Programmieren in der IBM/360 ASSEMBLER-Sprache

IBM

ASS-PU
Text 2
Programmierte Unterweisung

IBM Schule für Datenverarbeitung

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

Eine Programmierte Unterweisung

Text 2

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:	Der Hauptspeicher des Datenverarbeitungssystems (Einführung)	5
Kapitel II:	Die Verschlüsselung der Zeichen (Buchstaben, Ziffern und Sonderzeichen) im EBCD-Code	33
Kapitel III:	Die Darstellung dezimaler Zahlen im Hauptspeicher	99
Kapitel IV:	Die Dualdaten und ihre Formate	139
Kapitel V:	Die Adressierung des Hauptspeichers	185
Kapitel VI:	Instruktionsformate	217

KAPITEL I

DER HAUPTSPEICHER DES DATENVERARBEITUNGSSYSTEMS

(Einführung)

Lernstoff:

Die Eingabe- und Ausgabebereiche

Der Aufbau des Hauptspeichers

DIE EINHEITEN EINES DATENVERARBEITUNGSSYSTEMS

(Kartensystem)

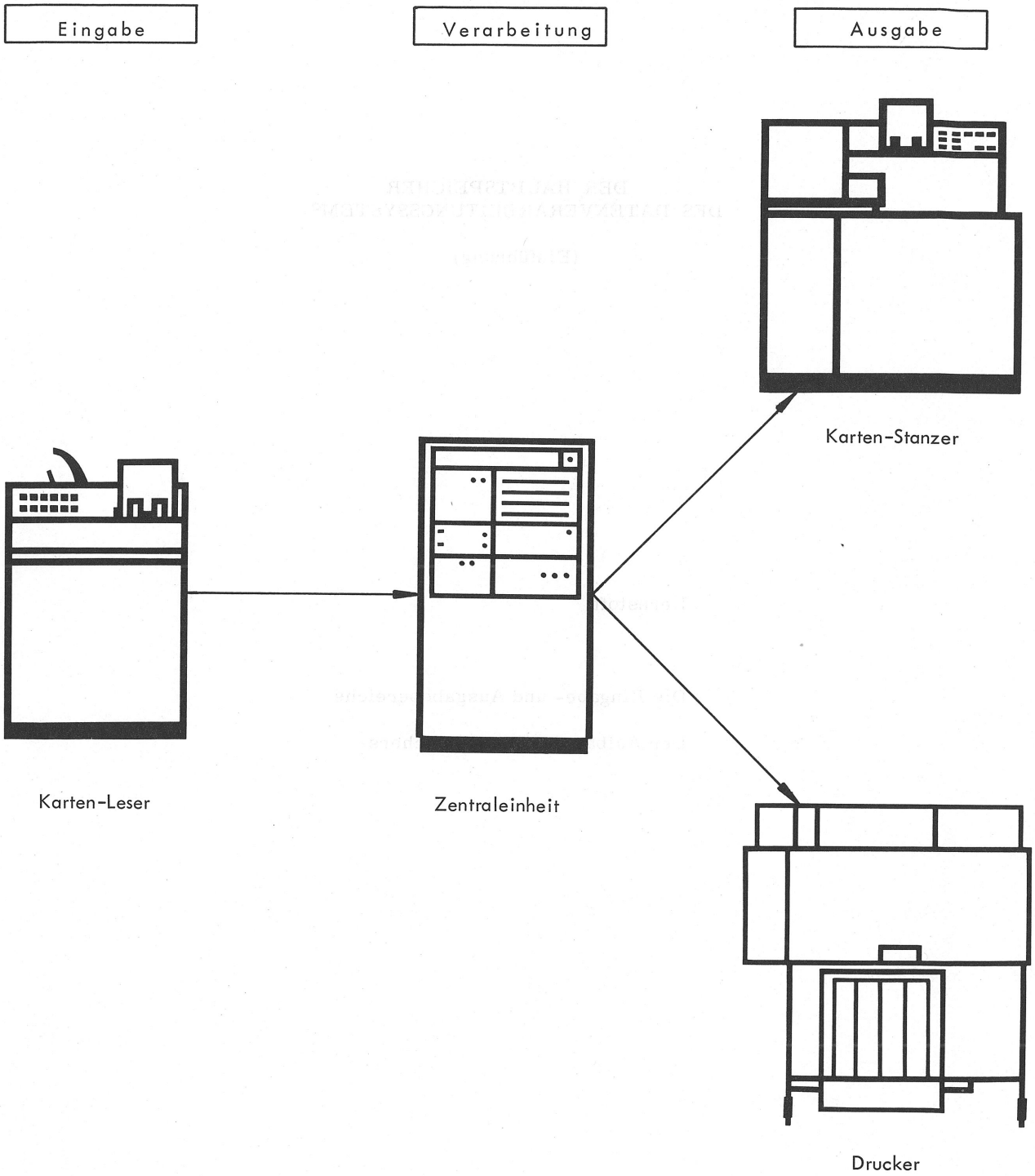


Abb. 1

Dieses Kapitel enthält als erstes einen Überblick über den Aufbau eines Datenverarbeitungssystems und über die allgemeinen Funktionen des Hauptspeichers. Dieser Abschnitt ist für die meisten von Ihnen im wesentlichen eine Wiederholung schon bekannten Stoffes.

Weiterhin wird in diesem Kapitel dargelegt, wie die Ein- und Ausgabedaten in Bereichen im Hauptspeicher gespeichert werden und wie es möglich ist, die Daten im Hauptspeicher aufzufinden.

Ein Datenverarbeitungssystem (DVS) besteht aus mehreren Einheiten, die in der Bewältigung eines Problemes zusammenwirken. Die Ausstattung eines Systems, die Zusammensetzung seiner Einheiten also, ist abhängig von den Aufgaben und Problemen, die zu lösen sind.

So kann z. B. ein DVS die vier Einheiten umfassen, die in Abb. 1 dargestellt sind: Karten-Leser, Zentraleinheit, Karten-Stanzer und Drucker.

Bei der Verarbeitung von Daten werden die Stufen Eingabe, Verarbeitung, Ausgabe durchlaufen. Dabei fällt jeder Einheit im DVS eine bestimmte Funktion zu (Abb. 1):

Karten-Leser	——	Eingabe
Zentraleinheit	——	Verarbeitung
Karten-Stanzer	\	Ausgabe
Drucker	/	

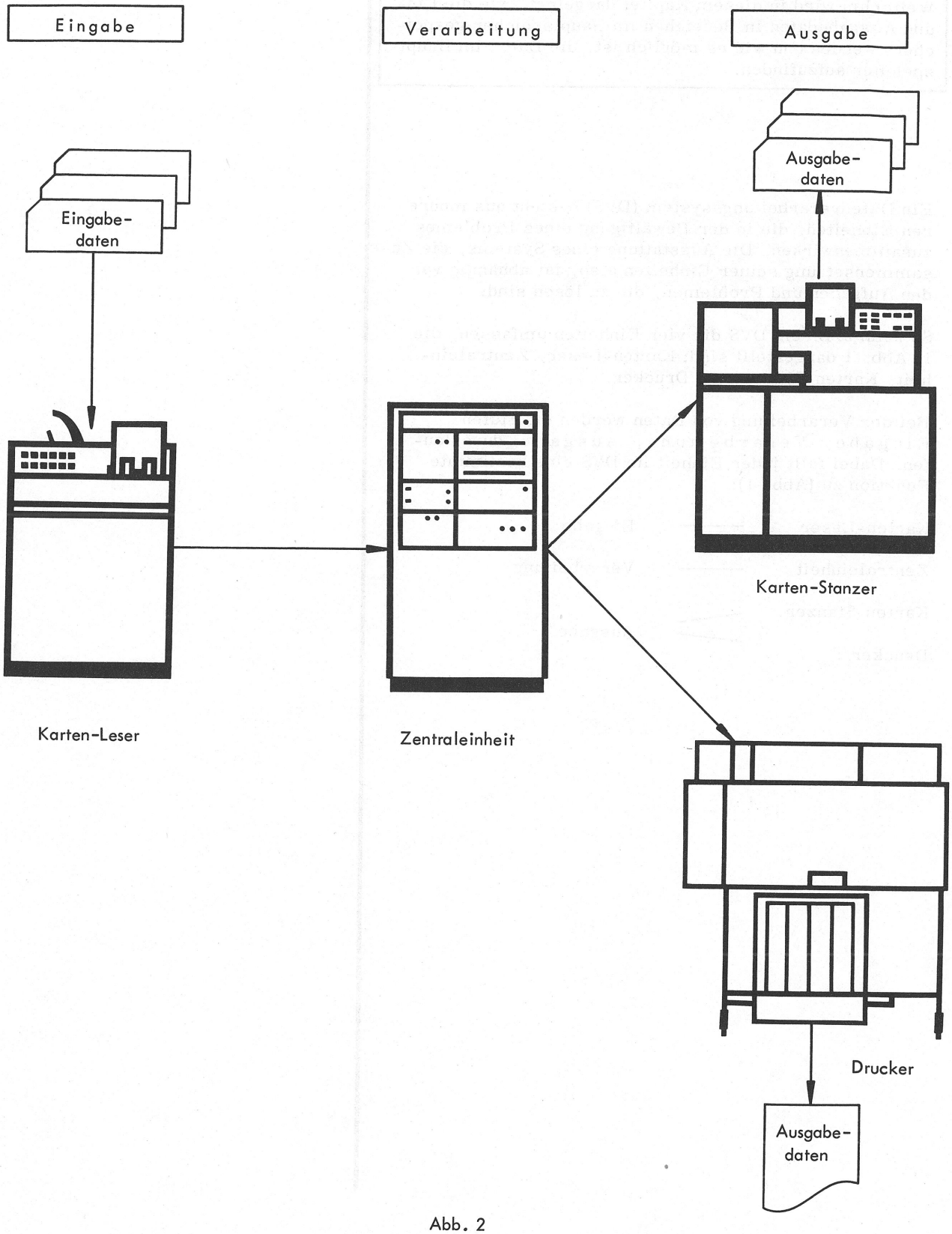

Eingabe
Verarbeitung
Ausgabe
Eingabe-
daten
Ausgabe-
daten
Karten-Stanzer
Karten-Leser
Zentraleinheit
Drucker
Ausgabe-
daten

Abb. 2

Betrachten Sie bitte Abb. 2. -

Die Abbildung zeigt den Datenfluß in dem dargestellten System. Wir verfolgen den Weg der Eingabedaten.

Die Eingabekarten werden vom Karten-Leser gelesen, und die Daten gelangen direkt in die Zentraleinheit. In der Zentraleinheit erfolgt die Verarbeitung der eingegebenen Daten zu den gewünschten Ausgabedaten. Diese können sowohl in Lochkarten gestanzt als auch auf Formulare oder Listen gedruckt werden. Zum Stanzen und/oder Drucken werden die Ausgabedaten von der Zentraleinheit zum Karten-Stanzer und/oder Drucker geleitet.

Wir haben uns kurz mit dem Aufbau eines DVS und mit dem Datenfluß in dem System befaßt. Doch nun tun wir einen Blick in das Innere der Zentraleinheit.

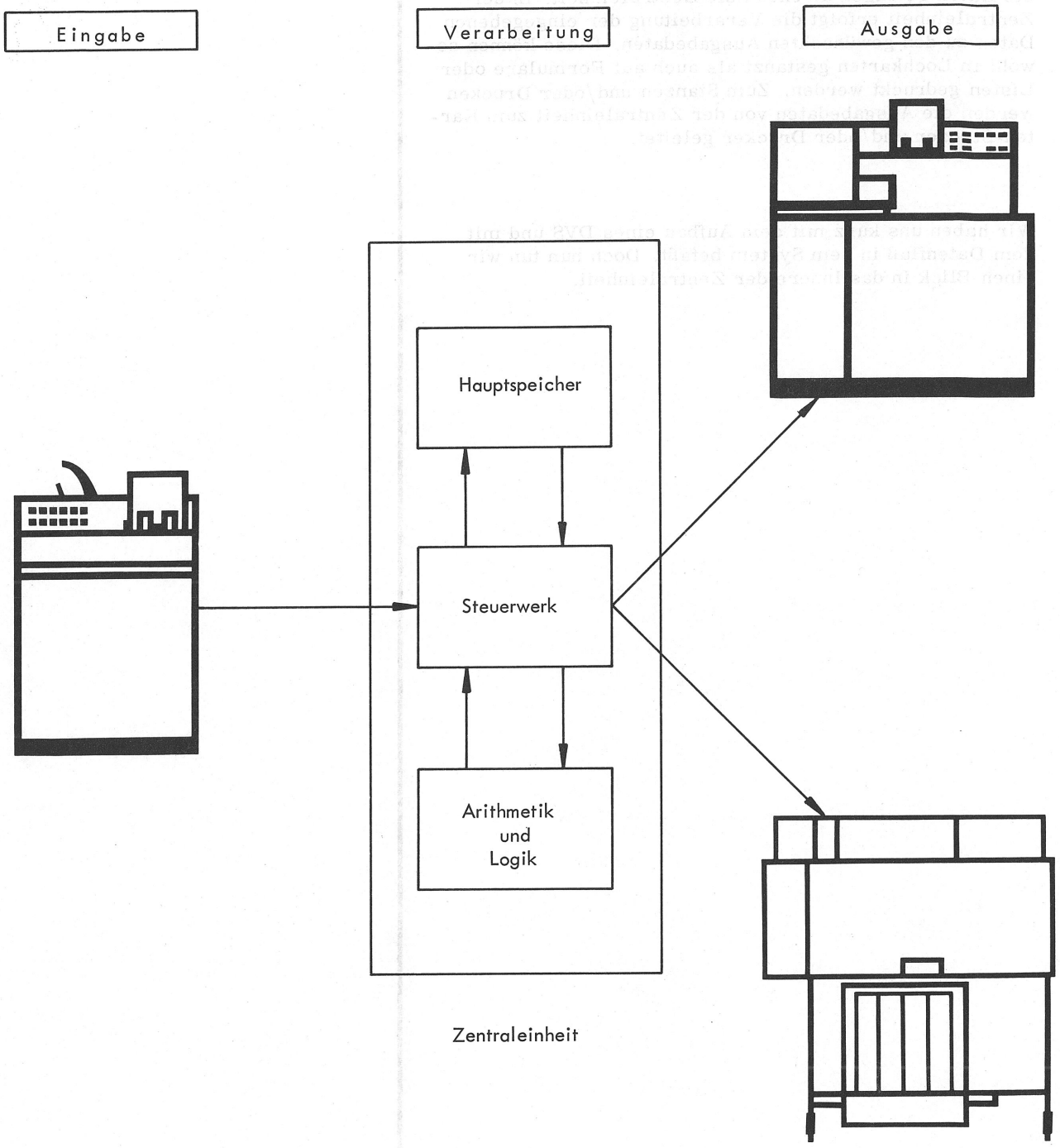

Eingabe
Verarbeitung
Ausgabe
Hauptspeicher
Steuerwerk
Arithmetik
und
Logik
Zentraleinheit

Abb. 3

Betrachten Sie bitte Abb. 3. -

Sie sehen, die Zentraleinheit setzt sich aus drei Elementen zusammen, aus

a) Hauptspeicher,

b) Steuerwerk,

c) Arithmetik und Logik.

Zwischen der Eingabe, den Elementen der Zentraleinheit und der Ausgabe bewegt sich der Informationsfluß.

Wir befassen uns nun mit demjenigen Element der Zentraleinheit, dessen Aufgabe es ist, die Daten im System aufzubewahren (zu speichern). Das ist der Hauptspeicher.

Ein Speicher kann Daten aufnehmen, aufbewahren und abgeben. Dementsprechend hat der Hauptspeicher in der Zentraleinheit eines DVS die Aufgabe:

a) die Eingabedaten über die Eingabe-Einheit aufzunehmen,

b) die Eingabedaten während der Verarbeitung aufzubewahren (zu speichern),

c) die verarbeiteten Daten an die Ausgabe-Einheit abzugeben.

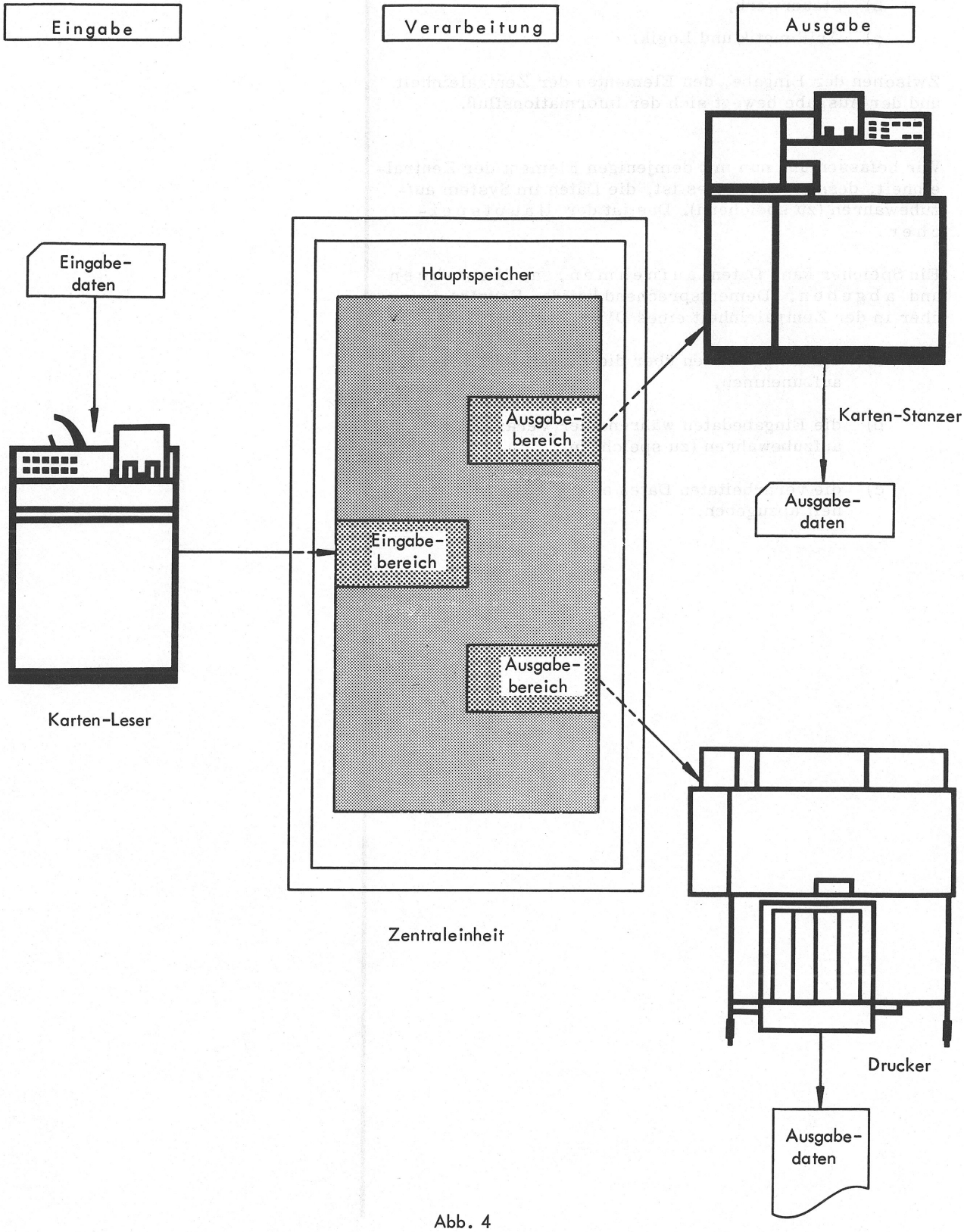

Eingabe
Verarbeitung
Ausgabe
Eingabe-daten
Hauptspeicher
Ausgabe-bereich
Karten-Stanzer
Ausgabe-daten
Eingabe-bereich
Ausgabe-bereich
Karten-Leser
Zentraleinheit
Drucker
Ausgabe-daten

Abb. 4

Abb. 4 veranschaulicht den Datenfluß zwischen Eingabeeinheit und Hauptspeicher sowie zwischen Hauptspeicher und Ausgabeeinheit.

Anhand der Abb. 4 verfolgen wir den Weg der Daten von der Eingabe bis zur Ausgabe in Lochkarte und auf Liste.

Die Eingabedaten werden über den Karten-Leser in den Eingabebereich des Hauptspeichers eingelesen. Im Eingabebereich stehen die Daten für die Verarbeitung zur Verfügung. Sind die Ausgabedaten nach den Anweisungen des Programmes erstellt, dann stehen sie in den Ausgabebereichen für den Karten-Stanzer und den Drucker bereit.

1 Für die Ein-/Ausgabebereiche sind im Hauptspeicher keine festen Plätze vorgesehen.

Die Speicherbereiche sind frei wählbar.
Der Programmierer bestimmt also, wo die Bereiche reserviert werden und wie groß sie sind.

Wir merken uns:
Die Ein-/Ausgabebereiche sind ___(WW)___.

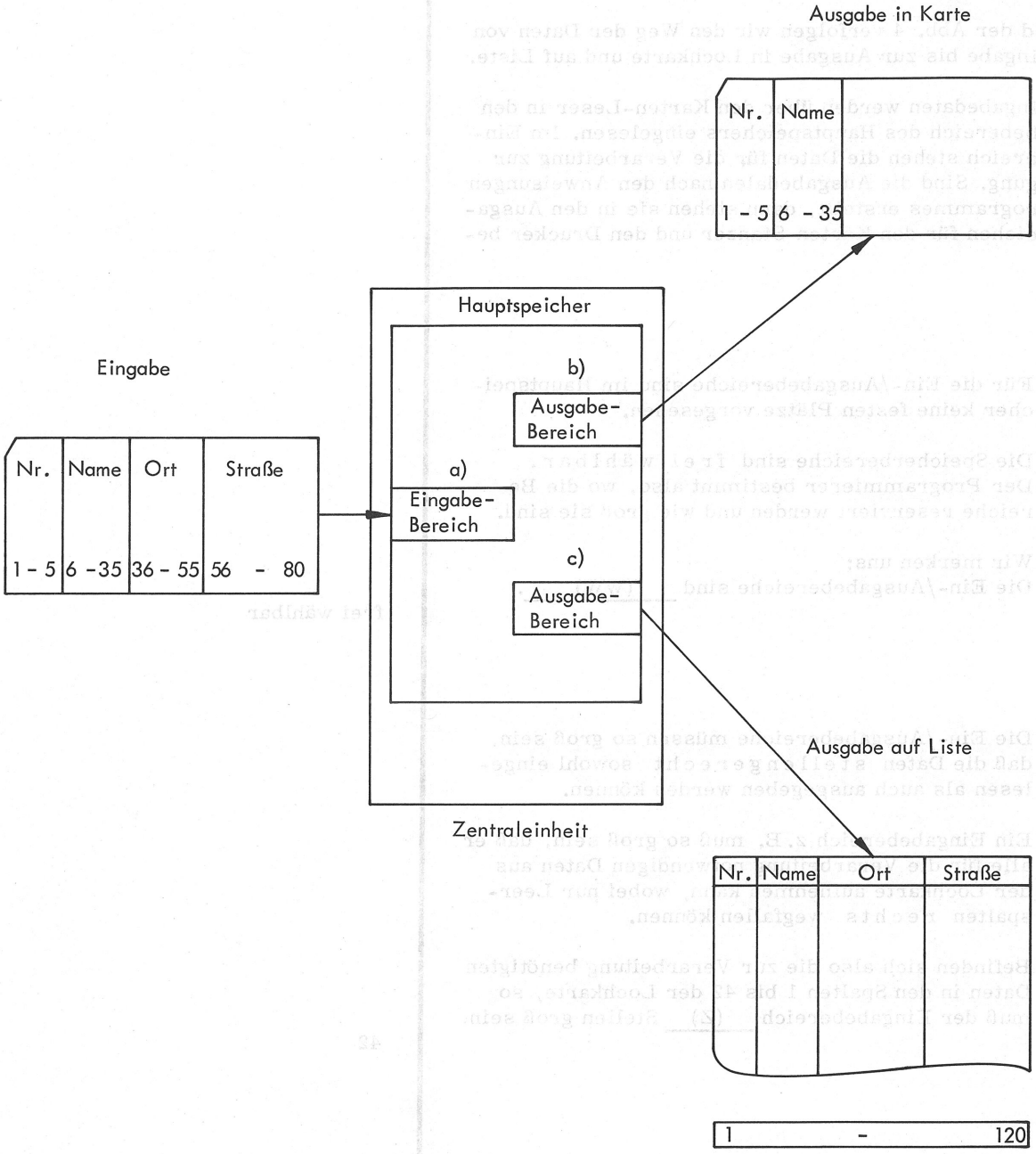
frei wählbar

2 Die Ein-/Ausgabebereiche müssen so groß sein, daß die Daten stellengerecht sowohl eingelesen als auch ausgegeben werden können.

Ein Eingabebereich z. B. muß so groß sein, daß er alle für die Verarbeitung notwendigen Daten aus der Lochkarte aufnehmen kann, wobei nur Leerspalten rechts wegfallen können.

Befinden sich also die zur Verarbeitung benötigten Daten in den Spalten 1 bis 42 der Lochkarte, so muß der Eingabebereich ___(Z)___ Stellen groß sein.

42



a) Von einer Lochkarte sollen 80 Spalten eingelesen werden.
Ausgegeben sollen werden:

b) 35 Spalten in Karte

c) 120 Druckstellen auf Liste

Abb. 5

3 Versuchen Sie die Größe der Ein-/Ausgabebereiche für die in Abb. 5 dargestellte Aufgabe zu ermitteln.

Die einzelnen Bereiche sind:

a) __(Z)__ Stellen groß,

b) __(Z)__ Stellen groß,

c) __(Z)__ Stellen groß.

a) 80

b) 35

c) 120

4 Wie Abb. 5 zeigt, sind bei der Ausgabe-Karte nur die ersten 35 Spalten zu stanzen.

Der Ausgabebereich braucht also auch nur __(Z)__ Speicherstellen zu umfassen. Damit ist genügend Platz für die Daten der Stanzer-Ausgabe reserviert.

35

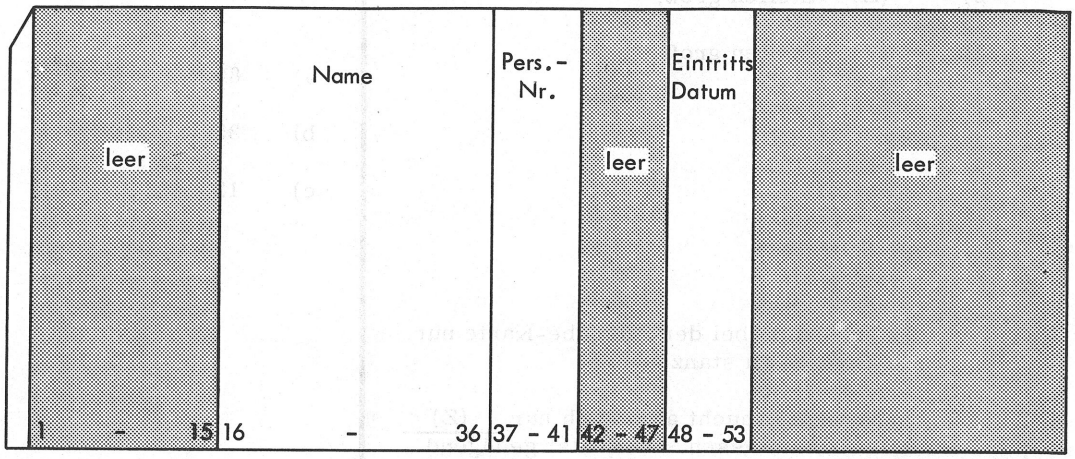

leer
Name
Pers.-Nr.
leer
Eintritts Datum
leer
1 - 15
16 - 36
37 - 41
42 - 47
48 - 53

Abb. 6

5 Sehen Sie sich bitte Abb. 6 an. -

Nur die rechts in der Lochkarte nicht benötigten Spalten bleiben für die Ein-/Ausgabebereiche unberücksichtigt.

Leerspalten am Anfang oder zwischen den zu verarbeitenden Feldern dürfen nicht vernachlässigt werden.

Dasselbe gilt analog für Druckstellen auf Listen.

Für die Lochkarte in Abb. 6 muß der Bereich also __(Z)__ Stellen groß sein.

