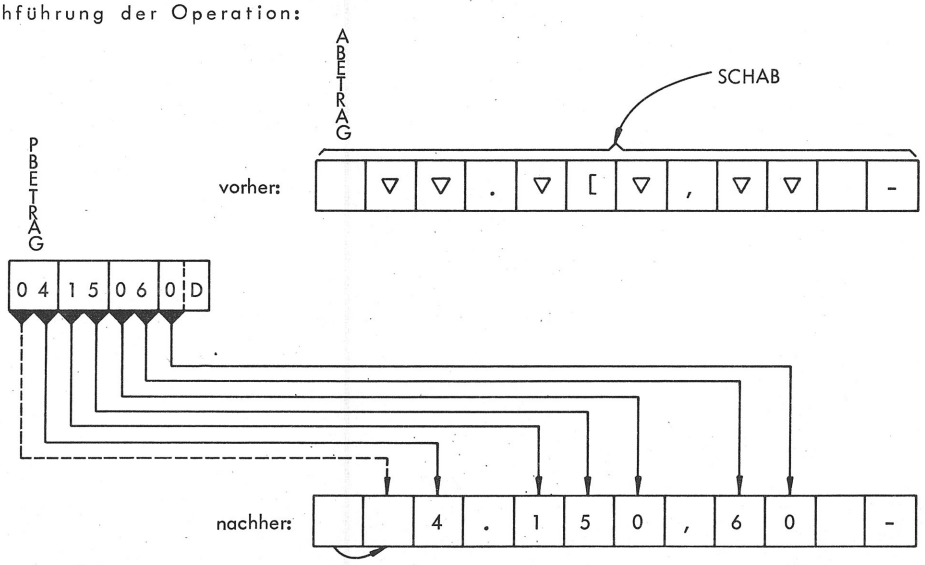


Abb. 1

Teil I

1 In Abb. 1 sind die Eigenschaften der Operation "Aufbereiten zum Drucken" beschrieben.

Vor der Ausführung der Operation enthält der erste Operand eine _____.

Er hat in unserem Beispiel eine Länge von __(ZW)__ und heißt _____.

Schablone

12 Bytes

ABETRAG

2 Der erste Buchstabe "A" deutet darauf hin, daß es sich bei ABETRAG um ein Ausgabefeld handelt.

Der betreffende Ausgabebereich ist ein (Stanzbereich, Druckbereich) _____.

Die 12 Bytes des Feldes ABETRAG werden in __(Z)__ Druckstellen ausgegeben.

Druckbereich

12

3 In unserem Beispiel wird der Inhalt des Feldes _____ aufbereitet.

Dieses Feld enthält __(Z)__ Ziffern und __(WW)__.

Die Speicherungsform ist _____.

PBETRAG

7

ein Vorzeichen

gepackt

4 Nur gepackte Zahlen können mit ED zum Drucken aufbereitet werden.

Bei der Ausführung der Operation werden in jedem Byte des zweiten Operanden die linken Halbbytes geprüft. Sie müssen eine der Dezimalziffern __(Z)__ bis __(Z)__ enthalten.

0 (bis) 9

ED

AUFBEREITEN ZUM DRUCKEN

(EDit)

Diese Operation bewirkt, daß eine gepackte Zahl mit Hilfe einer Schablone ein durch Punkt, Komma, Minuskennzeichnung usw. aufbereitetes Druckbild erhält.

Beispiel einer Instruktion:

Name (1 … 8)	Operation (10 … 14)	Operand und Bemerkungen (20 … 30)
SCHAB	DC	C' ∇∇.∇[∇,∇∇-'
	MVC	ABETRAG,SCHAB
	ED	ABETRAG,PBETRAG

Wirkungsweise:

Diese Operation setzt voraus, daß vor ihrer Durchführung eine entsprechende Schablone (im Beispiel SCHAB) im Ergebnisfeld (im Beispiel ABETRAG) steht.

Die Schablone wird als Konstante definiert. Zur Übertragung dient die MVC-Instruktion.

Der Inhalt des zweiten Operanden (im Beispiel PBETRAG) wird durch ED vom gepackten in das ungepackte Format überführt und unter Kontrolle der Schablone im ersten Operanden aufbereitet.

Die Schablone im ersten Operanden wird nicht auf Gültigkeit geprüft. Sie kann beliebige Zeichen enthalten. Der zweite Operand muß gepacktes Format haben. Genauer gesagt: Jedes linke Halbbyte muß eine der Ziffern 0 bis 9 enthalten. In jedem rechten Halbbyte darf eine der Ziffern 0 bis 9 oder ein Vorzeichen stehen.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Der Bedingungsschlüssel wird neu gesetzt - abhängig vom Inhalt des zweiten Operanden - auf positiv, negativ oder null.

Ein Überlauf wird nicht angezeigt. Ist der erste Operand zu kurz, so wird der rechte Teil des zweiten Operanden nicht aufbereitet.

Der Inhalt des zweiten Operanden bleibt erhalten.

Der erste Operand darf maximal 256 Bytes lang sein.

Bei Gültigkeitsfehler tritt Programmunterbrechung ein.

Durchführung der Operation:

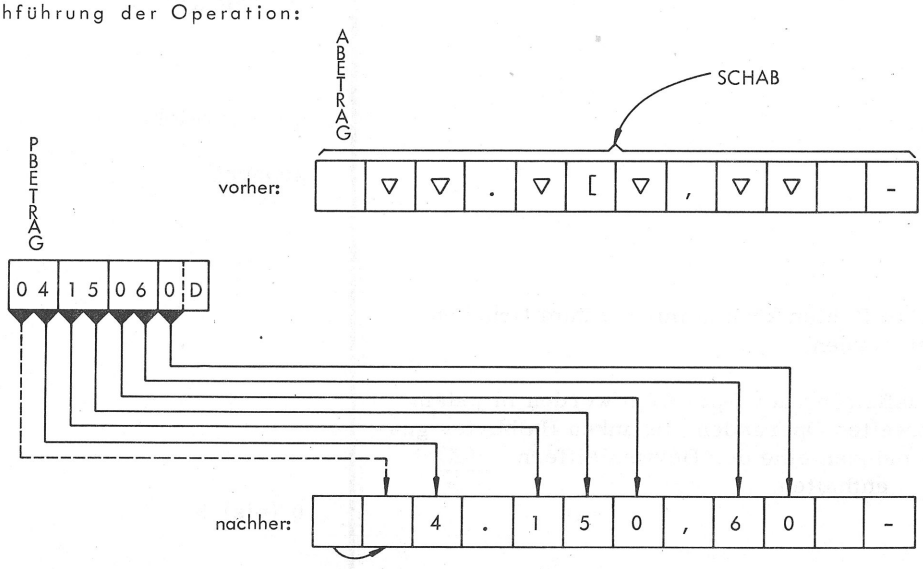


Abb. 1

5 Die rechten Halbbytes des aufzubereitenden Feldes dürfen beliebigen Inhalt haben. Sie dürfen eine der Bit-Kombinationen 0000 bis 1111 enthalten.

In ihnen ist also entweder eine der Ziffern 0 bis 9 oder ein _____ dargestellt.

Vorzeichen

(Anmerkung: Es ist erlaubt, daß der 2. Operand mehr als 1 Vorzeichen enthält. Es können also mehrere nebeneinanderstehende Felder durch eine einzige ED-Operation aufbereitet werden.)

Aus der gepackten Zahl im zweiten Operanden werden nur die Ziffern in die Schablone eingefüllt (soweit Platz vorhanden ist). Vorzeichen werden n i c h t eingefüllt.

Allerdings haben auch die Vorzeichen Einfluß auf das Aufbereiten (im Beispiel auf den Inhalt der rechten zwei Bytes im Druckbild). Bei negativem Vorzeichen bleibt der Inhalt dieser Bytes erhalten (im Beispiel "b-"). Bei positivem Vorzeichen wird der alte Inhalt jedes dieser Bytes durch den Inhalt des linken Bytes der Schablone (im Beispiel "b") ersetzt.

Wir halten fest:

Ins Ausgabefeld werden nur Ziffern und keine Vorzeichen eingefüllt.

6 Mit der ED-Operation in unserem Beispiel wird der Inhalt des 4 Bytes großen Feldes PBETRAG in ein 12 Bytes großes Feld eingefüllt.

Nach dem Aufbereiten stehen die Ziffern (gepackt, entpackt) _____ in ABETRAG.

Die Ziffern werden in rechte Halbbytes eingefüllt. Die entsprechenden linken Halbbytes erhalten die Zonenverschlüsselung __(Z)__.

entpackt

F/1111

ED

AUFBEREITEN ZUM DRUCKEN

(EDit)

Diese Operation bewirkt, daß eine gepackte Zahl mit Hilfe einer Schablone ein durch Punkt, Komma, Minuskennzeichnung usw. aufbereitetes Druckbild erhält.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
SCHAB	DC	C' ∇∇.∇[∇,∇∇-'
	MVC	ABETRAG,SCHAB
	ED	ABETRAG,PBETRAG

Wirkungsweise:

Diese Operation setzt voraus, daß vor ihrer Durchführung eine entsprechende Schablone (im Beispiel SCHAB) im Ergebnisfeld (im Beispiel ABETRAG) steht.

Die Schablone wird als Konstante definiert. Zur Übertragung dient die MVC-Instruktion.

Der Inhalt des zweiten Operanden (im Beispiel PBETRAG) wird durch ED vom gepackten in das ungepackte Format überführt und unter Kontrolle der Schablone im ersten Operanden aufbereitet.

Die Schablone im ersten Operanden wird nicht auf Gültigkeit geprüft. Sie kann beliebige Zeichen enthalten. Der zweite Operand muß gepacktes Format haben. Genauer gesagt: Jedes linke Halbbyte muß eine der Ziffern 0 bis 9 enthalten. In jedem rechten Halbbyte darf eine der Ziffern 0 bis 9 oder ein Vorzeichen stehen.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Der Bedingungsschlüssel wird neu gesetzt - abhängig vom Inhalt des zweiten Operanden - auf positiv, negativ oder null.

Ein Überlauf wird nicht angezeigt. Ist der erste Operand zu kurz, so wird der rechte Teil des zweiten Operanden nicht aufbereitet.

Der Inhalt des zweiten Operanden bleibt erhalten.

Der erste Operand darf maximal 256 Bytes lang sein.

Bei Gültigkeitsfehler tritt Programmunterbrechung ein.

Durchführung der Operation:

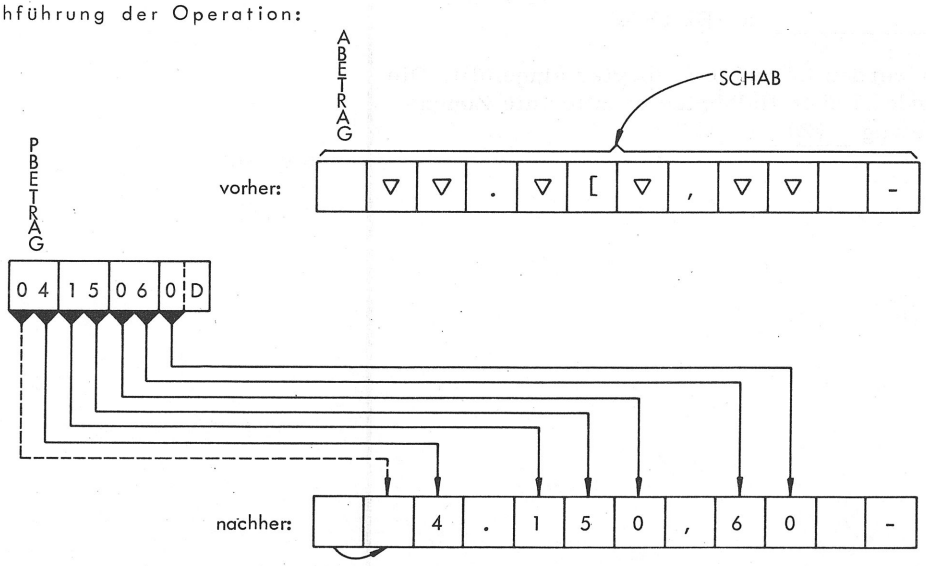


Abb. 1

7 Bei der Durchführung der ED-Operation bleiben nur einige Zeichen im Ausgabefeld erhalten. Im Beispiel in Abb. 1 sind es von links nach rechts die Zeichen:

ƀ . , ƀ -

Alle anderen Zeichen werden zerstört. Wir müssen daher vor jedem Aufbereiten die _____ erneut in das Ausgabefeld übertragen.

Schablone

8 Eine besondere Bedeutung haben die Zeichen " ∇ " und " [" in der Schablone. Sie dienen der Aufnahme der entpackten Ziffern aus PBETRAG.

Die Schablone in Abb. 1 enthält 7 solcher Zeichen. In ihr kann daher eine Zahl mit __(Z)__ Ziffern aufbereitet werden.

7

9 In unserem Beispiel hat die aufbereitete Zahl eine führende Null. Diese führende Null wird unterdrückt. An ihrer Stelle wird das sogenannte F ü l l z e i c h e n eingefüllt.

Das Füllzeichen steht immer im linken Byte der Schablone. In unserem Beispiel ist das Füllzeichen ein _____.

Blank

10 Betrachten Sie das folgende Beispiel einer Schablone:

*	∇	∇	∇	∇	.	∇	[	∇	-

Hier ist das Zeichen "*" das _____.

Bei einer solchen Schablone für das Beispiel in Abb. 1 würde statt der führenden Null ein _____ eingefüllt.

Füllzeichen

*

ED

AUFBEREITEN ZUM DRUCKEN

(EDit)

Diese Operation bewirkt, daß eine gepackte Zahl mit Hilfe einer Schablone ein durch Punkt, Komma, Minuskennzeichnung usw. aufbereitetes Druckbild erhält.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
SCHAB	DC	C' ∇∇.∇[∇,∇∇.-'
	MVC	ABETRAG,SCHAB
	ED	ABETRAG,PBETRAG

Wirkungsweise:

Diese Operation setzt voraus, daß vor ihrer Durchführung eine entsprechende Schablone (im Beispiel SCHAB) im Ergebnisfeld (im Beispiel ABETRAG) steht.

Die Schablone wird als Konstante definiert. Zur Übertragung dient die MVC-Instruktion.

Der Inhalt des zweiten Operanden (im Beispiel PBETRAG) wird durch ED vom gepackten in das ungepackte Format überführt und unter Kontrolle der Schablone im ersten Operanden aufbereitet.

Die Schablone im ersten Operanden wird nicht auf Gültigkeit geprüft. Sie kann beliebige Zeichen enthalten. Der zweite Operand muß gepacktes Format haben. Genauer gesagt: Jedes linke Halbbyte muß eine der Ziffern 0 bis 9 enthalten. In jedem rechten Halbbyte darf eine der Ziffern 0 bis 9 oder ein Vorzeichen stehen.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Der Bedingungsschlüssel wird neu gesetzt - abhängig vom Inhalt des zweiten Operanden - auf positiv, negativ oder null.

Ein Überlauf wird nicht angezeigt. Ist der erste Operand zu kurz, so wird der rechte Teil des zweiten Operanden nicht aufbereitet.

Der Inhalt des zweiten Operanden bleibt erhalten.

Der erste Operand darf maximal 256 Bytes lang sein.

Bei Gültigkeitsfehler tritt Programmunterbrechung ein.

Durchführung der Operation:

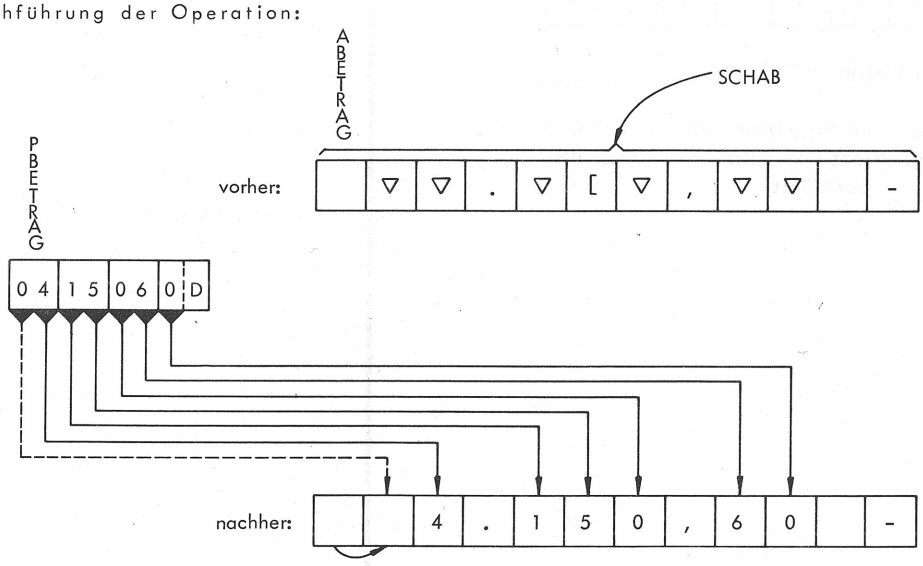


Abb. 1

11 Sie kennen eine Reihe von Operationen (PACK, UNPK, AP, SP usw.), die mit gepackten Daten zu tun haben. Bei diesen Operationen geht die Verarbeitungsrichtung von rechts nach links.

Auch die ED-Operation verarbeitet gepackte Daten. Ihre Verarbeitungsrichtung ist (genauso, entgegengesetzt) _____.

Die Ziffern aus PBETRAG in Abb. 1 werden den Schablone-Zeichen " ∇ " und " [" in der Reihenfolge __(ZZ)__ zugeordnet.

entgegengesetzt

0, 4, 1, 5, 0, 6, 0

12 Bezüglich ihrer Verarbeitungsrichtung gehört die ED-Operation in die gleiche Gruppe wie die Übertragungsoperationen und die logischen Vergleichsoperationen (MVC, CLC usw.).

Auch in bezug auf die Operations-Länge gehört ED zu dieser Gruppe. Die Länge der Verarbeitung richtet sich nach der Länge ___(WW)___.

des ersten Operanden
(im Beispiel ABETRAG)

13 Inwiefern ist die Form des folgenden Instruktionsbeispiels fehlerhaft?

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
. . . . .	E,D. . .	A,F,E,L,D,(,6,),,P,W,E,R,T,+,1,(,3,),

Sinngemäß:

Der 2. Operand darf keine explizite Längenangabe haben.

14 Nach Abb. 1 bewirkt die ED-Operation außerdem etwas, das wir bisher außer acht gelassen haben.

Bei der Ausführung der Operation wird der _____ neu eingestellt.

Im Beispiel in Abb. 1 wird er auf _____ gesetzt.

Bedingungsschlüssel

negativ

ED

AUFBEREITEN ZUM DRUCKEN

(EDit)

Diese Operation bewirkt, daß eine gepackte Zahl mit Hilfe einer Schablone ein durch Punkt, Komma, Minuskennzeichnung usw. aufbereitetes Druckbild erhält.

Beispiel einer Instruktion:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
S,C,H,A,B,	D,C,	C,',▽,▽,.,▽,I,▽,,▽,▽,-,'
	M,V,C,	A,B,E,T,R,A,G,,S,C,H,A,B,
	E,D,	A,B,E,T,R,A,G,,P,B,E,T,R,A,G,

Wirkungsweise:

Diese Operation setzt voraus, daß vor ihrer Durchführung eine entsprechende Schablone (im Beispiel SCHAB) im Ergebnisfeld (im Beispiel ABETRAG) steht.

Die Schablone wird als Konstante definiert. Zur Übertragung dient die MVC-Instruktion.

Der Inhalt des zweiten Operanden (im Beispiel PBETRAG) wird durch ED vom gepackten in das ungepackte Format überführt und unter Kontrolle der Schablone im ersten Operanden aufbereitet.

Die Schablone im ersten Operanden wird nicht auf Gültigkeit geprüft. Sie kann beliebige Zeichen enthalten. Der zweite Operand muß gepacktes Format haben. Genauer gesagt: Jedes linke Halbbyte muß eine der Ziffern 0 bis 9 enthalten. In jedem rechten Halbbyte darf eine der Ziffern 0 bis 9 oder ein Vorzeichen stehen.

Verarbeitet wird byteweise von links nach rechts in der Länge des ersten Operanden.

Der Bedingungsschlüssel wird neu gesetzt - abhängig vom Inhalt des zweiten Operanden - auf positiv, negativ oder null.

Ein Überlauf wird nicht angezeigt. Ist der erste Operand zu kurz, so wird der rechte Teil des zweiten Operanden nicht aufbereitet.

Der Inhalt des zweiten Operanden bleibt erhalten.

Der erste Operand darf maximal 256 Bytes lang sein.

Bei Gültigkeitsfehler tritt Programmunterbrechung ein.

Durchführung der Operation:

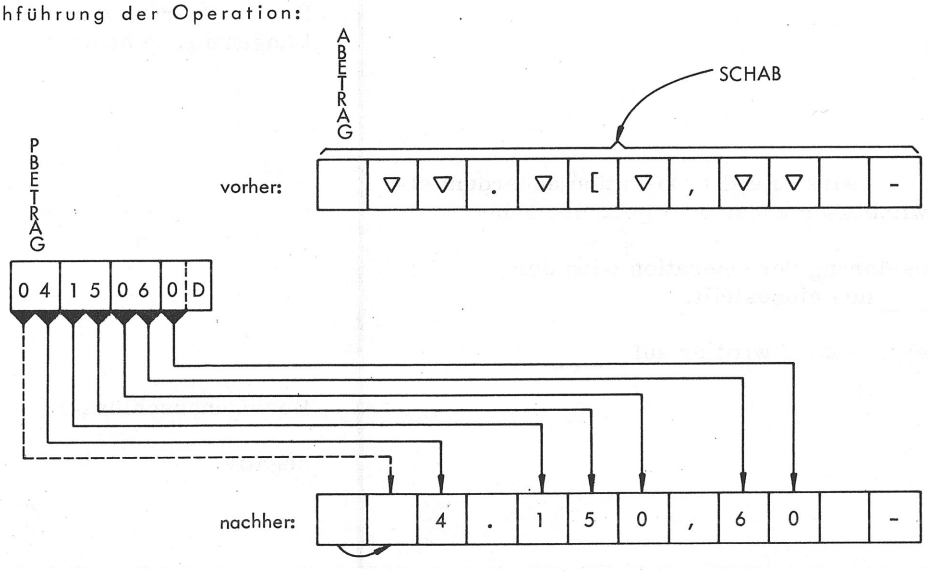


Abb. 1

Merkblatt 420, Seite 1 enthält die Beschreibung der Operation "Aufbereiten zum Drucken" (vgl. Abb. 1).

Wir wenden uns jetzt dem 2. Teil der Behandlung von ED zu.

Teil II

In diesem Teil werden wir uns vorwiegend mit dem Aufbau der Schablone beschäftigen.
Von ihrer Form hängt es ab, wie eine bestimmte Dezimalzahl durch Punkte, Zwischenräume, eingeschobene Bezeichnungen usw. gegliedert wird, wie Minusbeträge gekennzeichnet und wann führende Nullen unterdrückt werden.

Anmerkung:
Falls Sie den RPG kennen, beachten Sie bitte folgenden Hinweis:

Im Assembler sind die Regeln über die Schablone a n d e r s als im RPG.