53

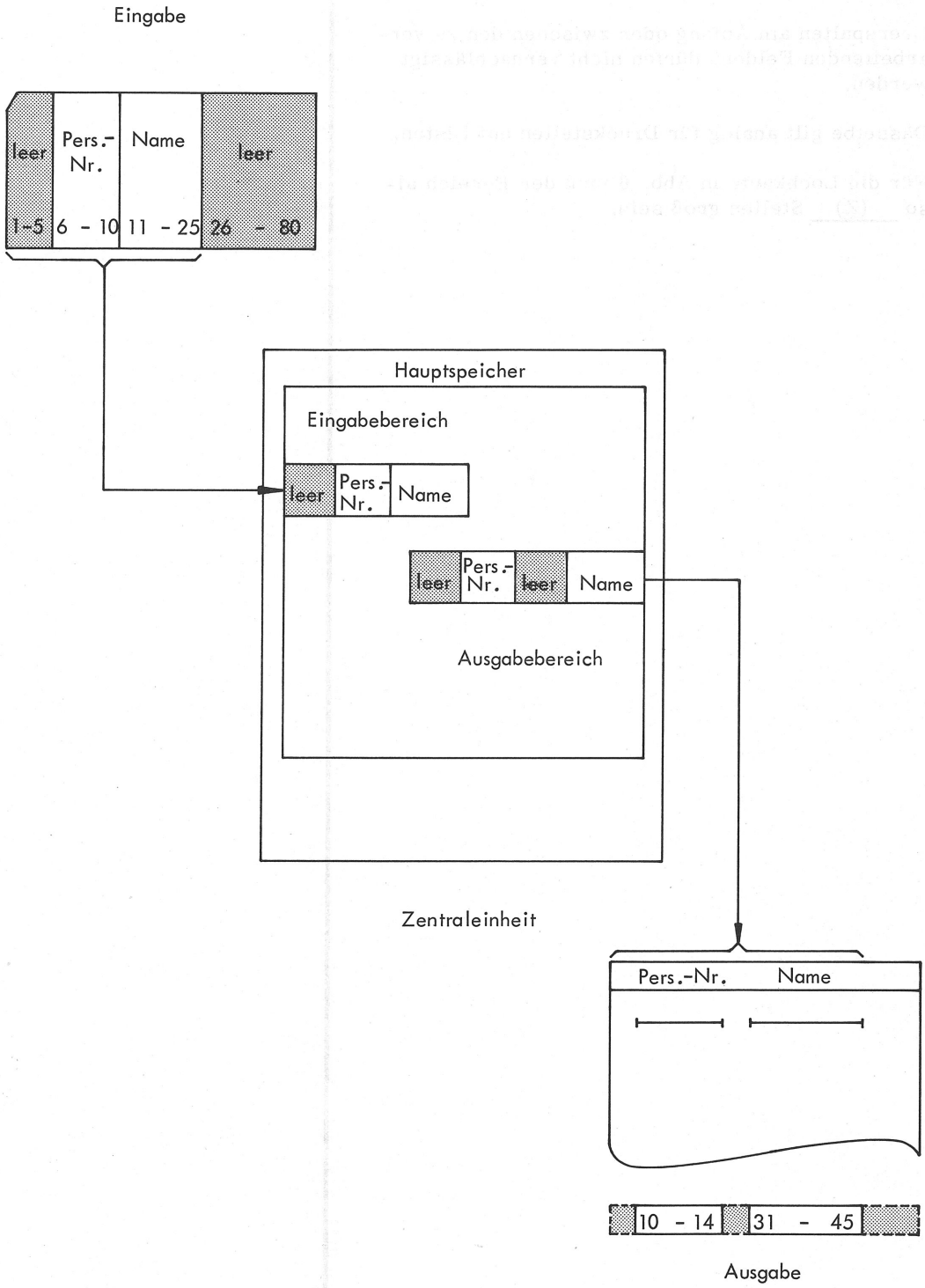

Eingabe
leer
Pers.-Nr.
Name
leer
1-5
6 - 10
11 - 25
26 - 80
Hauptspeicher
Eingabebereich
leer
Pers.-Nr.
Name
leer
Pers.-Nr.
leer
Name
Ausgabebereich
Zentraleinheit
Pers.-Nr.
Name
10 - 14
31 - 45
Ausgabe

Abb. 7

6 Aus der Abb. 7 wird die Abhängigkeit der Ein-/Ausgabebereiche von den ein- oder auszugebenden Daten noch einmal deutlich sichtbar. Eingabe- und Ausgabebereich sind gleichsam ein Abbild der auf Lochkarte und Liste angeordneten Ein-/Ausgabedaten.

Schauen Sie sich Eingabekarte und Eingabebereich der Abb. 7 genau an!

Damit alle für die Verarbeitung benötigten Daten stellengerecht in den Hauptspeicher eingelesen werden können, müssen wir die Eingabekarte von Spalte __(Z)__ bis Spalte __(Z)__ im Eingabebereich "abbilden"

Der Eingabebereich muß also __(Z)__ Stellen groß sein.

1

25

25

7 Damit die Daten bei der Ausgabe auf die Liste stellengerecht gedruckt werden können, müssen __(Z)__ Speicherstellen für den _____ reserviert werden.

45

Ausgabebereich

Wie Sie wissen, werden die Eingabedaten in den Eingabebereich eingelesen und stehen hier für die Verarbeitung bereit. Sollen nun während der Verarbeitung mit diesem Bereich oder mit einzelnen Feldern desselben bestimmte Operationen durchgeführt werden, so muß es auch eine Möglichkeit geben, z. B. das Feld Pers. -Nr. aufzufinden.

Wie dies ermöglicht wird, erfahren Sie in den folgenden Lernelementen.

8 Der Hauptspeicher besteht aus einzelnen Speicherstellen.

Eine Speicherstelle wird auch Byte genannt (gesprochen: bait). 5 Bytes sind also 5 _____ im Hauptspeicher.

Speicherstellen

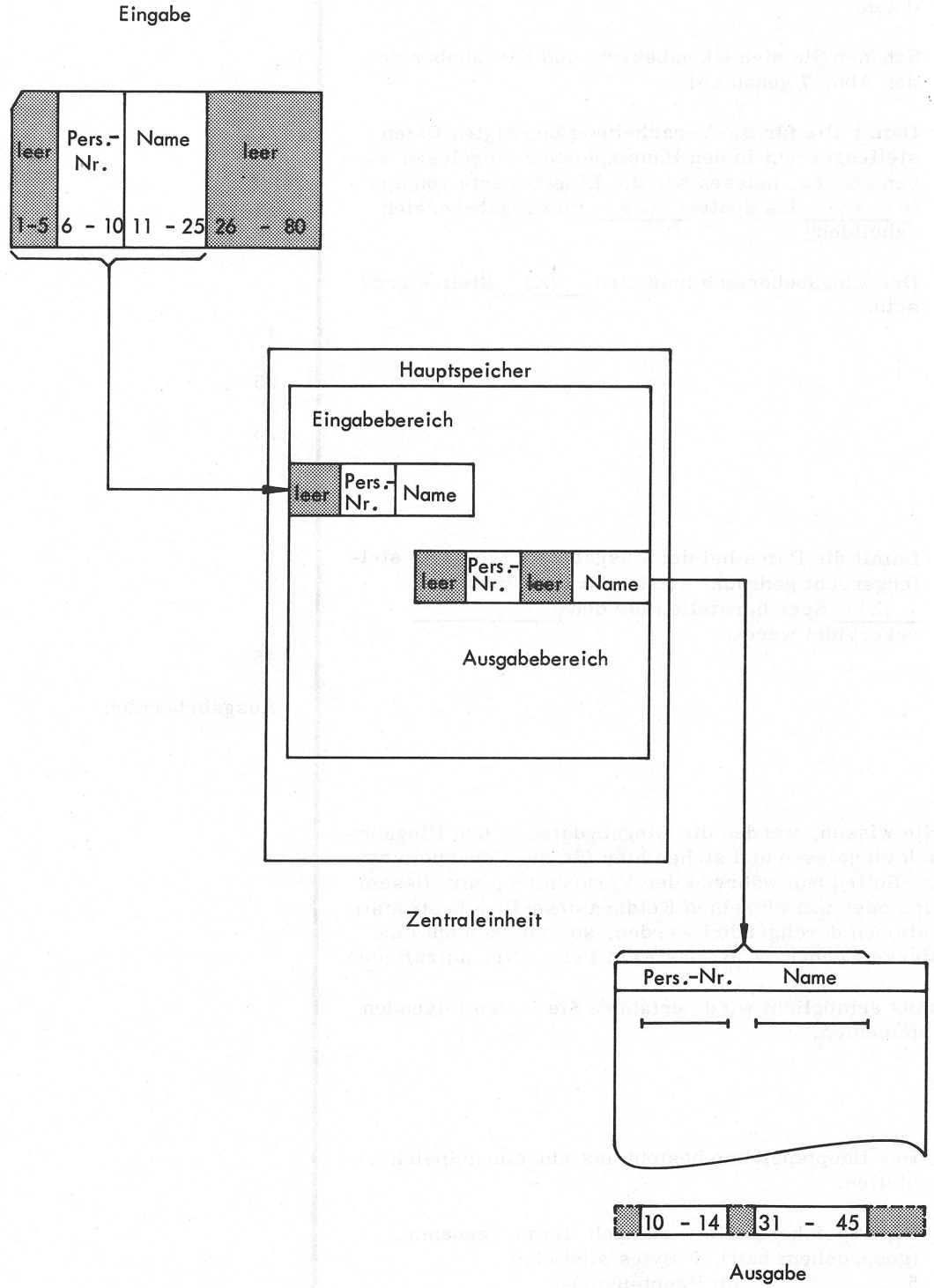

Eingabe
leer
Pers.-Nr.
Name
leer
1-5
6 - 10
11 - 25
26 - 80
Hauptspeicher
Eingabebereich
leer
Pers.-Nr.
Name
leer
Pers.-Nr.
leer
Name
Ausgabebereich
Zentraleinheit
Pers.-Nr.
Name
10 - 14
31 - 45
Ausgabe

Abb. 7

9 Die Speicherstellen (Bytes) sind, von Null beginnend, fortlaufend numeriert.

Jedes Byte hat also seine eigene _____.

Nummer

10 Stellen Sie sich eine unübersehbar lange Straße vor, in der sich Haus an Haus reiht.

Indem man die Häuser fortlaufend numeriert, ist es möglich, ein ganz bestimmtes Haus in dieser fast endlosen Straße aufzufinden.

So ermöglicht auch die fortlaufende _____ im Hauptspeicher, daß eine bestimmte _____ angesprochen werden kann.

Numerierung

Speicherstelle

11 Wir bezeichnen die fortlaufenden Nummern der Speicherstellen als Adressen.

Jede Speicherstelle im Hauptspeicher kann also durch ihre _____ aufgesucht werden.

Adresse

12 So wie in Lochkarten mehrere Spalten, so können im Speicher mehrere Stellen zu einem Feld zusammengefaßt werden.

Im Eingabebereich der Abb. 7 finden Sie z. B. zwei Felder, die sich jeweils aus mehreren Speicherstellen zusammensetzen:

a) das Feld _____ ist __(Z)__ Stellen groß,

b) das Feld _____ ist __(Z)__ Stellen groß.

Pers.-Nr.

5

Name

15

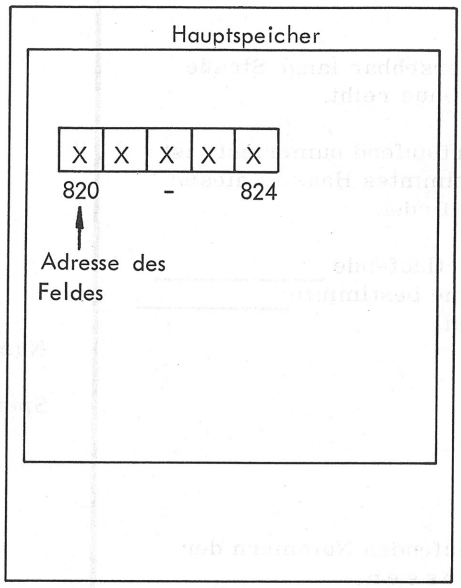


Abb. 8

13 Ein solches aus mehreren Speicherstellen gebildetes Feld ist ein geschlossenes Ganzes, eine neue Einheit.

Bei der Verarbeitung muß diese neue _____ auch als Ganzes angesprochen werden können.

Einheit

14 Um ein Feld im Hauptspeicher während der Verarbeitung aufsuchen zu können, muß es wie jede einzelne Speicherstelle ebenfalls eine _____ haben.

Adresse

15 Betrachten Sie bitte Abb. 8. -

Ein fünfstelliges Feld aus der Lochkarte ist in die Speicherstellen __(Z)__ bis __(Z)__ eingelesen worden.

Wie lautet die Adresse des Feldes im Hauptspeicher?

820 (bis) 824

820

16 Die Adresse eines Feldes ist die Nummer seines (rechten, linken) _____ Bytes.

linken

17 Felder und Bereiche im Hauptspeicher werden also (links, rechts) _____ adressiert.

links

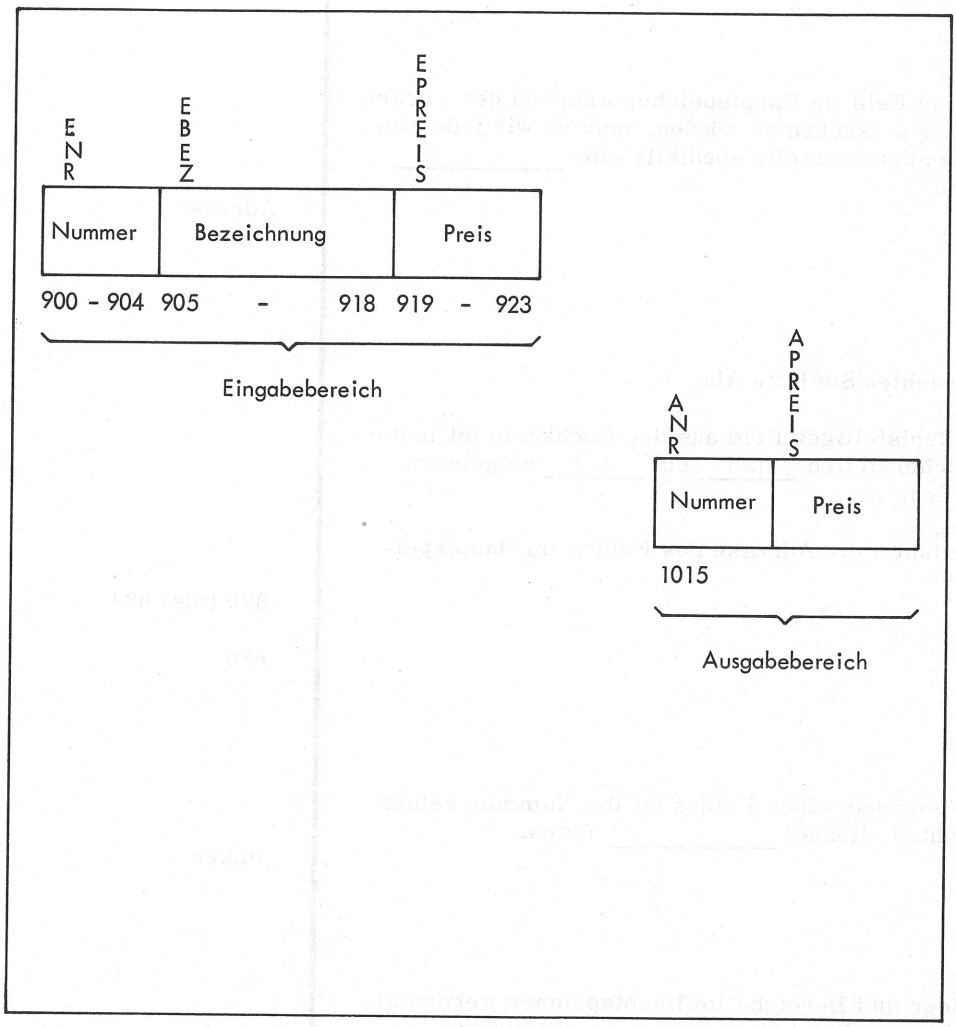

ENR
EBEZ
EPREIS
Nummer
Bezeichnung
Preis
900 - 904 905 - 918 919 - 923
Eingabebereich
ANR
APREIS
Nummer
Preis
1015
Ausgabebereich
Hauptspeicher

Abb. 9

18 Abb. 9 zeigt einen Eingabe- und einen Ausgabebereich im Hauptspeicher.

Der Eingabebereich umfaßt die Speicherstellen
__(Z)__ bis __(Z)__.

Geben Sie die Adresse des Eingabebereiches an:
__(Z)__

900 (bis) 923

900

19 Die Adresse des Eingabebereiches ist gleichzeitig auch die Adresse des Feldes mit dem symbolischen Namen _____.

ENR

20 Nennen Sie auch die Adressen der Felder mit den Namen EBEZ und EPREIS im Eingabebereich:

__(Z)__

__(Z)__

905

919

21 Sehen Sie sich nun den Ausgabebereich in Abb. 9 an.

Das Feld ANR ist 5 Stellen groß. Nennen Sie die Adressen der beiden Felder im Ausgabebereich:

__(Z)__ (ANR)

__(Z)__ (APREIS)

1015

1020

22 Wir merken uns:

Die Adresse eines Feldes im Hauptspeicher ist die Nummer des _____ Bytes des Feldes.

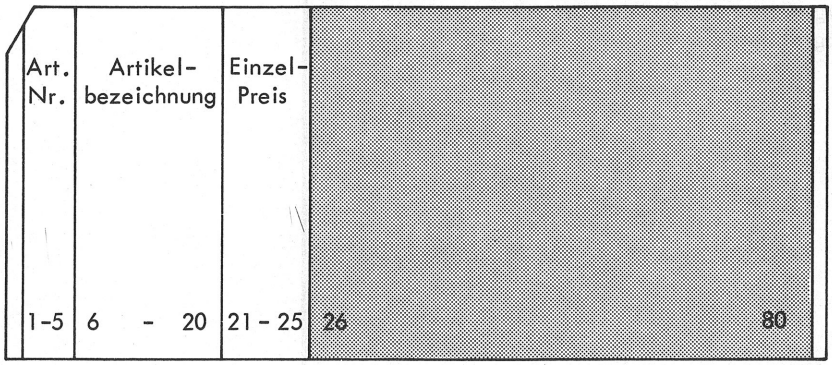
Die L i n k s a d r e s s i e r u n g ist ein Merkmal des Systems /360.

linken

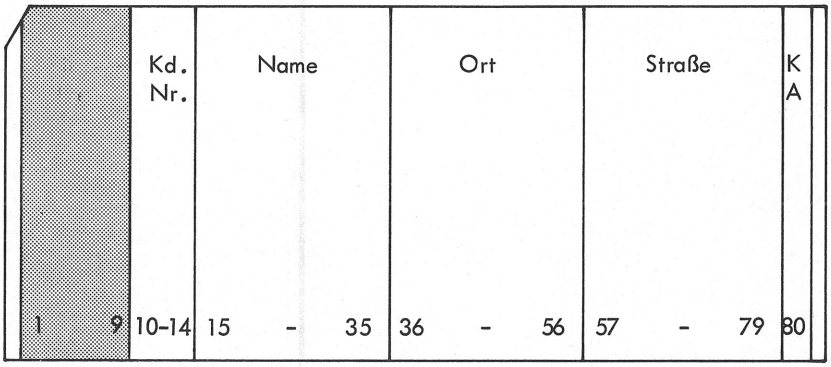
Sie wissen nun, daß aus einzelnen Bytes im Hauptspeicher Felder und Bereiche gebildet werden können und daß diese links adressiert werden.

Wie man Platz für Felder und Bereiche im Hauptspeicher reserviert und wie die Adressierung derselben im System /360 durchgeführt wird, wird an anderer Stelle in diesem Lehrgang behandelt.

a)


Art. Nr.
Artikel-bezeichnung
Einzel-Preis
1-5
6 - 20
21 - 25
26
80

b)


Kd. Nr.
Name
Ort
Straße
K A
1 9
10-14
15 - 35
36 - 56
57 - 79
80

c)

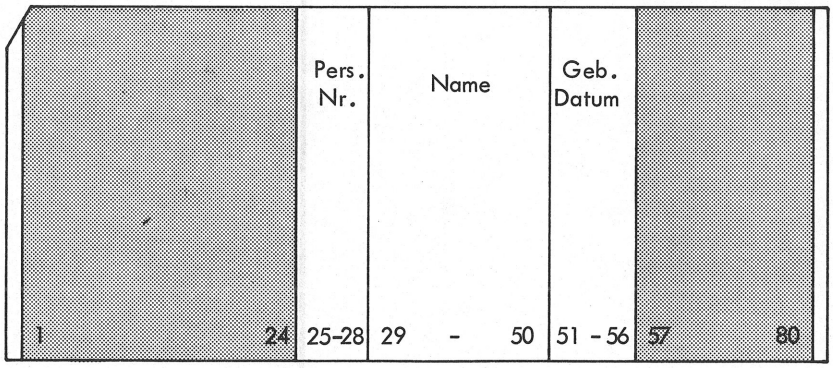

Pers. Nr.
Name
Geb. Datum
1 24
25-28
29 - 50
51 - 56
57 80

Abb. 10

FRAGEN ZUM KAPITEL I

Prüfen Sie anhand der folgenden Fragen, ob Sie das erste Kapitel verstanden haben.

Beantworten Sie erst alle Fragen, ehe Sie Ihre Lösung mit den Antworten auf der folgenden Seite vergleichen.

Benutzen Sie bitte für die Antworten ein Extrablatt.

1. Bestimmen Sie für die Eingabedaten in Abb. 10 die Größe der Eingabebereiche:

   a) __(Z)__ Stellen

   b) __(Z)__ Stellen

   c) __(Z)__ Stellen

2. Wie nennt man die Nummer einer Speicherstelle oder eines Feldes im Hauptspeicher, die es ermöglicht, Speicherstelle oder Feld anzusprechen?

3. Wie lautet die Adresse eines Hauptspeicherbereiches, der die Speicherstellen 840 bis 865 umfaßt?

4. Felder und Bereiche im Hauptspeicher werden stets _____ adressiert.

ANTWORTEN ZU KAPITEL I

Wenn Ihnen Fehler unterlaufen sind, sollten Sie nicht fortfahren, ehe Sie alle Unklarheiten beseitigen konnten. Am besten wiederholen Sie die betreffenden Lernelemente; die Nummern der Lernelemente sind nach jeder Antwort in Klammern angeführt.

1. a) 25
   b) 80 (LE 1 - 7)
   c) 56

2. Adresse (LE 9-11, 14-15)

3. 840 (LE 16 - 21)

4. Felder und Bereiche im Hauptspeicher werden stets links adressiert.
   (LE 15 - 22)



Band 3, Seite 47 (Kapitel II)

KAPITEL II

VERSCHLÜSSELUNG DER ZEICHEN (BUCHSTABEN, ZIFFERN UND SONDERZEICHEN) IM EBCD - CODE

Lernstoff:

Magnetkern - Bit - Binärziffer

Dezimalziffern 0 - 9 in 4 Bits

Hexadezimalziffern 0 - F

Speicherstelle - Byte - Zeichen

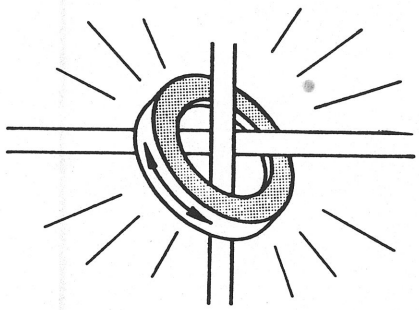
Buchstaben, Dezimalziffern und Sonderzeichen in 8 Bits

Wie Ihnen schon bekannt ist, muß der Hauptspeicher verschiedene Arten von Daten speichern. Dazu gehören sowohl Programme mit ihren Instruktionen und Konstanten als auch Daten für die Verarbeitung.

Alle diese Daten haben eine gemeinsame Grundlage: sie setzen sich zusammen aus einzelnen Z e i c h e n. Diese Zeichen können Buchstaben, Ziffern oder Sonderzeichen sein.

In diesem Kapitel erfahren Sie, wie die einzelnen Zeichen im Hauptspeicher des Systems /360 dargestellt werden.

DER MAGNETKERN



EIN - AUS

Abb. 1

Der Hauptspeicher ist in seinem technischen Aufbau ein "Kernspeicher", genauer gesagt ein "Magnetkernspeicher". Er besteht aus einer Vielzahl kleiner magnetisierbarer Ringe (Magnetkerne, Abb. 1). Jeder dieser Magnetkerne kann in 2 entgegengesetzten Richtungen magnetisiert werden. Ein Magnetkern kann eine bestimmte Magnetisierungsrichtung annehmen, beliebig lange beibehalten und an andere Magnetkerne weitergeben.

Die Magnetisierungsrichtungen eines Magnetkerns kann man mit den 2 Stellungen eines Lichtschalters vergleichen. Man sagt: der Magnetkern ist "EIN" oder "AUS".

Statt der technischen Bezeichnung "Magnetkern" benutzt der Programmierer meistens den Ausdruck "Bit", und zwar aus folgendem Grunde:

Unser normales Zahlensystem ist das Dezimalsystem oder Zehnersystem. Es verwendet 10 Ziffern, die auch "Dezimalziffern" genannt werden.

Daneben gibt es andere Zahlensysteme, so z. B. das Binärsystem oder Zweiersystem. Dieses System kennt nur die 2 Ziffern 0 und 1. Im Binärsystem heißen diese beiden Ziffern "Binärziffern".

Jeder Magnetkern kann durch seine Stellung EIN oder AUS eine Binärziffer (1 oder 0) speichern. Die Binärziffern 0 und 1 bilden die Grundelemente bei der Speicherung aller Zeichen im Hauptspeicher.

"Binärziffer" heißt englisch "binary digit" (sprich: baineri didschit), abgekürzt "Bit". Im Hinblick auf den gespeicherten Inhalt hat der Ausdruck "Bit" größere Aussagekraft als der Name Magnetkern und wird daher häufiger benutzt.

Wir halten fest:

1 In diesem Kapitel geht es um das Speichern von _____ im Hauptspeicher.

Dazu gehören Buchstaben, _____ und _____.

Zeichen

Ziffern

Sonderzeichen

2 Der Hauptspeicher ist ein Magnetkernspeicher. Er besteht aus einzelnen _____, die in __(Z)__ entgegengesetzten Richtungen magnetisiert werden können.

Magnetkernen

2

3 Man sagt: ein Magnetkern hat die Stellung _____ oder _____.

Er speichert damit eine der beiden Binärziffern __(Z)__ und __(Z)__.

Im Hinblick auf diesen Speicherinhalt "Binärziffer" (= binary digit) benutzt man statt "Magnetkern" häufiger den Ausdruck "_____".

EIN (oder) AUS

1 (und) 0

Bit

ein Bit (Magnetkern)	Dezimal-Ziffer
AUS	0
EIN	1
	2
	3
	4
	5
	6
	7
	8
	9

AUS - Binärziffer 0

EIN - Binärziffer 1

Abb. 2

Wir werden im Verlauf dieses Kapitels untersuchen, wie die Binärziffern 0 und 1 in einem Bit bzw. die Stellungen AUS und EIN des Magnetkerns zur Speicherung der Buchstaben, Ziffern und Sonderzeichen benutzt werden.

Zunächst beschäftigen wir uns mit der Darstellung der Dezimalziffern 0 bis 9.

4 Betrachten Sie bitte Abb. 2.

Wir nehmen an, daß nur ein Bit zum Speichern zur Verfügung steht.

Mit diesem Bit können __(Z)__ Ziffern von den 10 Dezimalziffern 0 bis 9 eindeutig dargestellt werden.

2

5 Die Zuordnung zwischen Bit-Stellung und Dezimalziffer ist so festgelegt, wie Abb. 2 zeigt.

Der Ziffer 0 entspricht die Stellung _____
und der Ziffer 1 die Stellung _____.

AUS

EIN

6 Zur Darstellung der weiteren 8 Dezimalziffern reicht ein Bit nicht aus.

Man muß ein zweites Bit hinzunehmen. Auch dieses zweite Bit kann __(Z)__ verschiedene Stellungen einnehmen.

2

ein Bit (Magnetkern)	ein Bit (Magnetkern)	Dezimal-Ziffer
AUS	AUS	0
AUS	EIN	1
EIN	AUS	2
EIN	EIN	3
		4
		5
		6
		7
		8
		9

Abb. 3

ein Bit	ein Bit	Dezimal-Ziffer
0	0	0
0	1	1
1	0	2
1	1	3
		4
		5
		6
		7
		8
		9

Abb. 4

7 Abb. 3 zeigt, wie die Stellungen EIN bzw. AUS der beiden Bits kombiniert werden.

Bei der Dezimalziffer 2 haben die beiden Bits die Stellungen _____ und _____.

Die höchste dargestellte Dezimalziffer ist __(Z)__. Bei ihr sind beide Bits _____.

EIN (und) AUS

3

EIN

8 Anstelle der Bezeichnungen AUS und EIN werden wir im folgenden in den Tabellen die Binärziffern 0 und 1 verwenden.

Abb. 4 zeigt die vorige Tabelle mit den entsprechenden Angaben.

Für die Ziffer 3 haben die beiden Bits den Inhalt __(ZZ)__ .

11

STELLENWERTSYSTEME

I Dezimalsystem:

Dezimalzahl
"88"

Ziffernwert	8	8
Stellenwert	10	1
dargestellter Wert	8 x 10	8 x 1

Gesamtwert : 80 + 8

II Binärsystem:

Binärzahl
"11"

Ziffernwert	1	1
Stellenwert	2	1
dargestellter Wert	1 x 2	1 x 1

Gesamtwert : 2 + 1

Abb. 5

Der Dezimalzahl 3 steht also die Binärzahl 11 (sprich "eins eins", nicht "elf") gegenüber.

Bevor wir uns die Darstellung der weiteren Dezimalziffern ansehen, untersuchen wir den grundsätzlichen Aufbau des Binärsystems.

Das Binärsystem ist ein Stellenwertsystem - genau wie das Ihnen vertraute Dezimalsystem. Es ist also ein Zahlensystem, bei dem der Wert einer Ziffer nicht nur von der Ziffer selbst, sondern auch von der Stelle abhängt, an der die Ziffer innerhalb der Zahl steht.

9 Betrachten Sie Abb. 5.

Wir erinnern uns zunächst an den Aufbau des Dezimalsystems.