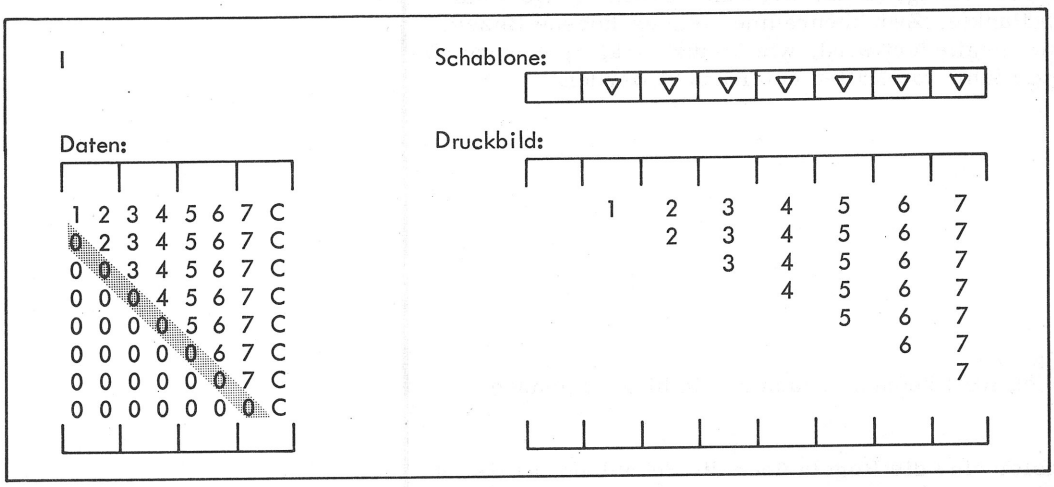

I
Schablone:
Daten:
1 2 3 4 5 6 7 C
0 2 3 4 5 6 7 C
0 0 3 4 5 6 7 C
0 0 0 4 5 6 7 C
0 0 0 0 5 6 7 C
0 0 0 0 0 6 7 C
0 0 0 0 0 0 7 C
0 0 0 0 0 0 0 C
Druckbild:
1 2 3 4 5 6 7
2 3 4 5 6 7
3 4 5 6 7
4 5 6 7
5 6 7
6 7
7

Abb. 2

15 Betrachten Sie Abb. 2.

Es soll der Inhalt von 4 Bytes im Speicher zum Drukken aufbereitet werden.

Die 4 Bytes enthalten __(Z)__ Ziffern.

Die Darstellungsform ist _____.

Die wechselnden Zahlen in diesen 4 Bytes haben verschieden viele __(WW)__.

7

gepackt

führende Nullen

16 Sehen Sie sich das Druckbild in Abb. 2 an.

Für dieses Druckbild muß das Ausgabefeld eine Länge von __(ZW)__ haben.

Vor der ED-Operation muß das Ausgabefeld __(WW)__ enthalten.

Nach der ED-Operation haben die Daten im Ausgabefeld _____ Speicherungsform.

8 Bytes

die Schablone

ungepackte/entpackte

17 Die Schablone wird als Konstante definiert.

Sie hat eine Länge von __(ZW)__.

Sie enthält zwei verschiedene Zeichen, nämlich ein _____ und 7 Zeichen _____.

8 Bytes

Blank

▽

Name	Symbol	Ver-schlüsselung	Lochkarten-Code
Ziffern-Auswahl-Zeichen	▽	0010 0000	11,0,9,8,1

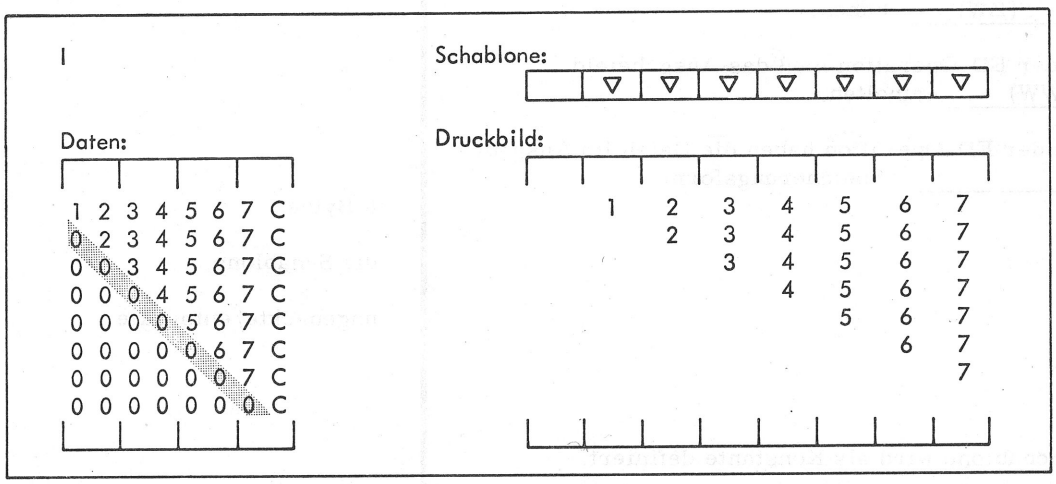


Abb. 3

18 Betrachten Sie die kleine Tabelle oben in Abb. 3.

Das Zeichen " ▽ " heißt "_____".

Im Speicher hat es die Verschlüsselung _(ZZ)_ in einem Byte.

Diese Verschlüsselung wird durch die Lochungen _(ZZ)_ erzeugt.

Ziffern-Auswahl-Zeichen

0010 0000

11, 0, 9, 8 und 1

19 Wie der Name sagt, tritt an die Stelle eines Ziffern-Auswahl-Zeichens im Ausgabefeld eine _____ aus dem Datenfeld.

Ausnahme (vgl. Abb. 3):
Die Ziffer kommt nicht ins Ausgabefeld, wenn sie eine _(WW)_ ist.

Ziffer

führende Null

20 Das Ziffern-Auswahl-Zeichen dient der Aufnahme von Ziffern aus dem Datenfeld. Führende Nullen werden jedoch unterdrückt.

An ihre Stelle tritt das _____ aus dem linken Byte der Schablone.

Füllzeichen

21 Im Beispiel der Abb. 3 ist das Füllzeichen ein _____.

Bei einem Inhalt null des Datenfeldes ersetzt dieses Füllzeichen _(Z)_ Nullen.

Im Druckbild erscheinen daher _(Z)_ Leerstellen.

Blank

7

8

Name	Symbol	Ver-schlüsselung	Lochkarten-Code
Ziffern-Auswahl-Zeichen	▽	0010 0000	11,0,9,8,1

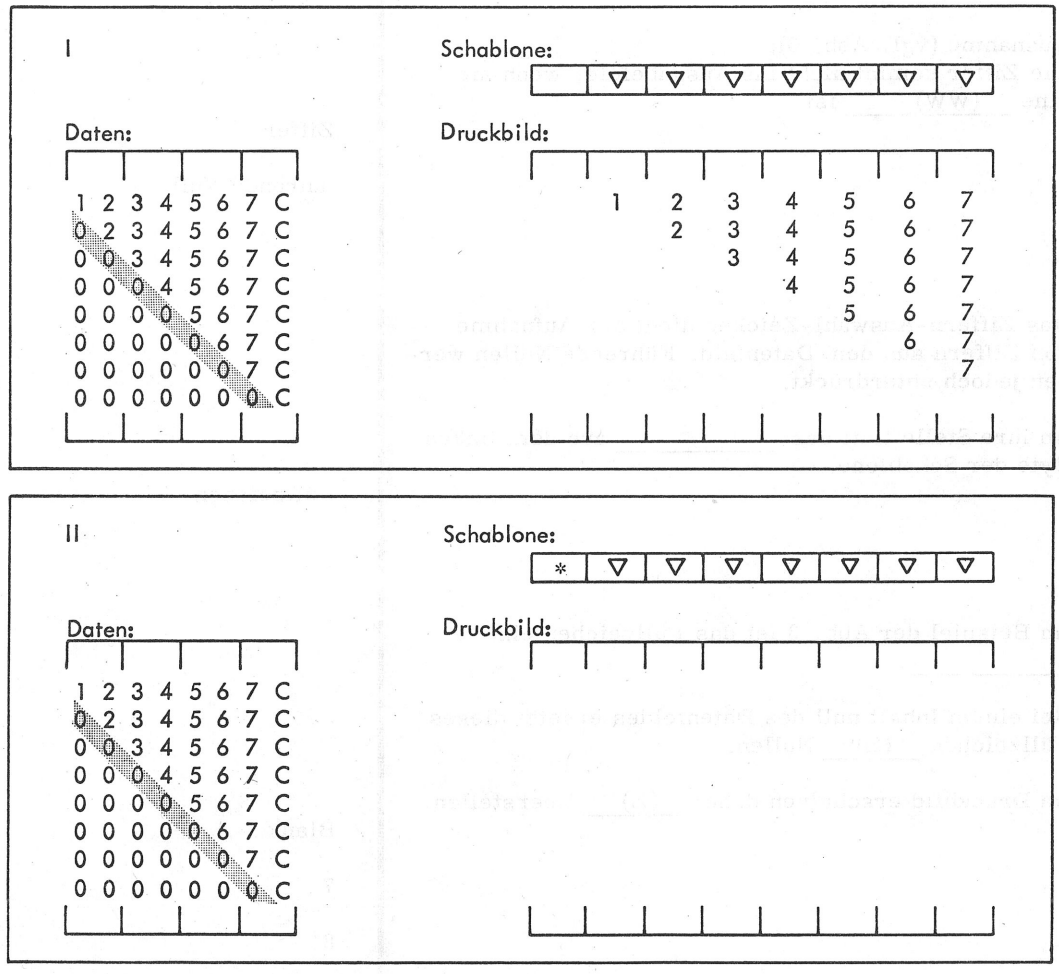


Abb. 4

22 Als Füllzeichen kann jedes beliebige Zeichen verwendet werden. Im allgemeinen wird das Blank gewählt.

Im Beispiel II in Abb. 4 ist das Füllzeichen ein "*".

Entnehmen Sie Ihren Arbeitsmitteln Übung 421.

Tragen Sie ein, wie die Schablone die wechselnden Zahlen des Datenfeldes aufbereitet. Prüfen Sie Ihre Lösung an Abb. 5.

Name	Symbol	Ver-schlüsselung	Lochkarten-Code
Ziffern-Auswahl-Zeichen	▽	0010 0000	11,0,9,8,1

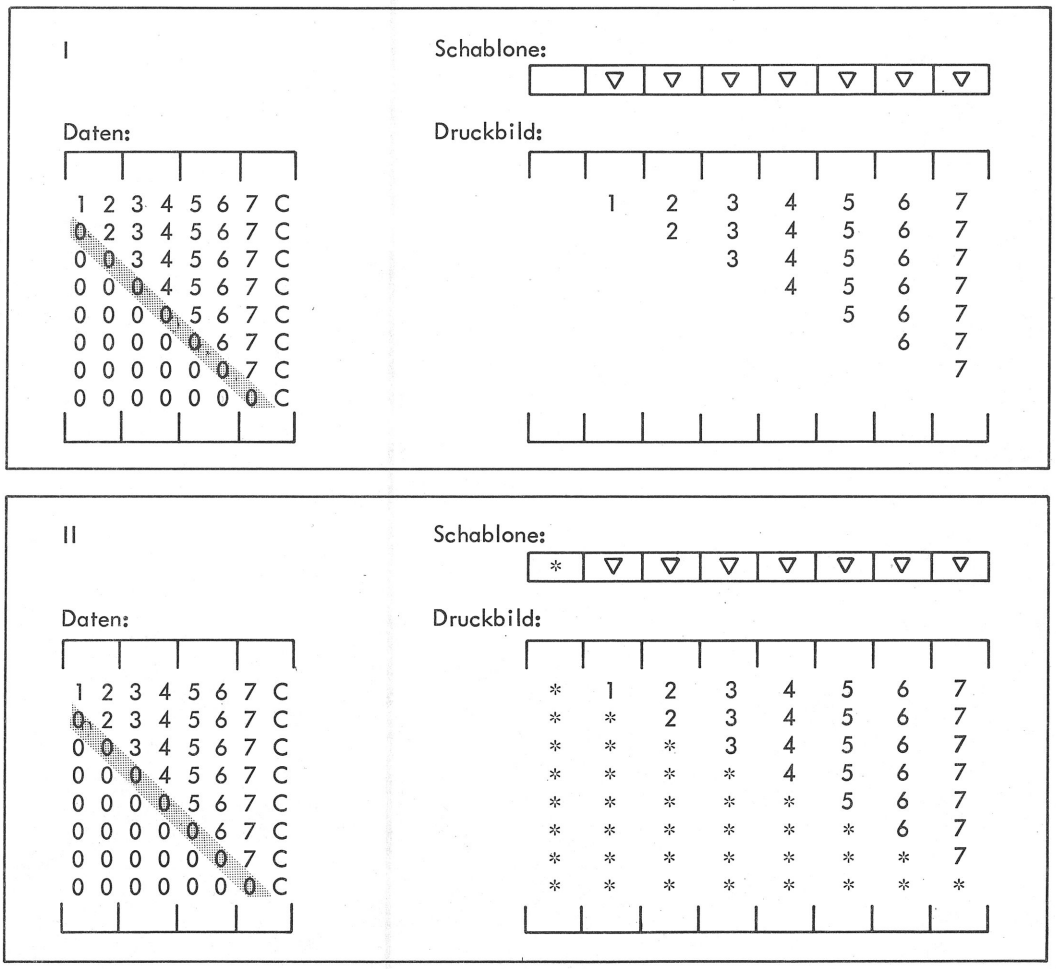


Abb. 5

Wir sehen:
Enthält die Schablone als Füllzeichen einen Stern und sonst nur Ziffern-Auswahl-Zeichen, so werden bei der Aufbereitung die führenden Nullen unterdrückt und durch Sterne ersetzt.

Eine solche Schreibweise nennt man "Schutz-Stern-Schreibung". Sie verhindert späteres unberechtigtes Einfügen von Ziffern anstelle der unterdrückten führenden Nullen.

23 Betrachten wir die Länge der Schablone, so stellen wir fest:

Wegen des Füllzeichens muß die Schablone um eine Stelle _____ sein, als von den Daten her erforderlich ist.

In die Stelle des Füllzeichens werden _____ Daten eingegeben.

länger

keine

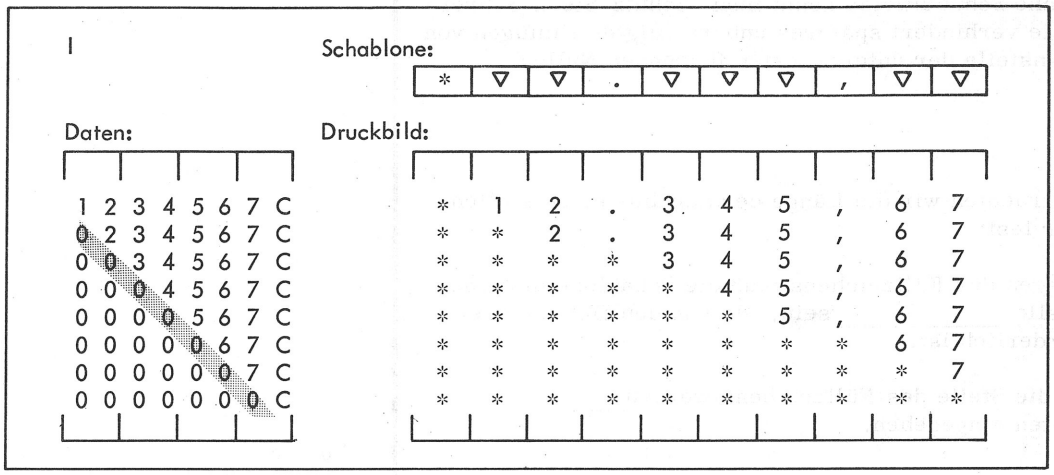

I
Schablone:
* ▽ ▽ . ▽ ▽ ▽ , ▽ ▽
Daten:
1 2 3 4 5 6 7 C
0 2 3 4 5 6 7 C
0 0 3 4 5 6 7 C
0 0 0 4 5 6 7 C
0 0 0 0 5 6 7 C
0 0 0 0 0 6 7 C
0 0 0 0 0 0 7 C
0 0 0 0 0 0 0 C
Druckbild:
* 1 2 . 3 4 5 , 6 7
* * 2 . 3 4 5 , 6 7
* * * * 3 4 5 , 6 7
* * * * * 4 5 , 6 7
* * * * * * 5 , 6 7
* * * * * * * * 6 7
* * * * * * * * * 7
* * * * * * * * * *

Abb. 6

24 Wir kommen jetzt zu dem Beispiel in Abb. 6.

Die hier gezeigte Schablone dient zum Aufbereiten von Geldbeträgen.

Zwischen Tausendern und Hundertern erscheint ein _____.

Die Dezimalstellen werden durch ein _____ abgetrennt.

Der Stern wird als _____ verwendet.

Punkt

Komma

Füllzeichen

25 Durch die beiden Sonderzeichen " · " und " , " wird die Schablone um __(Z)__ Bytes länger.

Zum Aufbereiten von 7 Ziffern hat die Schablone jetzt eine Länge von __(ZW)__.

2

10 Bytes

26 Die eingeschobenen Sonderzeichen Punkt und Komma bleiben beim Aufbereiten der Daten nicht immer erhalten.

Teilweise werden sie durch das jeweilige _____ ersetzt.

Werden links von dem eingefügten Sonderzeichen eine oder mehrere Ziffern aus dem Datenfeld eingefüllt, so bleibt das Sonderzeichen für das Druckbild (erhalten, nicht erhalten) _____.

Füllzeichen

erhalten

Wie Punkt und Komma können auch Bezeichnungen wie DM und KG oder andere alphabetische, numerische oder Sonder-Zeichen in die Schablone eingefügt werden.

Diese eingeschobenen Zeichen werden nicht durch Ziffern aus dem Datenfeld ersetzt.
Bei der Durchführung der ED-Operation bleiben sie im allgemeinen erhalten und erscheinen im Druckbild.

Werden aber links von solchen Zeichen nur Füllzeichen eingefüllt, so bleiben sie nicht erhalten, sondern werden auch durch Füllzeichen ersetzt.

Name	Symbol	Ver-schlüsselung	Lochkarten-Code
Ziffern-Auswahl-Zeichen	▽	0010 0000	11,0,9,8,1
Start-Zeichen für geltende Ziffern	[	0010 0001	0,9,1

I

Schablone:

	▽	▽	.	▽	[	▽	,	▽	▽

Daten:

1	2	3	4	5	6	7	C
0	2	3	4	5	6	7	C
0	0	3	4	5	6	7	C
0	0	0	4	5	6	7	C
0	0	0	0	5	6	7	C
0	0	0	0	0	6	7	C
0	0	0	0	0	0	7	C
0	0	0	0	0	0	0	C

Druckbild:

	1	2	.	3	4	5	,	6	7
		2	.	3	4	5	,	6	7
				3	4	5	,	6	7
					4	5	,	6	7
						5	,	6	7
						0	,	6	7
						0	,	0	7
						0	,	0	0

II

Schablone:

	▽	▽	.	▽	▽	[	,	▽	▽

Daten:

1	2	3	4	5	6	7	C
0	2	3	4	5	6	7	C
0	0	3	4	5	6	7	C
0	0	0	4	5	6	7	C
0	0	0	0	5	6	7	C
0	0	0	0	0	6	7	C
0	0	0	0	0	0	7	C
0	0	0	0	0	0	0	C

Druckbild:

Abb. 7

27 In Abb. 7 wird ein weiteres Zeichen für die Schablone eingeführt.

Es ist das "_____".

Start-Zeichen für geltende Ziffern

28 Das Zeichen "[" entspricht der Lochkombination __(ZZ)__ in einer Spalte.

Diese Lochkombination ruft im Speicher die Verschlüsselung _(ZZ)_ in einem Byte hervor.

0, 9 und 1

0010 0001

25

29 Beispiel I in Abb. 7 läßt die Bedeutung des Zeichens "[" in einer Schablone erkennen.

In die entsprechende Stelle wird eine Ziffer aus dem Datenfeld (nicht eingegeben, eingegeben) _____.

eingegeben

30 Zwischen den beiden Zeichen ▽ und [besteht aber ein Unterschied.

Welches Druckbild erhält die Zahl "+7"

a) bei der Schablone im Beispiel I in Abb. 7,

b) bei der gleichen Schablone, wenn das Zeichen "[" durch ein " ▽ " ersetzt wird?

a) 0,07

b) 7

31 Steht das Zeichen "[" in einer Schablone, so wird von einer bestimmten Stelle an kein Füllzeichen mehr eingefüllt, weder für führende Nullen noch für eingeschobene Sonderzeichen oder Bezeichnungen.

Aus dem Beispiel I in Abb. 7 geht hervor:

Das Füllzeichen wird nicht mehr eingesteuert ab __(WW)__.

Sinngemäß:

der Stelle rechts vom "["

Name	Symbol	Ver-schlüsselung	Lochkarten-Code
Ziffern-Auswahl-Zeichen	▽	0010 0000	11,0,9,8,1
Start-Zeichen für geltende Ziffern	[	0010 0001	0,9,1

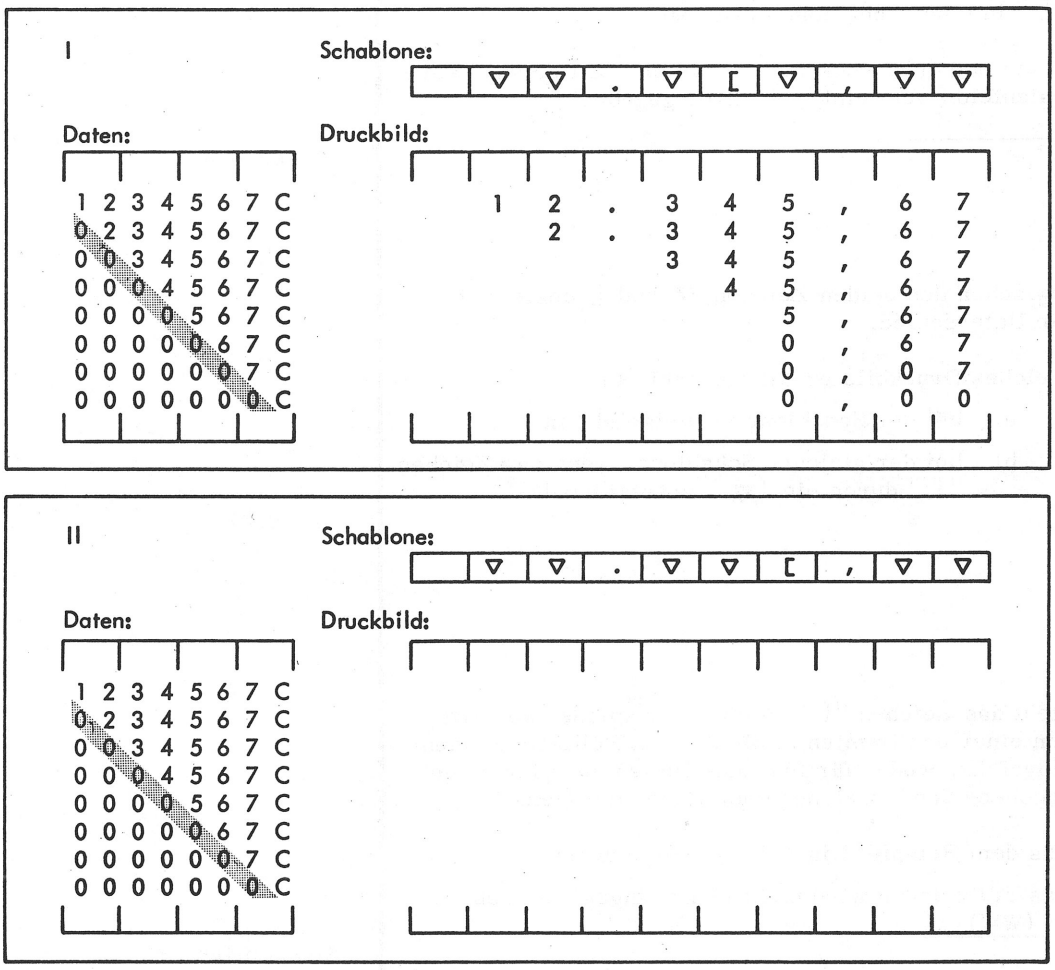


Abb. 7

32 Im Beispiel II in Abb. 7 steht das Zeichen "[" eine Stelle weiter rechts.

Entnehmen Sie Ihren Arbeitsmitteln Übung 422.