Die Dezimalzahl 88 enthält zweimal die Ziffer 8. Diese Ziffer hat in beiden Stellen den gleichen Ziffernwert, aber unterschiedlichen _____.

Stellenwert

10 Um zu ermitteln, welchen Wert eine Ziffer an einer bestimmten Stelle darstellt, müssen wir den Stellenwert und den _____ miteinander multiplizieren.

Um den Gesamtwert der Zahl zu ermitteln, muß man dann die Einzelwerte _____.

Ziffernwert

addieren

11 Das Binärsystem ist genauso aufgebaut.

Als Beispiel dient die Binärzahl "11".

Die Binärziffern 0 und 1 haben an jeder beliebigen Stelle in einer Binärzahl immer die Ziffernwerte ___(Z)___ bzw. ___(Z)___.

0 (bzw.) 1

STELLENWERTSYSTEME

I Dezimalsystem:

Dezimalzahl
"88"

Ziffernwert	8	8
Stellenwert	10	1
dargestellter Wert	8 x 10	8 x 1

Gesamtwert : 80 + 8

II Binärsystem:

Binärzahl
"11"

Ziffernwert	1	1
Stellenwert	2	1
dargestellter Wert	1 x 2	1 x 1

Gesamtwert : 2 + 1

Abb. 5

12 Obwohl der Ziffernwert einer Ziffer immer der gleiche bleibt, gibt sie an verschiedenen Stellen in einer Zahl verschiedene Werte an. Jede solche Stelle hat nämlich einen anderen _____.

Wie Sie aus Abb. 5 ersehen, hat die rechte Stelle in einer Binärzahl den Stellenwert __(Z)__ und die zweite Stelle von rechts den _____(WZ)_____.

Der Wert, den die Binärziffer 1 an einer bestimmten Stelle darstellt, ergibt sich als Produkt aus _____ und _____.

Stellenwert

1

Stellenwert 2

Ziffernwert (und) Stellenwert

13 Der Gesamtwert, den die Binärzahl 11 darstellt, ergibt sich aus der Rechnung __(Z)__ + __(Z)__ = __(Z)__

2 + 1 = 3

14 Die Stellenwerte ändern sich im Binärsystem anders als im Dezimalsystem.

Die rechte Stelle hat immer den Stellenwert 1.

In der zweiten Stelle von rechts ist der Stellenwert beim Binärsystem __(Z)__ mal so groß und beim Dezimalsystem __(Z)__ mal so groß wie in der rechten Stelle.

2

10

15 Beantworten Sie bitte folgende Fragen mit Hilfe der Abb. 5:

Welchen Wert stellt die Binärziffer 0 in einer Binärzahl

a) in der rechten Stelle

b) in der zweiten Stelle von rechts

dar ?

a) den Wert 0 (0 x 1)

b) den Wert 0 (0 x 2)

ein Bit	ein Bit	Dezimal-Ziffer
0	0	0
0	1	1
1	0	2
1	1	3
		4
		5
		6
		7
		8
		9
2	1	Stellenwert

Abb. 6

16 Wir fassen zusammen:

In einer Binärzahl stellt die Ziffer 0 in jeder Stelle den Wert __(Z)__ dar und die Ziffer 1 in jeder Stelle den Wert des _____.

In Abb. 6 finden Sie eine frühere Abbildung, ergänzt um die Angabe der Stellenwerte.

0

Stellenwertes

17 Sie kennen bisher die binäre Darstellung der Dezimalzahlen 0 bis 3.

Zu diesen Zahlen gehören die zwei "geraden" Zahlen __(Z)__ und __(Z)__ und die zwei "ungeraden" Zahlen __(Z)__ und __(Z)__.

Eine ungerade Zahl ist dadurch gekennzeichnet, daß sich bei einer Division durch 2 ein Rest von __(Z)__ ergibt.

0 (und) 2

1 (und) 3

1

18 Werden die ungeraden Zahlen 1 und 3 als Binärzahlen dargestellt, so enthält das rechte Bit eine __(Z)__.

Enthält die rechte Stelle in einer Binärzahl eine 0, so handelt es sich um eine _____ Zahl.

1

gerade

ein Bit	ein Bit	ein Bit	Dezimal-Ziffer
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1			4
1			5
1			6
1			7
			8
			9
4	2	1	Stellenwert

Abb. 7

19 Wir werden jetzt die binäre Darstellung der Ziffern 4 bis 7 entwickeln.

Zu den bisherigen 2 Bits nehmen wir ein drittes Bit hinzu (vgl. Abb. 7).

Eine Binärziffer hat in diesem Bit den Stellenwert __(Z)__.

Der Stellenwert hat sich also vom zweiten zum dritten Bit wieder _____.

4

verdoppelt

20 Ist das linke (dritte) Bit auf EIN, so hat die dargestellte Dezimalziffer sicher einen Wert, der größer oder gleich __(Z)__ ist.

4

21 Für die ungeraden Dezimalziffern 5 und 7 muß jeweils der Stellenwert 1 herangezogen werden.

Bei beiden Ziffern enthält also das (rechte, mittlere) _____ Bit eine 1.

rechte

ein Bit	ein Bit	ein Bit	Dezimal-Ziffer
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1		0	4
1		1	5
1		0	6
1		1	7
			8
			9
4	2	1	Stellenwert

Abb. 8

22 Offen ist noch der Inhalt des mittleren Bits.

Der Stellenwert 2 muß hinzugenommen werden für die Dezimalziffern ___(Z)___ und ___(Z)___.

Bei diesen beiden Ziffern hat das mittlere Bit den Inhalt ___(Z)___.

6 (und) 7

1

ein Bit	ein Bit	ein Bit	Dezimal-Ziffer
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7
			8
			9
4	2	1	Stellenwert

Abb. 9

23 Die höchstmögliche Dezimalziffer ist dann ausgedrückt, wenn alle verfügbaren Bits den Inhalt __(Z)__ haben.

Bei den 3 Bits in Abb. 9 ergibt die Addition der Stellenwerte __(Z)__, __(Z)__ und __(Z)__ den Wert für die Dezimalziffer __(Z)__.

1

4, 2 (und) 1

7

24 Für die Ziffern 8 und 9 reichen die 3 Bits nicht mehr aus. Ein 4. Bit kommt (links, rechts) _____ hinzu.

Wie Abb. 9 zeigt, verdoppeln sich die Stellenwerte nach links ständig. Im 4. Bit wird eine Binärziffer also den Stellenwert __(Z)__ haben.

links

8

ein Bit	ein Bit	ein Bit	ein Bit	Dezimal-Ziffer
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
8	4	2	1	Stellenwert

Abb. 10

25 Den Inhalt der 4 Bits für die Ziffern 8 und 9 sehen Sie in Abb. 10.

Die Ziffer 9 hat die Darstellung ____(ZZ)____.

1 0 0 1

26 Zur Darstellung der 10 Dezimalziffern 0 bis 9 braucht man also im Hauptspeicher 4 Bits.

Betrachten Sie nochmals die Darstellung der Dezimalziffer 9 in Abb. 10.

Kann man mit Hilfe der 4 Bits höhere Ziffernwerte als 9 darstellen ?

Ja

27 Es gilt hier wieder ein sehr einfaches Prinzip:

In 1 Bit können ___(Z)___ Ziffern dargestellt werden.
In 2 Bits können ___(Z)___ Ziffern dargestellt werden.
In 3 Bits können ___(Z)___ Ziffern dargestellt werden.

2

4

8

28 Die Anzahl der darstellbaren Ziffern (erhöht sich um 2, verdoppelt sich) _____ ständig mit der Erhöhung der Bit-Anzahl.

Wenden wir diese Regel auf 4 Bits an, so ergibt sich: in 4 Bits können ___(Z)___ Ziffern dargestellt werden.

verdoppelt sich

16

ein Bit	ein Bit	ein Bit	ein Bit	Ziffer
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	
8	4	2	1	Stellenwert

Abb. 11

Unser Dezimalsystem hat keine 16, sondern nur 10 Ziffern. Diese 10 Ziffern werden mit Hilfe der Binärziffern 0 und 1 in 4 Bits dargestellt.

Aber auch die weiteren 6 Darstellungen in 4 Bits treten im Hauptspeicher auf. Sie werden z. B. zur Darstellung der Buchstaben und Sonderzeichen und der Vorzeichen von Dezimalzahlen benutzt.

Die ersten 10 Kombinationsmöglichkeiten haben wir den Dezimalziffern 0 bis 9 zugeordnet. Auch für die restlichen 6 Kombinationsmöglichkeiten werden wir jetzt Namen festlegen. Diese Namen dienen der verkürzten Ausdrucksweise im Zusammenhang mit der Verschlüsselung der Zeichen im Hauptspeicher.

29 Abb. 11 zeigt die weiteren 6 Kombinationsmöglichkeiten in 4 Bits.

Bei der ersten Kombination 1010 sind die Bits mit den Stellenwerten 8 und 2 EIN. Als Wert ergibt sich daraus

(Z) + (Z) = (Z)

8 + 2 = 10

30 Berechnen Sie, welche Werte die restlichen Bitkombinationen darstellen bzw. verschlüsseln:

1011	(Z) + (Z) + (Z) = (Z)	8 + 2 + 1 = 11
1100	(Z) + (Z) = (Z)	8 + 4 = 12
1101	(Z) + (Z) + (Z) = (Z)	8 + 4 + 1 = 13
1110	(Z) + (Z) + (Z) = (Z)	8 + 4 + 2 = 14
1111	(Z) + (Z) + (Z) + (Z) = (Z)	8 + 4 + 2 + 1 = 15

ein Bit	ein Bit	ein Bit	ein Bit	Ziffer
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F
8	4	2	1	Stellenwert

Abb. 12

31 Es bietet sich an, die Bitkombinationen 1010 bis 1111 mit "10" bis "15" zu benennen.

Wie Sie aus Abb. 12 ersehen, werden jedoch andere Namen benutzt. Man nimmt die Buchstaben _____ bis _____.

Sie haben vor den Zahlen 10 bis 15 den Vorteil, daß sie _____-stellig sind.

A (bis) F

ein

32 Es gibt ein Zahlensystem, das auf 16 Ziffern aufbaut. Dieses System nennt sich Sechszehnersystem oder Hexadezimalsystem (Sedezimalsystem).

Zur Unterscheidung spricht man daher von den Dezimalziffern 0 bis 9 und den Hexadezimalziffern 0 bis F.

Die Hexadezimalziffer "D" entspricht also der Bitkombination ___(ZZ)___ und damit dem Wert __(Z)__.

1101

13

33 Es gibt __(Z)__ Hexadezimalziffern.

Sie stehen als Bezeichnungen für die Werte 0 bis __(Z)__.

Die Ziffer 6 kann ebenso als Dezimalziffer wie auch als _____ aufgefaßt werden.

16

15

Hexadezimalziffer

ein Bit	ein Bit	ein Bit	ein Bit	Ziffer
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F
8	4	2	1	Stellenwert

Abb. 12

34 Geben Sie an, welche Bit-Kombinationen in 4 Bits folgenden Ziffern entsprechen:

Hexadezimalziffer	9	(ZZ) 1001
Hexadezimalziffer	B	(ZZ) 1011
Hexadezimalziffer	2	(ZZ) 0010
Dezimalziffer	8	(ZZ) 1000
Dezimalziffer	5	(ZZ) 0101
Dezimalziffer	2	(ZZ) 0010

Mit der Verschlüsselung der Hexadezimalziffern 0 bis F sind alle Möglichkeiten, Zeichen in 4 Bits dazustellen, erschöpft.

Wir können nun in der Behandlung der Verschlüsselung der Zeichen im Hauptspeicher fortfahren.

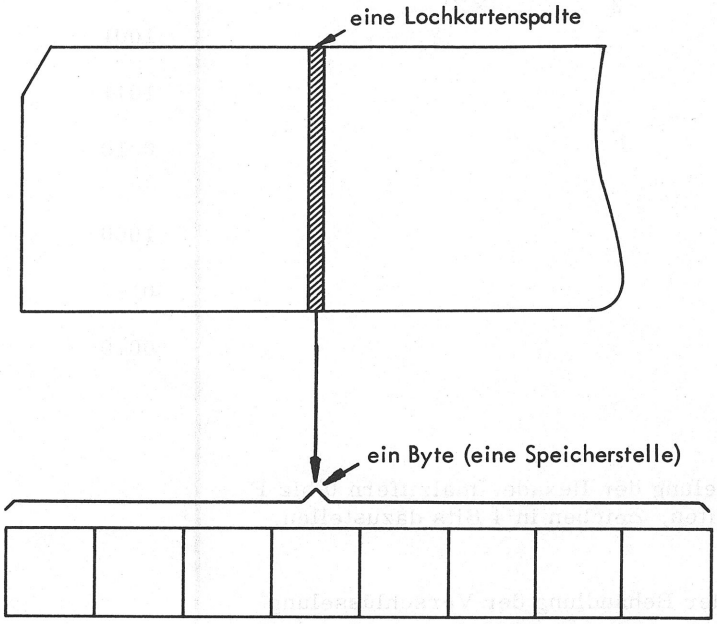


Abb. 13

35 Sie kennen bereits die Begriffe Speicherbereich, Speicherfeld und Speicherstelle bzw. Byte.

Wie uns Abb. 13 in Erinnerung bringt, kommt beim Einlesen der Daten in den Speicher der Inhalt einer Lochkartenspalte in ein _____.

Wie wir gerade besprochen haben, ist aber das kleinste Speicherelement im Speicher das _____.

Byte / Speicherstelle

Bit

36 Die Einheit "Byte" ist (kleiner, gleich, größer) _____ als die Einheit "Bit".

größer

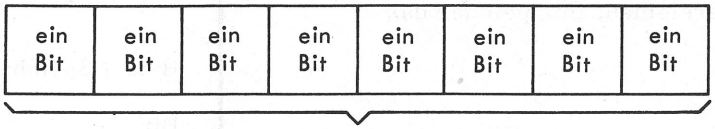

ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Byte (eine Speicherstelle)

Abb. 14

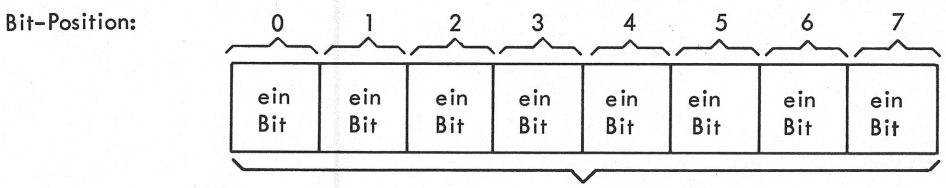

Bit-Position:
0
1
2
3
4
5
6
7
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Byte (eine Speicherstelle)

Abb. 15

37 Wie wir aus Abb. 14 ersehen, besteht ein Byte aus ___(ZW)___.

Eine Hauptspeicheradresse spricht demnach immer (ein Bit, ein Byte) _____ an.

8 Bits

ein Byte

Die kleinste Speichereinheit ist das Bit.

Die kleinste a d r e s s i e r b a r e Speichereinheit aber ist das Byte.

38 Um die einzelnen Bits in einem Byte eindeutig bezeichnen zu können, numerieren wir die Bits, wie es in Abb. 15 gezeigt ist.

Das 4. Bit von links hat also die Bit-Position ___(Z)___.

Die 8 Bits haben von links nach rechts die Bit-Positionen ___(Z)___ bis ___(Z)___.

3

0 (bis) 7

39 Wir wiederholen:

Ein Byte umfaßt ___(ZW)___.

Die Bits haben die _____ 0 bis 7.

8 Bits

Bit-Positionen

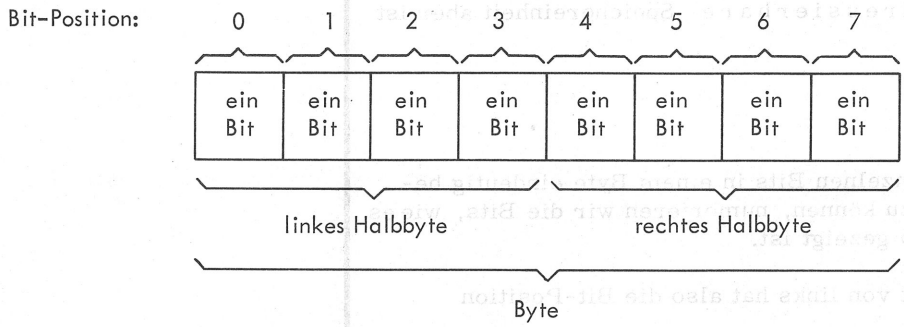

Bit-Position:
0
1
2
3
4
5
6
7
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
ein Bit
linkes Halbbyte
rechtes Halbbyte
Byte

Abb. 16

40 Der Ausdruck "Byte" kommt aus dem Englischen und bezeichnet eine Reihe von Bits, hier genauer die Reihe der 8 Bits einer _____.

Speicherstelle

41 Oft interessiert nicht ein ganzes Byte, sondern nur die Verschlüsselung in einem halben Byte. Darum hat man die beiden Bezeichnungen eingeführt, die Abb. 16 zeigt.

Ein Byte wird also unterteilt in ein ___(WW)___ und ein ___(WW)___.

linkes Halbbyte

rechtes Halbbyte

42 Jedes Halbbyte umfaßt __(Z)__ Bits.

Zum linken Halbbyte gehören die Bit-Positionen __(Z)__ bis __(Z)__ und zum rechten _____ die Bit-Positionen __(Z)__ bis __(Z)__.

4

0 (bis) 3

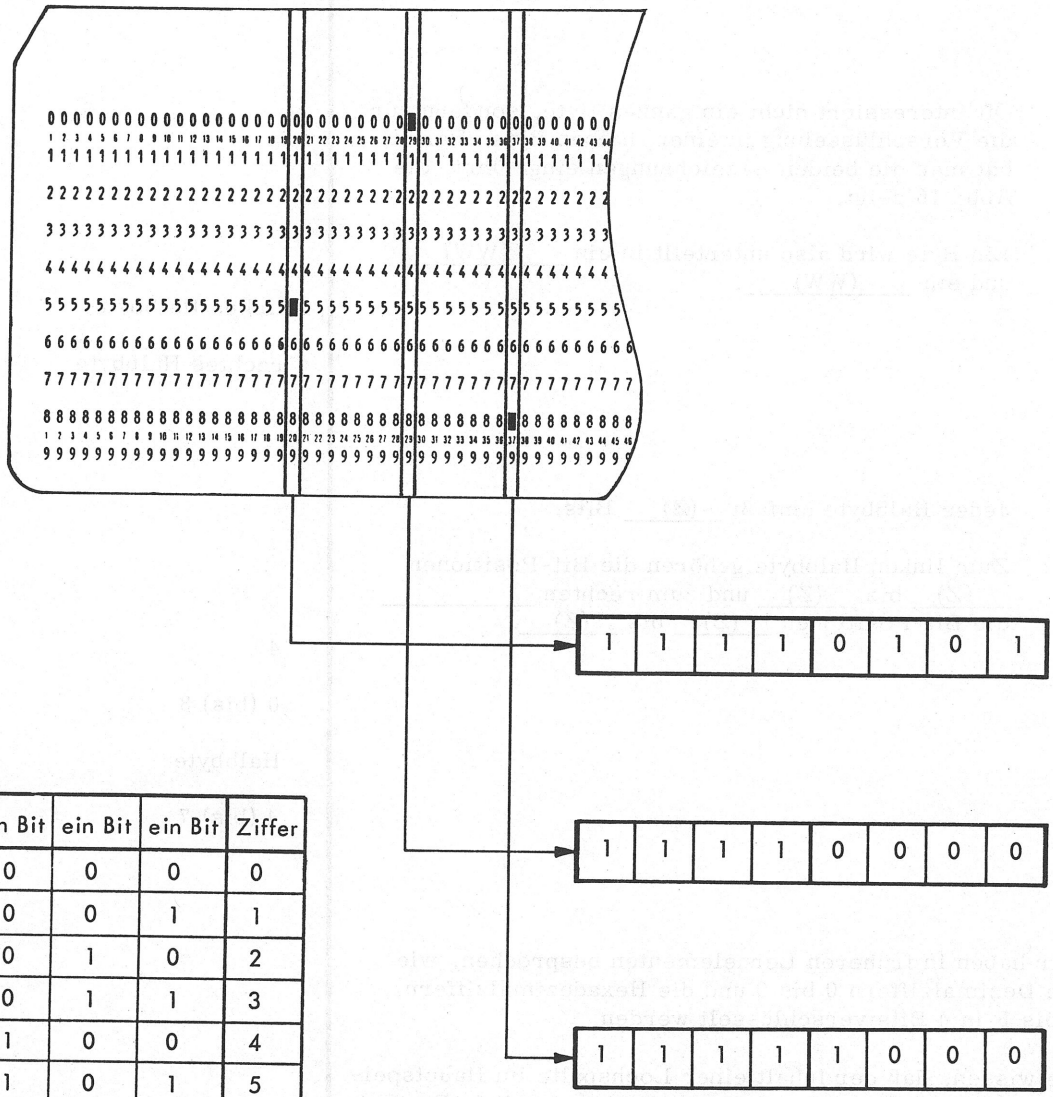
Halbbyte

4 (bis) 7

Wir haben in früheren Lernelementen besprochen, wie die Dezimalziffern 0 bis 9 und die Hexadezimalziffern 0 bis F in 4 Bits verschlüsselt werden.

Sie wissen, daß der Inhalt einer Lochspalte im Hauptspeicher in einem Byte oder 8 Bits gespeichert wird. Ein Zeichen wird also in 8 Bits im Hauptspeicher verschlüsselt.

Es bleibt nun zu klären, wie die Verschlüsselung der Ziffern, Buchstaben und Sonderzeichen in 8 Bits aussieht.



ein Bit	ein Bit	ein Bit	ein Bit	Ziffer
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Abb. 17

43 In Abb. 17 sehen Sie 3 Dezimalziffern jeweils in einer Lochspalte und einer Speicherstelle (8 Bits) verschlüsselt.

Es sind die Ziffern __(Z)__, __(Z)__ und __(Z)__.

5, 0 (und) 8

44 Die Tabelle unten in Abb. 17 zeigt noch einmal die Darstellung von Ziffern in 4 Bits.

In welchem Halbbyte tauchen jeweils die entsprechenden 4-Bit-Verschlüsselungen auf?

im rechten Halbbyte

45 Das linke Halbbyte ist zur Unterscheidung der Ziffern 5, 0 und 8 (nötig, nicht nötig) _____.

nicht nötig

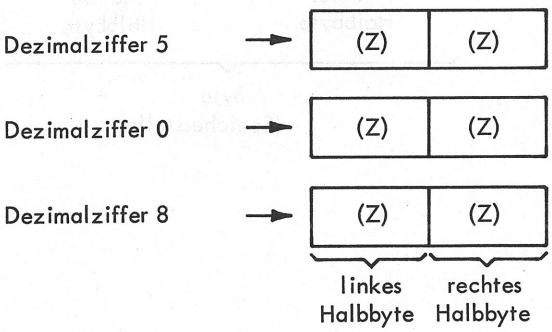
46 Für Dezimalziffern wurde für die linken 4 Bits die Verschlüsselung __(Z)__ festgelegt.

Bei den Dezimalziffern enthalten also die linken 4 Bits die Hexadezimalziffer __(Z)__.

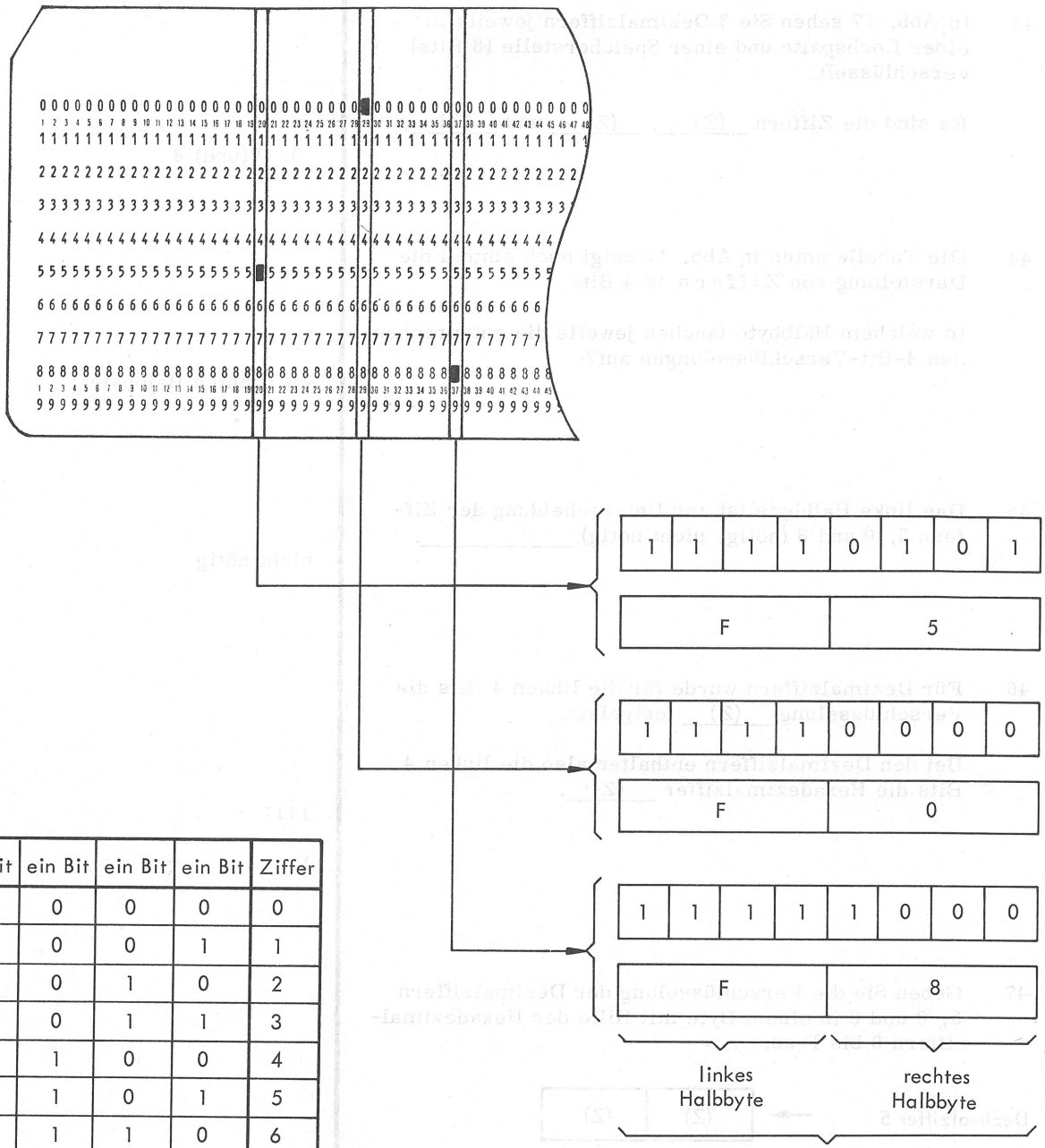
1111

F

47 Geben Sie die Verschlüsselung der Dezimalziffern 5, 0 und 8 in einem Byte mit Hilfe der Hexadezimalziffern 0 bis F an:



F	5
F	0
F	8



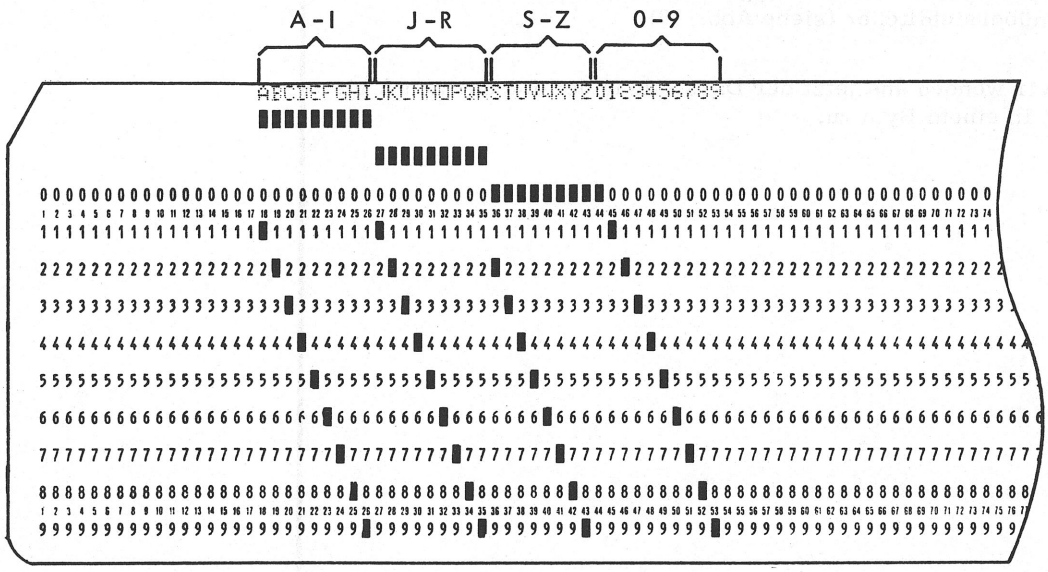
ein Bit	ein Bit	ein Bit	ein Bit	Ziffer
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Abb. 18

Die Hexadezimalziffern bieten die Möglichkeit, die Verschlüsselung, z.B. der Dezimalziffern 0 bis 9, in einem Byte verkürzt anzugeben. Jede Verschlüsselung in einem Byte kann mit 2 Hexadezimalziffern vollständig beschrieben werden. Eine binäre Ausgabe wäre genauso richtig, aber unübersichtlicher (siehe Abb. 18).