Tragen Sie den wechselnden Inhalt des Ausgabefeldes im Beispiel II ein.
Die Lösung finden Sie in Abb. 8.

Name	Symbol	Ver-schlüsselung	Lochkarten-Code
Ziffern-Auswahl-Zeichen	▽	0010 0000	11,0,9,8,1
Start-Zeichen für geltende Ziffern	[	0010 0001	0,9,1

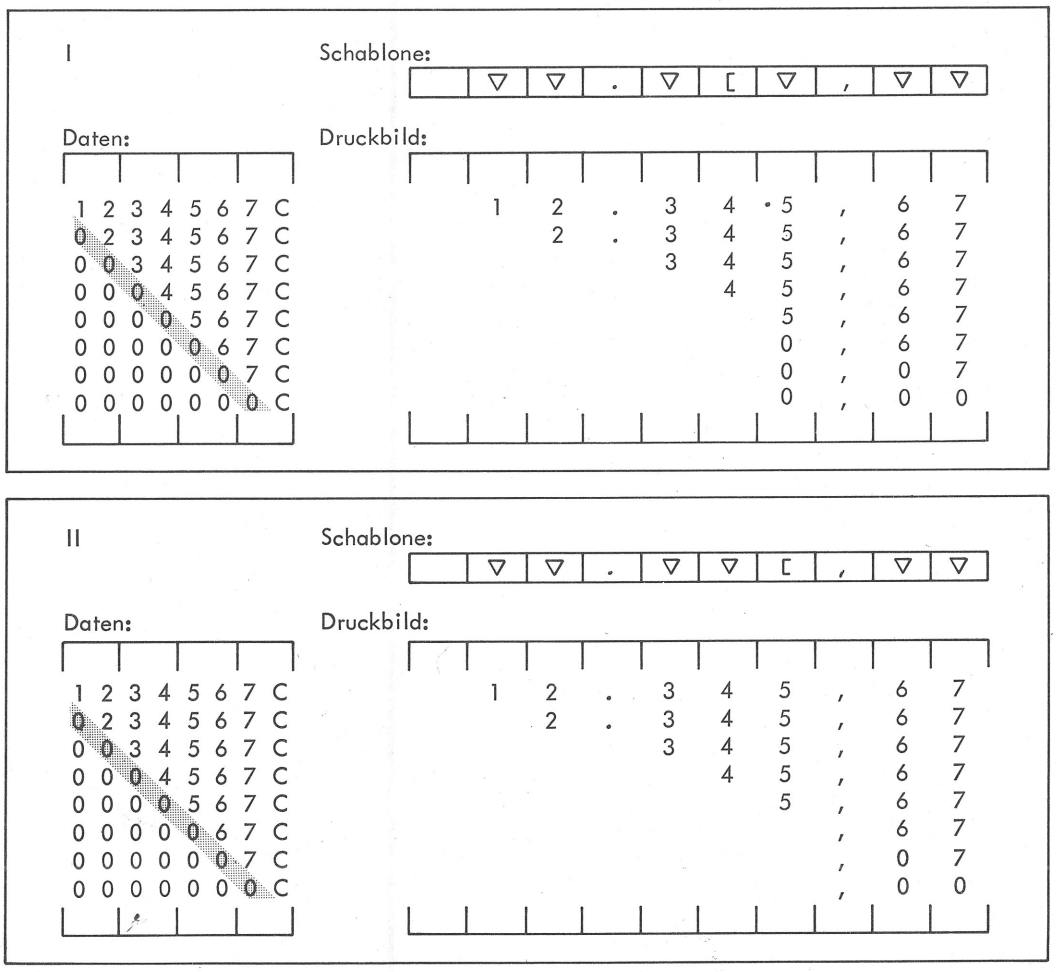


Abb. 8

33 Zum Aufbereiten von Geldbeträgen wird meist die Form im Beispiel I verwendet.

Wir fassen kurz zusammen:

In der Schablone können die verschiedensten Zeichen auftreten. Besondere Bedeutung hat das Zeichen in der linken Stelle. Es wird als _____ verwendet.

Das Füllzeichen wird statt führender Nullen oder eingeschobener Schablonenzeichen eingesteuert bis zu der Stelle einschließlich, die das Zeichen _____ enthält.

Das Einfüllen der Daten in die Schablone erfolgt von _____ nach _____.

Ziffern aus dem Datenfeld werden in solche Stellen entpackt, die entweder ein ____(WW)____ oder ein ____(WW)____ enthalten.

Füllzeichen

[

links (nach) rechts

Ziffern-Auswahl-Zeichen

Start-Zeichen für geltende Ziffern

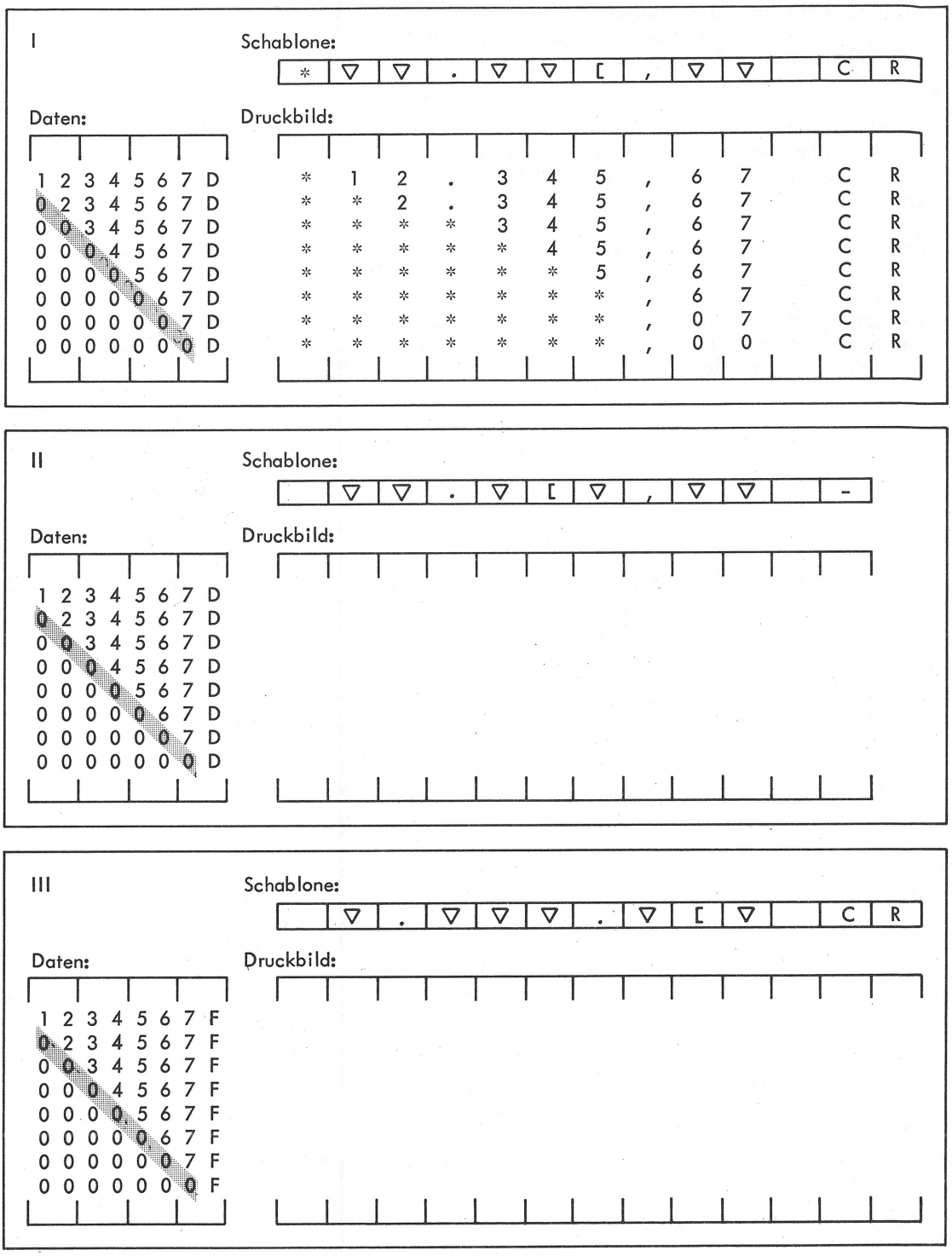

I
Schablone:
* ▽ ▽ . ▽ ▽ [, ▽ ▽ C R
Daten:
Druckbild:
1 2 3 4 5 6 7 D
0 2 3 4 5 6 7 D
0 0 3 4 5 6 7 D
0 0 0 4 5 6 7 D
0 0 0 0 5 6 7 D
0 0 0 0 0 6 7 D
0 0 0 0 0 0 7 D
0 0 0 0 0 0 0 D
* 1 2 . 3 4 5 , 6 7 C R
* * 2 . 3 4 5 , 6 7 C R
* * * * 3 4 5 , 6 7 C R
* * * * * 4 5 , 6 7 C R
* * * * * * 5 , 6 7 C R
* * * * * * * , 6 7 C R
* * * * * * * , 0 7 C R
* * * * * * * , 0 0 C R
II
Schablone:
▽ ▽ . ▽ [▽ , ▽ ▽ -
Daten:
Druckbild:
1 2 3 4 5 6 7 D
0 2 3 4 5 6 7 D
0 0 3 4 5 6 7 D
0 0 0 4 5 6 7 D
0 0 0 0 5 6 7 D
0 0 0 0 0 6 7 D
0 0 0 0 0 0 7 D
0 0 0 0 0 0 0 D
III
Schablone:
▽ . ▽ ▽ ▽ . ▽ [▽ C R
Daten:
Druckbild:
1 2 3 4 5 6 7 F
0 2 3 4 5 6 7 F
0 0 3 4 5 6 7 F
0 0 0 4 5 6 7 F
0 0 0 0 5 6 7 F
0 0 0 0 0 6 7 F
0 0 0 0 0 0 7 F
0 0 0 0 0 0 0 F

Abb. 9

34 In unseren bisherigen Beispielen hatte das Datenfeld immer positiven Inhalt.

Für den Fall, daß der Inhalt auch negativ werden kann und diese Tatsache dann im Druckbild deutlich werden soll, müssen wir in der betreffenden Schablone eine besondere Kennzeichnung vorsehen.

Wegen einer solchen Kennzeichnung wird die Schablone im Beispiel I in Abb. 9 um __(Z)__ Bytes länger.

Diese Bytes enthalten _____.

3

ƀ, C und R

35 "CR" ist die Abkürzung des Wortes "CREDIT". Diese Abkürzung wird häufig benutzt zur Kennzeichnung von Minus-Beträgen.

Statt der Zeichen "ƀCR" könnte der Schablone ebensogut irgendeine andere Bezeichnung, z.B. die Worte "KONTO UEBERZOGEN" angefügt werden oder, wie im Beispiel II, die Zeichen _____.

ƀ -

36 Wir müssen uns merken:

Alle Zeichen rechts vom letzten " ▽ " oder " [" werden beim Aufbereiten in gleicher Weise behandelt. Sie werden als Minus-Kennzeichnung aufgefaßt.

Steht im rechten Halbbyte des Datenfeldes ein positives Vorzeichen, so werden die rechts angefügten Zeichen in der Schablone durch Füllzeichen ersetzt. Die angefügten Zeichen bleiben nur dann erhalten, wenn die Zahl im Datenfeld kein _____ Vorzeichen hat.

positives

37 Überlegen Sie bitte, wie das Druckbild der verschiedenen Zahlen in den Beispielen II und III aussieht.

Benutzen Sie für Ihre Eintragungen Übung 423.

Die Lösungen finden Sie als Abb. 10 auf der folgenden Seite.

I

Schablone:

*	▽	▽	.	▽	▽	[	,	▽	▽		C	R

Daten:	Druckbild:
1 2 3 4 5 6 7 D	* 1 2 . 3 4 5 , 6 7 C R
0 2 3 4 5 6 7 D	* * 2 . 3 4 5 , 6 7 C R
0 0 3 4 5 6 7 D	* * * * 3 4 5 , 6 7 C R
0 0 0 4 5 6 7 D	* * * * * 4 5 , 6 7 C R
0 0 0 0 5 6 7 D	* * * * * * 5 , 6 7 C R
0 0 0 0 0 6 7 D	* * * * * * * , 6 7 C R
0 0 0 0 0 0 7 D	* * * * * * * , 0 7 C R
0 0 0 0 0 0 0 D	* * * * * * * , 0 0 C R

II

Schablone:

	▽	▽	.	▽	[	▽	,	▽	▽		-

Daten:	Druckbild:
1 2 3 4 5 6 7 D	1 2 . 3 4 5 , 6 7 -
0 2 3 4 5 6 7 D	2 . 3 4 5 , 6 7 -
0 0 3 4 5 6 7 D	3 4 5 , 6 7 -
0 0 0 4 5 6 7 D	4 5 , 6 7 -
0 0 0 0 5 6 7 D	5 , 6 7 -
0 0 0 0 0 6 7 D	0 , 6 7 -
0 0 0 0 0 0 7 D	0 , 0 7 -
0 0 0 0 0 0 0 D	0 , 0 0 -

III

Schablone:

	▽	.	▽	▽	▽	.	▽	[	▽		C	R

Daten:	Druckbild:
1 2 3 4 5 6 7 F	1 . 2 3 4 . 5 6 7
0 2 3 4 5 6 7 F	2 3 4 . 5 6 7
0 0 3 4 5 6 7 F	3 4 . 5 6 7
0 0 0 4 5 6 7 F	4 . 5 6 7
0 0 0 0 5 6 7 F	5 6 7
0 0 0 0 0 6 7 F	6 7
0 0 0 0 0 0 7 F	7
0 0 0 0 0 0 0 F	0

Abb. 10

Fahren Sie bitte auf der nächsten Seite fort.

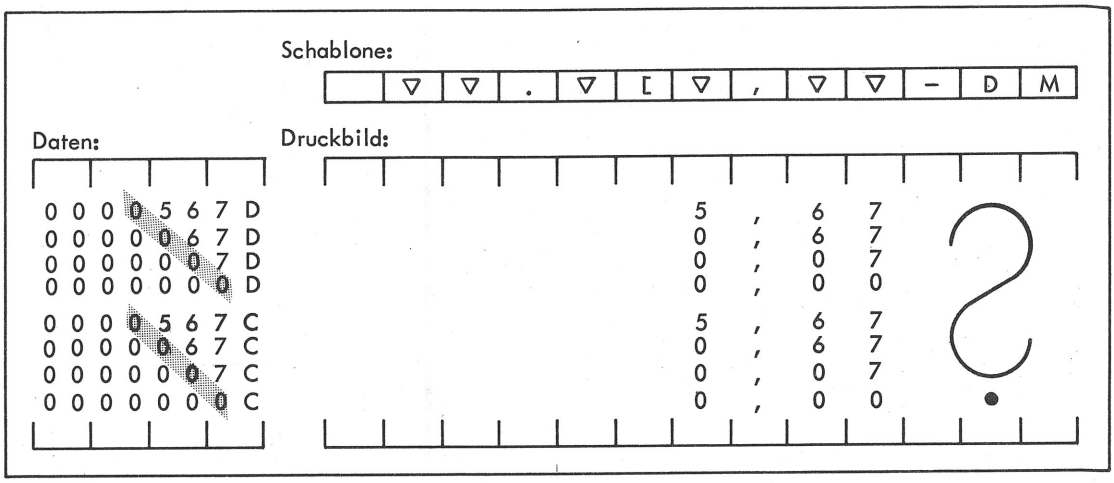

Schablone:
▽ ▽ . ▽ [▽ , ▽ ▽ – D M
Daten:
Druckbild:
0 0 0 0 5 6 7 D
0 0 0 0 0 6 7 D
0 0 0 0 0 0 7 D
0 0 0 0 0 0 0 D
0 0 0 0 5 6 7 C
0 0 0 0 0 6 7 C
0 0 0 0 0 0 7 C
0 0 0 0 0 0 0 C
5 , 6 7
0 , 6 7
0 , 0 7
0 , 0 0
5 , 6 7
0 , 6 7
0 , 0 7
0 , 0 0

Abb. 11

38 Betrachten Sie das Beispiel in Abb. 11.

Geben Sie an, welchen Inhalt die rechten 3 Bytes bei den 8 verschiedenen Zahlen erhalten.

-DM
-DM
-DM
-DM
ƀƀƀ
ƀƀƀ
ƀƀƀ
ƀƀƀ

39 Die Bezeichnung "-DM" rechts in der Schablone bleibt nicht erhalten, wenn die aufbereitete Zahl _____ ist.

Jedes der 3 Zeichen wird dann durch das _____ ersetzt.

positiv

Füllzeichen

Es stellt sich die Frage, ob es eine einfache Möglichkeit gibt, eine Bezeichnung wie "DM" unabhängig vom Vorzeichen der aufbereiteten Zahl immer rechts im Druckbild erscheinen zu lassen.

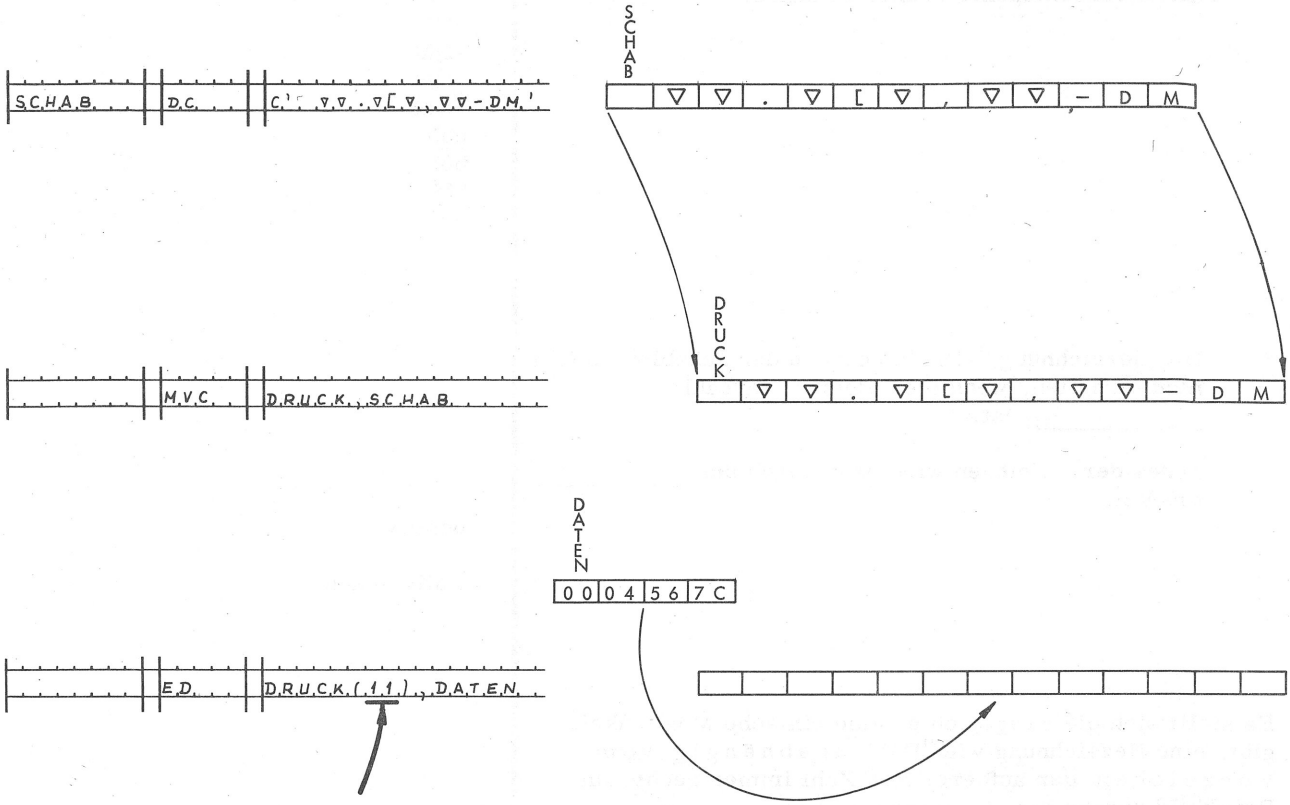


Abb. 12

40 Abb. 12 enthält eine Lösung des Problems.

Die Schablone ist in einer Länge von ___(ZW)___ definiert.

Vor der ED-Operation wird die Schablone in (verkürzter, voller) _____ Länge in das Ausgabefeld übertragen.

13 Bytes

voller

41 Die Daten werden durch eine ED-Operation in die Schablone im Ausgabefeld "DRUCK" eingegeben.

Eine positive Zahl wird aufbereitet. Betrachten Sie die betreffende Instruktion genau.

Welchen Inhalt hat das Feld DRUCK nach der Ausführung der ED-Instruktion?

45,67 DM

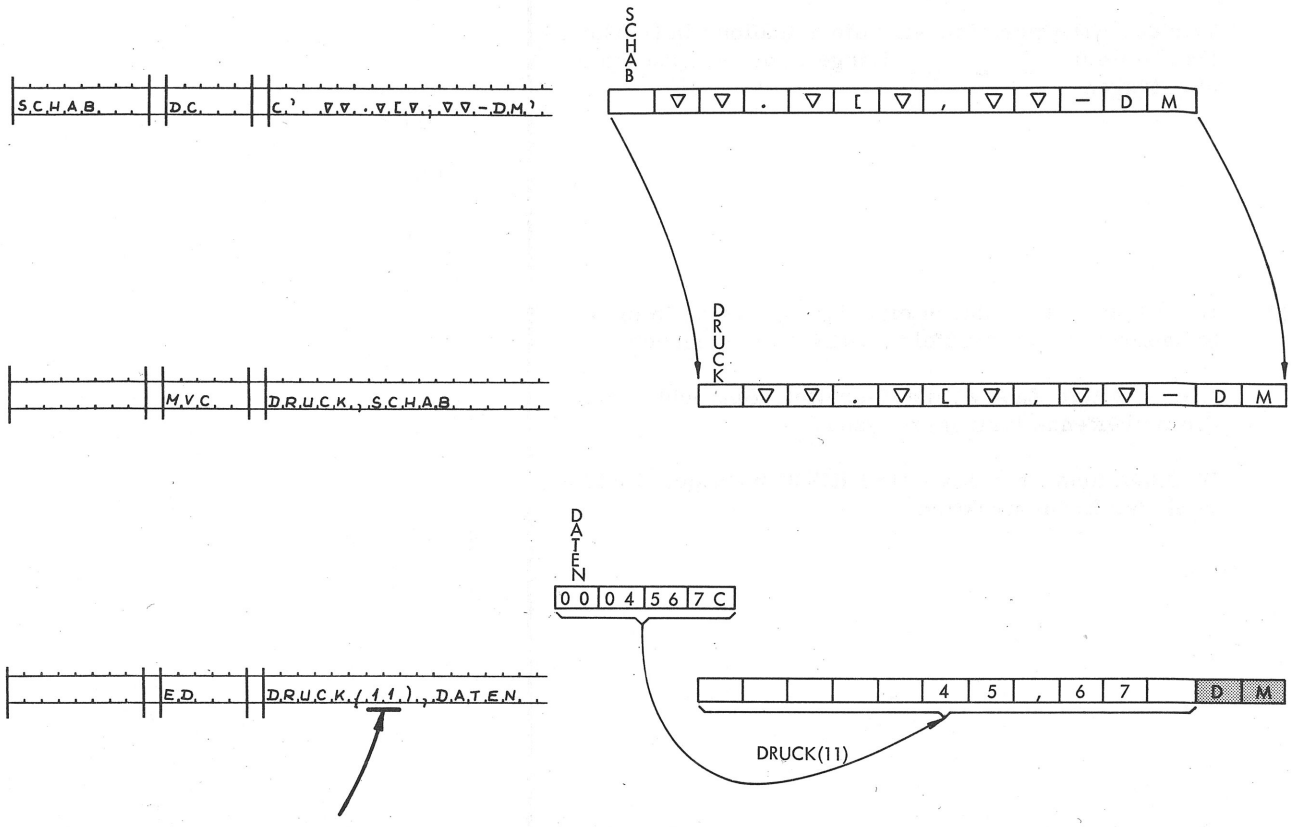


Abb. 13

42 Sehen Sie sich Abb. 13 an.

Durch die explizite Längenangabe "11" werden die rechten __(Z)__ Bytes des Ausgabefeldes von der ED-Operation ausgeschlossen.

Dadurch bleibt ihr Inhalt unverändert. Vor wie nach der ED-Operation enthalten diese Bytes die Zeichen _____.

2

DM

43 Die Methode in Abb. 13 bietet sich immer dann an, wenn rechts neben einer aufbereiteten Zahl eine konstante Bezeichnung im Druckbild erscheinen soll.

Diese Bezeichnung bleibt unabhängig vom Vorzeichen der Zahl erhalten, wenn die Länge des ersten Operanden der ED-Operation durch ____(WW)____ verkürzt wird.

eine explizite Längenangabe

Name	Symbol	Ver-schlüsselung	Lochkarten-Code
Ziffern-Auswahl-Zeichen	▽	0010 0000	11,0,9,8,1
Start-Zeichen für geltende Ziffern	[	0010 0001	0,9,1
Feld-Trennungs-Zeichen	]	0010 0010	0,9,2

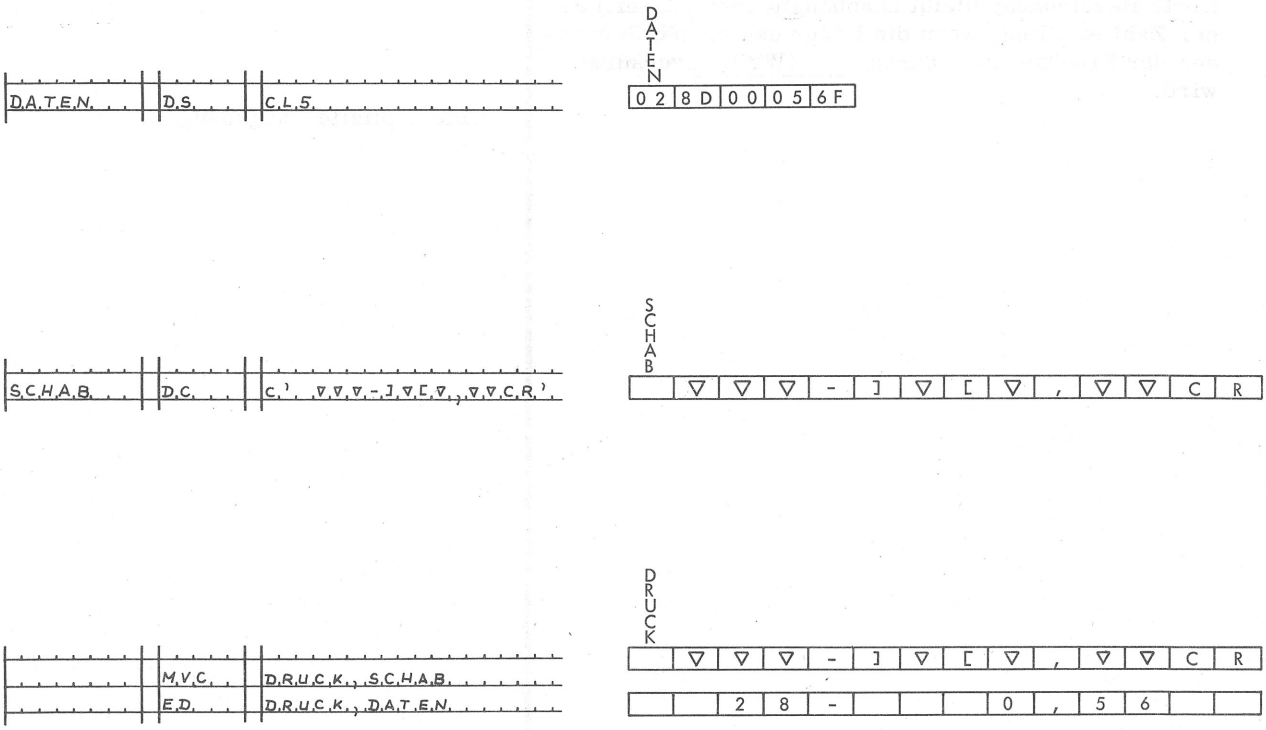


Abb. 14

44 Neben den Zeichen " ▽ " und " [" lernen Sie jetzt ein drittes Zeichen kennen, das beim Aufbau von Schablonen eine besondere Bedeutung hat. Es ist das Zeichen "] ".

Betrachten Sie Abb. 14.

Das Zeichen "] " heißt _____.

In der Schablone in Abb. 14 steht es im __(Z)__ Byte von links.

Feld-Trennungs-Zeichen

6.

45 Im Beispiel in Abb. 14 werden 2 Zahlenfelder mit einem einzigen MVC-Befehl und einem einzigen ED-Befehl aufbereitet. Das ist möglich, wenn die Felder sowohl vor als auch nach dem Aufbereiten dicht nebeneinander liegen.

Um die Wirkung des Zeichens "] " zu erkennen, stellen wir uns vor, die Schablone habe folgende veränderte Form:

	▽	▽	▽	-		▽	[	▽	,	▽	▽	C	R

An die Stelle des "] " ist ein _____ getreten.

Geben Sie an, welche Zeichen in diesem Fall im Druckbild erscheinen würden. (Benutzen Sie "ƀ" für "Blank"!)

Blank

ƀƀ2 8 - ƀ0 0 0, 5 6 bƀ

die Unterdrückung führen der Nullen wurde durch d Ziffer "2" für die ganze Schablone aufgehoben.

"-ƀ" wird als Minuskenn- zeichnung für die linke Z aufgefaßt und bleibt erhal

Name	Symbol	Ver-schlüsselung	Lochkarten-Code
Ziffern-Auswahl-Zeichen	▽	0010 0000	11,0,9,8,1
Start-Zeichen für geltende Ziffern	[	0010 0001	0,9,1
Feld-Trennungs-Zeichen	]	0010 0010	0,9,2

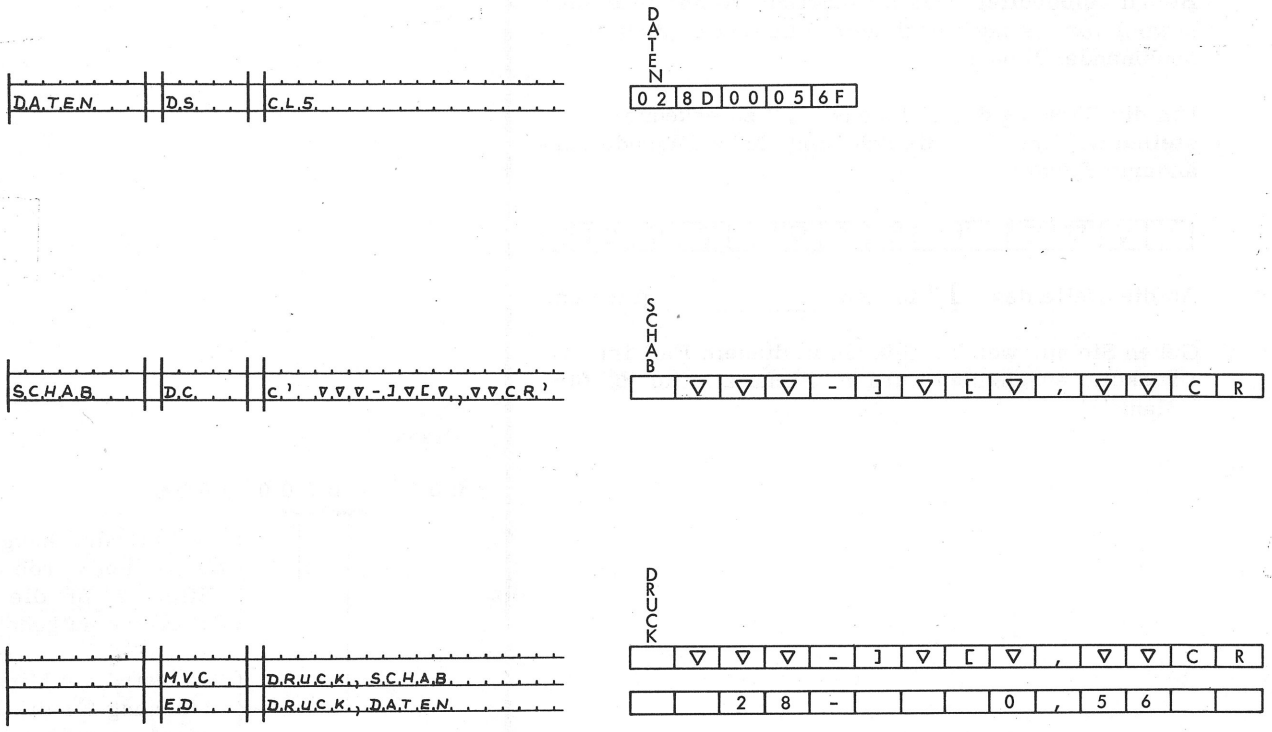


Abb. 14

Ist das Zeichen "] " in einer Schablone vertreten, so bewirkt es zweierlei:

a) Von der rechts folgenden Stelle an ist die Unterdrükkung führender Nullen (wieder) wirksam.

b) An die Stelle des Zeichens "]" tritt im Druckbild das Füllzeichen.

46 Im Beispiel in Abb. 14 hat die zweite Zahl im Feld DATEN drei führende Nullen.

Das Zeichen "]" bewirkt, daß von diesen drei Nullen _____ unterdrückt werden.

Für die rechts folgenden Stellen wird dann die Nullenunterdrückung wieder aufgehoben durch das Zeichen " _____ ".

zwei

[

47 Das Zeichen "] " erscheint wie die Zeichen " ▽ " und "[" nicht im Druckbild.

Anders als beim " ▽ " und " [" wird aber für das "] " keine ___(WW)___ eingefüllt, sondern das _____.

Sinngemäß:

Ziffer aus dem Datenfeld

Füllzeichen

Bezüglich der Nullenunterdrückung hat ein positives Vorzeichen im Datenfeld die gleiche Wirkung wie das "] ".

Wäre also die linke Zahl im Feld DATEN in Abb. 14 immer positiv, so wäre das Zeichen "] " in der Schablone überflüssig.

Im Merkblatt 420, Seite 2 sind die Regeln über die Schablone zusammengefaßt. Eventuell aufgetretene Unklarheiten wird dieses Merkblatt beseitigen. Achten Sie beim Durchlesen besonders auf die Wirkung des Vorzeichens.

Wir werden dieses Merkblatt zum Lösen der anschließenden Übungen benutzen.

Entnehmen Sie nun bitte Ihren Arbeitsmitteln Merkblatt 420, Seite 2, und machen Sie sich mit seinem Inhalt vertraut.

48 Wir wiederholen kurz, was Sie in diesem Merkblatt zum Vorzeichen gelesen haben.

Ein negatives Vorzeichen im Datenfeld hat überhaupt keinen Einfluß auf den Aufbereitungsvorgang.

Wir nehmen einmal an, daß beim Auftauchen des negativen Vorzeichens die Nullenunterdrückung nicht wirksam ist - wegen einer links vorangehenden Ziffer 1 bis 9 aus dem Datenfeld oder einem links vorangehenden Zeichen " [" in der Schablone -.

Was geschieht dann mit rechts folgenden "anderen" Zeichen in der Schablone?

Sinngemäß:

Die Nullenunterdrückung bleibt aufgehoben. Daher bleiben die "anderen" Zeichen erhalten und werden nicht durch das Füllzeichen ersetzt. - Häufigste Anwendung: Minuskennzeichnung.

49 Ein positives Vorzeichen dagegen läßt die Nullenunterdrückung wieder wirksam werden.

Daher bleiben in der Schablone rechts folgende "andere" Zeichen bei einem positiven Vorzeichen
(WW) .

nicht erhalten
(Sie werden durch das Füllzeichen ersetzt.)

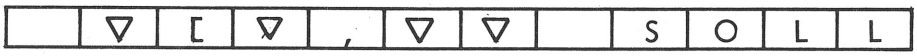
50 Benutzen Sie bitte Merkblatt 420, Seite 2 zur Lösung der folgenden Übungen. In diesen Übungen sollen Schablonen von Ihnen entwickelt werden. Die Daten vor und nach der Aufbereitung sind vorgegeben.

Entnehmen Sie bitte Ihren Arbeitsmitteln Übung 424 (2 Seiten).

Die Lösungen finden Sie auf dem folgenden Blatt.

Lösungen zu den Übungen

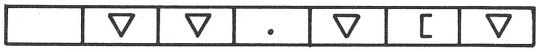
I



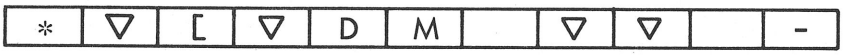
II



III



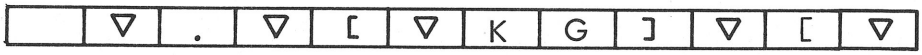
IV



V



VI



Wir haben die Druckaufbereitung und besonders den Aufbau der Schablone sehr eingehend behandelt.

Mit den folgenden Lernelementen, die noch einige Schlußfolgerungen und Ergänzungen enthalten, beenden wir dieses Kapitel.

51 In unseren Überlegungen gehen wir zunächst von folgendem Problem aus:

Bei einer Rechnungsschreibung soll das Datum,- z.B. 4. Mai 1990 - in der Form

0	4	.	0	5	.	9	0

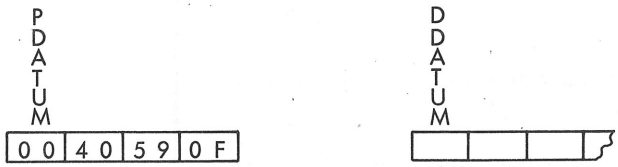
gedruckt werden. Es sei in gepackter Form im Hauptspeicher gespeichert.

Das entsprechende Speicherfeld muß ___(ZW)___ lang sein.

Es hat für das obige Beispiel den Inhalt ___(ZZ)___ (hexadezimal).

4 Bytes

0 0 4 0 5 9 0 F/0 0 4 0 5 9 0 C



Gewünschtes Druckbild:

0	4	.	0	5	.	9	0

Abb. 15

52 Betrachten Sie Abb. 15.

Das gepackte Datum steht in dem Feld _____.

Weil das Datum aus 6 Ziffern besteht, enthält das Feld links immer ___(WW)___.

PDATUM

eine führende Null

53 Die Daten werden von links nach rechts in die Schablone im Feld DDATUM eingefüllt.

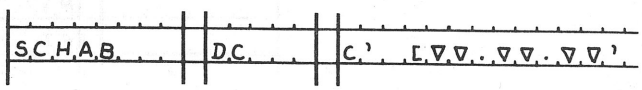
Darum muß für die linke führende Null aus PDATUM in der Schablone (keine, eine) _____ Stelle vorgesehen sein.

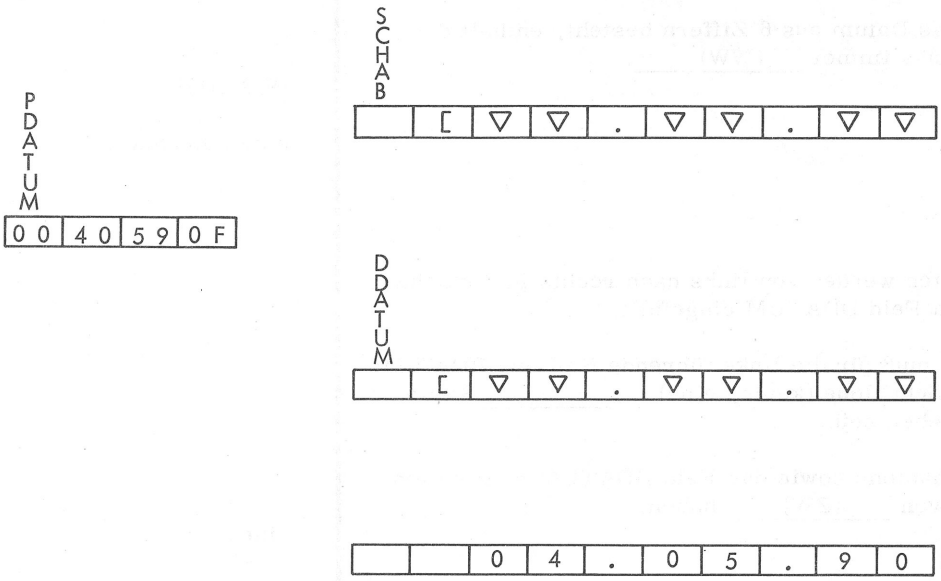
Die Schablone sowie das Feld DDATUM müssen eine Länge von ___(ZW)___ haben.

eine

10 Bytes

54 Wie lautet die Anweisung für die Definition der Schablone SCHAB für die Aufbereitung des Datums in Abb. 15? (Bedenken Sie bitte, daß nur die linke führende Null aus PDATUM beim Aufbereiten unterdrückt werden soll.)





Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	:	
SCHAB	DC	C' [▽▽.▽▽.▽▽'
	:	
	MVC	DDATUM,SCHAB
	ED	DDATUM,PDATUM
	:	

Abb. 16

55 Betrachten Sie Abb. 16.

Sie enthält alle Angaben zur Lösung des gestellten Problems.

Das Zeichen " [" in der Schablone erfüllt einen doppelten Zweck:

a) das Auffangen und Unterdrücken der linken führenden Null

b) das ____(WW)____ für die rechts folgende Stelle.

Sinngemäß:

Aufheben der Nullenunterdrückung

56 Das Datum in Abb. 16 besteht aus 6 Ziffern.

Das gleiche Problem mit der zusätzlichen Stelle in der Schablone tritt immer auf, wenn ein Wert mit einer _____ Anzahl von Ziffern aufbereitet werden soll.

Das zusätzliche Byte in der Schablone muß dann eines der Zeichen _____ enthalten.

geraden

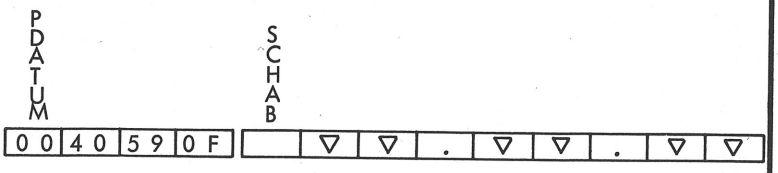
" ∇ " oder " ["

57 Grundsätzlich muß die Schablone so viele von den Zeichen " ∇ " und " [" aufweisen, wie der zweite Operand der ED-Instruktion Ziffern enthält.

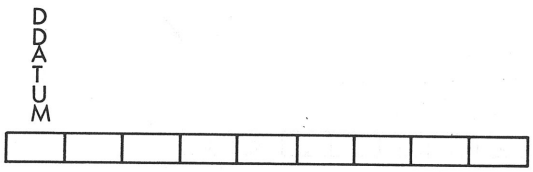
Ist der zweite Operand einer ED-Instruktion 6 Bytes lang und enthält er nur e i n Vorzeichen, so muß die Anzahl der Zeichen " ∇ " und " [" in der entsprechenden Schablone gleich ___(Z)___ sein.

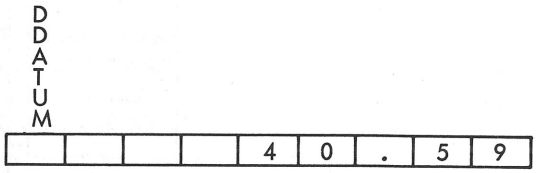
11

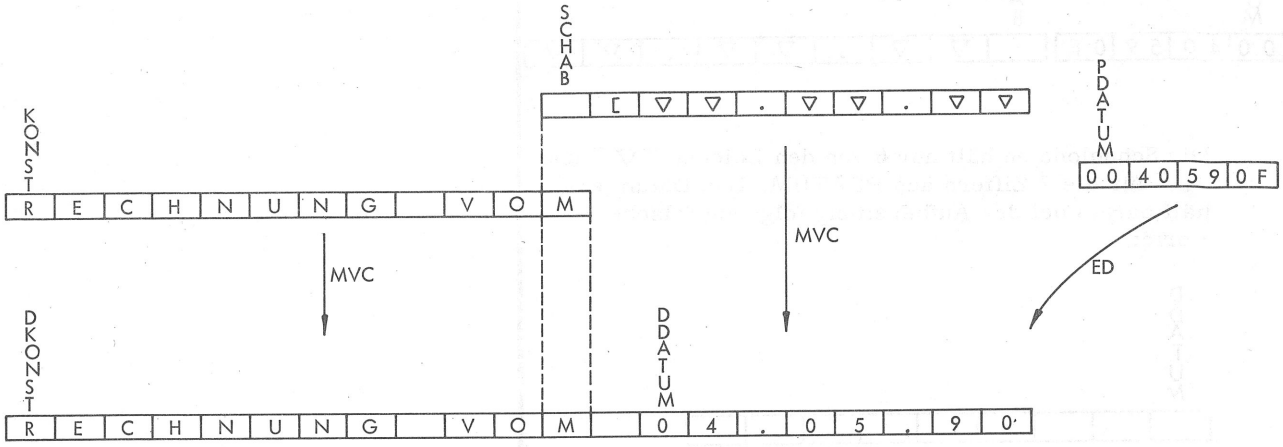
58 Betrachten Sie die folgende Schablone für das Datum aus dem Feld PDATUM.



Die Schablone enthält nur 6 von den Zeichen " ▽ " und "[" für die 7 Ziffern aus PDATUM. Das Datum erhält darum bei der Aufbereitung folgende falsche Form:







Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
P,D,A,T,U,M,	D,S,	C,L,4,
K,O,N,S,T,	D,C,	C,',R,E,C,H,N,U,N,G, ,V,O,M,'
S,C,H,A,B,	D,C,	C,', [,▽,▽, .,▽,▽, .,▽,▽,'
D,K,O,N,S,T,	D,S,	C,L,1,2,
	D,S,	C,L,1,
D,D,A,T,U,M,	D,S,	C,L,8,

Abb. 17

Wir haben gesehen:

Eine Schablone hat links eine oder sogar zwei Stellen, die nur aus programmiertechnischen Gründen erforderlich sind. Im Druckbild enthalten diese Stellen das Füllzeichen.

Im Anschluß daran stellt sich die Frage, ob wegen dieser zusätzlichen Stellen aufbereitete Zahlen nie direkt an andere Zahlen oder Konstanten anschließen können.