Wir wenden uns jetzt der Darstellung der Buchstaben A bis Z in einem Byte zu.

ZONENLOCHUNGEN BEI BUCHSTABEN UND ZIFFERN



ZONENVERSCHLÜSSELUNG BEI BUCHSTABEN UND ZIFFERN IM HAUPTSPEICHER

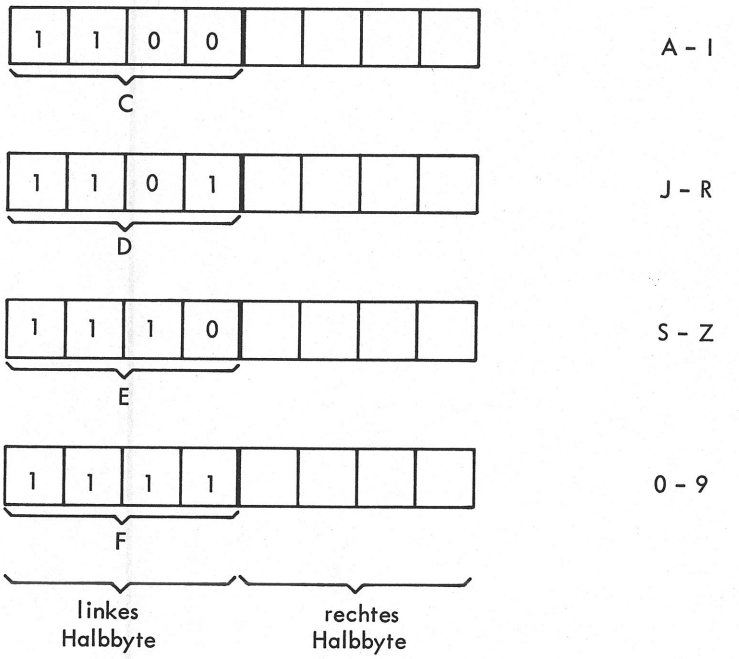


Abb. 19

48 Betrachten Sie bitte Abb. 19.

Die Buchstaben A - I haben die Zonenlochung (12, 11, 0, keine) _____.

Die Buchstaben J - R haben die Zonenlochung __(Z)__.

Die Buchstaben S - Z haben die Zonenlochung __(Z)__.

Die Ziffern 0 bis 9 haben _____ Zonenlochung.

12

11

0

keine

49 Jeder dieser Zonenlochungen entspricht im Speicher eine bestimmte Verschlüsselung in __(Z)__ Bits.

Sie steht im _____ Halbbyte.

4

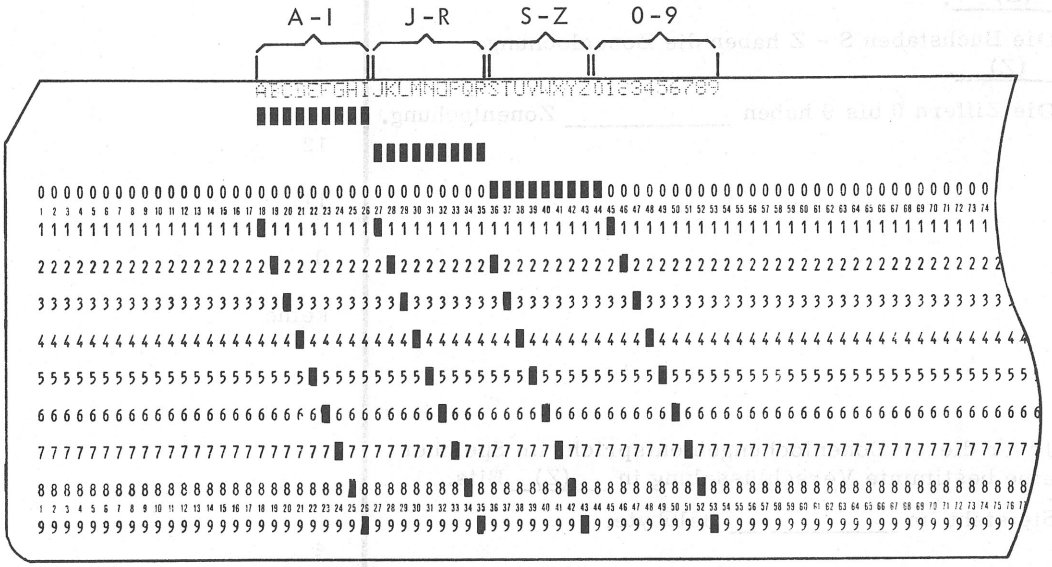
linken

50 Wir können die Beziehung zwischen Zonenlochung und Verschlüsselung im Speicher in einer Tabelle darstellen. Ergänzen Sie bitte auf einem Extrablatt (vgl. Abb. 19):

Zonenlochung	Verschlüsselung im linken Halbbyte	
	4 Binärziffern	1 Hexadezimalziffer
12		
11		
0		
keine		

Zonenlochung	Verschlüsselung im linken Halbbyte	
	4 Binärziffern	1 Hexadezimalziffer
12	1100	C
11	1101	D
0	1110	E
keine	1111	F

ZONENLOCHUNGEN BEI BUCHSTABEN UND ZIFFERN



ZONENVERSCHLÜSSELUNG BEI BUCHSTABEN UND ZIFFERN IM HAUPTSPEICHER

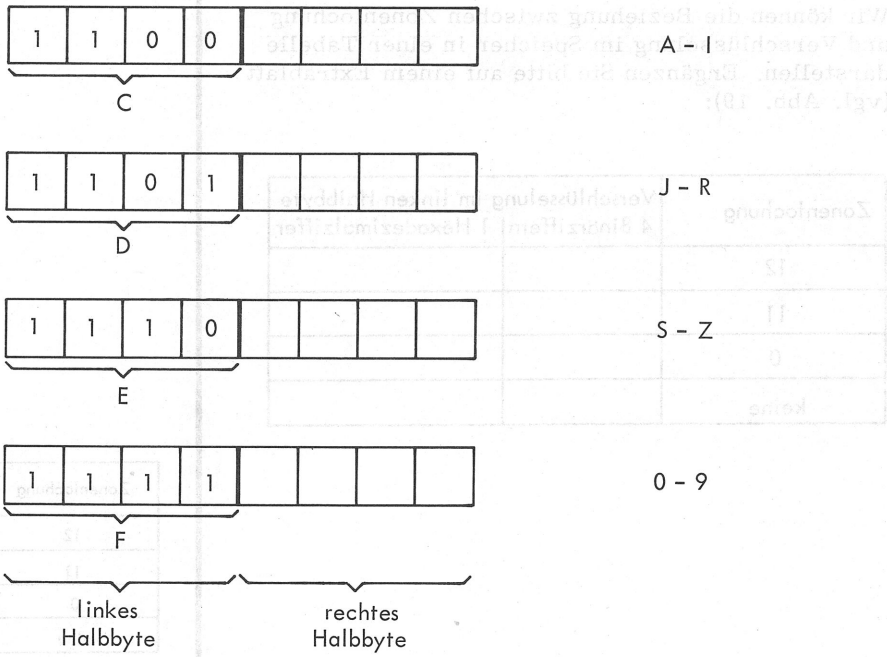


Abb. 19

51 Außer den Zonenlochungen gehört zu den Buchstaben und Ziffern in der Lochkarte je eine Ziffernlochung.

Im Speicher wird diese Ziffernlochung im (linken, rechten) _____ Halbbyte gespeichert.

rechten

52 Abb. 19 enthält 4 Zeichen, die eine 4-er Lochung und zusätzlich eine bzw. keine Zonenlochung haben.

Es sind dies die Zeichen _____, _____, _____ und __(Z)__.

D, M, U (und) 4

53 Werden die Zeichen D, M, U und 4 in den Hauptspeicher eingelesen, so wird jedes Zeichen in einem _____ gespeichert.

Dieses Byte zeigt links eine der Zonenverschlüsselungen C, D, E oder F, während das rechte Halbbyte bei allen 4 Zeichen die Hexadezimalziffer __(Z)__ enthält.

Byte

4

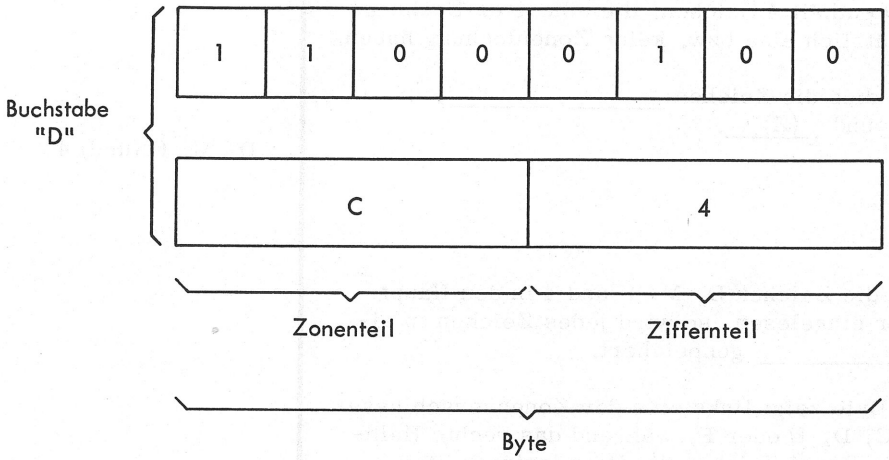


Abb. 20

54 In Abb. 20 finden Sie zwei neue Bezeichnungen:

Das Byte besteht aus dem linken Halbbyte oder _____ und aus dem rechten Halbbyte oder _____.

Zonenteil

Zifferntteil

55 Der Zonenteil im Byte entspricht der _____ -Lochung in der Lochspalte und der Zifferntteil der _____-Lochung.

Für den Buchstaben D enthält der Zonenteil die Hexadezimalziffer ___(Z)___ und der Zifferntteil die Hexadezimalziffer ___(Z)___.

Zonen

Ziffern

C

4

56 Wir halten fest:

Buchstaben und Ziffern werden im Hauptspeicher in einem Byte gespeichert.

Dabei enthält das linke Halbbyte oder der _____ eine der Hexadezimalziffern C, ___(Z)___, ___(Z)___ und ___(Z)___.

Das rechte Halbbyte oder der _____ enthält eine der Hexadezimalziffern 0, 1, 2 ... bis ___(Z)___.

Zonenteil

D, E (und) F

Zifferntteil

9

EBCD - CODE

		rechtes Halbbyte (Ziffernteil)															
	hexa-dezimal →	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
linkes Halbbyte (Zonenteil)	0 0000																
	1 0001																
	2 0010																
	3 0011																
	4 0100																
	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. 21

57 Abb. 21 zeigt die Verschlüsselung der Buchstaben und Ziffern in einer Tabelle.

Geben Sie an, wie der Buchstabe W verschlüsselt ist:

	linkes Halbbyte (Zonenteil)	rechtes Halbbyte (Ziffernteil)
a) binär:	(ZZ)	(ZZ)
b) hexadezimal:	(Z)	(Z)

a) 1110 0110

b) E 6

EBCD-CODE

linkes Halbbyte (Zonenteil) hexadezimal	binär	rechtes Halbbyte (Ziffernteil) 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		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																
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. 22

58 Wie Sie aus der Tabelle in Abb. 22 ersehen, gibt es außer W noch 7 andere Buchstaben mit der Zonenteilverschlüsselung E.

Es sind dies die Buchstaben _____.

S, T, U, V, X, Y, Z

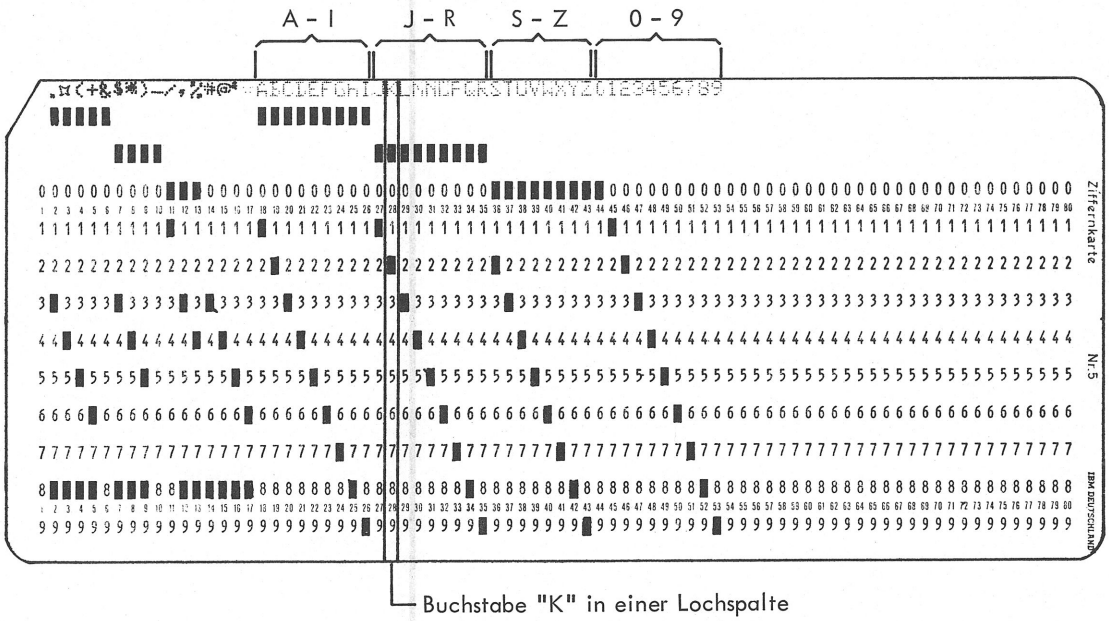
59 Es gibt auch mehrere Zeichen mit der gleichen Ziffernteilverschlüsselung wie der Buchstabe W.

Den gleichen Inhalt 6 im Ziffernteil haben die Zeichen _____.

F, O, 6

EBCD-CODE
(16 Sonderzeichen)

linkes Halbbyte (Zonenteil) hexadezimal	linkes Halbbyte (Zonenteil) binär	rechtes Halbbyte (Ziffernteil) 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		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						



Buchstabe "K" in einer Lochspalte

Abb. 23

60 In Abb. 23 ist die Tabelle um 16 Sonderzeichen und das Blank erweitert.
Das Zeichen % (Lochungen 0-4-8) hat hexadezimal die Verschlüsselung ____(ZZ)____.

Mit den Hexadezimalziffern "61" in einem Byte ist das Zeichen _____ dargestellt.

6 C

/

61 Ein Blank (eine Leerstelle) in der Lochkarte erhält im Hauptspeicher die Verschlüsselung:

a) binär: ____(ZZ)____

b) hexadezimal: ____(ZZ)____

01000000

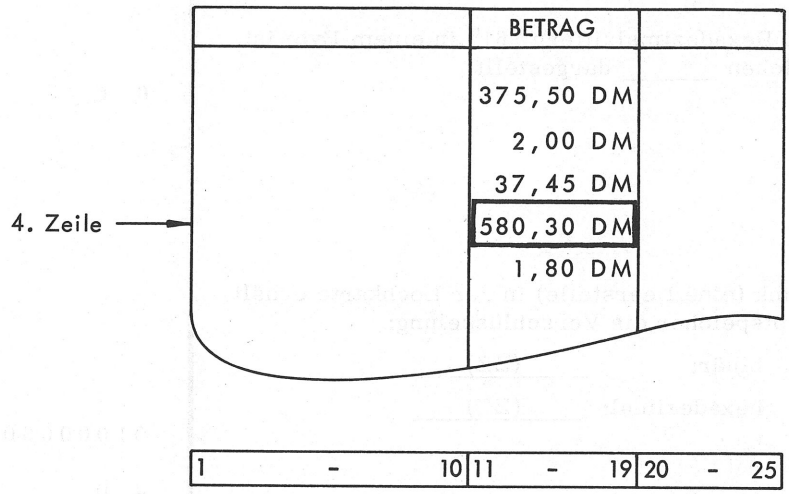
4 0

62 Zur Darstellung des Leerzeichens in einem Byte ist also nur das ___(Z)___ Bit von links auf EIN.

Anders ausgedrückt: Beim "blank" enthält nur Bit-Position ___(Z)___ eine binäre 1.

2.

1



		rechtes Halbbyte (Ziffernteil)															
hexa-dezimal →		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
linkes Halbbyte (Zonenteil)	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. 24

63 Zur Übung zeigt Abb. 24 einen Auszug aus einer Liste.

Der Inhalt des Feldes BETRAG soll in den Druckstellen __(Z)__ bis __(Z)__ gedruckt werden.

11 (bis) 19

Das Feld BETRAG im Hauptspeicher hat also eine Länge von ____(ZW)____.

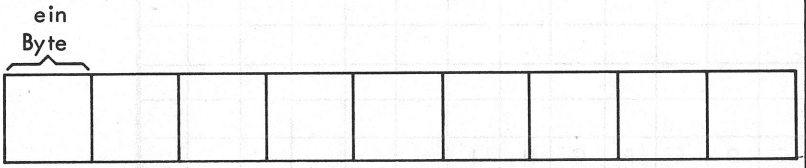
9 Bytes

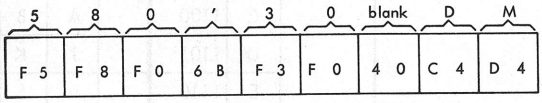
Es speichert __(Z)__ Zeichen.

9

64 Wir beschäftigen uns weiter mit der Liste in Abb. 24.

Geben Sie an, welchen Inhalt die 9 Bytes des Feldes BETRAG vor dem Drucken der 4. Zeile haben. (Hilfsmittel: Verschlüsselungstabelle. Angabe in Hexadezimalziffern.)





65 Wir werden die Übung jetzt umkehren. Wir nehmen an, daß 9 Bytes im Speicher folgenden Inhalt haben:

C 5	C 2	C 3	C 4	6 0	C 3	D 6	C 4	C 5

Geben Sie an, welches Wort beim Drucken dieses Feldes auf der Liste erscheint.

EBCD-CODE

EBCD-CODE

(Erweiterter 8-Bit IBM-BCD-CODE)

(16 Sonderzeichen)

			rechtes Halbbyte (Ziffernteil)															
	hexa-dezimal →		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
linkes Halbbyte (Zonenteil)	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. 25

"EBCD-CODE" heißt der Code, nach dem die Zeichen im System /360 verschlüsselt werden (vgl. Abb. 1). "EBCD" ist die Abkürzung von
"Erweiterter 8-Bit IBM- BCD - CODE"
(Binary Coded Decimal)

Es ist ein Code, der mit Hilfe des Binärsystems die Dezimalziffern, Buchstaben und Sonderzeichen jeweils in 8 Bits verschlüsselt. Der BCD-Code für das System /360 ist "erweitert" gegenüber einem anderen BCD-Code, der jedes Zeichen in 6 Bits verschlüsselt.

Wir haben uns hier also mit dem Aufbau des EBCD-Codes befaßt.

66 Die Verschlüsselungstabelle (Abb. 25) weist neben den besprochenen Zeichen viele leere Kästchen auf.

In jedem der beiden Halbbytes kann jede der 16 Hexadezimalziffern __(Z)__ bis __(Z)__ auftreten.

Bestimmen Sie aus der Tabelle, wieviele Bitkombinationen in einem Byte insgesamt möglich sind.

Sie kommen auf die Zahl __(Z)__.

0 (bis) F

256

Von den 256 (=16 x 16) Möglichkeiten wird nur ein kleiner Teil zur Darstellung der Buchstaben, Dezimalziffern und Sonderzeichen benutzt.

Aber auch die restlichen Möglichkeiten werden verwendet. Näheres hierüber erfahren Sie später in diesem Lehrgang.

FRAGEN ZUM KAPITEL II

Benutzen Sie für die Beantwortung der folgenden Fragen ein Extrablatt. Die richtigen Antworten finden Sie auf den anschließenden Seiten. Vergleichen Sie bitte erst, nachdem Sie alle Fragen beantwortet haben.

1. Wie heißt das kleinste Speicherelement im System /360 ?

2. Wie heißt die kleinste adressierbare Einheit im Speicher ?

3. Nennen Sie die Ziffern des Binärsystems.

4. a) Wieviel Hexadezimalziffern gibt es ?
   b) Wie heißen sie ?

5. Welche Stellenwerte sind den Bits in einem Halbbyte zugeordnet ?

6. Wieviel Bits genügen zum Speichern einer beliebigen Dezimalziffer 0 bis 9 ?

7. In wieviel Bits wird eine Dezimalziffer gespeichert, wenn sie aus einer Lochspalte eingelesen wird ?

8. Wie heißt der besprochene Code (Abkürzung) ?

EBCD-CODE

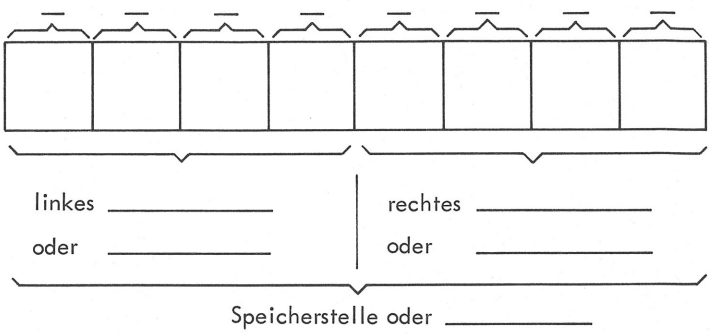
(Erweiterter 8-Bit IBM-BCD-CODE)
(16 Sonderzeichen)

			rechtes Halbbyte (Ziffernteil)															
	hexa-dezimal →		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
linkes Halbbyte (Zonenteil)	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. 25

9. Ergänzen Sie die fehlenden Bezeichnungen (aber bitte auf einem Extrablatt):

Bit-Position:



10. Ergänzen Sie folgende Tabelle mit Hilfe von Abb. 25:

2 Hexadezimal-ziffern		8 Binärziffern		gedrucktes Zeichen
6	C	0110	1100	%
6	0			
				3
		1110	0100	
		0100	0000	
				#

ANTWORTEN ZUM KAPITEL II

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

1. Magnetkern/Bit (Seite 35 - 39)

2. Speicherstelle/Byte (LE 35 - 42)

3. 0 und 1 (Seite 35 - 43)

4. a) 16 Hexadezimalziffern
   b) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
   (Seite 59 - 63)

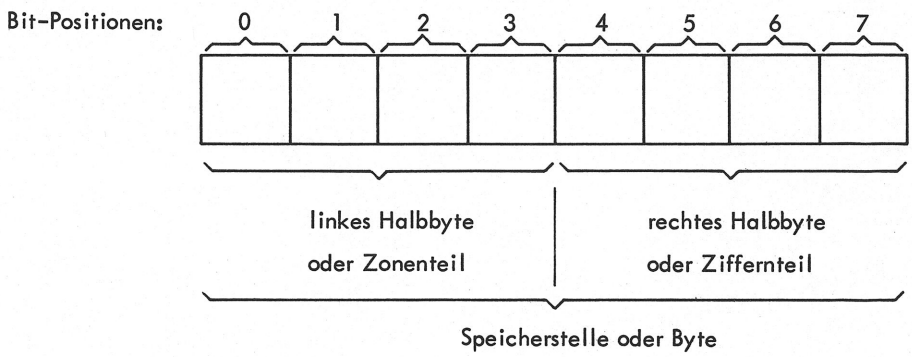
5. 1, 2, 4, 8 (Seite 45 - 55)

6. 4 Bits (Seite 41 - 57)

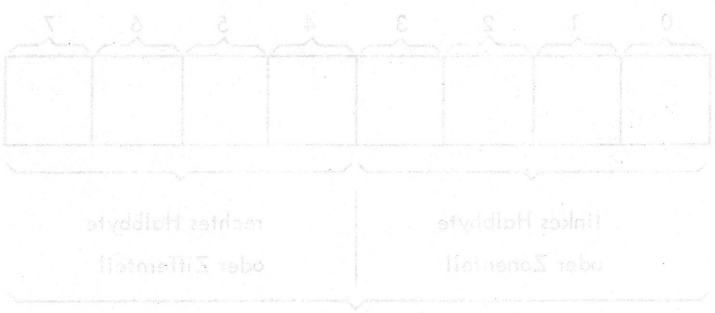
7. in 8 Bits (LE 35 - 36)

8. EBCD - CODE (LE 65 - 66)

9.



(LE 35 - 56)



10.

2 Hexadezimal-ziffern		8 Binärziffern		gedrucktes Zeichen
6	C	0110	1100	%
6	0	0110	0000	-
F	3	1111	0011	3
E	4	1110	0100	U
4	0	0100	0000	blank
7	B	0111	1011	#

(LE 57 - 66)

Schauen Sie sich nun noch das Merkblatt 201 (2 Seiten) an. -



Band 4, Seite 9 (MVC)

KAPITEL III

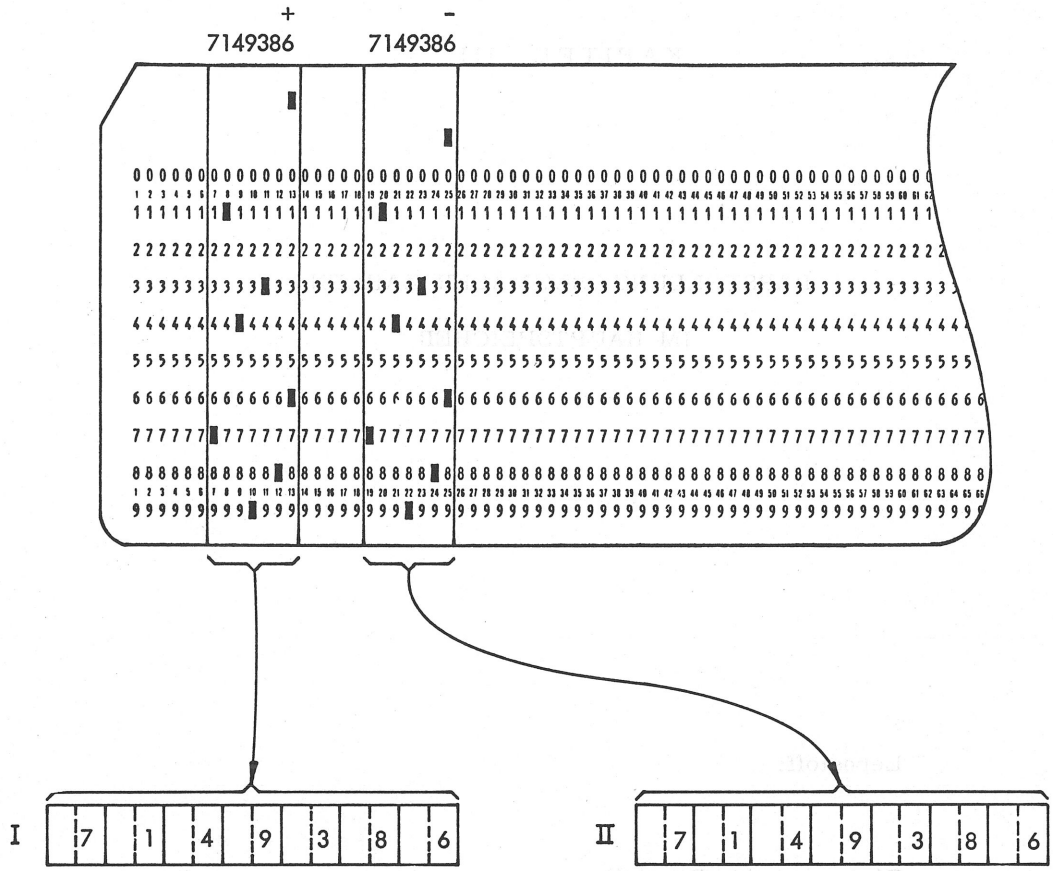
DARSTELLUNG DEZIMALER ZAHLEN IM HAUPTSPEICHER

Lernstoff:

Die ungepackte Darstellung

Die gepackte Darstellung

Vorzeichen



Zonen-lochung	Zonenteil Verschlüsselung
12	C
11	D
0	E
keine	F

Abb. 1

Wir haben uns schon einmal mit der Darstellung von Daten im Hauptspeicher beschäftigt. Von daher wissen wir:

Ein Zeichen aus einer Lochkartenspalte wird im Hauptspeicher nach dem EBCD-Code in einem Byte (= 8 Bits) verschlüsselt. Auch eine Dezimalziffer (0 bis 9) wird also in 8 Bits gespeichert.

Wir haben aber festgestellt, daß zur Darstellung einer der 10 Dezimalziffern schon 4 Bits genügen. Sogar jede der 16 Hexadezimalziffern 0 bis F kann in 4 Bits verschlüsselt werden.

Wir gehen jetzt von der Speicherung von Ziffern zur Speicherung von Z a h l e n über. Eine Dezimalzahl besteht aus einem Vorzeichen und verschieden vielen Dezimalziffern.

Wir lernen die zwei Formen der Speicherung von Dezimalzahlen im Hauptspeicher des Systems /360 kennen.

1 Betrachten Sie Abb. 1.

Die Lochkarte enthält 2 Zahlenfelder in den Spalten __(Z)__ bis __(Z)__ und __(Z)__ bis __(Z)__.

Beide Felder umfassen je __(Z)__ Spalten.

Die beiden Zahlen werden beim Einlesen der Karte in je __(ZW)__ gespeichert.

7 (bis) 13

19 (bis) 25

7

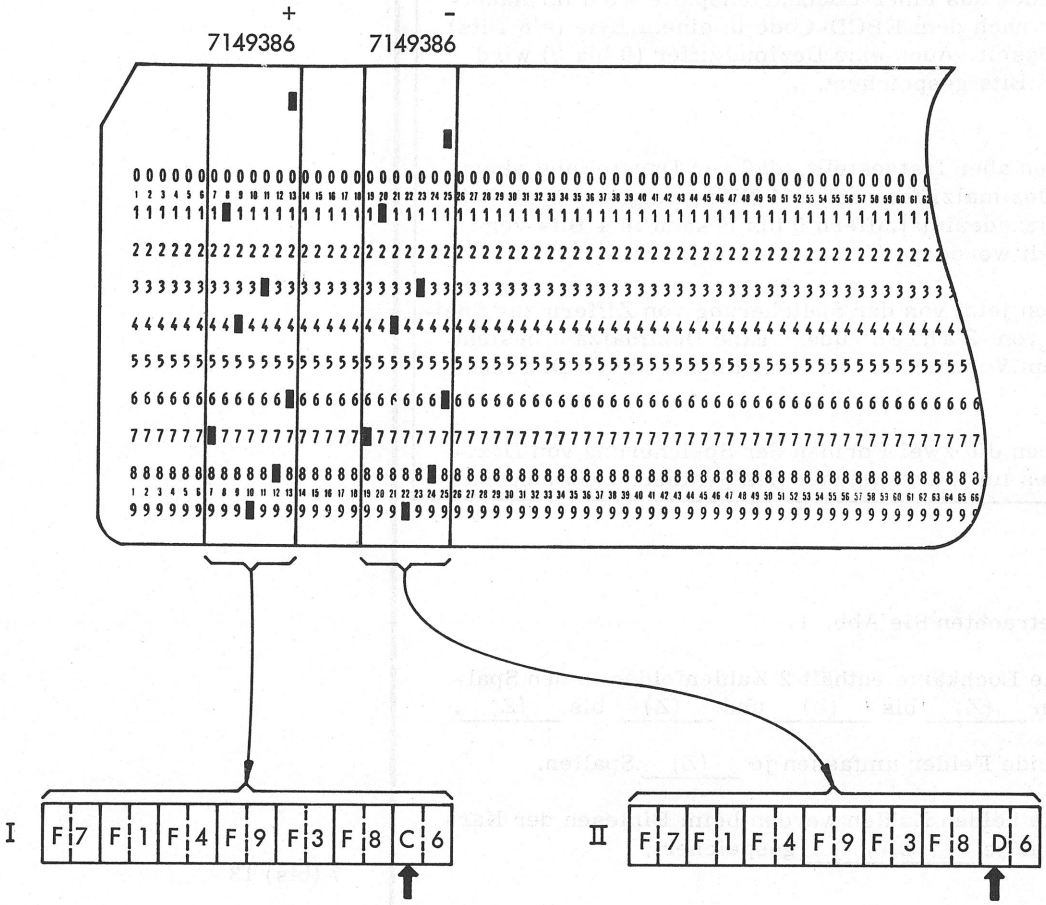
7 Bytes

2 Unterhalb der Karte in Abb. 1 finden Sie die beiden 7 Bytes großen Felder.
Die Ziffernteile sind schon ausgefüllt.

Nehmen Sie ein Extrablatt zur Hand. Schreiben Sie den Inhalt beider Felder auf. Die Tabelle unten links in Abb. 1 hilft Ihnen dabei.

Welche verschiedenen Hexadezimalziffern treten in den Zonenteilen auf?

C, D und F



Zonen-lochung	Zonenteil Verschlüsselung
12	C
11	D
0	E
keine	F

Abb. 2

3 Abb. 2 zeigt den vollständigen Inhalt der 7 Bytes. Die beiden Felder stimmen in allen 7 Ziffern und in 6 Zonen überein.

Nur die Zonen im _____ Byte sind verschieden.

Die Zonenlochung 12 ruft die Zonenteilverschlüsselung __(Z)__ hervor und die Zonenlochung 11 die Zonenteilverschlüsselung __(Z)__.

rechten

C

D

4 Sie kennen die Bedeutung der 12-er und 11-er Lochung in der Einerstelle einer Zahl.

Diese Lochungen kennzeichnen eine Zahl als positiv oder _____, stellen also das _____ der Zahl dar.

negativ

Vorzeichen

5 Nach dem Einlesen der Zahl in den Hauptspeicher ist das Vorzeichen in einem (Ziffernteil, Zonenteil) _____ verschlüsselt.

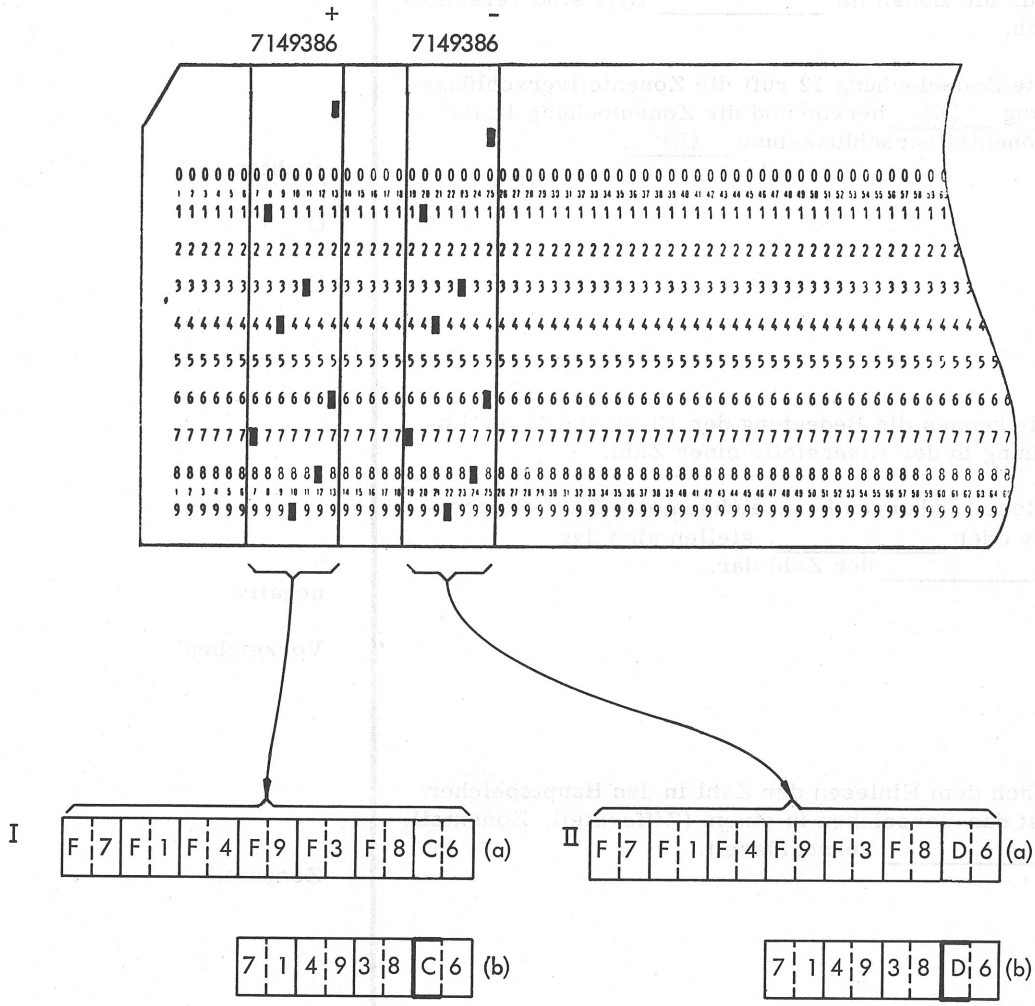
Zonenteil

6 Das Vorzeichen ist im Zonenteil der Einerstelle, also des rechten Bytes enthalten.

Das Vorzeichen im rechten Byte ist wichtig für spätere Rechnungen.

Der Inhalt F der übrigen Zonenteile ist (nötig, nicht nötig) _____ für die Charakterisierung der beiden Zahlen im Speicher.

nicht nötig



Zonen-lochung	Zonenteil Verschlüsselung
12	C
11	D
0	E
keine	F

Abb. 3

7 Abb. 3 zeigt die beiden Zahlen in einer neuen Form (b) gespeichert.

Diese Form enthält alle nötigen Angaben in __(Z)__ Bytes anstelle der __(Z)__ Bytes vorher bei Form (a).

Weggefallen sind die __(Z)__ Zonenteilverschlüsselungen F.

4

7

6

8 Diese neue Speicherungsform für Zahlen ist rationeller, weil sie Speicherplatz spart.

Für eine 7-stellige Zahl braucht man nur noch 4 Bytes, weil 6 Halbbytes = 3 Bytes fortfallen.

Für eine 11-stellige Zahl sind nur noch __(Z)__ Halbbytes, also __(Z)__ Bytes nötig.

12

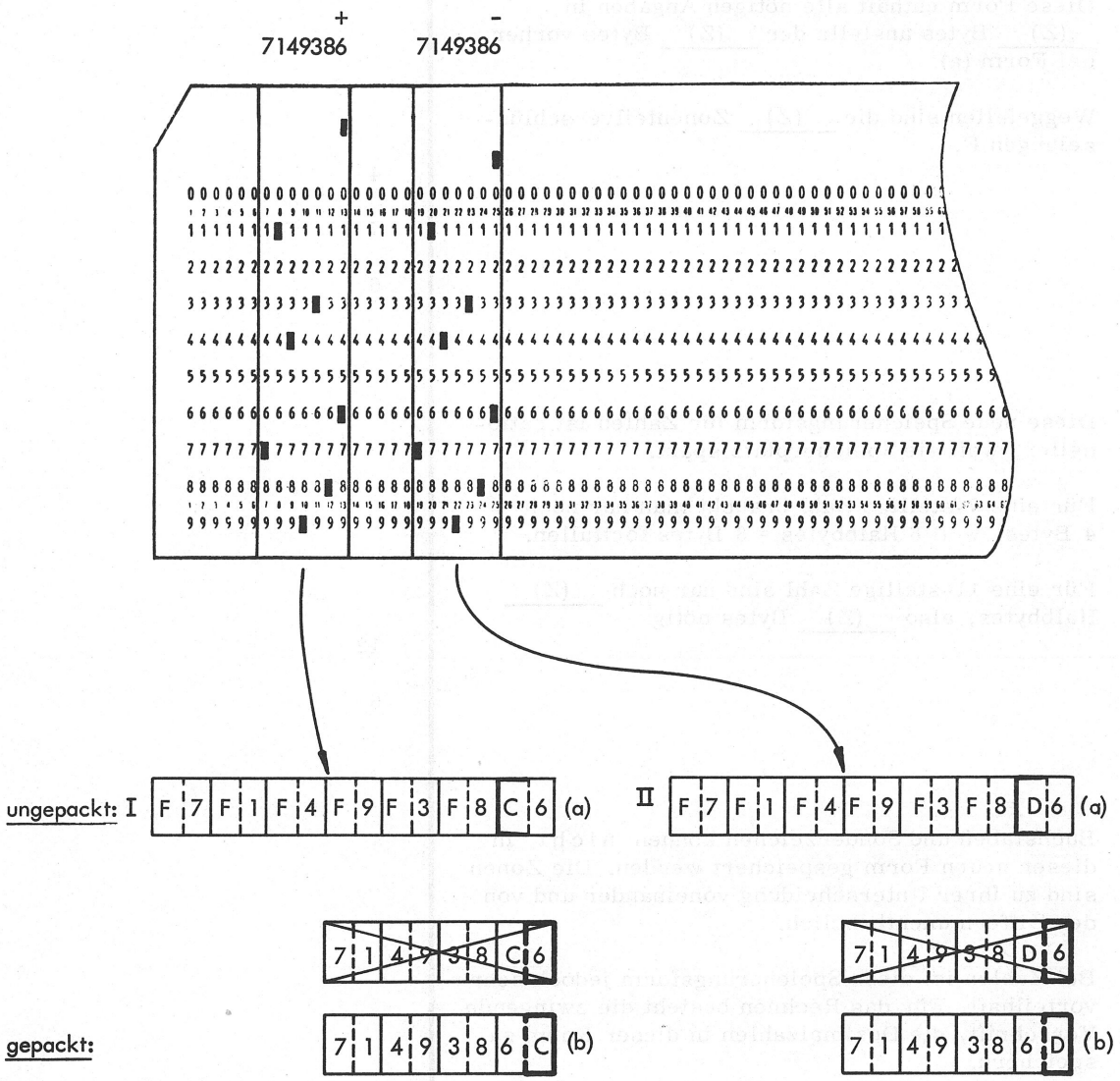
6

9 Buchstaben und Sonderzeichen können n i c h t in dieser neuen Form gespeichert werden. Die Zonen sind zu ihrer Unterscheidung voneinander und von den Ziffern unentbehrlich.

Bei Zahlen ist diese Speicherungsform jedoch sehr vorteilhaft. Für das Rechnen besteht die zwingende Vorschrift, die Dezimalzahlen in dieser Form zu speichern.

Vor arithmetischen Operationen müssen also die aus der Karte eingelesenen Zahlen in der Form (a, b) _____ gespeichert sein.

b



Zonen-lochung	Zonenteil Verschlüsselung
12	C
11	D
0	E
keine	F

Abb. 4

10 In Abb. 4 ist die bisherige Darstellung der Form b) durchgestrichen. Sie enthielt noch einen kleinen Fehler, der unten in Abb. 4 berichtigt ist.

Die Vorzeichenangaben C und D sind bei der neuen Speicherungsform im _____ Halbbyte des rechten Bytes gespeichert.

rechten

11 Abb. 4 zeigt Ihnen auch, wie man die beiden Speicherungsformen nennt.

Beim Einlesen aus der Lochkarte sind die Zahlen zunächst _____.

Zum Rechnen müssen die Zahlen dann _____ vorliegen.

ungepackt

gepackt

12 Das ungepackte Format heißt auch g e z o n t, weil jeder Ziffer eine Hexadezimalziffer im _____ beigegeben ist.

Zonenteil

13 Es ist Aufgabe des Programmierers, die Zahlen vor dem Rechnen in die _____ Speicherungsform zu bringen.

Für dieses "Packen" steht eine entsprechende Instruktion zur Verfügung.

gepackte

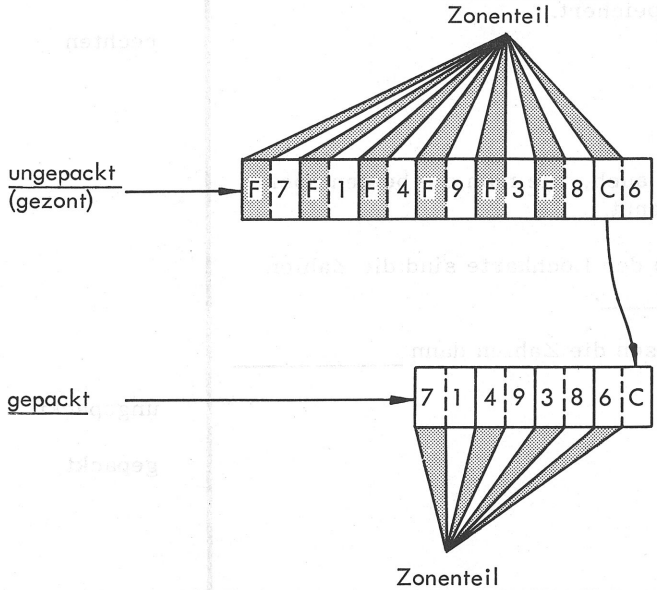

Zonenteil
ungepackt
(gezont)
F 7 F 1 F 4 F 9 F 3 F 8 C 6
gepackt
7 1 4 9 3 8 6 C
Zonenteil

Abb. 5

14 In Abb. 5 sind die beiden Speicherungsformen noch einmal gegenübergestellt.

Wir haben früher für die beiden Teile eines Bytes die Bezeichnungen rechtes Halbbyte oder Zifernteil und linkes Halbbyte oder Zonenteil eingeführt. Bei "ungepackt" enthält der Zonenteil immer eine Zone und der Zifernteil immer eine Ziffer.

Bei "gepackt" dagegen ist das anders. Hier enthält der Zonenteil (immer, manchmal, nie) _____ eine Zone.

Der Zifernteil enthält ___(WW)___ eine Ziffer.

nie

Sinngemäß:
immer außer im rechten Byte

STANZEN GEPACKTER DEZIMALZAHLEN

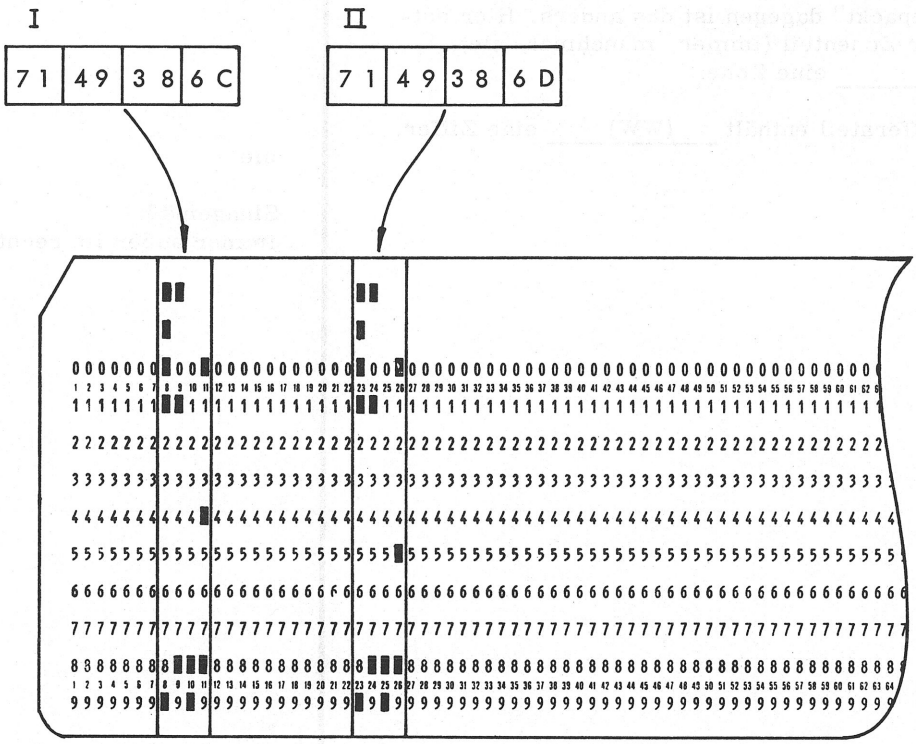


Abb. 6

Gepackte Daten können auch von externen Speichern, wie z. B. Lochkarten eingelesen und wieder ausgegeben werden. Wie Sie in den folgenden Lernelementen sehen werden, ist jedoch bei Lochkarten eine gepackte Speicherung im allgemeinen nicht sinnvoll.

15 In Abb. 6 sehen Sie zwei 4 Bytes große Speicherfelder mit gepackten Daten.

Der Inhalt dieser Felder wird in eine Lochkarte gestanzt.

Jedes der beiden Lochfelder umfaßt __(Z)__ Spalten.

4

16 Abb. 6 zeigt Ihnen den Inhalt der beiden Lochfelder nach dem Stanzen.

Die linke Spalte enthält __(Z)__ Lochungen und auch die übrigen Spalten weisen Mehrfachlochungen auf.

5

erweiterter 8-Bit IBM-BCD -Code	Loch-kartencode	Druck-zeichen	Hexa-dezimal
00000000	12,0,9,8,1		00
00000001	12,9,1		01
00000010	12,9,2		02
00000011	12,9,3		03
00000100	12,9,4		04
00000101	12,9,5		05
00000110	12,9,6		06
00000111	12,9,7		07
00001000	12,9,8		08
00001001	12,9,8,1		09
00001010	12,9,8,2		0A
00001011	12,9,8,3		0B
00001100	12,9,8,4		0C
00001101	12,9,8,5		0D
00001110	12,9,8,6		0E
00001111	12,9,8,7		0F
00010000	12,11,9,8		10
00010001	11,9		11
00010010	11		12
00010011			
000			
	12,11		
11	0,8,3	,	
01101100	0,8,4	%	6C
01101101	0,8,5		6D
01101110	0,8,6		6E
01101111	0,8,7		6F
01110000	12,11,0		70
01110001	12,11,0,9,1		71
01110010	12,11,0,9,2		72
01110011	12,11,0,9,3		73
01110100	12,11,0,9,4		74
01110101	12,11,0,9,5		75
01110110	12,11,0,9,6		76
01110111	12,11,0,9,7		77
01111000	12,11,0,9,8		78
01111001	8,1		79
01111010	8,2		7A
01111011	8,3	#	7B
01111100	8,4	@	7C
01111101	8,5	' (Anführungszeichen)	7D
01111110	8,6	=	7E
01111111	8,7		7F
10000000	12,0,8,1		80

Abb. 7

17 Aus einer Tabelle (vgl. den Auszug in Abb. 7) können Sie ablesen:

Dem Inhalt

7	1

bzw.

0111	0001

in einem Byte entsprechen die Lochungen (ZZ)
in einer Lochspalte und umgekehrt.

12, 11, 0, 9, 1

Aus diesem Beispiel ersehen Sie, daß es Mühe macht, gepackte Daten aus der Lochkarte abzulesen. Lochkarten werden daher im allgemeinen ungepackte Daten enthalten.

18 Wir überlegen uns, ob gepackte Daten auf Listen ausgegeben werden können.

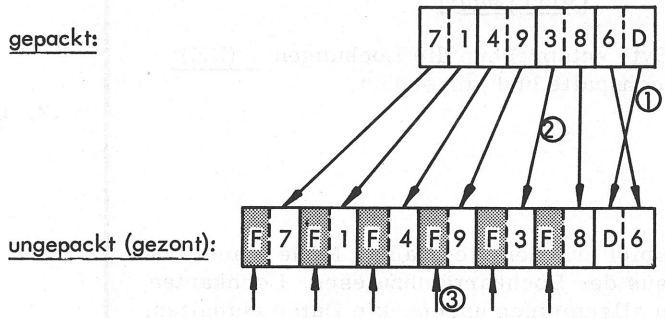
In einem gepackten Feld habe z. B. ein Byte den Inhalt "71". Wie Sie aus Abb. 7 ersehen, entspricht diesem Byte-Inhalt im Druckbild (die Ziffer 7, die Ziffer 1, kein Druckzeichen) _____.

kein Druckzeichen

19 Aus den genannten Gründen werden Zahlen im allgemeinen vor der Ausgabe in Karten oder Listen in die _____ Speicherungsform gebracht.

Hierfür steht dem Programmierer wieder eine entsprechende Instruktion zur Verfügung.

ungepackte/gezonte



Zonen-lochung	Zonenteil-Verschlüsselung
12	C
11	D
0	E
keine	F

Abb. 8

20 Abb. 8 zeigt, wie aus einer gepackten eine ungepackte Dezimalzahl entsteht.

① Die Vorzeichenverschlüsselung im rechten Byte wechselt vom _____ Halbbyte ins _____ Halbbyte.

② Die Ziffernverschlüsselungen werden in die (Ziffernteile, Zonenteile) _____ der 7 Bytes eingesteuert.

③ Die restlichen Zonenteile werden mit der Hexadezimalziffer __(Z)__ aufgefüllt.

rechten

linke

Ziffernteile

F

21 Wenn die ungepackte Zahl in Abb. 8 gestanzt werden soll, muß dafür in der Lochkarte ein Feld mit __(Z)__ Spalten zur Verfügung stehen.

Das Vorzeichen der Zahl ist (positiv, negativ) _____.

Die Einerstelle erhält also als Zonenlochung eine __(Z)__-er Lochung.

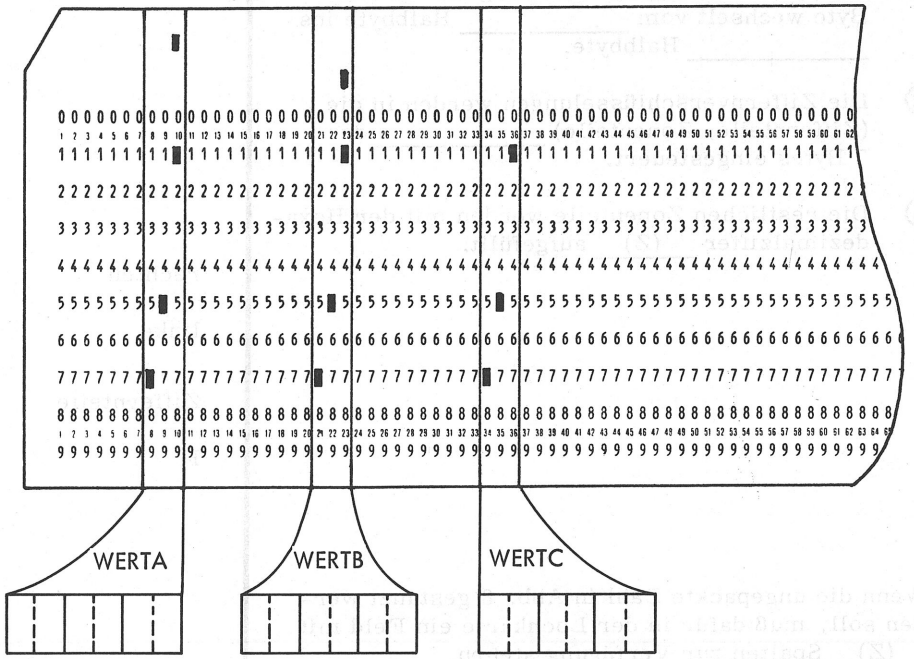
Alle anderen Spalten erhalten _____ Zonenlochung.