59 Betrachten Sie Abb. 17.

3 Instruktionen sind erforderlich, um den abgebildeten Inhalt nach DKONST und DDATUM zu bringen.

Wie heißt

a) der MVC-Befehl

b) der ED-Befehl

zur Druckaufbereitung des Datums?

a)

MVC	DDATUM-2(1Ø),SCHAB

b)

ED	DDATUM-2(1Ø),PDATUM

60 In welchem der folgenden Beispiele (a oder b) wird der MVC-Befehl für KONST richtig eingefügt?

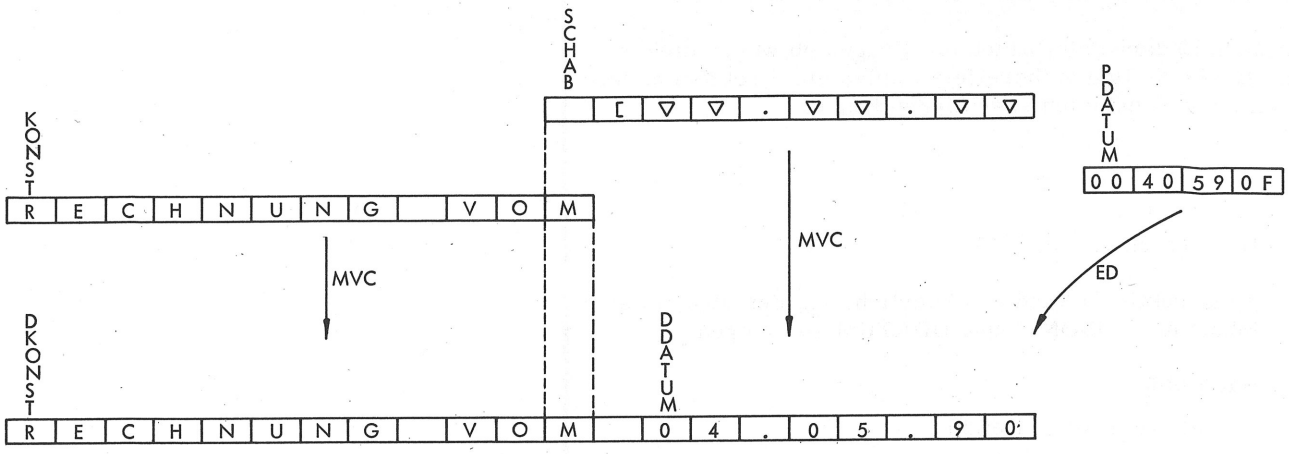
a)

MVC	DKONST,KONST
MVC	DDATUM-2(1Ø),SCHAB
ED	DDATUM-2(1Ø),PDATUM

b)

MVC	DDATUM-2(1Ø),SCHAB
ED	DDATUM-2(1Ø),PDATUM
MVC	DKONST,KONST

b)



Name	Operation	Operand und Bemerkungen
P,D,A,T,U,M,	D,S,	C,L,4,
K,O,N,S,T,	D,C,	C,',R,E,C,H,N,U,N,G, ,V,O,M,',
S,C,H,A,B,	D,C,	C,', ,[,▽,▽, ,▽,▽, ,▽,▽,'
D,K,O,N,S,T,	D,S,	C,L,1,2,
	D,S,	C,L,1,
D,D,A,T,U,M,	D,S,	C,L,8,

Abb. 17

61 Verfolgen Sie genau, warum die Reihenfolge der Befehle in diesem Beispiel f a l s c h ist:

```
MVC   DKONST,KONST
MVC   DDATUM-2(10),SCHAB
ED    DDATUM-2(10),PDATUM
```

Es geht um den Inhalt von einem Byte. Dieses Byte liegt ganz rechts im Feld _____.

Nach dem ersten MVC-Befehl hat dieses Byte den Inhalt _____.

Durch den zweiten MVC-Befehl ändert sich dieser Inhalt in _____.

Die falsche Reihenfolge der Befehle bewirkt also, daß der Buchstabe _____ nicht im endgültigen Druckbild erscheint.

DKONST

M

ö

M

62 Wir ziehen daraus den Schluß:

Sollen aufbereitete Daten ohne Zwischenräume gedruckt werden, so müssen sie im Druckbild von _____ nach _____ aufbereitet werden. Das am weitesten rechts stehende Feld muß _____ bearbeitet werden.

rechts

links

zuerst

Übungen

Entnehmen Sie Ihren Arbeitsmitteln Übung 425, 3 Seiten.

Tragen Sie Ihre Ergebnisse ein. Prüfen Sie zum Schluß die Richtigkeit mit Hilfe der Lösungen auf den folgenden Seiten.

Treten Fragen auf, so benutzen Sie bitte Merkblatt 420, Seite 2.

1. Entwickeln Sie das Druckbild einiger Zahlen:

Schablone:

*	▽	[	,	▽	▽	▽	K	G

(In der entsprechenden ED-Instruktion sollen die rechten 2 Bytes des ersten Operanden n i c h t durch eine explizite Längenangabe von der Operation ausgeschlossen sein.)

	Daten:	Druckbild:
a)	0 8 1 9 2 D	* * 8 , 1 9 2 K G
b)	0 0 0 0 8 C	* * * , 0 0 8 * *
c)	F 3 F 9 C 6	ungültige Daten
d)	3 0 0 0 2 6	* 3 0 , 0 0 2 K G

* Da rechts neben der Ziffer "2" im Feld DATEN kein positives Vorzeichen erscheint, bleibt "KG" erhalten.

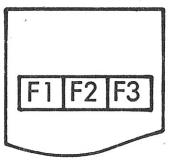
2. Wie viele der Zeichen " ▽ " und " [" muß die Schablone enthalten, wenn aufbereitet werden soll:

a) eine in 4 Bytes gepackte Dezimalzahl

b) eine 9-stellige Dezimalzahl ?

Antwort: a) 7 b) 9

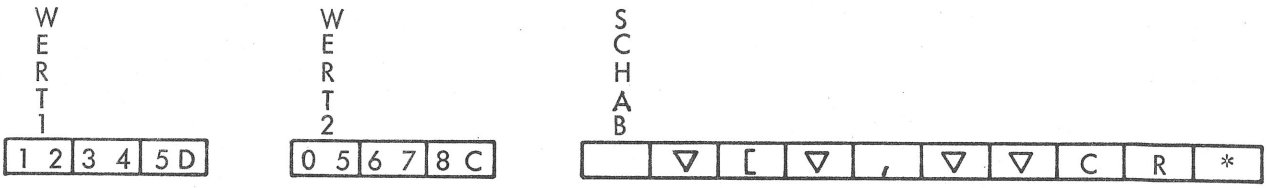
3.



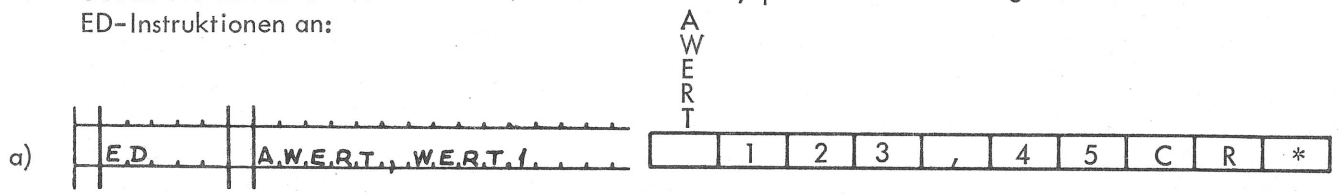
Die Druckfelder F1, F2 und F3 sollen in einer Liste so angeordnet werden, daß beim Aufbereiten keine Zwischenräume (durch Füllzeichen etc.) entstehen. In welcher Reihenfolge müssen die Felder aufbereitet werden ?

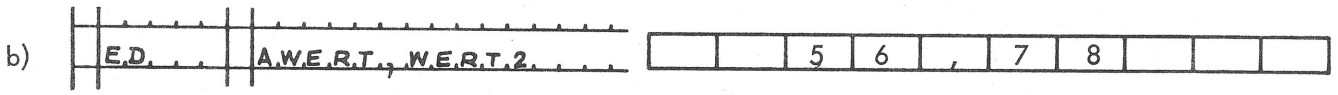
Antwort: F3, F2, F1

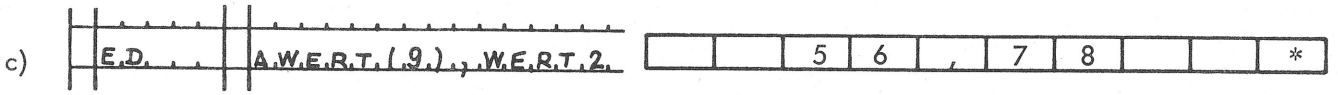
4. Folgende Daten und folgende MVC-Instruktion liegen vor:



Geben Sie den Inhalt von AWERT nach verschiedenen, jeweils auf MVC folgenden ED-Instruktionen an:







5. Geben Sie für die folgenden 2 Beispiele die Definition der Schablone SCHAB und die MVC- und ED-Instruktion an:

a)

WERT

1	2	3	4	5	D
0	2	3	4	5	D
0	0	3	4	5	C
0	0	0	4	5	C
0	0	0	0	5	D
0	0	0	0	0	C

AWERT

	1	2	.	3	4	5		-
		2	.	3	4	5		-
				3	4	5		
					4	5		
						5		-
						0		

SCHAB	DC	C' ∇∇.∇[∇ -'
	⋮	
	MVC	AWERT,SCHAB
	ED	AWERT,WERT

b)

WERT

1	2	3	4	5	D
0	2	3	4	5	D
0	0	3	4	5	C
0	0	0	4	5	C
0	0	0	0	5	D
0	0	0	0	0	C

AWERT

*	1	2	3	,	4	5	-	D	M
*	*	2	3	,	4	5	-	D	M
*	*	*	3	,	4	5	*	D	M
*	*	*	0	,	4	5	*	D	M
*	*	*	0	,	0	5	-	D	M
*	*	*	0	,	0	0	*	D	M

SCHAB	DC	C'*∇[∇,∇∇-DM'
	MVC	AWERT,SCHAB
	ED	AWERT(8),WERT

Merkblatt 420, Seite 3 enthält Beispiele mit zwei typischen Schablonen.

Entnehmen Sie bitte das Merkblatt Ihren Arbeitsmitteln und betrachten Sie die Beispiele.



Band 1, Seite 337

KAPITEL VI

DIE OPERATIONEN
BALR, A, AH, AR, LR, L UND LA

Lernstoff:

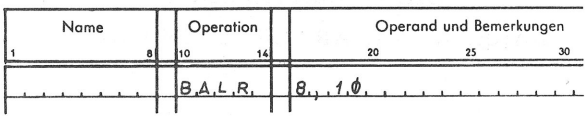
BALR	-	Verzweigen und Laden Folgeadresse - Verzweigadresse im Register - (einschließlich Verbindung von BALR und USING)
A	-	Addieren
AH	-	Addieren Halbwort
AR	-	Addieren Register
LR	-	Laden Register
L	-	Laden
LA	-	Laden Adresse

BALR VERZWEIGEN UND LADEN FOLGEADRESSE - VERZWEIGADRESSE IM REGISTER -

(Branch And Link Register)

Diese Operation bewirkt das Verzweigen zu einer Adresse im Hauptspeicher und das Laden (Speichern) der Folgeadresse (Rücksprungadresse) in einem allgemeinen Register.

Beispiel einer Instruktion:



Wirkungsweise:

Die beiden Operanden sind 2 allgemeine Register (im Beispiel Register 8 und Register 10).

Der zweite Operand enthält die Verzweigadresse. In den ersten Operanden wird die Folgeadresse (Adresse der folgenden Instruktion) geladen.

Die Operation wird in 3 Schritten durchgeführt:

a) Zwischenspeichern der Verzweigadresse aus dem zweiten Operanden

b) Laden der Folgeadresse in den ersten Operanden

c) Verzweigen

Besonderheit:

Es wird nicht verzweigt, wenn der zweite Operand 0 ist, z.B.:

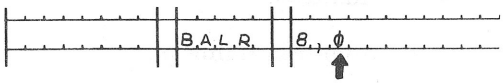


Abb. 1

1 Der symbolische Operationsschlüssel für die Instruktion "Verzweigen und Laden Folgeadresse" lautet _____.

Im Operandenfeld dieser Instruktion werden 2 ___(WW)___ angegeben.

BALR

allgemeine Register

2 In Abb. 1 oben gibt der 1. Operand der BALR-Instruktion das allgemeine Register __(Z)__ an.

In dieses Register wird die _____ geladen.

8

Folgeadresse

3 Im 2. Operanden dieser Instruktion ist das allgemeine Register __(Z)__ angegeben.

Dieses Register enthält die Hauptspeicheradresse, zu der _____ werden soll.

10

verzweigt

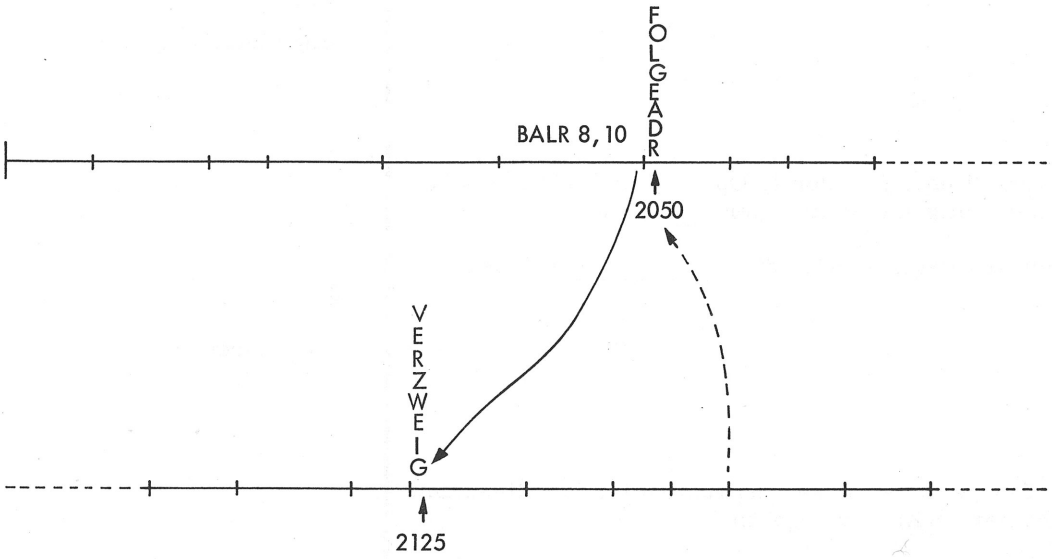


Abb. 2

4 Betrachten Sie bitte Abb. 2. -

Die Instruktionen des Programmes werden im Hauptspeicher der Reihenfolge der Codierung entsprechend gespeichert. In Abb. 2 sind die gespeicherten Instruktionen auf 2 Linien abgetragen.

Durch die BALR-Instruktion ist es möglich, zu irgendeiner Adresse im Programm zu verzweigen.

Bei der BALR-Operation in Abb. 2 enthält das Register 10 die Adresse __(Z)__, zu der verzweigt werden soll.

2125

5 Ab der Adresse 2125 folgen nun wiederum einige Instruktionen im Hauptspeicher, die zunächst durchgeführt werden.

Es kann sein, daß man an einer späteren Stelle wieder zurückverzweigen will, um mit der Instruktion fortzufahren, die der BALR-Instruktion folgt.

Dieses Zurückverzweigen ist jederzeit möglich, denn die Folgeadresse ist im Register __(Z)__ gespeichert. Es ist die Adresse __(Z)__. Bei dieser Adresse beginnt die Instruktion mit dem Namen _____.

Für das Rückverzweigen zur Adresse im Register 8 gibt es eine entsprechende Operation.

8

2050

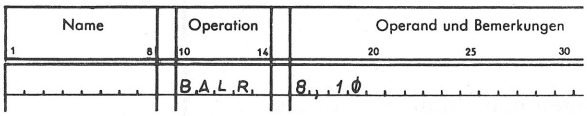
FOLGEADR

BALR VERZWEIGEN UND LADEN FOLGEADRESSE - VERZWEIGADRESSE IM REGISTER -

(Branch And Link Register)

Diese Operation bewirkt das Verzweigen zu einer Adresse im Hauptspeicher und das Laden (Speichern) der Folgeadresse (Rücksprungadresse) in einem allgemeinen Register.

Beispiel einer Instruktion:



Wirkungsweise:

Die beiden Operanden sind 2 allgemeine Register (im Beispiel Register 8 und Register 10).

Der zweite Operand enthält die Verzweigadresse. In den ersten Operanden wird die Folgeadresse (Adresse der folgenden Instruktion) geladen.

Die Operation wird in 3 Schritten durchgeführt:

a) Zwischenspeichern der Verzweigadresse aus dem zweiten Operanden

b) Laden der Folgeadresse in den ersten Operanden

c) Verzweigen

Besonderheit:

Es wird nicht verzweigt, wenn der zweite Operand 0 ist, z.B.:

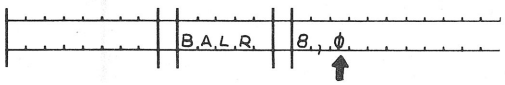
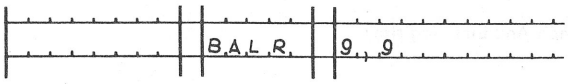


Abb. 1

6 In Abb. 1 ist gesagt, daß die BALR-Operation in 3 Schritten durchgeführt wird.
Es liege folgende Instruktion vor:



Vor der Ausführung dieser Instruktion habe Register 9 den Inhalt 1053 und die auf BALR folgende Instruktion die Adresse 3011.

a) Welche Adresse steht nach der Ausführung der BALR-Instruktion in Register 9?

b) Zu welcher Adresse verzweigt das Programm?

c) Zu welcher Adresse würde das Programm verzweigen, wenn der erste Schritt (das Zwischenspeichern der Verzweigadresse, vgl. Abb. 1) wegfallen würde?

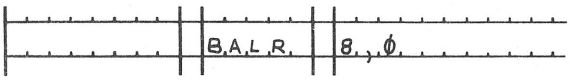
3011

1053

3011 (Damit wäre eine BALR-Instruktion mit 2 gleichen Operanden sinnlos.)

Soweit zur normalen Wirkungsweise der BALR-Instruktion. -

Der Grund, warum wir die Instruktion BALR als Lernstoff in diesen Lehrgang aufgenommen haben, liegt jedoch nicht bei der normalen Wirkungsweise, sondern bei der Besonderheit der BALR-Instruktion:



Wir wenden uns in den folgenden Lernelementen nun dieser Besonderheit zu.

7 Wird im 2. Operanden einer BALR-Instruktion eine (Z) angegeben, so wird nur die _____ in das Register im 1. Operanden geladen. Es erfolgt (WW).

0

Folgeadresse

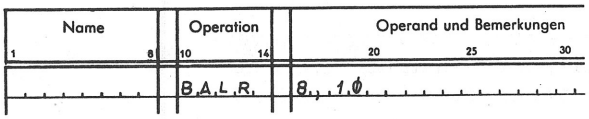
keine Verzweigung
(weil 0 nicht "Register 0", sondern "kein Register" bedeutet)

BALR VERZWEIGEN UND LADEN FOLGEADRESSE - VERZWEIGADRESSE IM REGISTER -

(Branch And Link Register)

Diese Operation bewirkt das Verzweigen zu einer Adresse im Hauptspeicher und das Laden (Speichern) der Folgeadresse (Rücksprungadresse) in einem allgemeinen Register.

Beispiel einer Instruktion:



Wirkungsweise:

Die beiden Operanden sind 2 allgemeine Register (im Beispiel Register 8 und Register 10).

Der zweite Operand enthält die Verzweigadresse. In den ersten Operanden wird die Folgeadresse (Adresse der folgenden Instruktion) geladen.

Die Operation wird in 3 Schritten durchgeführt:

a) Zwischenspeichern der Verzweigadresse aus dem zweiten Operanden

b) Laden der Folgeadresse in den ersten Operanden

c) Verzweigen

Besonderheit:

Es wird nicht verzweigt, wenn der zweite Operand 0 ist, z.B.:

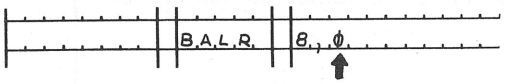
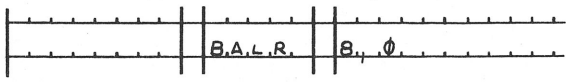


Abb. 1

8



Diese Instruktion kann also dazu benutzt werden, einen bestimmten Inhalt in das Register __(Z)__ zu _____.

8

laden

9 Wie Sie bereits wissen, wird dem Assembler durch eine USING-Anweisung mitgeteilt, welches Basisregister und welche Basisadresse für die Adreßbildung verwendet werden sollen. Durch die USING-Anweisung wird das Basisregister jedoch n i c h t mit der Basisadresse g e l a d e n .

Wir benötigen also eine Instruktion, mit deren Hilfe zum Zeitpunkt der Ausführung des Programms das vorgesehene Basisregister mit der vorgesehenen Basisadresse geladen werden kann.

Hierzu können wir nun die _____-Instruktion benutzen.

BALR

Name 1 8	Operation 10 14	Operand und Bemerkungen 20 25 30
	S,T,A,R,T	8,ϕ,ϕ,
A,N,F,A,N,G,	B,A,L,R,	8,,ϕ,
	U,S,I,N,G	*,,8,
	:	
	:	
E,B,E,R,	D,S,	ϕ,C,L,1,2,
	D,S,	C,L,2,
E,F,E,L,D,A,	D,S,	C,L,5,
E,F,E,L,D,B,	D,S,	C,L,5,
	:	
	:	
	E,N,D,	A,N,F,A,N,G,

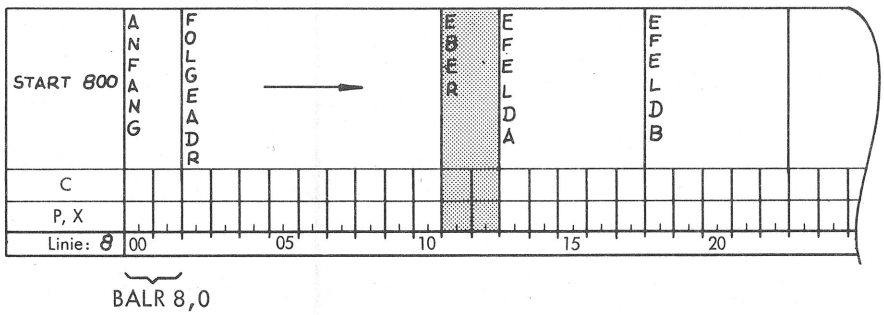


Abb. 3

10 Betrachten Sie bitte Abb. 3. -

Durch die START-Anweisung wird der Zuordnungszähler auf 800 eingestellt. Die BALR-Instruktion mit dem Namen Anfang wird im Hauptspeicher (siehe Speicherbelegung Abb. 3) in __(Z)__ Bytes gespeichert. Sie steht also in den Speicherstellen __(Z)__ bis __(Z)__.

2

800 (bis) 801

11 Die BALR-Instruktion in Abb. 3 bewirkt, daß die Folgeadresse in das allgemeine Register __(Z)__ geladen wird.

Eine Verzweigung erfolgt nicht, da im zweiten Operandenfeld der BALR-Instruktion eine __(Z)__ steht.

8

0

12 Nach der BALR-Instruktion folgen im Hauptspeicher einige Maschineninstruktionen, die wir nicht im einzelnen aufgezeichnet haben (Pfeil in Abb. 3).
Die USING-Anweisung in der Codierung der Abb. 3 wird nicht im Hauptspeicher gespeichert, da sie als Assemblerinstruktion nur der Umwandlung dient.

Die Folgeadresse der BALR-Instruktion ist die Adresse der nächsten Instruktion, die sich an die BALR-Instruktion anschließt.

Durch die BALR-Instruktion in Abb. 3 wird also die Folgeadresse __(Z)__ in das allgemeine Register __(Z)__ geladen.

802

8

13 Die USING-Anweisung teilt dem Assembler mit, daß das allgemeine Register __(Z)__ als _____ benutzt werden soll und daß der augenblickliche Wert des Zuordnungszählers als _____ zu verwenden ist.