7

negativ

11

keine

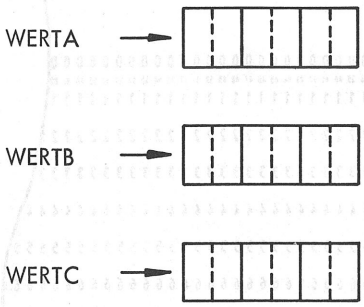


Zonen-lochung	Zonenteil-Verschlüsselung	Vorzeichen
12	C	+
11	D	-
keine	F	(+)

Abb. 9

22 Das Beispiel in Abb. 9 gibt Ihnen Gelegenheit, Ihre Kenntnisse über gepackte und ungepackte Dezimalzahlen anzuwenden.

Geben Sie an, welchen Inhalt die Felder WERTA, WERTB und WERTC im Hauptspeicher erhalten (hexadezimal):



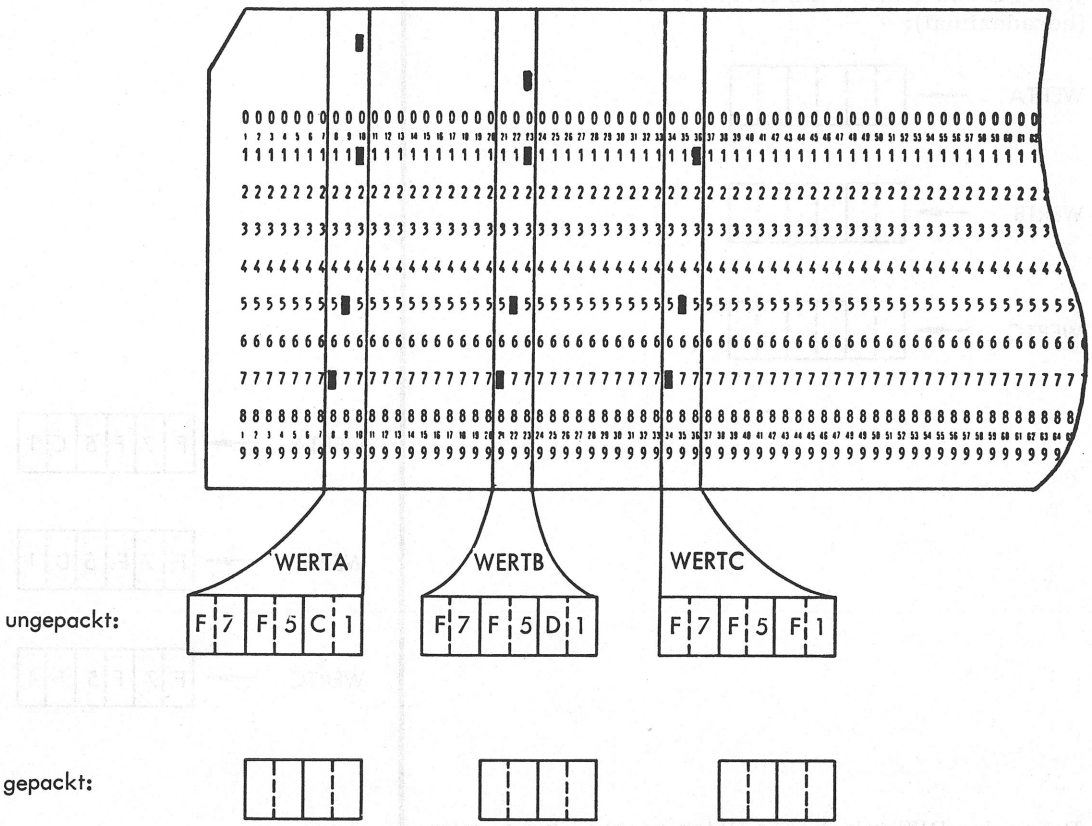
WERTA → F 7 F 5 C 1

WERTB → F 7 F 5 D 1

WERTC → F 7 F 5 F 1

23 Bevor das DVS mit diesen Daten rechnen kann, müssen sie in die gepackte Speicherungsform gebracht werden.

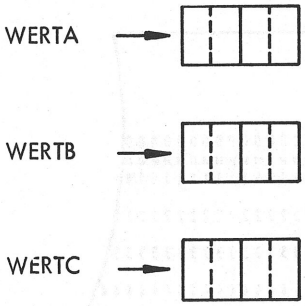
Dann benötigen die Zahlen nicht mehr 3 Bytes, sondern nur noch 2 Bytes (ZW)

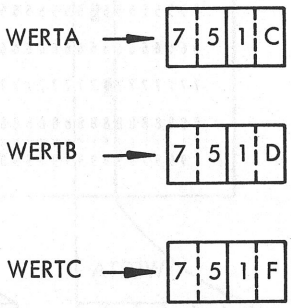


Zonen-lochung	Zonenteil-Verschlüsselung	Vorzeichen
12	C	+
11	D	-
keine	F	(+)

Abb. 10

24 Entwickeln Sie die gepackte Form von WERTA, WERTB und WERTC.



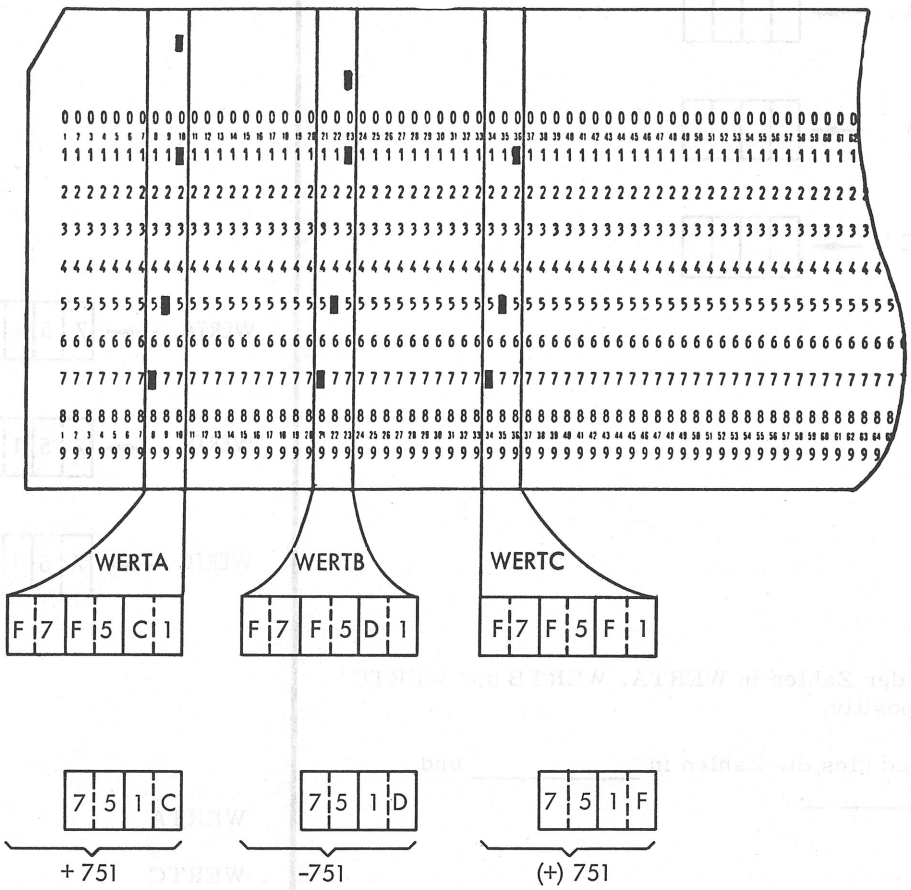


25 Zwei der Zahlen in WERTA, WERTB und WERTC sind positiv.

Es sind dies die Zahlen in _____ und _____.

WERTA

WERTC



Zonen-lochung	Zonenteil-Verschlüsselung	Vorzeichen
12	C	+
11	D	-
keine	F	(+)

Abb. 11

26 In Abb. 11 sehen Sie die gepackten Zahlen.

Das Feld WERTC hat keine Vorzeichenlochung in der Lochkarte. Kommt es aber zu einer Rechenoperation, so gilt die Zahl in WERTC als positiv.

Es ist völlig ohne Einfluß auf das Rechenergebnis, ob die Vorzeichenverschlüsselung einer positiven Zahl __(Z)__ oder __(Z)__ lautet.

C (oder) F

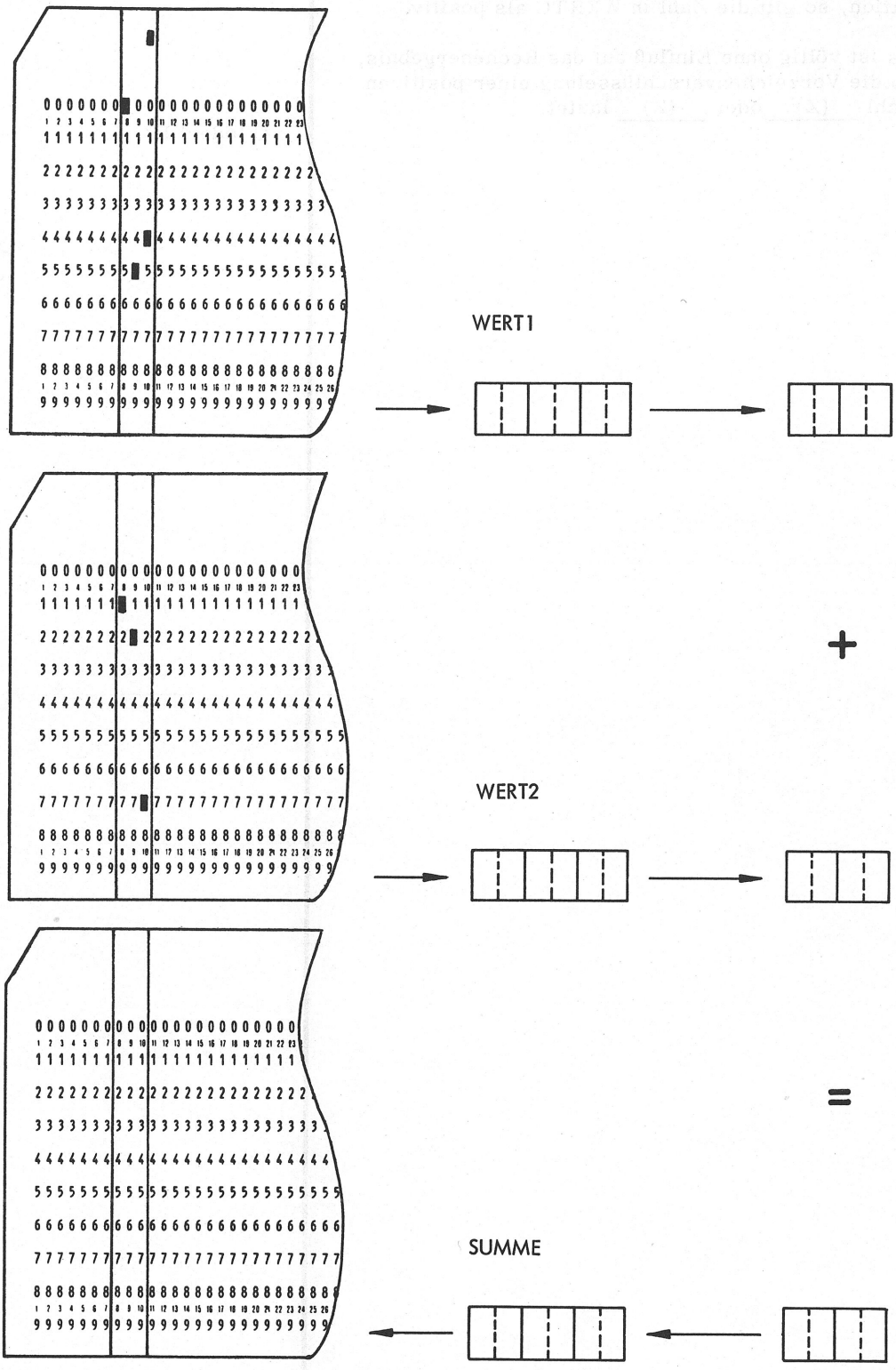


Abb. 12

27 Betrachten Sie Abb. 12.

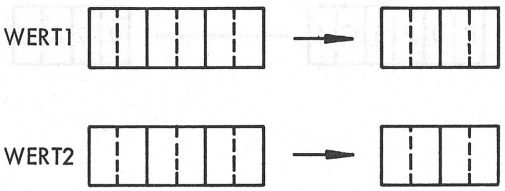
Die Werte aus den beiden oberen Lochkarten sollen addiert werden.

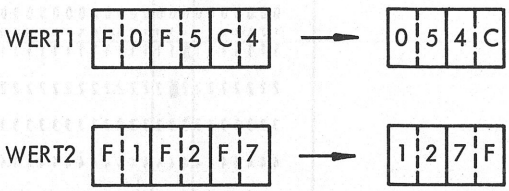
Es sind dies die beiden (positiven, negativen) _____ Zahlen __(Z)__ und __(Z)__.

positiven

54 (und) 127 / +54 (und) (+)127

28 Geben Sie bitte den ungepackten und den gepackten Inhalt der Felder im Speicher an:





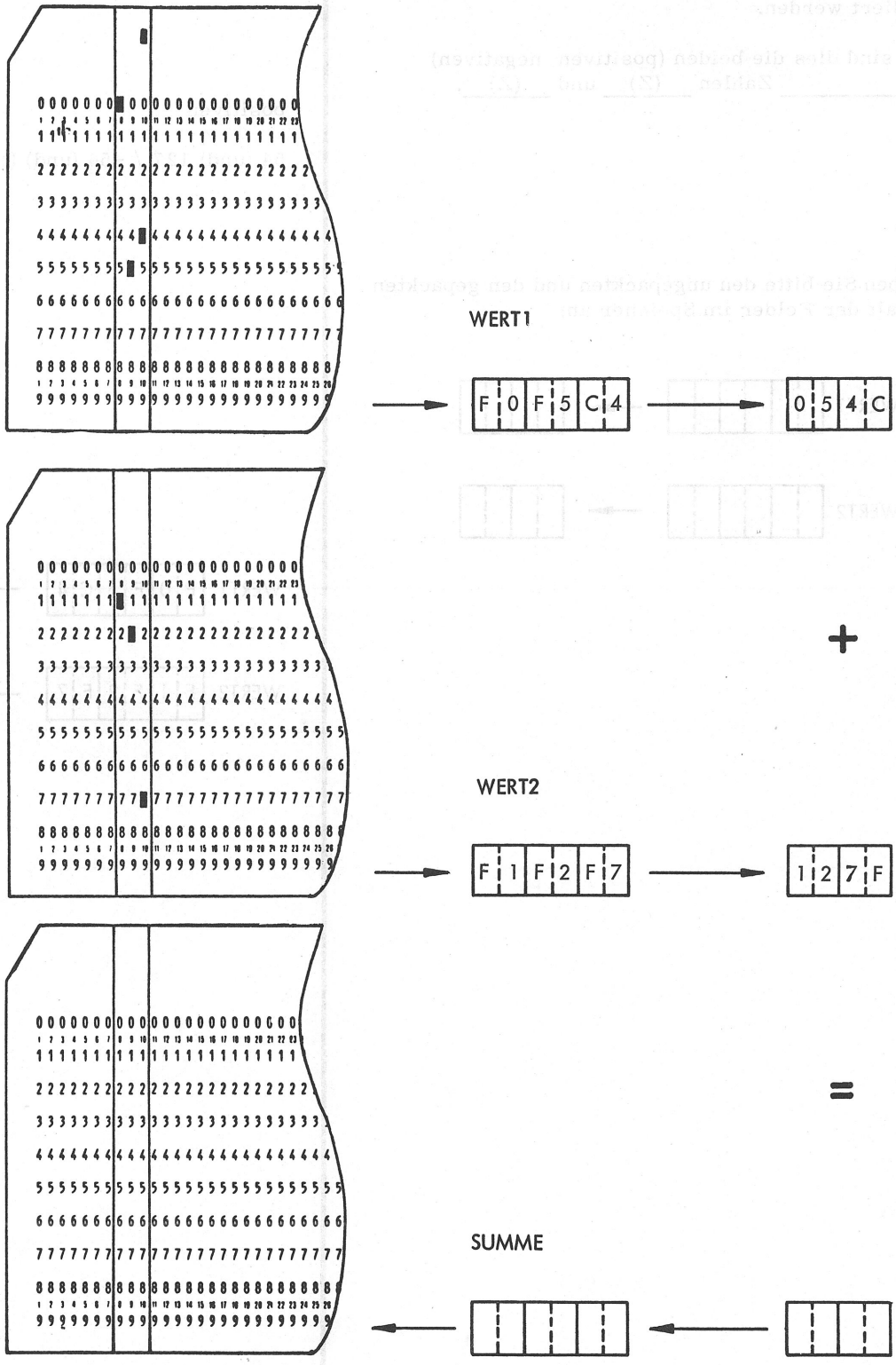


Abb. 13

29 Bis hierher ist das Beispiel in Abb. 13 ergänzt.

Die Summe der beiden positiven Zahlen 54 und 127 ist __(Z)__.

Dieses Ergebnis ist (positiv, negativ) _____.

Zur Verschlüsselung eines positiven Vorzeichens kommen die Hexadezimalziffern __(Z)__ und __(Z)__ in Frage.

181

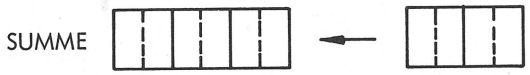
positiv

C (und) F

30 Für Rechenergebnisse, die im DVS gewonnen werden, gilt die Regel:

Ein positives Ergebnis hat immer die Hexadezimalziffer C als Vorzeichenverschlüsselung.

Jetzt können Sie die untere Zeile in Abb. 13 ergänzen:





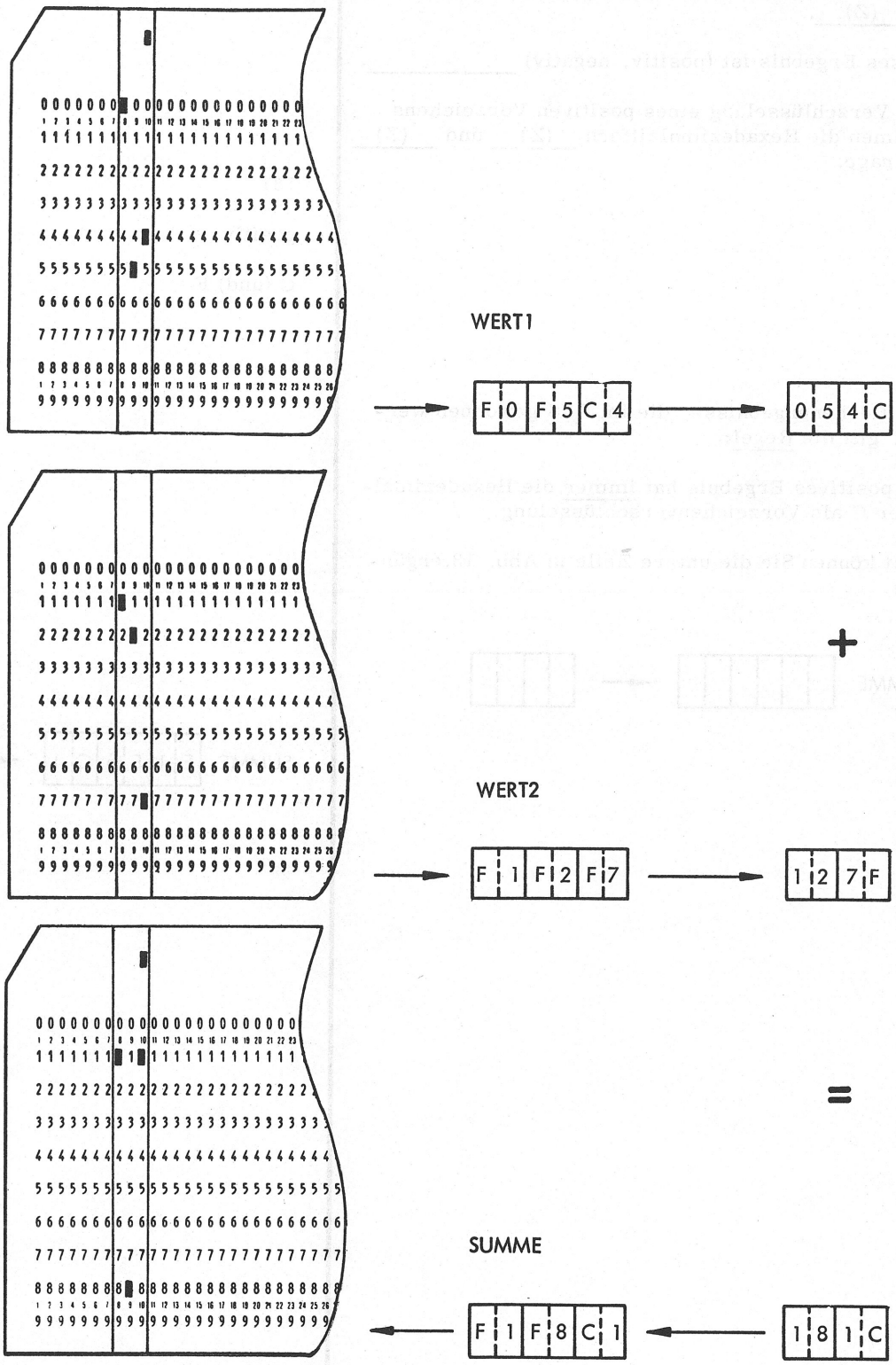


Abb. 14

31 Abb. 14 zeigt den vollständigen Ablauf der Lösung dieser Aufgabe.

Die Summe erhält beim Stanzen der Ausgabekarte die Zonenlochung __(Z)__.

12

32 Wir halten fest:

Beim Stanzen eines positiven Rechenergebnisses (Vorzeichenverschlüsselung "C") erhält das Lochfeld die Vorzeichenlochung 12.

Ein negatives Rechenergebnis (Vorzeichenverschlüsselung "D") erhält die Vorzeichenlochung __(Z)__.

11

VOLLSTÄNDIGE TABELLE DER VORZEICHENVERSCHLÜSSELUNGEN BEI DEZIMALZAHLEN

Verschlüsselung hexadezimal	Verschlüsselung binär	Vorzeichen	entsteht bei Rechenergebnis	entspricht Zonenlochung in Zeile
A	1010	+		
B	1011	-		
C	1100	+	positiv	12
D	1101	-	negativ	11
E	1110	+		
F	1111	(+)		(keine)

Abb. 15

Abb. 15 führt alle Verschlüsselungen auf, die bei einer Dezimalzahl als Vorzeichen gedeutet werden.

An sich genügen C, D und F, um ein positives oder negatives Vorzeichen einschließlich seiner Verschlüsselung in der Lochkarte (12-er, 11-er oder keine Zonenlochung) eindeutig darzustellen.

Um jedoch zu erreichen, daß jede der 16 Hexadezimalziffern 0 bis F, also jede Verschlüsselung in einem Halbbyte, entweder eine Dezimalziffer oder ein Vorzeichen ausdrückt, wird auch den Hexadezimalziffern A, B und E eine Bedeutung als positives bzw. negatives Vorzeichen gegeben.

Diese Zuordnung hat maschinentechnische Gründe. Für uns ist nur der schraffierte Teil der Abb. 15 von praktischer Bedeutung.

33 Wir halten fest:

Wird eine Dezimalzahl aus der Lochkarte eingelesen, so wird das Vorzeichen durch eine der Hexadezimalziffern (ZZ) verschlüsselt.

C, D, F

Entsteht eine Dezimalzahl als Rechenergebnis im DVS, so ist das Vorzeichen durch eine der Hexadezimalziffern (ZZ) dargestellt.

C, D

FRAGEN ZUM KAPITEL III

Bei den folgenden Fragen ist jeweils nur eine der Aussagen A, B, C und D richtig. Lesen Sie erst eine Frage vollständig durch, ehe Sie sich entscheiden. Notieren Sie sich dann jeweils den entsprechenden Buchstaben (bitte auf einem Extrablatt).

Beantworten Sie erst alle Fragen, ehe Sie Ihre Lösungen mit den Antworten auf den folgenden Seiten vergleichen.

1. A Jeder beliebige Inhalt eines Feldes im Speicher kann sowohl gepackt als auch ungepackt gespeichert werden.

   B Die gepackte Darstellung unterscheidet sich von der ungepackten durch Vertauschen der Zonen- und Ziffernteile in jedem Byte.

   C Die gepackte Speicherungsform ist nur für Dezimalzahlen sinnvoll.

   D "Gezont" ist das Gegenteil von "ungepackt".

2. A Das Umwandeln der Zahlen von der ungepackten in die gepackte Darstellung und umgekehrt führt die Maschine ohne Zutun des Programmierers aus, sobald es nötig wird.

   B Beim Übergang von der gepackten in die ungepackte Darstellung ändert sich die Vorzeichenverschlüsselung der Zahl.

   C Beim Übergang von der ungepackten in die gepackte Darstellung fällt der Inhalt aller Zonenteile außer dem Zonenteil des rechten Bytes fort.

   D Gepackte Zahlen werden beim Stanzen automatisch entpackt.

3. A Eine gepackte Zahl hat kein Vorzeichen.

B Die Hexadezimalziffern C und D haben als Vorzeichen beide die Bedeutung " + ".

C Die Hexadezimalziffern C und F haben als Vorzeichen beide die Bedeutung " + " und rufen beim Stanzen eine Zonenlochung hervor.

D Der Hexadezimalziffer C als Vorzeichendarstellung entspricht eine Zonenlochung in Zeile 12.

4. A Als Vorzeichendarstellung erkennt das DVS nur die Hexadezimalziffern C, D und F an.

B Jede der Hexadezimalziffern A bis F stellt eines der Vorzeichen " + " oder " - " dar.

C Zum Rechnen im DVS benötigt eine Zahl kein Vorzeichen.

D Ob eine Hexadezimalziffer ein Vorzeichen oder eine Dezimalziffer darstellt, richtet sich nach ihrer Stellung innerhalb des Zahlenfeldes.

5. Entnehmen Sie Ihren Arbeitsmitteln Übung 201.

Geben Sie den Inhalt der 6 eingezeichneten Speicherfelder an und die Lochungen für das Feld "SUMME" in der Ausgabekarte.

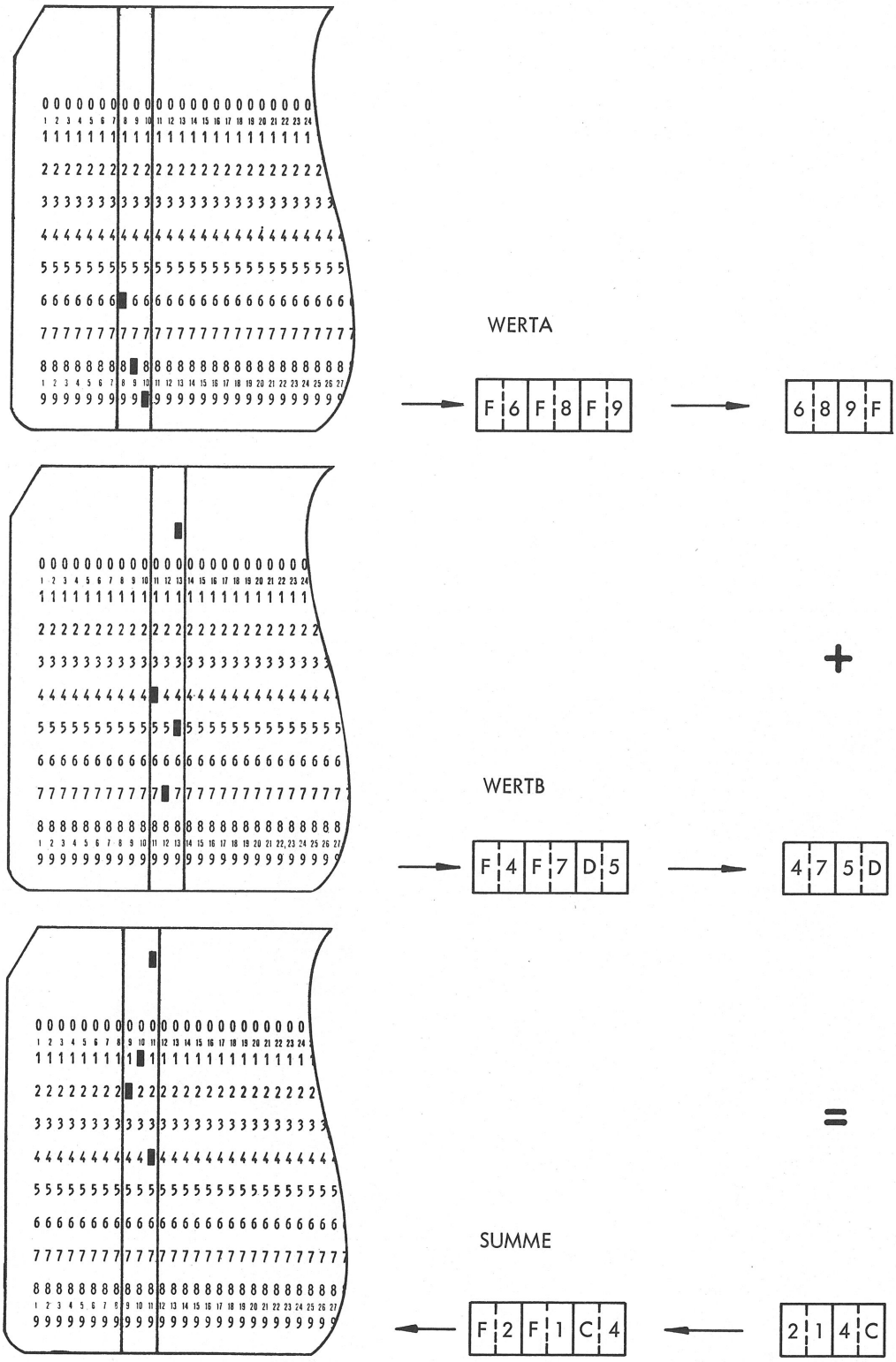
Hinweis:
Denken Sie bitte an die Rechenregel:

$$a + (-b) = a - b$$

ANTWORTEN ZUM KAPITEL III

1.	C	(LE 7 - 14)
2.	C	(LE 13 - 21)
3.	D	(LE 32 - 33)
4.	B	(LE 32 - 33)
5.	siehe folgendes Blatt	(LE 22 - 31)

5.



Merkblatt 202 (3 Seiten) gibt Ihnen eine Zusammenfassung des Lernstoffes zu diesem Kapitel. -

Band 4, Seite 146 (PACK)



KAPITEL IV

DIE DUALDATEN UND IHRE FORMATE

Lernstoff:

Das duale Zahlensystem

Das Prinzip des dualen Rechnens

Die Datenformate der Dualdaten

Die Register in der Zentraleinheit

Duales Rechnen im System /360 am Beispiel der Subtraktion

ein Bit	ein Bit	ein Bit	ein Bit
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
Stellenwerte: 8	4	2	1

Abb. 1

Wir befassen uns in diesem Kapitel mit dem dualen Zahlensystem und mit den Formaten, die es für Dualdaten im System /360 gibt.

Sie lernen in diesem Lehrgang das duale Zahlensystem kennen, weil es Voraussetzung für das Verständnis der Adressierung des Hauptspeichers ist. Außerdem ist es im System /360 möglich, neben dezimalen Daten auch duale Daten zu verarbeiten. Einige der hierfür zur Verfügung stehenden Operationen lernen Sie noch in diesem Lehrgang kennen.

Wenn Sie mit dem dualen Zahlensystem und dem Rechnen (Addition) in diesem Zahlensystem schon vertraut sind, dann gehen Sie gleich zu LE 16. Dort können Sie Ihre Kenntnisse überprüfen.

Das Binär- oder Dualsystem entspricht der Grundkonzeption des Hauptspeichers in ganz besonderer Weise. Die Magnetisierung des Magnetkerns in die eine oder die andere Richtung, das "EIN" oder "AUS" eines Bits, das sind jeweils zwei Möglichkeiten, die sich durch die zwei Ziffern (0 und 1) des Dualsystems ausdrücken lassen.

Betrachten Sie bitte Abb. 1. -

Aus dem Kapitel II ("Verschlüsselung der Zeichen") dieses Bandes kennen Sie bereits die duale Darstellung der Dezimalziffern 0 - 9 in vier Bits sowie die Stellenwerte 1, 2, 4, 8 dieser Bits.

In den folgenden Lernelementen behandeln wir nun das duale Zahlensystem als solches, ohne die Begrenzung auf ein Halbbyte.

$$345 = 3 \cdot 100 + 4 \cdot 10 + 5 \cdot 1$$

$$345 = 3 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$$

Anmerkung:
10^0 ist eine mathematische Schreibweise für 1.

Abb. 2

1 Die Bedeutung einer Ziffer in einer Dezimalzahl hängt davon ab, welche Stellung sie innerhalb der Zahl einnimmt.

Die Zahl 345 setzt sich zusammen aus den Werten 300 + 40 + 5. Die Ziffern 3, 4, 5 haben also einen unterschiedlichen S t e l l e n w e r t .

Die Ziffer 5 bezeichnet die Anzahl der Einer.
Die Ziffer 4 bezeichnet die Anzahl der _____.
Die Ziffer 3 bezeichnet die Anzahl der _____.

Zehner

Hunderter

2 Betrachten Sie bitte Abb. 2. -

Statt der Einer, Zehner, Hunderter, usw. kann man auch die Zehnerpotenzen 10^0, 10^1, 10^2, usw. verwenden.

Bestimmen Sie den Stellenwert der Ziffern in den Zahlen 829 und 3 517. Benutzen Sie die Zehnerpotenzen:

$829 = 8 \cdot 10^2 +$ (ZZ)

3517 = (ZZ)

$8 \cdot 10^2 + 2 \cdot 10^1 + 9 \cdot 10^0$

$3 \cdot 10^3 + 5 \cdot 10^2 + 1 \cdot 10^1 + 7 \cdot 10^0$

3 Das Zehner- oder Dezimalsystem baut auf der Basis 10 auf. Man kann nun auch Zahlensysteme bilden, die nicht auf der Basis 10, sondern auf einer beliebigen anderen Basis aufbauen.

So kannten z. B. die Babylonier das Sexagesimalsystem, ein Zahlensystem, das auf der Basis 60 aufbaute und 60 verschiedene Zahlzeichen benutzte.

Es gibt das Hexadezimal- oder Sedezimalsystem, ein Sechzehnersystem, das auf der Basis (Z) aufbaut und zur Darstellung seiner Zahlen 16 Ziffern benötigt.

16

4 Das Binär- oder Dualsystem ist ein "Zweiersystem".

Das Binärsystem baut auf der Basis __(Z)__ auf und benötigt für die Darstellung seiner Zahlen __(Z)__ Ziffern. Es benutzt die Ziffern __(Z)__ und __(Z)__.

2

2

0 (und) 1

5 Im Dezimalsystem bezeichnen die Ziffern je nach ihrer Stellung die Anzahl der Einer, Zehner, Hunderter, Tausender usw. - Im Dualsystem bezeichnen die Ziffern die Anzahl der Einer, Zweier, Vierer, Achter usw.

Wie im Dezimalsystem kann man auch im Dualsystem die Stellenwerte in Potenzen ausdrücken.

Schreiben Sie bitte die Stellenwerte 1, 2, 4, 8 im Dualsystem als Zweierpotenzen:

_____, _____, _____, _____.

2^0, 2^1, 2^2, 2^3

6

8	4	2	1
2^3	2^2	2^1	2^0

Sie kennen nun bereits 4 Stellenwerte im Dualsystem. Wir können also schon die Dualzahl 1111 in das Dezimalsystem umwandeln.

1111 das ist: $1 \cdot 2^0 = 1$

$1 \cdot 2^1 = 2$

$1 \cdot 2^2 = 4$

$1 \cdot 2^3 = 8$

Summe: __(Z)__

15

UMWANDELN

DUAL ⟶ DEZIMAL

DUAL		DEZIMAL
100101_2	⟶	37_{10}

Der Index 2 bzw. 10 kennzeichnet eine Zahl als Dualzahl bzw. Dezimalzahl.

Stellenwerte im Dualsystem:

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

100101_2 das ist:

$$
\begin{aligned}
1 \cdot 2^0 &= 1 \\
0 \cdot 2^1 &= 0 \\
1 \cdot 2^2 &= 4 \\
0 \cdot 2^3 &= 0 \\
0 \cdot 2^4 &= 0 \\
1 \cdot 2^5 &= \underline{32} \\
&\quad 37_{10}
\end{aligned}
$$

Abb. 3

7 In Abb. 3 ist die Umwandlung einer Dualzahl in eine Dezimalzahl noch einmal systematisch dargestellt.

Verfolgen Sie bitte die Umwandlung der Dualzahl 100101 in die Dezimalzahl 37 in Abb. 3. -

Wandeln Sie nun selbst nach dem Muster in Abb. 3 die Dualzahl 10101 in eine Dezimalzahl um. Benutzen Sie ein Extrablatt.

$1 \cdot 2^0 = 1$

$0 \cdot 2^1 = 0$

$1 \cdot 2^2 = 4$

$0 \cdot 2^3 = 0$

$1 \cdot 2^4 = 16$

21

8 Wandeln Sie in gleicher Weise die folgenden Dualzahlen in Dezimalzahlen um:

a) 11111

b) 100111

c) 11010001

a) 31

b) 39

c) 209

UMWANDELN

DEZIMAL ⟶ DUAL

DEZIMAL 137_{10} ⟶ DUAL 10001001_2

Stellenwerte im Dualsystem:

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$137_{10} = 1\cdot2^7 + 0\cdot2^6 + 0\cdot2^5 + 0\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 0\cdot2^1 + 1\cdot2^0$$

$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

$$1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1$$

Abb. 4

9 In Abb. 4 ist die Umwandlung der Dezimalzahl 137 in die Dualzahl 10001001 dargestellt. Verfolgen Sie bitte den Gang der Umwandlung. -

In 137 ist 2^7 1 mal enthalten, als Rest bleibt 9.

In 9 ist 2^6 0 mal enthalten, ebenso 2^5 und 2^4. Wir müssen also für diese Stellenwerte jeweils eine 0 schreiben.

2^3 ist in 9 1 mal enthalten, als Rest bleibt 1.

In 1 ist 2^2 und 2^1 jeweils 0 mal enthalten.

2^0 jedoch ist in 1 1 mal enthalten, also schreiben wir eine 1.

Auf diese Weise haben wir die Dezimalzahl 137 in die Stellenwerte des Dualsystems aufgegliedert und erhalten als Ergebnis die Dualzahl __(Z)__.

10001001

10 Erarbeiten Sie sich bitte auf die gleiche Art (vgl. Abb. 4) die binäre Darstellung der Dezimalzahl 38.

100110

11 Wandeln Sie bitte auf die gleiche Art folgende Dezimalzahlen um:

a) 19

b) 73

c) 159

a) 10011

b) 1001001

c) 10011111

ADDITION IM DUALEN ZAHLENSYSTEM

Arithmetische Regeln:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 (und Übertrag)

Beispiel einer Addition:

	1001011_2	≙	75_{10}
	$+\ 1011110_2$	≙	$+\ 94_{10}$
(Überträge:)	$1\ 1111_2$		1_{10}
Ergebnis:	10101001_2	≙	169_{10}

Abb. 5

In den folgenden Lernelementen befassen wir uns mit dem Rechnen im Dualsystem. Sie sollen nur einen kleinen Einblick in dieses Gebiet erhalten. Wir beschränken uns daher auf die Addition im Dualsystem.

In Abb. 5 finden Sie die arithmetischen Regeln, nach denen im Dualsystem addiert wird. Rechnen Sie bitte anhand dieser Regeln das Beispiel in Abb. 5 durch. -

Wenn Sie glauben, daß Sie eine Addition wie in dem nebenstehenden Beispiel schon selbständig durchführen können, dann fahren Sie bitte mit LE 15 fort.

12 Betrachten Sie bitte noch einmal die arithmetischen Regeln in Abb. 5.

1 + 1 = 0 (und Übertrag) ist hier zu lesen.

Bedenken Sie, daß wir im Dualsystem nur die Ziffern 0 und 1 benutzen. - Im Dezimalsystem schreiben wir beim Zusammenzählen von 5 und 5 nur die 0 und behalten eine "1" im Sinn. Diese "1" nennen wir den Übertrag.

Wenn wir im Dualsystem 1 und 1 addieren, schreiben wir ebenfalls die 0 und erhalten einen

_____ Übertrag

13 Addieren Sie nun nach den arithmetischen Regeln in Abb. 5 die beiden Dualzahlen.

```
  11111
+  1100
------
```

Ergebnis: (Z) 101011

14 In unserer kleinen Aufgabe ist durch einen Übertrag der Fall eingetreten, daß 1 + 1 + 1 addiert werden mußte.

1 + 1 + 1 = (Z) 1 und (ZW) 1 Übertrag

ADDITION IM DUALEN ZAHLENSYSTEM

Arithmetische Regeln:

0	+	0	=	0	
0	+	1	=	1	
1	+	0	=	1	
1	+	1	=	0	(und Übertrag)

Beispiel einer Addition:

	1001011_2	$\hat{=}$	75_{10}
	$+\ 1011110_2$	$\hat{=}$	$+\ 94_{10}$
(Überträge:)	1 1111		1
Ergebnis:	10101001_2	$\hat{=}$	169_{10}

Abb. 5

15 Addieren Sie bitte die beiden Dualzahlen

```
  1001111
+  111100
---------
```

Ergebnis: (Z)

10001011

Wenn Sie das Ergebnis nicht richtig ermittelt haben, dann gehen Sie bitte zurück zu LE 12.

16 Prüfen Sie nun selbst durch diesen kleinen Test, ob Sie mit dem dualen Zahlensystem und mit der Addition in diesem Zahlensystem vertraut sind.

Lösen Sie bitte erst alle Aufgaben, bevor Sie die Ergebnisse auf der folgenden Seite überprüfen.

1. Wandeln Sie die Dezimalzahl 20 in eine Dualzahl um.

2. Addieren Sie die Dualzahlen 101110 und 1111.

3. Wie lauten die folgenden Dualzahlen im Dezimalsystem?

   a) 101110

   b) 1111

   c) 111101

1. 10100 (LE 9 - 11)

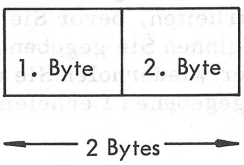
2. 111101 (LE 12 - 15)

3. a) 46
   b) 15 } (LE 7 - 8)
   c) 61

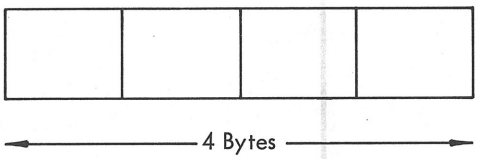
Sollten Ihnen Fehler unterlaufen sein, dann beseitigen Sie zunächst alle Unklarheiten, bevor Sie fortfahren. Beginnen Sie gegebenenfalls mit LE 1 oder wiederholen Sie die in Klammern angegebenen Lernelemente.

DIE DATENFORMATE FÜR DUALDATEN

Halbwort



Wort



Doppelwort

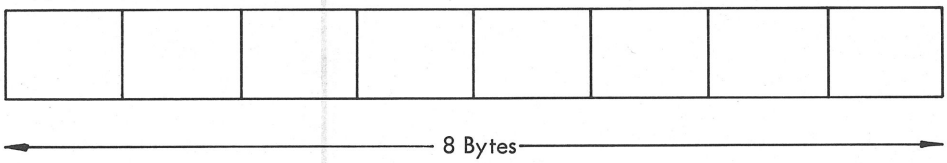


Abb. 6

Nachdem wir uns mit Dualzahlen allgemein befaßt haben, sehen wir uns nun an, wie Dualzahlen im System /360 gespeichert werden müssen.

Für Dualdaten (duale Zahlen) gibt es einen besonderen Instruktionssatz, der Instruktionen für das Addieren, Subtrahieren, Vergleichen, Multiplizieren, Dividieren von Dualdaten enthält. Einen Teil dieser Instruktionen werden Sie im Band 4 kennenlernen. -

Sie wissen bereits, daß Dezimaldaten in Feldern variabler Länge gespeichert werden. Das heißt, der Programmierer bestimmt durch Definitionen die Größe eines Feldes z. B. für die Kunden-Nr., den Namen, den Einzelpreis, usw.

Für Dualdaten hingegen sind Formate mit f e s t e r L ä n g e vorgegeben.

Betrachten Sie Abb. 6. -

Datenformate der Dualdaten im Hauptspeicher

17 Abb. 6 zeigt die drei Formate für Dualdaten.

Die drei Datenformate heißen:

a) _____

b) _____

c) _____

a) Halbwort

b) Wort

c) Doppelwort

18 Aus Abb. 6 können Sie auch die Größe der einzelnen Datenformate ersehen.

Das Halbwort ist __(Z)__ Bytes groß, das Wort __(Z)__ Bytes und das Doppelwort __(Z)__ Bytes.

2

4

8

DIE DATENFORMATE FÜR DUALDATEN

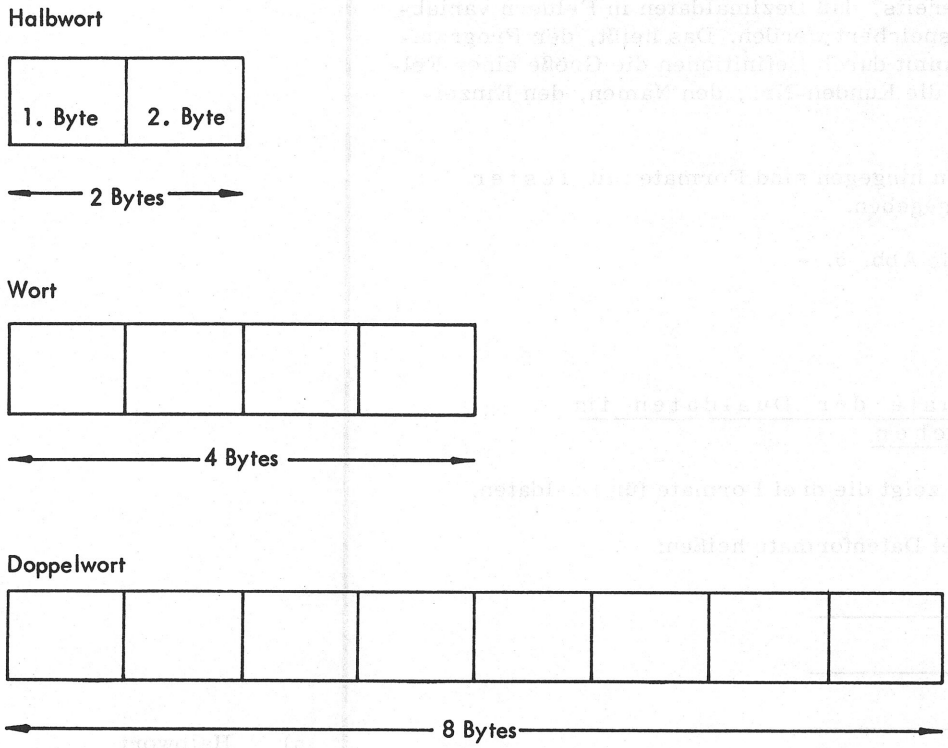


Abb. 6

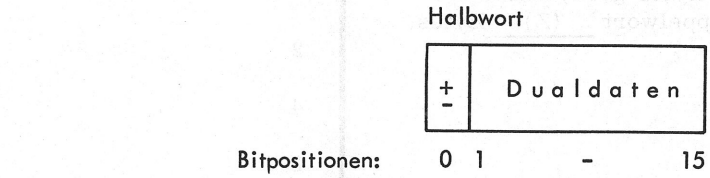


Abb. 7

19 Befassen wir uns zunächst einmal mit dem Halbwort.

Das Halbwort ist __(Z)__ Bytes oder __(Z)__ Bits groß.

2

16

20 Betrachten Sie sich Abb. 7. -

Die 16 Bits des Halbwortes sind hier als Bitpositionen __(Z)__ bis __(Z)__ ausgedrückt.

0 (bis) 15

21 Für das Vorzeichen der Dualdaten ist das linke Bit, nämlich Bitposition __(Z)__ vorgesehen.

Für die Dualdaten selbst bleiben also noch __(Z)__ Stellen.

0

15

22 Positive Dualzahlen haben die Vorzeichenverschlüsselung "0", negative Dualzahlen die Vorzeichenverschlüsselung "1".

V

0	0 0 0 0 0 0 1 1 0 1 1 1 0 0 1

In dem hier dargestellten Halbwort ist also eine _____ Dualzahl gespeichert.

positive

23 Die Dualdaten werden rechtsbündig im Halbwort gespeichert.

V

0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1

Die abgebildete (positive, negative) _____ Dualzahl entspricht der Dezimalzahl __(Z)__.

Die größte positive Dualzahl in einem Halbwort entspricht der Dezimalzahl 32767.

positive

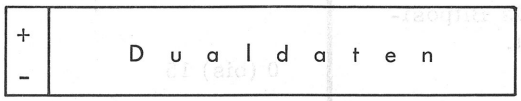
5

Halbwort



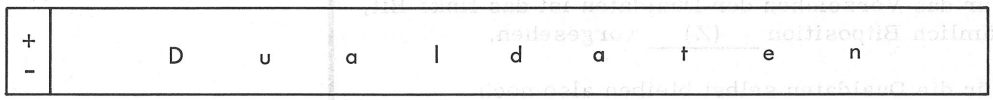
0 1 - 15

Wort



0 1 - 31

Doppelwort



0 1 - 63

Abb. 8

24 In gleicher Weise wie das Halbwort sind auch die beiden anderen Formate für Dualdaten aufgebaut. Nur in der Größe unterscheiden sich Wort und Doppelwort vom Halbwort.

In Abb. 8 sind Halbwort, Wort und Doppelwort einander gegenübergestellt.

Das Wort ist __(Z)__ Bytes lang, d.h. es umfaßt __(Z)__ Bits.

Da das linke Bit für das _____ reserviert ist, kann ein Wort maximal eine __(Z)__-stellige Dualzahl speichern.

Die größte positive Dualzahl in einem Wort entspricht der Dezimalzahl 2 147 483 647.

4

32

Vorzeichen

31

25 Das Doppelwort ist __(Z)__ Bytes groß.

Das bedeutet:

Es stehen für die Dualdaten ohne Vorzeichenstelle __(Z)__ Bits zur Verfügung.

8

63

26 Wir fassen zusammen:

Dualdaten werden in Feldern (fester Länge, variabler Länge) _____ gespeichert.

Drei Formate stehen uns hierzu zur Verfügung:

a) Das _____ mit __(Z)__ Bytes.

b) Das _____ mit __(Z)__ Bytes.

c) Das _____ mit __(Z)__ Bytes.

fester Länge

a) Halbwort 2

b) Wort 4

c) Doppelwort 8

Die Begrenzungszuordnung

Sie wissen bereits, daß die Hauptspeicherstellen mit null beginnend fortlaufend durchnumeriert sind. Die Nummer einer Speicherstelle bezeichnen wir als ihre Adresse. Die Adresse eines Feldes oder Bereiches im Hauptspeicher ist die Adresse des linken Bytes. Felder und Bereiche variabler Länge können an einer beliebigen Speicherstelle beginnen.

Felder fester Länge im Hauptspeicher, nämlich Halbworte, Worte und Doppelworte, die wir für das Speichern von Dualdaten benötigen, werden ebenfalls durch die Adresse des linken Bytes des Feldes adressiert. Die Felder fester Länge für Dualdaten dürfen jedoch nicht an einer beliebigen Speicherstelle beginnen.

Ein Feld fester Länge muß ab einer Speicherstelle gespeichert werden, deren Adresse ein Vielfaches der Länge des Feldes ist. Oder anders ausgedrückt: die Adresse eines Feldes fester Länge muß immer durch die Länge des Feldes (Anzahl der Bytes) teilbar sein.

Die Adresse eines Halbwortes muß also durch 2, die Adresse eines Wortes durch 4 und die Adresse eines Doppelwortes durch 8 teilbar sein.

Diese Zuordnung bestimmter Adressen für Felder fester Länge nennt man Begrenzungszuordnung.

Die Instruktionen für duale Daten können nur dann richtig ausgeführt werden, wenn die Begrenzungszuordnung eingehalten wird. Die Begrenzungszuordnung stellt beim Programmieren kein Problem dar, da sie bei der Umwandlung automatisch durchgeführt wird.

ADRESSIERBARE EINHEITEN DER ZENTRALEINHEIT

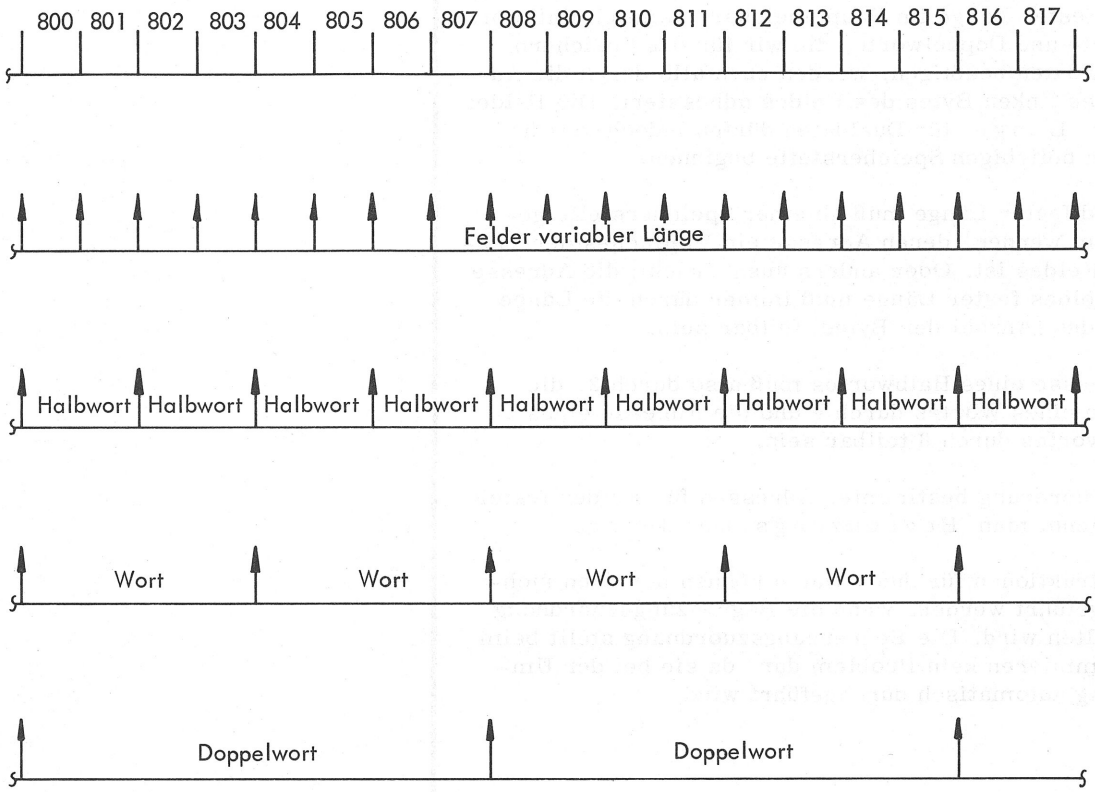


Abb. 9

27 Betrachten Sie bitte Abb. 9. -

In Abb. 9 ist ein Ausschnitt aus dem Hauptspeicher dargestellt, der die Speicherstellen 800 - 817 umfaßt. Durch Pfeile ist angedeutet, ab welchen Speicherstellen Felder variabler Länge und Felder fester Länge (Halbwort, Wort, Doppelwort) gespeichert werden dürfen.

Ein Feld variabler Länge setzt sich aus einer variablen Anzahl Bytes zusammen. Ein solches Feld kann z. B. 1 Byte, 5 Bytes, 8 Bytes oder auch 35 Bytes groß sein.

Ein solches Feld kann an einer _____ Stelle im Hauptspeicher beginnen.

beliebigen

28 Anders hingegen verhält es sich mit Feldern fester Länge für Dualdaten.

Ein Halbwort ist __(Z)__ Bytes groß und muß ab einer Stelle im Hauptspeicher gespeichert werden, die durch __(Z)__ teilbar ist.

2

2

29 Eine durch 2 teilbare Adresse im Hauptspeicher liegt an einer Halbwortgrenze. Halbworte müssen also an einer Halbwortgrenze gespeichert werden.

Welche der folgenden Adressen liegen an einer Halbwortgrenze und können einem Halbwort zugeordnet werden?

a) 805

b) 807

c) 812

d) 813

e) 814

c) 812

e) 814

ADRESSIERBARE EINHEITEN DER ZENTRALEINHEIT

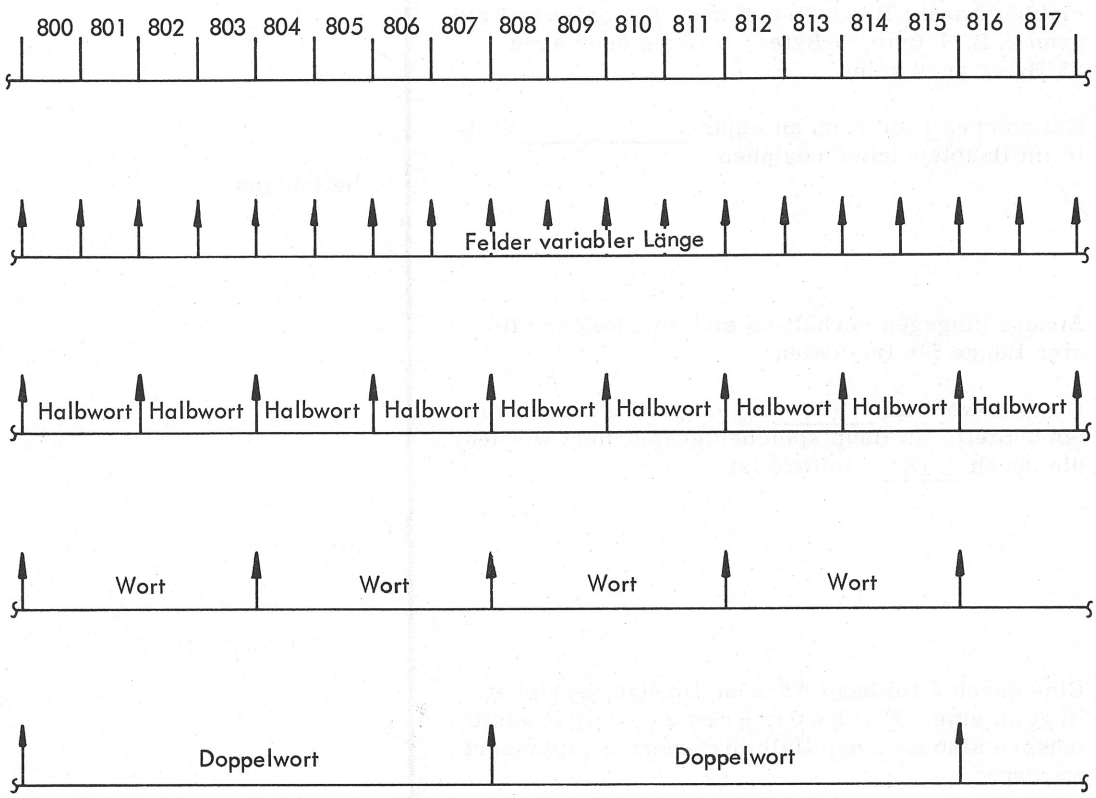


Abb. 9

30 Ein Wort ist __(Z)__ Bytes groß, seine Adresse muß also durch __(Z)__ teilbar sein.

Welche Adressen in Abb. 9 liegen an einer Wortgrenze und könnten einem Wort zugeordnet werden?

4

4

800, 804, 808, 812, 816

31 Ein Doppelwort muß an einer _____-Grenze beginnen. Seine Adresse muß durch __(Z)__ teilbar sein.

Welche der folgenden Adressen können einem Doppelwort zugeordnet werden?

a) 4000

b) 4500

c) 2011

d) 1632

Doppelwort

8

a) 4000

d) 1632

32 Wir merken uns:

Zur Erreichung eines möglichst hohen Nutzeffektes und um zu garantieren, daß bei allen Modellen des Systems /360 Daten fester Länge immer an ganz bestimmten Stellen im Hauptspeicher adressiert werden können, hat man die Begrenzungszuordnung eingeführt: Halbworte, Worte und Doppelworte müssen an einer Halbwort-, Wort- bzw. Doppelwortgrenze im Hauptspeicher gespeichert werden.

Felder fester Länge müssen ab einer Stelle im Hauptspeicher gespeichert werden, die ein Vielfaches der Länge eines Feldes darstellt. Die Adresse eines Halbwortes, Wortes oder Doppelwortes muß also durch __(Z)__, __(Z)__ bzw. __(Z)__ teilbar sein.

Diese Zuordnung nennt man _____.