8

Basisregister

Basisadresse

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	START	8,0,0
ANFANG	BALR	8,,0
	USING	*,,8
	:	
	:	
EBER	DS	0CL12
	DS	CL2
EFELDA	DS	CL5
EFELDB	DS	CL5
	:	
	:	
	END	ANFANG

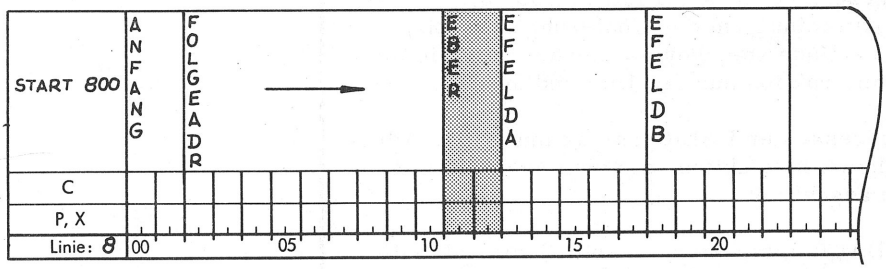


Abb. 3

14 Der Zuordnungszähler zeigt immer die nächste verfügbare Adresse im Hauptspeicher an. Der augenblickliche Wert des Zuordnungszählers ist also __(Z)__.

802

15 Sie sehen, mit Hilfe der BALR-Instruktion wird nun tatsächlich auch der Wert im Basisregister 8 geladen, der in der USING-Anweisung als _____ angegeben wird.

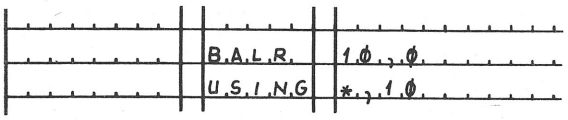
Basisadresse

16 Die BALR-Instruktion und die USING-Anweisung gehören zusammen und müssen aufeinander folgen. Denn dadurch ist sichergestellt, daß der Wert, der durch die BALR-Instruktion in das Register 8 geladen wird, und die durch USING angegebene Basisadresse g l e i c h sind.

Sowohl die Folgeadresse, die in das Register 8 geladen wird, als auch die durch USING angegebene Basisadresse haben den Wert __(Z)__ (siehe Abb. 3).

802

17



Durch diese beiden Anweisungen wird folgendes bewirkt:

a) durch die USING-Anweisung wird für den Zeitpunkt der Umwandlung das allgemeine Register 10 als _____ und der momentane Stand des Zuordnungszählers als _____ angegeben,

b) durch die BALR-Instruktion wird für den Zeitpunkt der Ausführung des Programms eine _____ in das allgemeine Register __(Z)__ geladen.

Weil die USING-Anweisung direkt auf die BALR-Instruktion folgt, enthalten Folgeadresse und Zuordnungszähler den gleichen Wert.

Basisregister

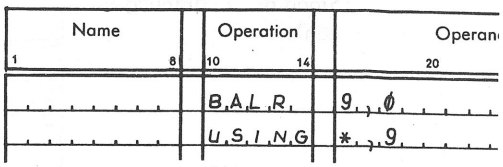
Basisadresse

Folgeadresse (d.h. die Adresse der nächsten Instruktion)

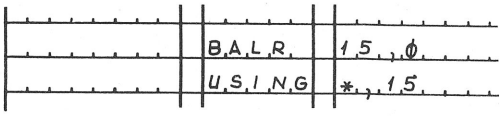
10

18 Codieren Sie nun bitte selbst:

Das Register 9 soll mit der Folgeadresse geladen werden. Für die Umwandlung sollen dasselbe Register und dieselbe Folgeadresse als Basisregister und Basisadresse zur Verfügung gestellt werden. - Benutzen Sie ein Programmschema.



19 Das Register 15 soll mit der Folgeadresse geladen und für die Umwandlung als Basisregister angegeben werden.
Wie muß die Codierung lauten?



20 Wir merken uns:

Da die Folgeadresse, die in das Register 15 geladen wird, gleichzeitig die Basisadresse des Basisregisters 15 sein muß, muß die _____-Anweisung direkt der _____-Instruktion folgen.

USING

BALR

Stand des Zuordnungszählers:	Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
		START	1600
1600	ANFANG	BALR	15,0
		USING	A1,15
1602	A1	B	B1
		:	
		:	

Abb. 4

21 Betrachten Sie bitte Abb. 4. -

Der Programmausschnitt in Abb. 4 beginnt mit der START-Anweisung. Durch diese Anweisung wird der Zuordnungszähler auf 1600 eingestellt. Die BALR-Instruktion mit dem Namen ANFANG belegt im Hauptspeicher 2 Bytes und wird somit in den Speicherstellen __(Z)__ bis __(Z)__ gespeichert.

1600 (bis) 1601

22 Durch die BALR-Instruktion in Abb. 4 wird das Register 15 mit der Adresse der nächsten Instruktion (Folgeadresse) geladen. Die Assembleranweisung USING dient nur der Umwandlung des Programms und ist nachher im umgewandelten Programm nicht mehr vorhanden.

Die nächste Instruktion, die nach der BALR-Instruktion im Hauptspeicher gespeichert wird, ist also die Instruktion mit dem Namen A1. A1 ist die symbolische Adresse der nächsten Instruktion.

Durch die BALR-Instruktion wird somit als Folgeadresse die Adresse von A1, nämlich die Adresse __(Z)__, in das Register __(Z)__ geladen.

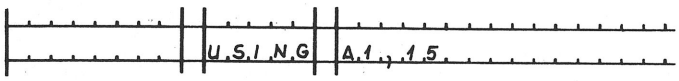
1602

15

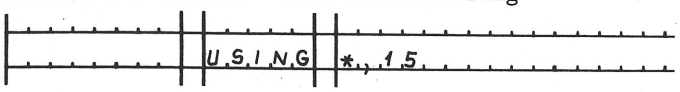
23 Die USING-Anweisung in Abb. 4 enthält statt des Sterns im 1. Operanden das Symbol A1.

Bisher hatten wir die nächste verfügbare Adresse durch einen Stern in der USING-Anweisung als Basisadresse angegeben. In Abb. 4 geben wir die zu verwendende Basisadresse durch das Symbol A1 an. Der Effekt ist derselbe.

Statt der Anweisung



könnten wir in Abb. 4 auch die Anweisung



verwenden.

Beide USING-Anweisungen teilen dem Assembler mit, daß Register 15 als _____ benutzt und als Basisadresse der Wert __(Z)__ verwendet werden soll.

Basisregister

1602

Es ist also durchaus möglich, im 1. Operanden einer USING-Anweisung die nächste verfügbare Adresse, die als Basisadresse verwendet werden soll, auch durch einen symbolischen Namen anzugeben.

Eine Forderung muß jedoch erfüllt sein, ob nun die Basisadresse in der USING-Anweisung durch den Stern oder durch einen symbolischen Namen angegeben wird:

Der Adreßwert, der als Basisadresse angegeben wird, muß mit dem Wert, der durch die BALR-Instruktion ins Basisregister geladen wird, übereinstimmen!

Die Erfahrungen haben gezeigt, daß die falsche Anwendung von BALR und USING eine der häufigsten Fehlerquellen beim Programmieren in der Assemblersprache ist. In den folgenden Lernelementen soll deshalb aufgezeigt werden, wie man es nicht machen darf!

Stand des Zuordnungszählers:

1000

1002

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	S,T,A,R,T	1,Ø,Ø,Ø,
	U,S,I,N,G	*,,5,
	B,A,L,R,	5,,Ø,
	:	
	:	

FALSCH!

Abb. 5

24 Betrachten Sie bitte Abb. 5. -

In der Codierung der Abb. 5 wird dem Assembler durch die USING-Anweisung Register 5 als Basisregister zur Verfügung gestellt. Der momentane Wert des Zuordnungszählers soll als Basisadresse dienen.

Welcher Wert wird aufgrund der Codierung in Abb. 5 vom Umwandlungsprogramm als Basisadresse benutzt?

1000

25 Die BALR-Instruktion nimmt im Hauptspeicher 2 Bytes ein; sie wird also in den Speicherstellen __(Z)__ bis __(Z)__ gespeichert.

Welcher Wert wird also als Folgeadresse in das Register 5 geladen?

1000 (bis) 1001

1002

26 Sie sehen:
In der Codierung der Abb. 5 wird durch die USING-Anweisung die Basisadresse 1000 angegeben, während durch die BALR-Instruktion der Wert 1002 in das als Basisregister verwendete allgemeine Register 5 geladen wird.

Ordnen Sie bitte die Anweisungen der Abb. 5 so um, daß die in das Register 5 geladene Folgeadresse und die durch USING angegebene Basisadresse übereinstimmen.

Name	Operation	Op
1 8	10 14	20
	START	1,Φ,Φ,Φ
	BALR	5,,Φ
	USING	*,,5
	⋮	
	⋮	

27 Nun wird in das Register 5 die Folgeadresse __(Z)__ geladen und durch USING wird als Basisadresse __(Z)__ angegeben.

1002

1002

Stand des Zuordnungszählers:	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)	
		START	1Ø5Ø	
1050	ANFANG	BALR	5,Ø	
1052		MVC	ANR,ENR	FALSCH!
1058		MVC	ABETR,EBETR	
1064		USING	*,5	
		⋮		
		⋮		

Abb. 6

28 Betrachten Sie bitte Abb. 6. -

a) Welcher Wert wird durch die BALR-Instruktion als Folgeadresse in das Register 5 geladen?

b) Welcher Wert wird durch die USING-Anweisung als Basisadresse angegeben?

a) 1052

b) 1064

29 Sie sehen, die durch BALR geladene Folgeadresse stimmt mit der durch USING angegebenen Basisadresse nicht überein.

Außerdem könnten die beiden MVC-Instruktionen gar nicht umgewandelt werden, da bei ihrer Umwandlung dem Umwandlungsprogramm bisher weder ein Basisregister noch eine Basisadresse bekannt sind.

Ordnen Sie bitte die Anweisungen in Abb. 6 so an, daß die Umwandlung richtig durchgeführt werden kann.

Name (1–8)	Operation (10–14)	Operand und Beme (20–25)
	START	10,50
ANFANG	BALR	5,0
	USING	*,5
	MVC	ANR,ENR
	MVC	ABETR,EBETR
	⋮	
	⋮	

Stand des Zuordnungszählers:	Name (1 … 8)	Operation (10 … 14)	Operand und Bemerkungen (20, 25, 30)	
		S,T,A,R,T	1,8,ϕ,ϕ,	
1800	E,B,E,R,	D,S,	ϕ,C,L,2,ϕ,	
1800	E,N,R,	D,S,	C,L,5,	
⋮		⋮		
⋮		⋮		FALSCH!
1851	A,B,E,T,R,	D,S,	C,L,6,	
1857	A,N,F,A,N,G,	B,A,L,R,	5,,ϕ,	
1859		U,S,I,N,G	*,,5,	

Abb. 7

30 Betrachten Sie bitte Abb. 7. -

Welcher Wert wird entsprechend der Codierung in Abb. 7 als Folgeadresse in das Register 5 geladen und dann durch USING als Basisadresse angegeben?

1859

31 Die Forderung, daß der in das Basisregister geladene Wert und die durch USING angegebene Basisadresse übereinstimmen müssen, ist also erfüllt.

Vergleichen Sie jedoch nun die Basisadresse 1859 mit den effektiven Adressen von EBER, ENR und ABETR in Abb. 7.

Sie stellen fest, daß die effektiven Adressen (größer, kleiner) _____ als die Basisadresse sind.

kleiner

32 Was bedeutet das für die Bildung der relativen Adressen in einer echten Maschineninstruktion?

Versuchen Sie, die relative Adresse von EBER zu bilden!

Sinngemäß:

Es lassen sich keine gültigen relativen Adressen bilden.
Für EBER z. B. ergäbe sich eine relative Adresse von -59.

33 Negative relative Adressen sind nicht möglich! Gültige relative Adressen sind 0 bis 4095.

Wir müssen also beim Programmieren dafür sorgen, daß die Basisadresse (größer, kleiner) _____ oder gleich der niedrigsten effektiven Adresse ist.

kleiner

Stand des Zuordnungszählers:	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)	
		START	1800	
1800	EBER	DS	ØCL2Ø	
1800	ENR	DS	CL5	
⋮		⋮		
⋮		⋮		FALSCH!
1851	ABETR	DS	CL6	
1857	ANFANG	BALR	5,Ø	
1859		USING	*,5	

Abb. 7

34 Ordnen Sie bitte die Anweisungen in Abb. 7 so an, daß sich gültige relative Adressen für EBER, ENR und ABETR bilden lassen!

Name (1–8)	Operation (10–14)	Oper (20)
	START	18,00
ANFANG	BALR	5,0
	USING	*,5
	⋮	
EBER	DS	0CL20
ENR	DS	CL5
	⋮	
	⋮	
ABETR	DS	CL6

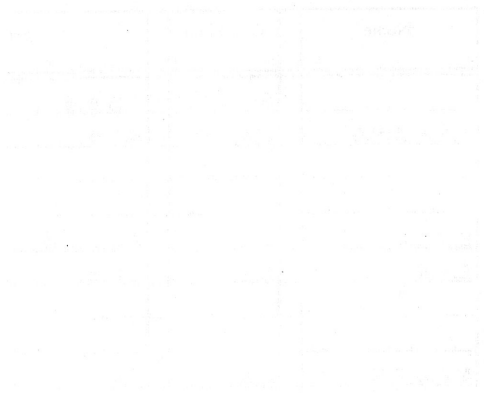
Wir merken uns:

a) Die BALR-Instruktion und die USING-Anweisung müssen im Programm so angeordnet sein, daß der Wert, der durch BALR geladen wird, mit der durch USING angegebenen Basisadresse übereinstimmt.

b) Die BALR-Instruktion und die USING-Anweisung müssen gleich zu Anfang des Programmes gegeben werden, damit die Basisadresse kleiner oder gleich den effektiven Adressen ist. Nur so können gültige relative Adressen errechnet werden.



Band 1, Seite 413



Wir kommen nun zur Besprechung der Instruktionen für Dualdaten. -

Es gibt einen ganzen Satz von Instruktionen, der zum Addieren, Subtrahieren, Dividieren, Multiplizieren, Laden usw. von Dualdaten vorgesehen ist. Wir besprechen jedoch in diesem Lehrgang nur einige der Instruktionen, die es für Dualdaten gibt.

Für das Speichern von Dualdaten gibt es Formate fester Länge, nämlich Halbwort(2 Bytes), Wort (4 Bytes) und Doppelwort (8 Bytes). Das Vorzeichen der dualen Daten wird im linken Bit eines Halbwortes, Wortes bzw. Doppelwortes durch 0 (positiv) oder 1 (negativ) verschlüsselt dargestellt.

Für das Rechnen mit Dualdaten werden Register benötigt. Die 16 allgemeinen Register haben Wortgröße, die Vorzeichendarstellung erfolgt wie in den Formaten fester Länge.

Soweit zur Auffrischung Ihres Wissens bezüglich der Dualdaten und ihrer Formate.

A

ADDIEREN

(Add full-word)

Diese Operation addiert eine Dualzahl aus einem Wort im Speicher zu einer Dualzahl in einem allgemeinen Register.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	A,	7,,W,O,R,T,A,

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel WORTA) wird zum Inhalt des ersten Operanden (im Beispiel Register 7) addiert und das Ergebnis im ersten Operanden gespeichert.

Der erste Operand ist immer ein allgemeines Register und der zweite Operand immer ein Feld im Hauptspeicher. Beide Operanden haben Wortlänge und enthalten dual verschlüsselte Zahlen.

Das Vorzeichen des Ergebnisses wird nach den algebraischen Regeln gebildet. Die Null hat immer positives Vorzeichen.

Der Bedingungsschlüssel wird eingestellt entweder auf "Überlauf"
oder auf "positiv", "negativ" oder "null".
Die Überlauf-Anzeige hat Vorrang.

Bei Überlauf erfolgt eine Programmunterbrechung.

Durchführung der Operation:

WORTA

0	0000000000000000	0000000000000001

Register 7 +

vorher:

0	0000000000000000	0000000000001110

Register 7 =

nachher:

0	0000000000000000	0000000000001111

Abb. 1

1 Der symbolische Operationsschlüssel für das Addieren von Dualdaten in Wortgröße lautet _____.

Mit Hilfe dieser Operation kann eine Dualzahl im _____ (2. Operand) zu dem Inhalt eines _____ (1. Operand) addiert werden.

A

Hauptspeicher

Registers

2 Das Ergebnis der Addition wird in dem Register gebildet, das im (1., 2.) _____ Operanden angegeben ist.

1.

3 Das Vorzeichen wird entsprechend den algebraischen Regeln gebildet. Ein Ergebnis von null wird als (positive, negative) _____ Null dargestellt.

positive

4 Betrachten Sie bitte die Durchführung der Operation in Abb. 1.

Der Inhalt von WORTA wird zu dem Inhalt des Registers 7 addiert. Die Addition erfolgt über die gesamte Länge, d.h. über 32 Bits (also einschließlich Vorzeichen).

Das Ergebnis wird in unserem Beispiel im _(WZ)_ gespeichert. Ins Dezimalsystem übertragen heißt das Ergebnis _(Z)_.

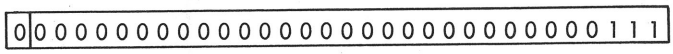
Register 7

15

5 Ist das Ergebnis in Register 7 in Abb. 1 positiv oder negativ?

positiv (0 im linken Bit)

Inhalt von WORTB



Inhalt von Register 11

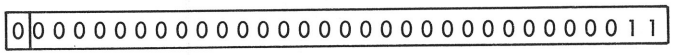
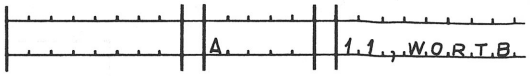


Abb. 2

6 Betrachten Sie bitte Abb. 2. -

Der Inhalt von WORTB soll nun zum Inhalt von Register 11 addiert werden. Wie heißt die entsprechende Instruktion?

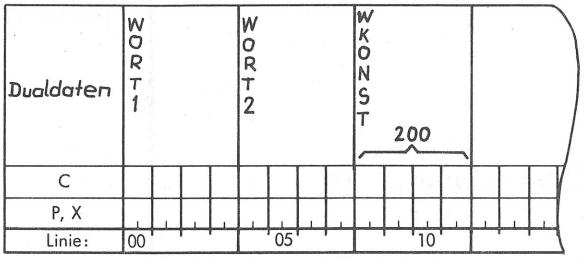


7 Das Ergebnis der Addition wird im (WZ) stehen.

Ins Dezimalsystem übertragen wird das Ergebnis den Wert (plus, minus) (Z) haben.

Register 11

plus 10



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
W,O,R,T,1,	D,S,	F,
W,O,R,T,2,	D,S,	F,
W,K,O,N,S,T,	D,C,	F,'1,2,Φ,Φ,'

Abb. 3

Übung

Die in Abb. 3 definierten Felder sowie die Konstante sollen zu allgemeinen Registern addiert werden. Wie müssen die Instruktionen für die folgenden Additionen lauten?

a) WORT1 zu Register 15

b) WORT2 zu Register 8

c) WKONST zu Register 10

Name	Operation	Operand und
1 8	10 14	20
	A	15,,WORT1
	A	8,,WORT2
	A	10,,WKONST

AH

ADDIEREN HALBWORT

(Add Half-word)

Diese Operation addiert den Inhalt eines Halbwortes im Hauptspeicher zum Inhalt eines allgemeinen Registers.

Beispiel einer Instruktion:

Name	Operation	Operand und Beme
1 8	10 14	20 25
. . . . .	A,H, . .	8, ,H,A,L,B,W,O,R,T, .

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel HALBWORT) wird zum Inhalt des ersten Operanden (im Beispiel Register 8) addiert und das Ergebnis im ersten Operanden gespeichert.

Der erste Operand ist immer ein allgemeines Register und der zweite Operand immer ein Halbwort im Hauptspeicher. Beide Operanden enthalten Dualzahlen.

Das Vorzeichen des Ergebnisses wird nach den algebraischen Regeln gebildet. Die Null hat immer positives Vorzeichen.

Der Bedingungsschlüssel wird eingestellt entweder auf "Überlauf"
oder auf "positiv", "negativ" oder "null".
Die Überlauf-Anzeige hat Vorrang.

Bei Überlauf erfolgt Programmunterbrechung.

Durchführung der Operation:

HALBWORT

0|0000000000001010

Register 8 +

vorher: 0|00000000000000000000000000000101

Register 8 =

nachher: 0|00000000000000000000000000001111

Abb. 1

1 Der symbolische Operationsschlüssel für die Operation "Addieren Halbwort" lautet _____.

Bei dieser Operation wird der Inhalt eines _____ im Hauptspeicher zum Inhalt eines _____ addiert.

AH

Halbwortes

Registers/allgemeinen Registers

2 Das Ergebnis der Addition wird im _____ Operanden gespeichert. Der Inhalt des (Registers, Halbwortes) _____ bleibt unverändert.

ersten

Halbwortes

3 Auch bei dieser Addition wird der Bedingungsschlüssel neu eingestellt.

Bei dem Beispiel in Abb. 1 wird der Bedingungsschlüssel auf _____ eingestellt.

positiv

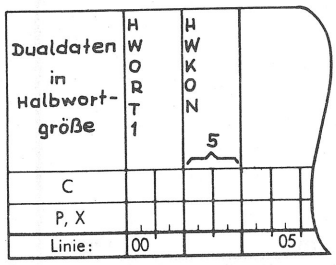
4 In Abb. 1 enthält das Halbwort als Dezimalzahl ausgedrückt den Wert + (Z) und das Register vor der Operation den Wert (Z).

Daher steht nach der Operation die Summe (Z) im Register.

10

+5

+15



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
H,W,O,R,T,1,	D,S,	H,
H,W,K,O,N,	D,C,	H,'5'

Register 10

vorher: 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0

Abb. 2

5 Abb. 2 zeigt die Definitionen und die Speicherbelegung für 2 Halbworte im Speicher.

Das erste Halbwort hat wechselnden Inhalt und das zweite enthält eine _____.

Konstante

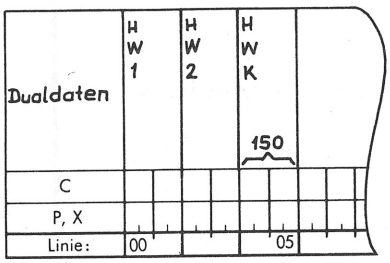
6 Schreiben Sie die Instruktion, die eine Addition von HWORT1 zu Register 10 bewirkt.

	A.H.	1Ø,HWORT1

7 Welcher duale Inhalt entsteht in Register 10 durch folgende Instruktion?

	A.H.	1Ø,HWKON

00000000000000000000000000001101



Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
H,W,1,	D,S,	H,
H,W,2,	D,S,	H,
H,W,K,	D,C,	H,'1,5ϕ,'

Abb. 3

Übungen

Die in Abb. 3 definierten Halbworte sollen zu dem Inhalt allgemeiner Register addiert werden.

Wie müssen die Instruktionen für die folgenden Additionen lauten?

a) HW1 zu Register 13

b) HW2 zu Register 14

c) HWK zu Register 15

```
        A,H,    1,3,,H,W,1
        A,H,    1,4,,H,W,2
        A,H,    1,5,,H,W,K
```

AR

ADDIEREN REGISTER

(Add Register)

Diese Operation addiert den Inhalt eines allgemeinen Registers zum Inhalt eines anderen allgemeinen Registers.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	A,R,	8,,1,1,

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel Register 11) wird zum Inhalt des ersten Operanden (im Beispiel Register 8) addiert.

Beide Operanden sind allgemeine Register und enthalten Dualzahlen.

Das Vorzeichen des Ergebnisses wird nach den algebraischen Regeln gebildet. Die Null hat immer positives Vorzeichen.

Der Bedingungsschlüssel wird eingestellt
entweder auf "Überlauf"
oder auf "positiv", "negativ" oder "null".
Die Überlauf-Anzeige hat Vorrang.

Bei Überlauf erfolgt Programmunterbrechung.

Durchführung der Operation:

Register 11

0 0000000000000000000000000001111

\+

Register 8

vorher: 0 0000000000000000000000000000011

=

Register 8

nachher: 0 0000000000000000000000000010010

Abb. 1

1 Der symbolische Operationsschlüssel für die Operation in Abb. 1 lautet _____.

AR

2 Wie bei dem Operationsschlüssel A, so wird auch bei AR der duale Inhalt eines Wortes zum dualen Inhalt eines anderen Wortes addiert.

Trotzdem besteht ein Unterschied. Der zweite Operand ist bei A ein Wort im _____ und bei AR ein Wort im _____. Der erste Operand ist in beiden Fällen ein _____.

Hauptspeicher

Register

Register

3 Das Ergebnis der Addition wird bei A und bei AR im _____ Operanden gespeichert. Es steht also in beiden Fällen in einem _____.

ersten

Register

Übungen

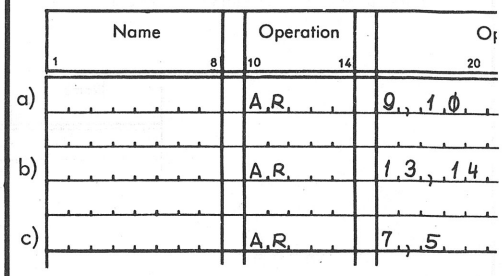
Benutzen Sie für die folgenden Codierungen ein Programmschema.

Die Inhalte der folgenden Register sollen addiert werden:

a) Register 9 und Register 10; Ergebnis in Register 9

b) Register 14 und Register 13; Ergebnis in Register 13

c) Register 5 und Register 7; Ergebnis in Register 7



LR

UMLADEN

(Load)

Mit dieser Operation wird der Inhalt eines allgemeinen Registers in ein anderes allgemeines Register geladen.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	L,R,	1,Ø,,8,

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel Register 8) wird in den ersten Operanden (im Beispiel Register 10) geladen.
Nach der Operation haben beide Register gleichen Inhalt.

Beide Operanden sind allgemeine Register und enthalten Dualdaten.

Der Bedingungsschlüssel wird nicht verändert.

Durchführung der Operation:

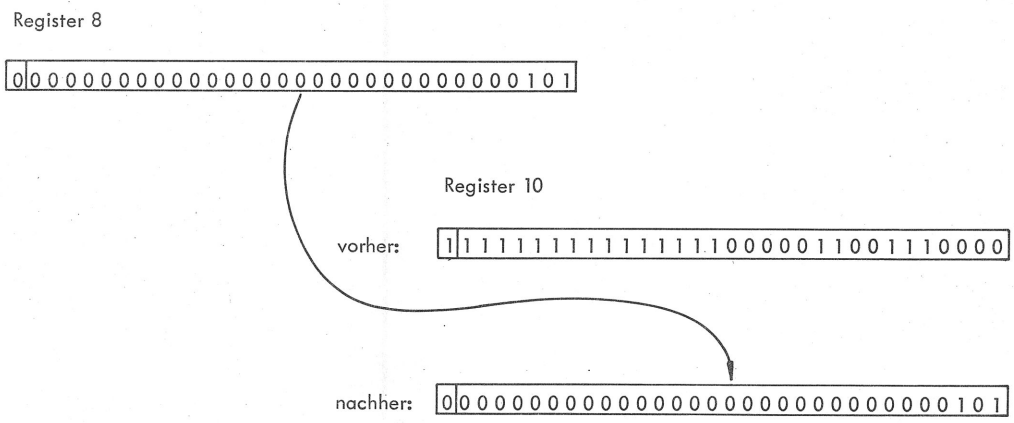


Abb. 1

1 LR ist der symbolische Operationsschlüssel für die Operation _____.

Wie bei AR sind auch hier beide Operanden _____.

Umladen

Register

2 Bei der Operation "Umladen" wird der Inhalt des zweiten Operanden ohne Veränderung oder Verschiebung in den ersten Operanden gebracht.

Um das gleiche Ziel für 2 Felder im Hauptspeicher zu erreichen, muß die Operation _____ benutzt werden.

Übertragen Zeichen/MVC

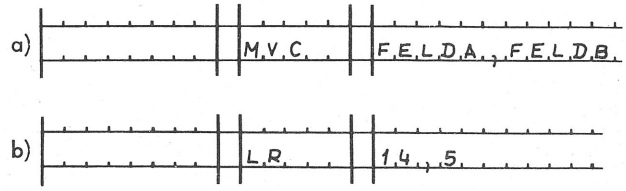
3 Die gleiche Funktion heißt "Übertragen", wenn es sich um Felder im Hauptspeicher mit variabler Länge handelt, und "Um-Laden", wenn es sich um _____ handelt.

Register

4 Schreiben Sie die Instruktionen für die folgenden Aufgaben:

a) Übertragen des Inhaltes von FELDB im Hauptspeicher nach FELDA im Hauptspeicher

b) Laden des Inhaltes von Register 5 nach Register 14.



Übungen

Codieren Sie bitte folgende Befehle:

a) Der Inhalt des Registers 5 soll in die Register 7, 9 und 10 geladen werden.

b) Der Inhalt der Register 13 und 14 soll durch den Inhalt des Registers 8 ersetzt werden.

	Name (1–8)	Operation (10–14)	Op (20)
a)		L,R,	7,,5,
		L,R,	9,,5,
		L,R,	1,0,,5,
b)		L,R,	1,3,,8,
		L,R,	1,4,,8,

L

LADEN

(Load full-word)

Mit dieser Operation wird der Inhalt eines Wortes im Hauptspeicher in ein allgemeines Register geladen.

Beispiel einer Instruktion:

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	L	8,,WORTA

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel WORTA) wird in den ersten Operanden (im Beispiel Register 8) geladen.

Beide Operanden enthalten Dualzahlen. Der zweite Operand ist immer ein Wort im Hauptspeicher und der erste Operand ein allgemeines Register.

Der Bedingungsschlüssel wird nicht verändert.

Durchführung der Operation:

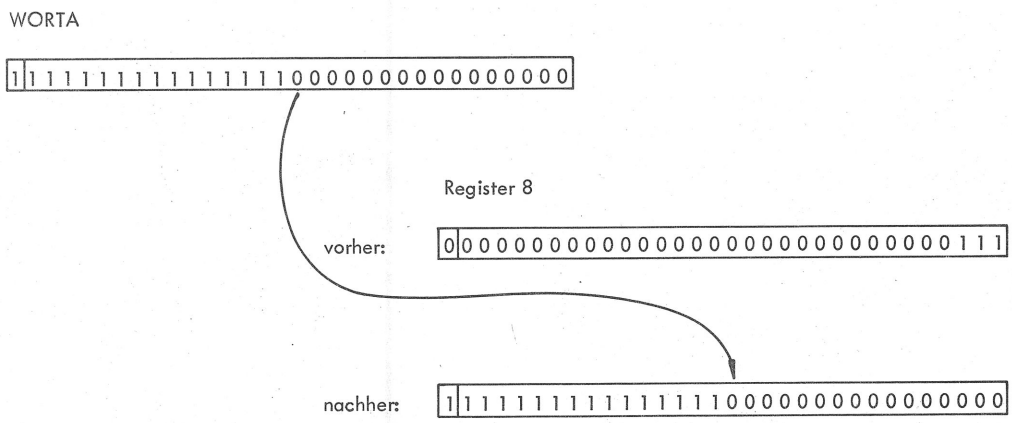


Abb. 1

1 Mit der Operation "Laden" werden Dualdaten aus ___(WW)___ in ___(WW)___ geladen.

einem Wort im Hauptspeicher/dem Hauptspeicher

ein Register

2 "Laden" hat den symbolischen Operationsschlüssel _____.

"Umladen" dagegen hat den Schlüssel _____.

L

LR

3 Der Unterschied zwischen beiden Operationen liegt beim _____ Operanden.

Dieser Operand ist beim Laden ein ___(WW)___, beim Umladen dagegen ein _____.

zweiten

Wort im Hauptspeicher

Register

4 Im übrigen sind die Operationen "Laden" und "Umladen" vollkommen gleich.

So bleibt z.B. der Inhalt des zweiten Operanden bei beiden Operationen _____.

Ein Überlauf kann bei beiden Operationen ___(WW)___.

unverändert

nicht auftreten

L

LADEN

(Load full-word)

Mit dieser Operation wird der Inhalt eines Wortes im Hauptspeicher in ein allgemeines Register geladen.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	L	8,,WORTA

Wirkungsweise:

Der Inhalt des zweiten Operanden (im Beispiel WORTA) wird in den ersten Operanden (im Beispiel Register 8) geladen.

Beide Operanden enthalten Dualzahlen. Der zweite Operand ist immer ein Wort im Hauptspeicher und der erste Operand ein allgemeines Register.

Der Bedingungsschlüssel wird nicht verändert.

Durchführung der Operation:

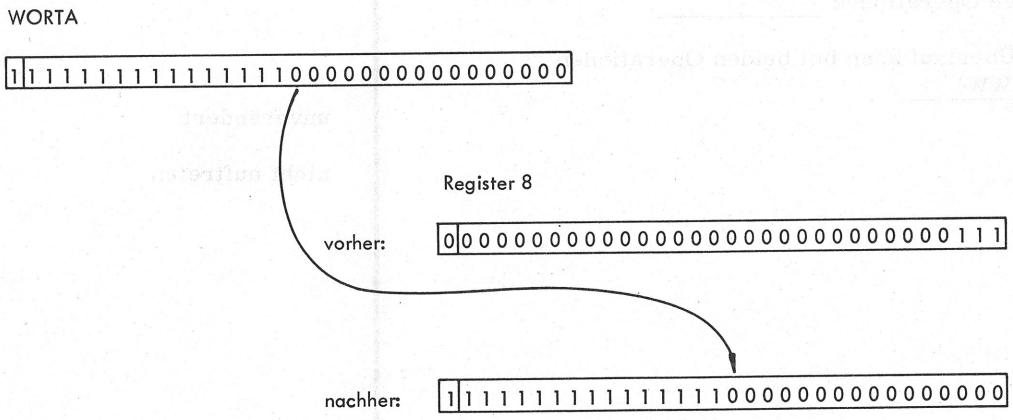


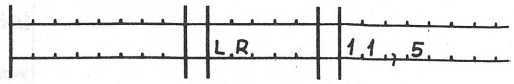
Abb. 1

5 Codieren Sie die folgenden Befehle:

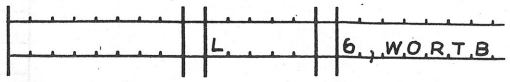
a) Der Inhalt von Register 5 soll nach Register 11 geladen werden.

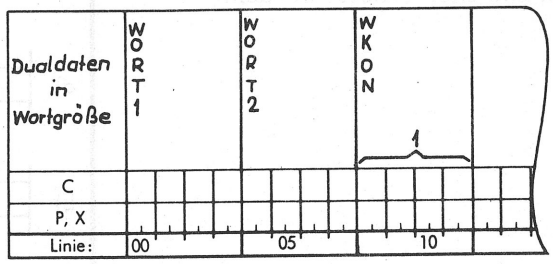
b) Der Inhalt von Register 6 soll durch den Inhalt von WORTB im Speicher ersetzt werden.

a)



b)





W.O.R.T.1. . .	D.S. . .	F. . . . .
W.O.R.T.2. . .	D.S. . .	F. . . . .
W.K.O.N. . . .	D.C. . .	F.'1.' . . . . .

Abb. 2

Übungen

In Abb. 2 sind 2 Felder für Dualdaten in Wortgröße und eine Wortkonstante definiert.

Mit Hilfe der Operation L können wir den Inhalt dieser Felder und der Konstanten in ein Register laden.

Codieren Sie nun bitte folgende Befehle:

a) Der Inhalt von WORT1 soll in das Register 10 geladen werden.

b) Der Inhalt von WORT2 soll in das Register 11 geladen werden.

c) Der Inhalt von WKON soll in das Register 12 geladen werden.

Name (1–8)	Operation (10–14)	Operand und Be (20–25)
	L	10,WORT1
	L	11,WORT2
	L	12,WKON

LA

LADEN ADRESSE

(Load Address)

Mit dieser Operation wird eine Hauptspeicheradresse in ein allgemeines Register geladen.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	L,A,	8,,E,F,E,L,D,

Wirkungsweise:

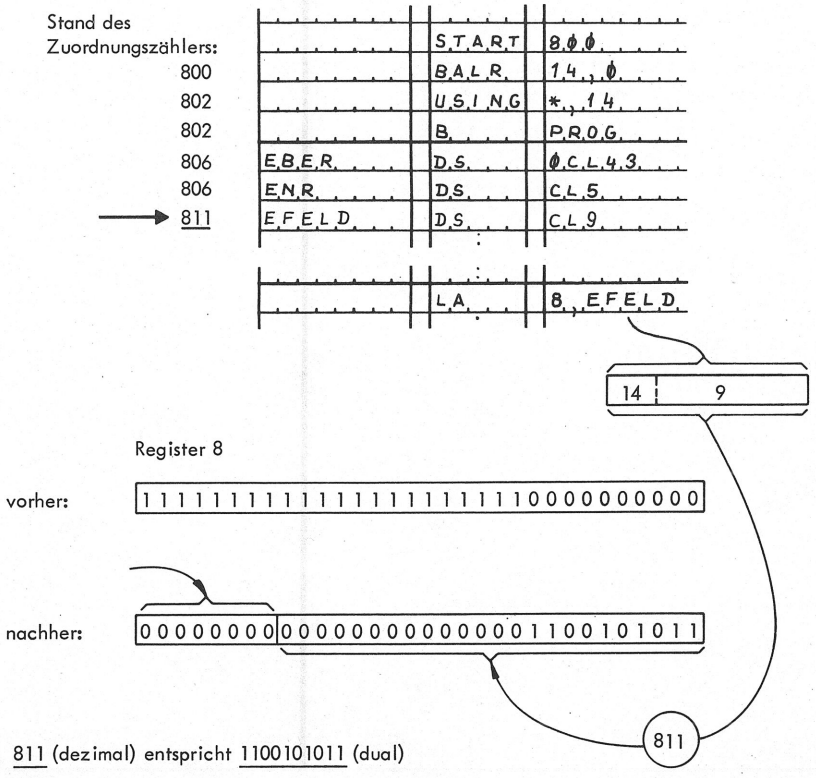
Die Adresse des zweiten Operanden (im Beispiel EFELD) wird in die rechten 24 Bits des ersten Operanden (im Beispiel Register 8) geladen. Die restlichen 8 Bits des Registers werden auf null gelöscht.

Der erste Operand ist immer ein allgemeines Register.

Die Adresse des zweiten Operanden, und zwar ihr effektiver Wert, wird in das Register geladen. Der Inhalt des zweiten Operanden wird nicht berührt.

Der Bedingungsschlüssel wird nicht verändert.

Durchführung der Operation:



811 (dezimal) entspricht 1100101011 (dual)

Abb. 1

1 Die Operation "Laden Adresse" hat zwei Operanden.

Der erste Operand ist ein _____ und der zweite Operand ein Feld im _____.

Register

Hauptspeicher

2 Die symbolische Adresse des Feldes im Hauptspeicher heißt im Beispiel der Abb. 1 _____.

Durch den Zuordnungszähler wird diesem Feld im Hauptspeicher die Adresse __(Z)__ zugeordnet.

EFELD

811

3 In das Feld EFELD werden während des Programmablaufs wechselnde Daten eingegeben. So enthält EFELD z.B. zu einem bestimmten Zeitpunkt den Namen "MUELLERßß".

Wird jetzt die Operation LA ausgeführt, so wird die __(WZ)__ nach Register 8 geladen.

Hauptspeicheradresse 811

4 Mit LA wird nie der Inhalt eines Speicherfeldes, sondern immer eine Speicheradresse geladen.

Die Adresse wird _____ verschlüsselt im Register gespeichert. Sie belegt die __(WZ)__ Bits des Registers.

dual

rechten 24

5 Bevor das Programm ausgeführt wird, ist es in echter Form im DVS gespeichert.

In der echten Form des LA-Befehls in Abb. 1 ist die symbolische Adresse EFELD ersetzt durch die Nummer des Basisregisters __(Z)__ und die relative Adresse __(Z)__.

14

9 (Inhalt des Basisregisters 14 ist 802)

LA

LADEN ADRESSE

(Load Address)

Mit dieser Operation wird eine Hauptspeicheradresse in ein allgemeines Register geladen.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	LA	8,EFELD

Wirkungsweise:

Die Adresse des zweiten Operanden (im Beispiel EFELD) wird in die rechten 24 Bits des ersten Operanden (im Beispiel Register 8) geladen. Die restlichen 8 Bits des Registers werden auf null gelöscht.

Der erste Operand ist immer ein allgemeines Register.

Die Adresse des zweiten Operanden, und zwar ihr effektiver Wert, wird in das Register geladen. Der Inhalt des zweiten Operanden wird nicht berührt.

Der Bedingungsschlüssel wird nicht verändert.

Durchführung der Operation:

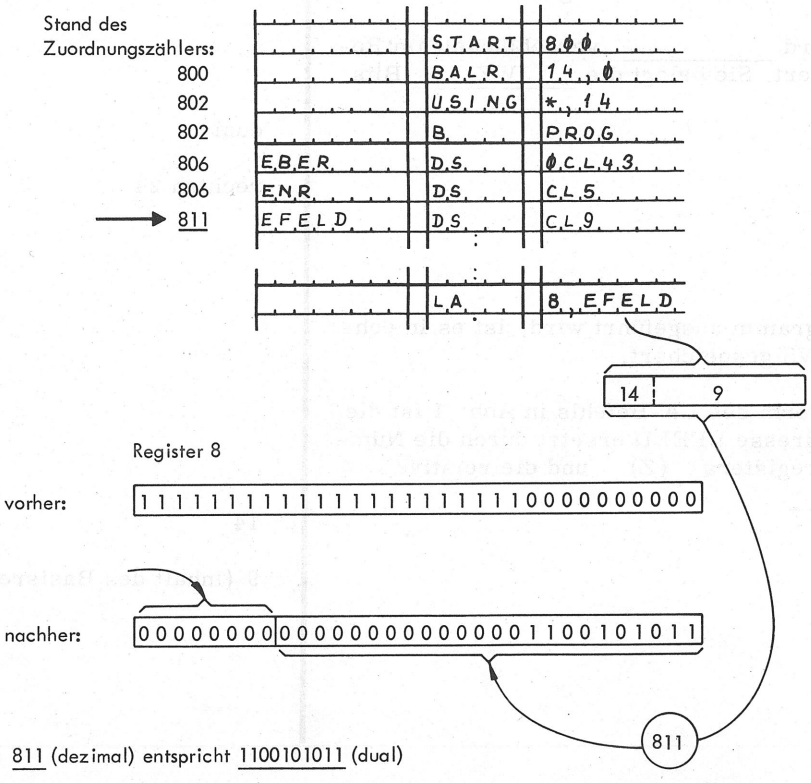


Abb. 1

6 Aus den dual verschlüsselten Adreßangaben

14	9

des zweiten Operanden wird bei der Ausführung des LA-Befehls die effektive Adresse __(Z)__ errechnet und nach Register __(Z)__ geladen.

811/0...01100101011

8

7 Eine Basisadresse belegt im Basisregister immer nur die rechten 24 Bits.

Auch für eine effektive Adresse werden nie mehr als __(ZW)__ benötigt.

Die linken 8 Bits im Register werden bei der Ausführung des LA-Befehls __(WW)__.

24 Bits

auf Null gelöscht

8 Für die folgende Aufgabe nehmen wir an, daß ein Wort im Speicher mit dem Namen WORTA dual verschlüsselt die Zahl 518 enthält. WORTA belege die Bytes 904, 905, 906 und 907.

Welche Werte (dezimal) werden durch folgende Befehle nach Register 12 geladen?

```
L     12,WORTA
LA    12,WORTA
```

518

904

LA

LADEN ADRESSE

(Load Address)

Mit dieser Operation wird eine Hauptspeicheradresse in ein allgemeines Register geladen.

Beispiel einer Instruktion:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	LA	8,EFELD

Wirkungsweise:

Die Adresse des zweiten Operanden (im Beispiel EFELD) wird in die rechten 24 Bits des ersten Operanden (im Beispiel Register 8) geladen. Die restlichen 8 Bits des Registers werden auf null gelöscht.

Der erste Operand ist immer ein allgemeines Register.

Die Adresse des zweiten Operanden, und zwar ihr effektiver Wert, wird in das Register geladen. Der Inhalt des zweiten Operanden wird nicht berührt.

Der Bedingungsschlüssel wird nicht verändert.

Durchführung der Operation:

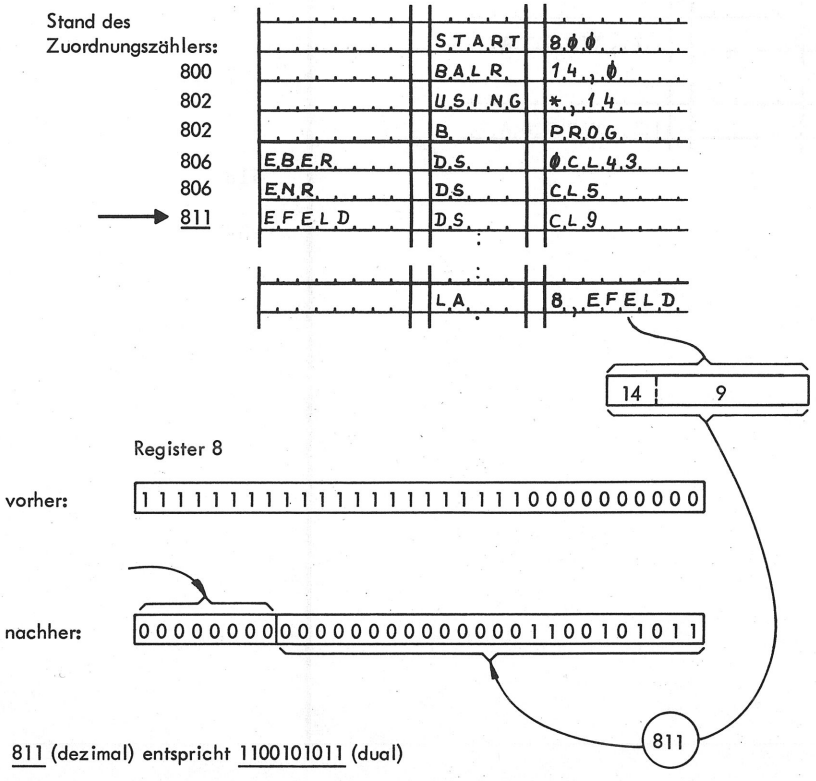
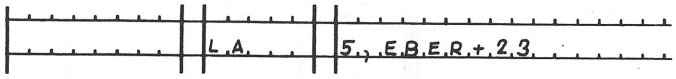


Abb. 1

9 Im zweiten Operandenfeld der LA-Instruktion kann natürlich auch ein zusammengesetzter Ausdruck stehen.

Betrachten Sie folgenden Befehl, bezogen auf das Beispiel in Abb. 1:



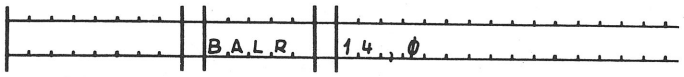
Durch diesen Befehl wird die Adresse __(Z)__ nach __(WZ)__ geladen.

829

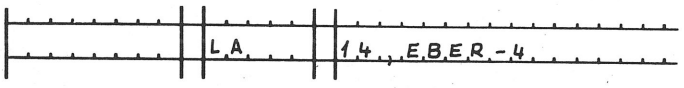
Register 5

10 Vielleicht fragen Sie sich, warum wir zum Laden des Basisregisters am Anfang eines Programms BALR und nicht LA benutzen.
Wir überlegen uns dieses Problem am Beispiel in Abb. 1.

Wir ersetzen die erste Instruktion



durch die Instruktion



Der Bereich EBER hat die Adresse __(Z)__.
"EBER -4" bezeichnet also die Folgeadresse __(Z)__, die durch BALR nach __(WZ)__ geladen wird.

806

802 (806 - 4)

Register 14

Obwohl LA und USING sich auf dieselbe Adresse 802 als Basisadresse beziehen, kann jedoch LA schon aus folgendem Grund nicht benutzt werden: Die USING-Anweisung folgt erst nach dem LA-Befehl, so daß die symbolische Adresse EBER-4 bei der Umwandlung gar nicht in die Nummer eines Basisregisters und eine relative Adresse aufgespaltet werden kann.

Wir merken uns:

Am Anfang des Programms darf keine Instruktion stehen, die sich auf eine Hauptspeicheradresse bezieht.

Zunächst muß eine Basisadresse geladen sein. Erst dann kann eine Hauptspeicheradresse mit Hilfe dieser Basisadresse gebildet werden.

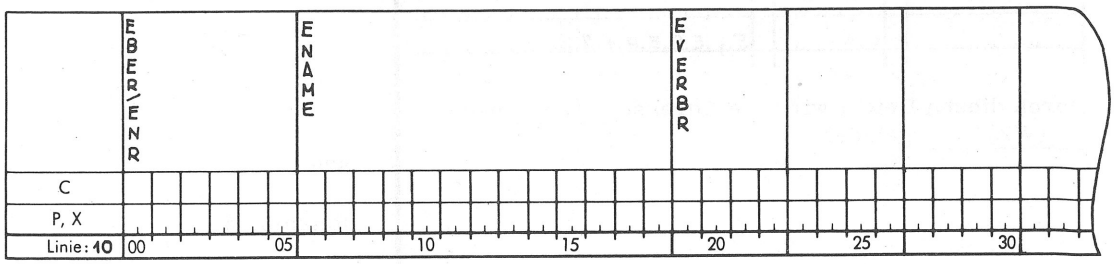


Abb. 2

Übungen

1. Anhand der Speicherbelegung in Abb. 2 sollen Sie nun selbst auf einem Programmschema einige LA-Instruktionen codieren. Benutzen Sie bitte die in Klammern angegebenen Symbole, wenn nötig mit Adreßrechnung.

   a) Die Adresse 1000 soll in das Register 13 geladen werden (EBER).

   b) Die Adresse 1027 soll in das Register 12 geladen werden (EVERBR).

   c) Die Adresse 1017 soll in das Register 9 geladen werden (EVERBR).

	Operation (10–14)	Operand und Bemerkun (20–25)
a)	L,A, . .	1,3,,,E,B,E,R, ...
b)	L,A, . .	1,2,,,E,V,E,R,B,R,+,8, . .
c)	L,A, . .	9,,,E,V,E,R,B,R,-,2, . . .

2. Welche Hauptspeicheradressen werden in welche Register geladen (vgl. Abb. 2)?

a)	. . . . .	L,A, . .	7,,,E,N,A,M,E, ...
b)	. . . . .	L,A, . .	6,,,E,B,E,R,+,1,0,0, .

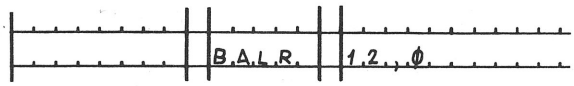
   a) Die Adresse 1006 wird in das Register 7 geladen.

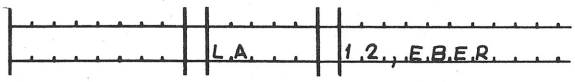
   b) Die Adresse 1100 wird in das Register 6 geladen.

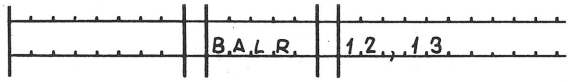
FRAGEN ZUM KAPITEL VI

Beantworten Sie bitte erst alle Fragen, ehe Sie Ihre Ergebnisse anhand der Lösungen überprüfen.

1. Welche der folgenden Instruktionen bewirkt eine Verzweigung und wohin?

a)


b)


c)


2. Was wird bei den drei obigen Instruktionen in das Register 12 geladen?

3. Für ein Programm soll Register 7 als Basisregister mit der Basisadresse 802 benutzt werden. Wie heißen die erforderlichen drei Anweisungen (einschließlich START) zu Beginn des Programmes?

4. Welche Operation steht Ihnen zur Verfügung

   a) zur Addition zweier gepackter Dezimalzahlen,

   b) zur Addition zweier Dualzahlen in Registern?

5. Eine Dualzahl aus einem Wort im Speicher (symbolische Adresse WORT) soll

   a) in das Speicherfeld ZWORT,

   b) in das Register 11

   gebracht werden. Wie heißen die beiden Instruktionen dazu?

6. Codieren Sie folgende Instruktionen:

   a) Der Inhalt des Wortes WORT1 im Speicher soll zum Inhalt des Registers 2 addiert werden.

   b) Der Inhalt von Register 3 soll zum Inhalt von Register 4 addiert werden.

   c) Die Dualzahl in Register 5 soll durch die Dualzahl in Register 6 ersetzt werden.

   d) Die Halbwortkonstante HWKON im Speicher soll zum Inhalt des Registers 7 addiert werden.

   e) Die Adresse von EBER soll in das Register 8 geladen werden.

   f) Der Wert 440 der Wortkonstanten WORTKON im Speicher soll in das Register 9 geladen werden.

ANTWORTEN ZUM KAPITEL VI

Überprüfen Sie nun bitte Ihre Ergebnisse und beseitigen Sie eventuell aufgetretene Unklarheiten.

Wiederholen Sie gegebenfalls den in Klammern angegebenen Lernstoff.

1. Instruktion c) bewirkt eine Verzweigung zu der Hauptspeicheradresse, die in Register 13 steht.

(Seite 432)

2. a) die Folgeadresse (Seite 432)

   b) die Adresse von EBER (Seite 486)

   c) die Folgeadresse (Seite 432)

3.

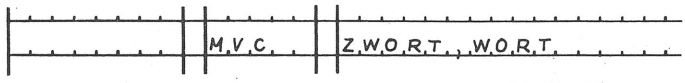
Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	S,T,A,R,T	B,Ø,Ø,
	B,A,L,R,	7,,Ø,
	U,S,I,N,G	*,,7,

(Seite 441)

4. a) AP / Addieren Dezimal (Gepackt) (Seite 184)

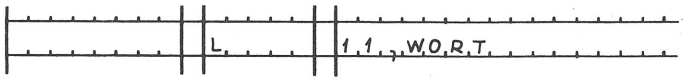
   b) AR / Addieren Register (Seite 472)

5. a)



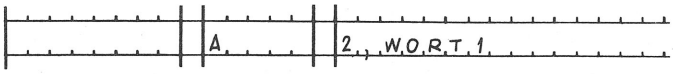
(Seite 8)

b)



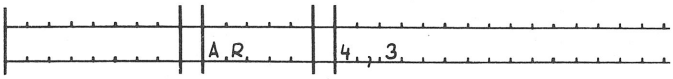
(Seite 480)

6. a)



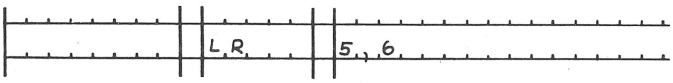
(Seite 460)

b)



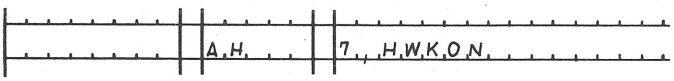
(Seite 472)

c)



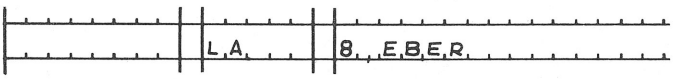
(Seite 476)

d)



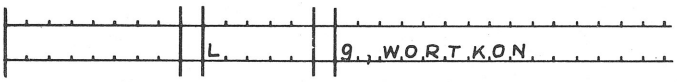
(Seite 466)

e)



(Seite 486)

f)



(Seite 480)

Entnehmen Sie nun Ihren Arbeitsmitteln das Merkblatt 421.
In diesem Merkblatt sind die Instruktionen dieses Kapitels zusammenfassend dargestellt.



Band 2, Seite 217 (Kapitel VI)

ANHANG ZU DEN INSTRUKTIONEN

DIE ZUORDNUNG DER INSTRUKTIONEN ZU DEN INSTRUKTIONSFORMATEN

Entnehmen Sie bitte Ihren Arbeitsmitteln Übung 426 und Merkblatt 207. -

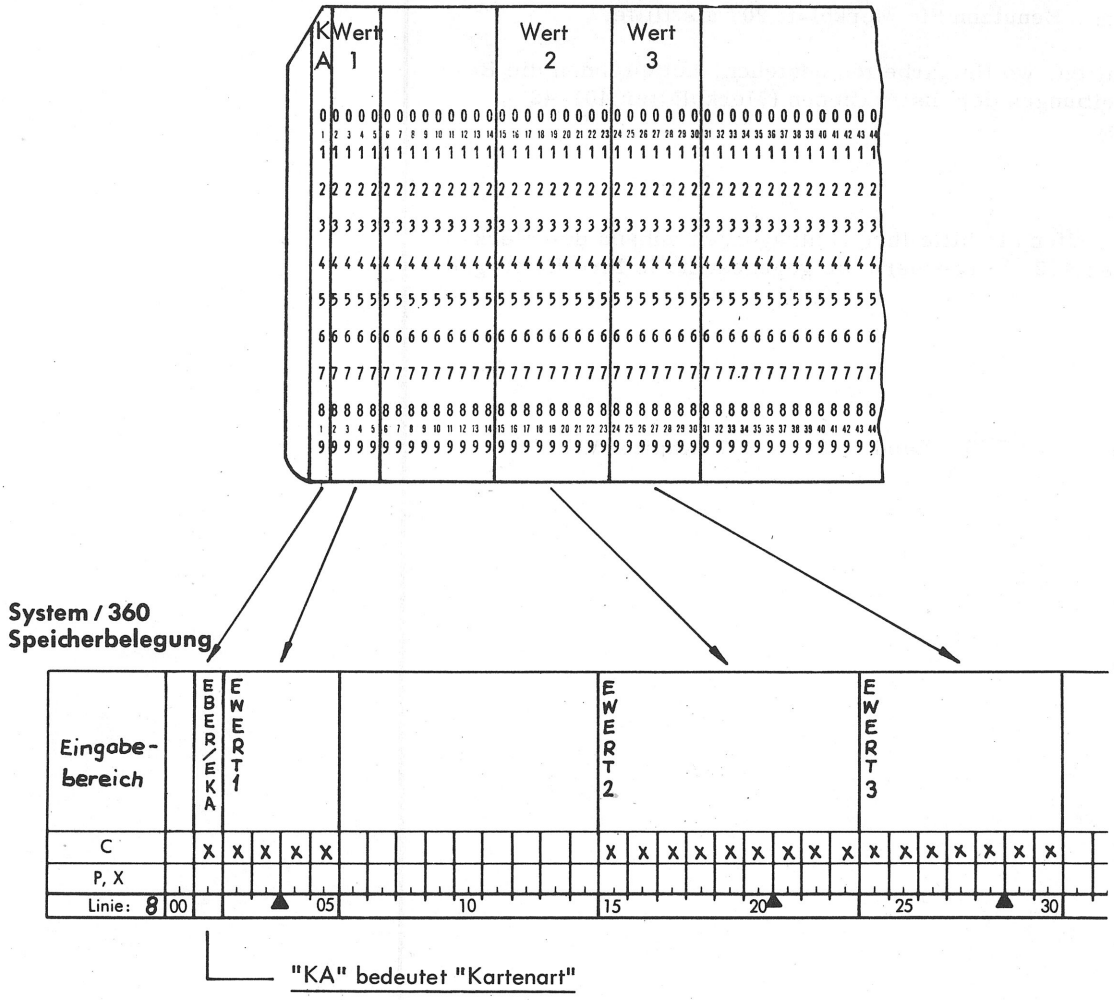
Ordnen Sie den in der Übung alphabetisch angeordneten Operationen die entsprechenden Formate (SS, SI, RX oder RR) zu. Benutzen Sie Merkblatt 207 als Hilfe.

In Fällen, wo Unklarheiten bestehen, helfen Ihnen die Beschreibungen der Instruktionen (Merkblätter 401-421) weiter.

Überprüfen Sie bitte Ihre Eintragungen anhand des Merkblattes 422. Verbessern Sie gegebenenfalls Ihre Eintragungen.



Band 3, Seite 307 (Kapitel VI)



Zur Problemstellung: bei KA "1": Addition von WERT1 und WERT3

bei KA "2": Multiplikation von WERT2 mit einer Konstanten "65".

Beide Ergebnisse sollen gedruckt werden.

Abb. 1

FRAGEN ZUM BAND 4

Beantworten Sie bitte erst alle Fragen, ehe Sie Ihre Antworten überprüfen.

Betrachten Sie Abb. 1.
Mit den Zahlen Wert 1, Wert 2 und Wert 3 soll gerechnet werden.
Die Ergebnisse sollen in Listen gedruckt werden.
Nach der Eingabe stehen alle Daten ungepackt im Hauptspeicher.

1. Wert 1, Wert 2 und Wert 3 müssen vor dem Rechnen gepackt werden.
   Schreiben Sie die Instruktionen zum Packen der Werte in die Eingabefelder (ohne Adreßrechnung und explizite Längenangabe).

2. Wie viele Bytes enthalten nach dem Packen immer führende Nullen

   a) bei EWERT1

   b) bei EWERT2

   c) bei EWERT3 ?

3. Schreiben Sie die 3 PACK-Instruktionen noch einmal. Verwenden Sie Adreßrechnung und explizite Längenangabe so, daß in die entsprechenden linken Bytes keine führenden Nullen eingesteuert werden.

4. Nach dem Packen soll eine Addition oder Multiplikation durchgeführt werden, je nach der Kartenart (vgl. Abb. 1). Zum Prüfen der Kartenart soll die Konstante KON1 benutzt werden.

K.O.N.1.	D.C.	C.'1.'

   Wie lautet der symbolische Vergleichsbefehl? (Das Vergleichsergebnis soll sich auf die Kartenart beziehen.)

5. Welche Voraussetzung müßten die Kartenart und die Konstante erfüllen, damit der Vergleich mit der Operation "Vergleichen Dezimal (Gepackt)" durchgeführt werden könnte?

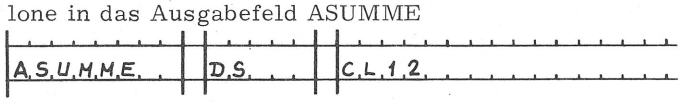
6. Bei dem Vergleichsergebnis "gleich" soll zu der Additions-Instruktion mit dem Namen A1 verzweigt werden.
Wie heißt der Verzweig-Befehl?

7. Schreiben Sie den Additions-Befehl (Wert 1 + Wert 3) mit dem Namen A1.
Das Ergebnis wird nicht größer als 7 Ziffernstellen und soll im Feld EWERT3 gespeichert werden.
Schließen Sie alle überflüssigen Bytes durch Adreßrechnung und explizite Längenangabe von der Operation aus. (Skizzieren Sie sich evtl. den Inhalt der beiden Felder vor und nach der Operation.)

8. Haben die 4 rechten Bytes von EWERT3 nach der Addition z.B. den negativen Inhalt "0283517D", dann soll die Summe folgendes Druckbild erhalten:
2.835,17 -
Führende Nullen sollen bis zur 2. Stelle links vom Komma einschließlich unterdrückt werden.
Wie lauten

   a) die Definition der Schablone SCHAB

   b) die Instruktion zur Übertragung der Schablone in das Ausgabefeld ASUMME

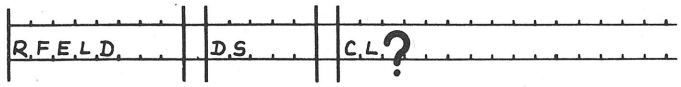


   c) die ED-Instruktion?

9. Bei der Kartenart "2" soll der gepackte Inhalt von EWERT2 mit der Konstanten "65" multipliziert werden.

   a) Wie lautet die Definition dieser Konstanten mit dem Namen KON65 ?

   b) Die unvollständige Definition des Rechenfeldes für die Multiplikation sei:

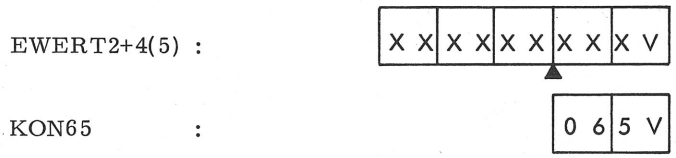


   Die fragliche Länge des Rechenfeldes in Bytes ergibt sich für dieses Beispiel durch die Rechnung:

   (Z) + (Z) = (Z)

   c) Wie lauten die beiden Befehle für die Multiplikation von EWERT2 und KON65 in RFELD ?
   (In RFELD stehe das Ergebnis einer vorhergehenden Multiplikation.)

   d) Multiplikand und Multiplikator haben folgenden Inhalt:

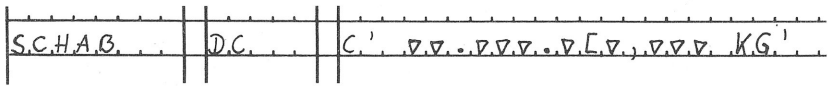


   Welchen Inhalt hat RFELD nach der Multiplikation?

10. Das (immer positive) Produkt in RFELD soll zum Drucken aufbereitet werden.

    Wieviel Ziffern muß die Schablone aufnehmen können?

11. Die Schablone zur Ausgabe von RFELD sei wie folgt definiert:



    RFELD enthalte das Produkt "00 00 00 00 36 7C".

    Wie heißen die nötigen MVC- und ED-Instruktionen, damit dieses Produkt im Ausgabefeld APROD (Länge 18 Bytes) aufbereitet und wie folgt ausgedruckt wird:

    0,367 KG

12. Zu den Instruktionen für duale Daten:

    Nach Register 7 soll die Adresse eines Feldes mit dem Namen ENR geladen werden. Anschließend soll diese Adresse im Register 7 um 300 erhöht werden.

    Schreiben Sie unter Verwendung

    a) einer Halbwortkonstanten (KHW300)

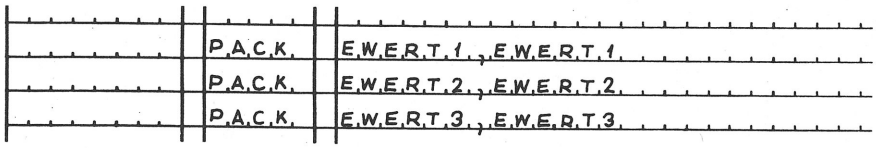
    b) einer Wortkonstanten (KW300)

    die Definition der Konstanten 300 und die beiden erforderlichen Instruktionen.

ANTWORTEN ZUM BAND 4

Wenn Sie Fehler gemacht haben, sollten Sie zur Beseitigung der Unklarheiten zum Zwischentest des betreffenden Kapitels (in Klammern) zurückschlagen. Kommt es auch dort zu Fehlern, so sollten Sie den betreffenden Lernstoff wiederholen.

1.

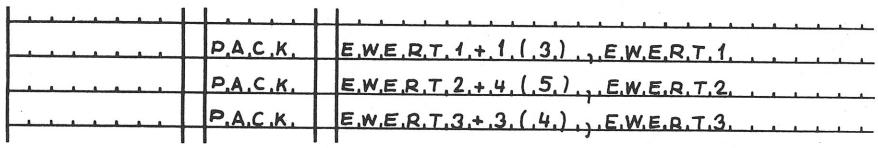


(Fragen zum Kapitel III, Seite 237 ff.)

2. a) 1 Byte
   b) 4 Bytes
   c) 3 Bytes

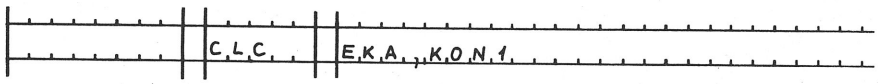
(Fragen zum Kapitel III, Seite 237 ff.)

3.



(Fragen zum Kapitel III, Seite 237 ff.)

4.

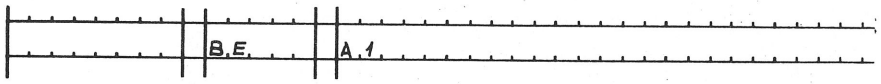


(Fragen zum Kapitel II, Seite 141 ff.)

5. Sinngemäß: Beide müssen gepackte Darstellungsform haben.

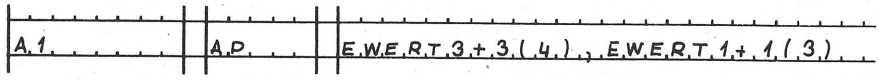
(Fragen zum Kapitel IV, Seite 349 ff.)

6.



(Fragen zum Kapitel II, Seite 141 ff.)

7.



(Fragen zum Kapitel III, Seite 237 ff.)

8. a)

S.C.H.A.B.	D.C.	C.' ∇∇.∇C∇,∇∇ -'

b)

	M.V.C.	A.S.U.M.M.E., S.C.H.A.B.

c)

	E.D.	A.S.U.M.M.E., E.W.E.R.T.3.+3.

(Übungen, Seite 421 ff.)

9. a)

K.O.N.6.5.	D.C.	P.'6.5.'
oder:		
K.O.N.6.5.	D.C.	x.'6.5.F.'

(statt "F" auch "C")

b) 5 + 2 = 7

c)

	Z.A.P.	R.F.E.L.D., E.W.E.R.T.2.+4.(5.)
	M.P.	R.F.E.L.D., K.O.N.6.5.

d) RFELD:

0 0	X X	X X	X X	X X	X X	X V

(Fragen zum Kapitel VI, Seite 495 ff.)

10. 11 Ziffern (das linke Byte von RFELD enthält immer 2 führende Nullen, die nicht mit aufbereitet zu werden brauchen).

(Übungen, Seite 421 ff.)

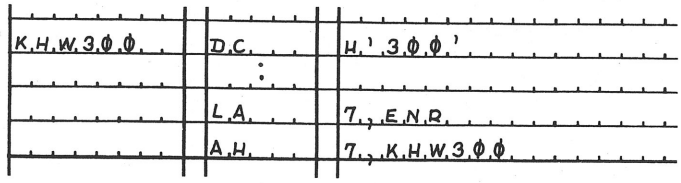
11.

	M.V.C.	A.P.R.O.D., S.C.H.A.B.
	E.D.	A.P.R.O.D.(1.5.), R.F.E.L.D.+1.

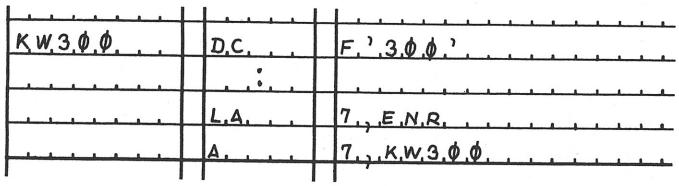
(um "KG"
zu erhalten)

(Übungen, Seite 421 ff.)

12. a)



b)



(Fragen zum Kapitel VI, Seite 495 ff.)

Wenden Sie sich jetzt noch einmal dem Band 1 (Aufgabenband) zu.

Arbeiten Sie abschließend die Seiten 421 bis 497 durch. An einigen Stellen finden Sie dort Verweise zu den Bänden 2, 3 und 4. Beachten Sie diese Verweise bitte nicht, da Ihnen der betreffende Lernstoff schon bekannt ist.