2, 4 (bzw.) 8

Begrenzungszuordnung

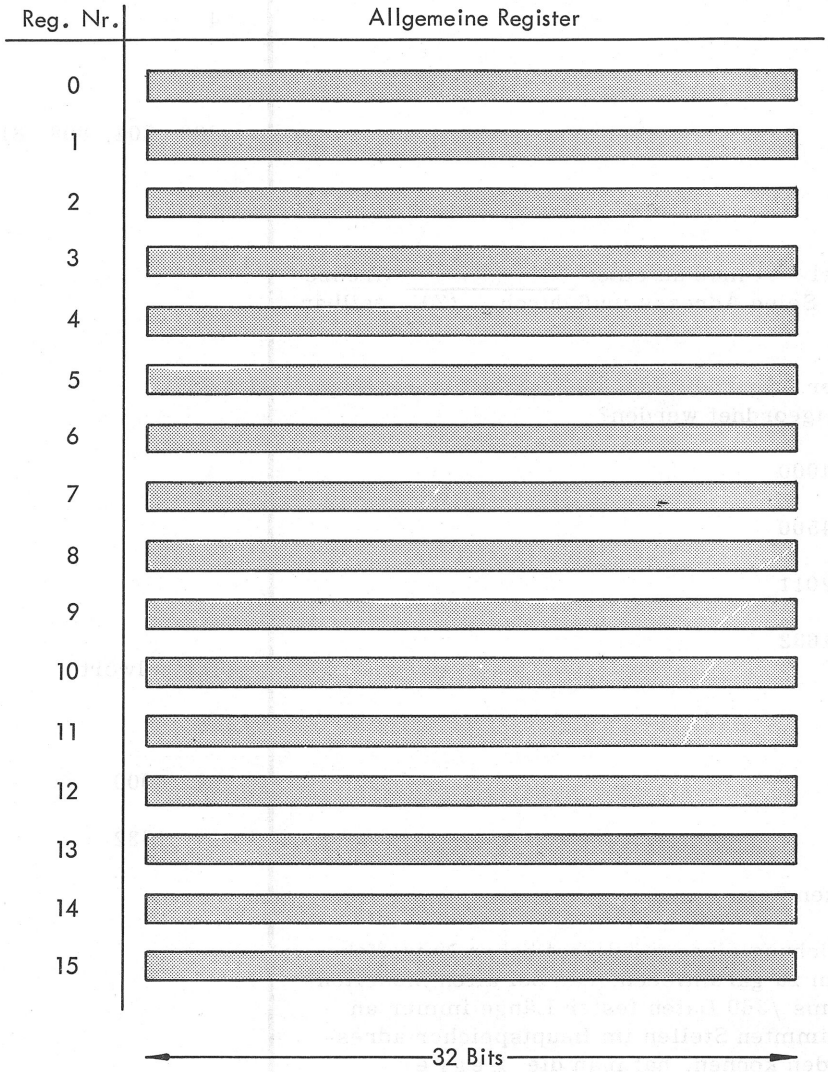

Reg. Nr.
Allgemeine Register
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
32 Bits

Abb. 10

Der Hauptspeicher kann Dualdaten zwar speichern, aber keine dualen Rechenoperationen durchführen.

Um mit Dualdaten rechnen zu können, benötigen wir sogenannte Register. Die Register sind Bestandteil der Zentraleinheit.

Die folgenden Lernelemente befassen sich mit diesen Registern.

Die Register in der Zentraleinheit

Die Register dienen u. a. zur Aufnahme von Dualdaten aus dem Hauptspeicher und zur Abgabe von Dualdaten an den Hauptspeicher. Sie werden z. B. benötigt, wenn mit Dualdaten gerechnet werden muß.

Es gibt 16 Register in der Zentraleinheit. Man bezeichnet diese Register auch als Allgemeine Register.

33 Betrachten Sie bitte Abb. 10. -

In Abb. 10 sind die 16 __(WW)__ der Zentraleinheit dargestellt.

Wie Sie sehen, haben die Register eine feste Grösse von jeweils __(Z)__ Bits, das sind jeweils __(Z)__ Bytes.

Allgemeinen Register

32

4

34 Die 16 Register haben also eine feste Länge von einem (Halbwort, Wort, Doppelwort) _____.

Wort

35 Jedes Register ist mit einer Register-Nummer versehen (Abb. 10). -

Die Register sind gekennzeichnet durch die Reg.-Nr. __(Z)__ bis __(Z)__.

0 (bis) 15

36 Die Reg. Nr. sind die A d r e s s e n , durch die jedes einzelne der Register angesprochen werden kann.

Die Adressierung der 16 Allgemeinen Register erfolgt also über die festen Adressen __(Z)__ bis __(Z)__.

0 (bis) 15

37 Wir fassen zusammen:

Die Register dienen dem Rechnen und Speichern von _____.

Die __(Z)__ Allgemeinen Register der Zentraleinheit haben eine feste Länge von jeweils __(Z)__ Bytes, das ist ein _____.

Die Adressierung der Register erfolgt über die festen Adressen (Reg. Nr.) __(Z)__ bis __(Z)__.

Dualdaten

16

4

Wort

0 (bis) 15

Über einige wenige Lernelemente hinweg soll Ihnen nun am Beispiel der Subtraktion ein Einblick in das duale Rechnen im System /360 gegeben werden.

Sie brauchen diese Art des dualen Rechnens nicht zu beherrschen, denn dieses Rechnen erfolgt in der Zentraleinheit ohne Ihr Zutun.

DARSTELLUNG NEGATIVER DUALZAHLEN

IM SYSTEM /360

Die positive Dezimalzahl "4" wird in einem Register folgendermaßen dargestellt:

\+

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 0 1 0 0

Eine negative Dualzahl wird gebildet:

1. durch KONVERTIEREN (Umkehren) der positiven Dualzahl:

\-

1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1

und

2. durch ADDIEREN einer dualen "1" zu der konvertierten Dualzahl:

\-

1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0

(negative Dezimalzahl "4" dual dargestellt)

Durch KONVERTIEREN der positiven Dualzahl und durch das ADDIEREN einer dualen "1" wird das sogenannte ZWEIERKOMPLEMENT gebildet.

Abb. 11

38 Die Grundrechenarten Subtraktion, Multiplikation und Division werden in der /360 auf die Addition zurückgeführt. Soll z.B. ein Betrag mit x multipliziert werden, so wird dieser Betrag x-mal in einen Zähler addiert. Auch die Subtraktion erfolgt durch eine _____.

Addition

39 Die Maschine hat eine besondere Art zu subtrahieren. Sie bildet sich für Dualzahlen, die negativ sind, das sogenannte Zweierkomplement.

Abb. 11 zeigt, wie eine negative Dualzahl gebildet wird. Verfolgen Sie bitte Schritt für Schritt die Umkehrung der positiven Dualzahl 4 in die negative Dualzahl 4 in Abb. 11.

Alle negativen Dualzahlen werden in der Maschine durch das Zweierkomplement der entsprechenden Dualzahl dargestellt.

Das Zweierkomplement einer Dualzahl wird gebildet durch:

1. _____ der positiven Dualzahl
2. Addieren einer ____(WZ)____.

Konvertieren / Umkehren

dualen 1

40 Wir wollen als Beispiel eine Subtraktion von Dualzahlen im System /360 durchführen.

4 soll von 9 subtrahiert werden:
(+)
9: 00000000000000000000000000001001
(+)
4: 00000000000000000000000000000100

In je einem Wort sind die Daten zunächst als (positive, negative) _____ Dualzahlen gespeichert.

positive

DARSTELLUNG NEGATIVER DUALZAHLEN
IM SYSTEM /360

Die positive Dezimalzahl "4" wird in einem Register folgendermaßen dargestellt:

\+

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 0 1 0 0

Eine negative Dualzahl wird gebildet:

1. durch KONVERTIEREN (Umkehren) der positiven Dualzahl:

\-

1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1

und

2. durch ADDIEREN einer dualen "1" zu der konvertierten Dualzahl:

\-

1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0

(negative Dezimalzahl "4" dual dargestellt)

Durch KONVERTIEREN der positiven Dualzahl
und durch das ADDIEREN einer dualen "1"
wird das sogenannte ZWEIERKOMPLEMENT
gebildet.

Abb. 11

41 Die Subtraktion erfolgt durch die Addition der negativen Dualzahl.

Wir müssen also von der Dualzahl
00000000000000000000000000000100 das sogenannte _____ bilden.

Zweierkomplement

42 Das Zweierkomplement der Dualzahl
(+)
00000000000000000000000000000100 ist:
(-)
11111111111111111111111111111100
(Siehe Abb. 11).

Wir können nun die Addition durchführen:

```
(+)
00000000000000000000000000001001   (+9)
(-)
+ 11111111111111111111111111111100   (-4)
```

Ergebnis:

(Z)

(+)
00000000000000000000000000000101

Die Addition läuft durch alle 32 Bits, also einschließlich Vorzeichenstelle. Der letzte Übertrag (Überlauf) bleibt unberücksichtigt.

43 Übertragen Sie nun noch bitte das Ergebnis
00000000000000000000000000000101
mit dem Vorzeichen in das Dezimalsystem, um die Lösung auf ihre Richtigkeit zu überprüfen.

Das Ergebnis lautet dezimal dargestellt (plus, minus) _____ (Z).

plus 5

FRAGEN ZUM KAPITEL IV

Schreiben Sie Ihre Antworten auf ein Extrablatt. - Wenn Sie alle Fragen beantwortet haben, finden Sie die richtigen Antworten auf der nächsten Seite.

1. Das Binärsystem baut auf der Basis __(Z)__ auf. Für die Darstellung der Dualzahlen benutzt man die Ziffern __(Z)__ und __(Z)__.

2. Addieren Sie die beiden Dualzahlen:

   1111
   1101

   Ergebnis: __(Z)__

3. Wiederholen Sie bitte die Addition mit den entsprechenden Zahlen des Dezimalsystems.

4. Dualdaten verlangen Formate fester Länge. Nennen Sie die 3 Datenformate und ihre Länge in Bytes.

5. Beschreiben Sie bitte in einem knappen Satz, wo das Vorzeichen in den Formaten fester Länge untergebracht ist und wie es dargestellt wird.

6. Halbworte, Worte und Doppelworte müssen an _____-, _____- bzw. _____-Grenzen gespeichert werden. Das bedeutet, die Adressen von Halbworten, Worten und Doppelworten müssen durch __(Z)__, __(Z)__ bzw. __(Z)__ teilbar sein.

7. In der Zentraleinheit gibt es __(Z)__ Allgemeine Register. Die Adressierung dieser Register erfolgt über die festen Adressen __(Z)__ bis __(Z)__.

ANTWORTEN ZUM KAPITEL IV

Wenn Ihnen bei Ihren Antworten Fehler unterlaufen sind, sollten Sie die entsprechenden Lernelemente (in Klammern) nochmals wiederholen.

1. Das Binärsystem baut auf der Basis 2 auf. Für die Darstellung der Dualzahlen benutzt man die Ziffern 0 und 1.

   (LE 4)

2. Ergebnis: 11100

   (LE 12-15)

3. 15
   13

   28

   (LE 7-8)

4. Halbwort - 2 Bytes
   Wort - 4 Bytes
   Doppelwort - 8 Bytes

   (LE 17-18)

5. Das Vorzeichen ist in dem linken Bit untergebracht und wird durch 0 (positiv) und 1 (negativ) dargestellt.

   (LE 21-23)

6. Halbworte, Worte und Doppelworte müssen an Halbwort-, Wort- bzw. Doppelwort-Grenzen gespeichert werden. Das bedeutet: die Adressen von Halbworten, Worten und Doppelworten müssen durch 2, 4 bzw. 8 teilbar sein.

   (LE 27-32)

7. In der Zentraleinheit gibt es 16 Allgemeine Register. Die Adressierung der Register erfolgt über die festen Adressen 0 bis 15.
(LE 33-37)

Schauen Sie sich nun bitte die Merkblätter 203 (2 Seiten), 204 und 205 an. - Diese Merkblätter geben Ihnen noch einmal einen zusammenfassenden Überblick über den von Ihnen erarbeiteten Lernstoff dieses Kapitels.

KAPITEL V

DIE ADRESSIERUNG DES HAUPTSPEICHERS

Lernstoff:

Die Darstellung der Hauptspeicheradressen in den Maschineninstruktionen

Die relative Adresse

Die Basisadresse

Die Bildung der effektiven Hauptspeicheradresse

HAUPTSPEICHERKAPAZITÄTEN EINZELNER MODELLE DES SYSTEMS /360

Modell	20	25	30	40	44	50	65	75	85
	4 K								
	8 K		8 K						
	12 K								
	16 K	16 K	16 K	16 K					
	24 K	24 K	24 K						
	32 K	32 K	32 K	32 K	32 K				
		48 K							
			64 K	64 K	64 K	64 K			
				128 K	128 K	128 K	128 K		
				256 K	256 K	256 K	256 K	256 K	
						512 K	512 K	512 K	512 K
							1024 K	1024 K	1024 K
									2048 K
									4096 K

Anmerkung:

Die Tabelle der Hauptspeicherkapazitäten erhebt keinen Anspruch auf Vollständigkeit. Sie entspricht einem momentanen Stand und kann jederzeit Änderungen unterworfen werden.

Abb. 1

Wir befassen uns in diesem Kapitel mit den von der Konzeption des Systems /360 her gegebenen Voraussetzungen für die Adressierung.

In welcher Weise wir diese Voraussetzungen beim Programmieren berücksichtigen müssen, wird an anderer Stelle in diesem Lehrgang behandelt. Die in diesem Kapitel offen bleibenden Fragen finden dann ihre Klärung.

1 Der Hauptspeicher in der Zentraleinheit des Systems /360 kann verschieden groß sein. - Es gibt z. B. einen Hauptspeicher, der 8 192 Bytes groß ist.

Für 8 192 Bytes sagt man auch 8 K.

"K" ist eine Abkürzung für Kilo (griech. tausend). Die Bezeichnung "K" stellt eine Vereinfachung dar, denn 1 K bezeichnet entgegen seiner Wortbedeutung (Kilo = 1000) 1 0 2 4 Bytes.

Wir merken uns:

1 K Bytes ist eine vereinfachte Schreibweise für __(Z)__ Bytes.

1024

2 Eine Hauptspeicherkapazität von 16 K entspricht also einer Anzahl von 16 mal __(Z)__ Bytes, die im Speicher zur Verfügung stehen.

Der Speicher ist also __(Z)__ Bytes groß.

1024

16384

3 Abb. 1 zeigt eine Übersicht der Hauptspeicherkapazitäten für die jeweiligen Modelle.

Für das Modell 30 sind in der Tabelle die Kapazitäten _____, _____, _____, _____ vermerkt.

8 K, 16 K, 32 K, 64 K

HAUPTSPEICHERKAPAZITÄTEN EINZELNER MODELLE DES SYSTEMS /360

Modell	20	25	30	40	44	50	65	75	85
	4 K								
	8 K		8 K						
	12 K								
	16 K	16 K	16 K	16 K					
	24 K	24 K	24 K						
	32 K	32 K	32 K	32 K	32 K				
		48 K							
			64 K	64 K	64 K	64 K			
				128 K	128 K	128 K	128 K		
				256 K	256 K	256 K	256 K	256 K	
						512 K	512 K	512 K	512 K
							1024 K	1024 K	1024 K
									2048 K
									4096 K

Anmerkung:

Die Tabelle der Hauptspeicherkapazitäten erhebt keinen Anspruch auf Vollständigkeit. Sie entspricht einem momentanen Stand und kann jederzeit Änderungen unterworfen werden.

Abb. 1

4 Ein Hauptspeicher des Systems /360 Modell 30 kann 8 K groß sein. Das sind __(Z)__ Bytes.

8192

5 Der Hauptspeicher des Modells 85 z. B. kann eine maximale Größe von 4096 K haben. Das sind 4 194 304 Bytes.

Wir haben also Adressen von 0 bis __(Z)__.

4 194 303

(1. Adresse = 0,
4 194 304. Adresse = 4 194 303)

6 Wieviel Bytes benötigen wir zur Darstellung der Adresse (Zahl) 4 194 303 bei

a) ungepackter

b) gepackter

Darstellung ?

a) 7 Bytes
(|F,4|F,1|F,9|F,4|F,3|F,0|F,3|)

b) 4 Bytes
(|4,1|9,4|3,0|3,F|)

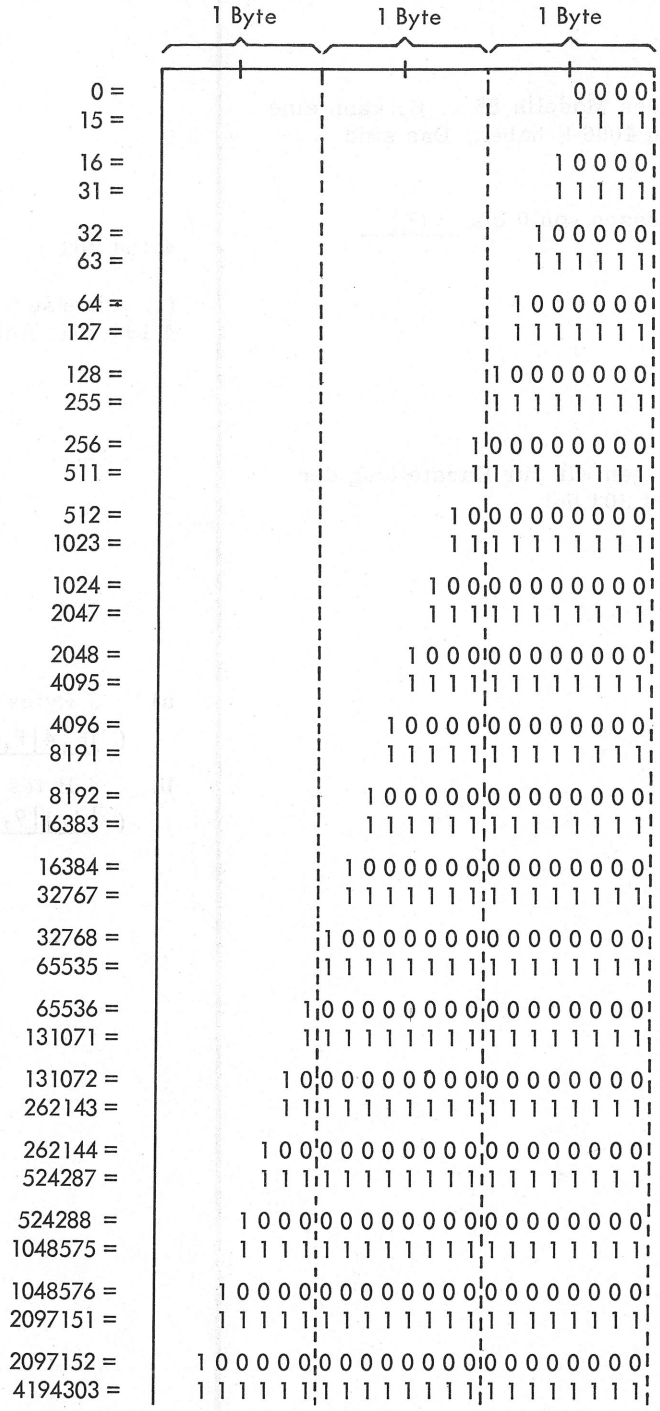

1 Byte
1 Byte
1 Byte
0 = 0 0 0 0
15 = 1 1 1 1
16 = 1 0 0 0 0
31 = 1 1 1 1 1
32 = 1 0 0 0 0 0
63 = 1 1 1 1 1 1
64 = 1 0 0 0 0 0 0
127 = 1 1 1 1 1 1 1
128 = 1 0 0 0 0 0 0 0
255 = 1 1 1 1 1 1 1 1
256 = 1 0 0 0 0 0 0 0 0
511 = 1 1 1 1 1 1 1 1 1
512 = 1 0 0 0 0 0 0 0 0 0
1023 = 1 1 1 1 1 1 1 1 1 1
1024 = 1 0 0 0 0 0 0 0 0 0 0
2047 = 1 1 1 1 1 1 1 1 1 1 1
2048 = 1 0 0 0 0 0 0 0 0 0 0 0
4095 = 1 1 1 1 1 1 1 1 1 1 1 1
4096 = 1 0 0 0 0 0 0 0 0 0 0 0 0
8191 = 1 1 1 1 1 1 1 1 1 1 1 1 1
8192 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0
16383 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1
16384 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
32767 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
32768 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
65535 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
65536 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
131071 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
131072 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
262143 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
262144 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
524287 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
524288 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1048575 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1048576 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2097151 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2097152 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4194303 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Abb. 2

7 Es besteht auch die Möglichkeit, die Adressen dual darzustellen.

Aus der Tabelle in Abb. 2 ersehen Sie, daß wir für die duale Darstellung der Adresse 4 194 303 __(Z)__ Bits benötigen, d. h. wir brauchen __(Z)__ ganze Bytes.

22

3

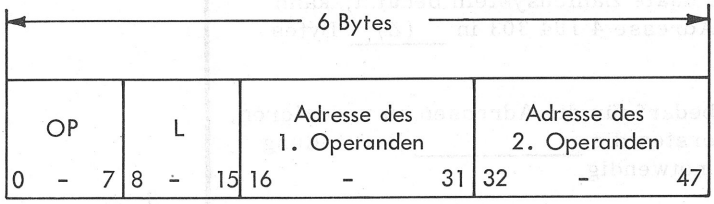
8 Wenn man das duale Zahlensystem benutzt, kann man also die Adresse 4 194 303 in __(Z)__ Bytes darstellen.

Um den Platzbedarf für die Adressen zu reduzieren, ist somit als erstes die _____ Darstellung der Adressen notwendig.

3

duale

FORMAT EINER MASCHINENINSTRUKTION



OP = Operationsschlüssel

L = Längenschlüssel

Abb. 3

9 Wie Sie wissen, teilt der Programmierer durch sein Programm der Zentraleinheit mit, was mit bestimmten Daten geschehen soll. Einmal soll addiert, ein andermal sollen Daten verglichen, multipliziert oder dividiert werden.

Die Anweisungen, die der Programmierer in leicht verständlicher Sprache (in der Programmiersprache) dem DVS mitteilt, werden durch das System in Maschineninstruktionen (Instruktionen in der Maschinensprache) umgewandelt und im Hauptspeicher gespeichert.

In Abb. 3 sehen Sie ein Beispiel für das Format einer _____.

Maschineninstruktion

10 Es gibt verschiedene Formate für Maschineninstruktionen. Im Laufe dieses Lehrganges werden Sie sich noch eingehend damit befassen.

Für das Beispiel in Abb. 3 haben wir ein Instruktionsformat mit einer Länge von __(Z)__ Bytes ausgewählt.

6

11 Betrachten Sie Abb. 3. -

In den Bitpositionen 0 - 7 befindet sich der _____. Die Bitpositionen 8 - 15 enthalten den _____ der Instruktion.

Durch die Angaben von Operations- und Längenschlüssel in der Maschineninstruktion ist sichergestellt, daß die Zentraleinheit weiß,

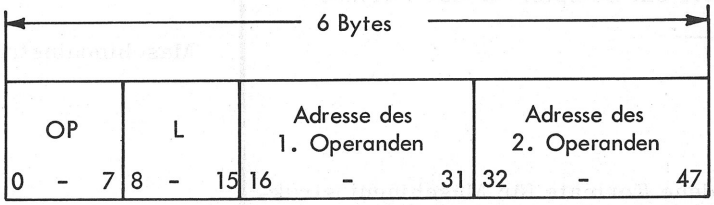
a) welche Operation durchzuführen ist,

b) auf wie viele Bytes sich die Operation erstrekken soll.

Operationsschlüssel

Längenschlüssel

FORMAT EINER MASCHINENINSTRUKTION



OP = Operationsschlüssel

L = Längenschlüssel

Abb. 3

12 Wo die Operanden im Hauptspeicher stehen, erfährt die Zentraleinheit ebenfalls aus der in Abb. 3 dargestellten Maschineninstruktion:

In der Maschineninstruktion sind Angaben über die Adresse des ___(ZW)___ und die Adresse des ___(ZW)___ gemacht.

1. Operanden

2. Operanden

13 Die höchste Adresse eines 4096 K großen Hauptspeichers ist 4 194 303. Wir haben festgestellt, daß wir für die duale Darstellung dieser Adresse 3 Bytes benötigen.

Aus Abb. 3 ersehen Sie, daß in der Maschineninstruktion für die Darstellung einer Adresse jedoch nur ___(Z)___ Bytes vorgesehen sind.

Die Adresse 4 194 303 muß sich also auch auf irgendeine Weise in diesen 2 Bytes der Maschineninstruktion darstellen lassen. - Wie das möglich gemacht wird, erfahren Sie in den folgenden Lernelementen.

2 (Bitpositionen 16 - 31 oder 32 - 47)

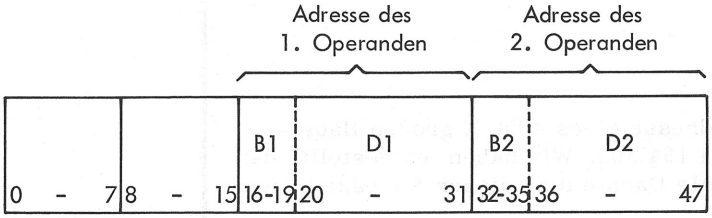

Adresse des
1. Operanden
Adresse des
2. Operanden
B1
D1
B2
D2
0 - 7
8 - 15
16-19
20 - 31
32-35
36 - 47

Abb. 4

Betrachten Sie bitte Abb. 4. -

Sie sehen, daß die Adresse eines Operanden aus 2 Komponenten gebildet wird. Die Adresse des 1. Operanden setzt sich aus B1 und D1 zusammen, die Adresse des 2. Operanden aus B2 und D2.

Wir befassen uns zunächst mit D1 und D2 in den Bitpositionen 20 - 31 bzw. 36 - 47.

Die Abkürzungen D1 und D2 bezeichnen die sogenannten r e l a t i v e n A d r e s s e n des 1. und des 2. Operanden. Im Englischen heißt die relative Adresse "displacement" (sprich: displ$\ddot{e}$isment). Von diesem englischen Ausdruck sind die Abkürzungen D1 und D2 für den 1. und 2. Operanden abzuleiten.

Die relativen Adressen D1 und D2 nehmen in der Maschineninstruktion jeweils 12 Bits ein und werden dual dargestellt.

14 D1 und D2 bezeichnen die _____ Adressen in der Maschineninstruktion.

Eine der beiden Komponenten, die für die Bildung der Hauptspeicheradressen benötigt werden, ist also die ___(WW)___.

relativen

relative Adresse

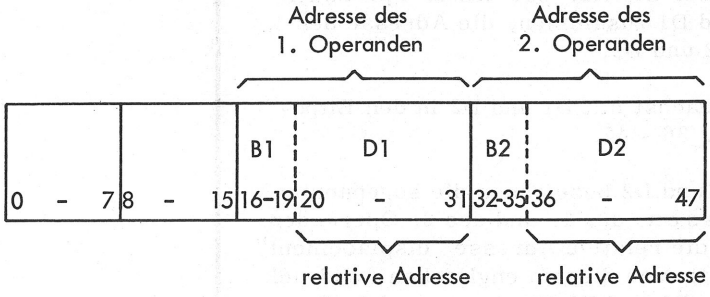

Adresse des
1. Operanden
Adresse des
2. Operanden
B1
D1
B2
D2
0 - 7
8 - 15
16-19
20 - 31
32-35
36 - 47
relative Adresse
relative Adresse

Abb. 5

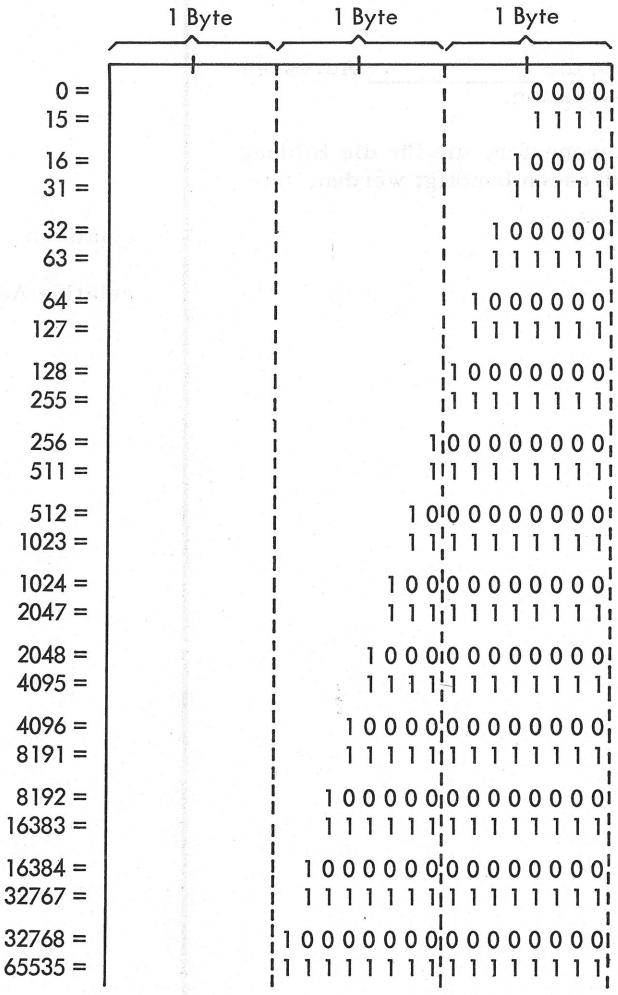

1 Byte
1 Byte
1 Byte
0 = 0 0 0 0
15 = 1 1 1 1
16 = 1 0 0 0 0
31 = 1 1 1 1 1
32 = 1 0 0 0 0 0
63 = 1 1 1 1 1 1
64 = 1 0 0 0 0 0 0
127 = 1 1 1 1 1 1 1
128 = 1 0 0 0 0 0 0 0
255 = 1 1 1 1 1 1 1 1
256 = 1 0 0 0 0 0 0 0 0
511 = 1 1 1 1 1 1 1 1 1
512 = 1 0 0 0 0 0 0 0 0 0
1023 = 1 1 1 1 1 1 1 1 1 1
1024 = 1 0 0 0 0 0 0 0 0 0 0
2047 = 1 1 1 1 1 1 1 1 1 1 1
2048 = 1 0 0 0 0 0 0 0 0 0 0 0
4095 = 1 1 1 1 1 1 1 1 1 1 1 1
4096 = 1 0 0 0 0 0 0 0 0 0 0 0 0
8191 = 1 1 1 1 1 1 1 1 1 1 1 1 1
8192 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0
16383 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1
16384 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
32767 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
32768 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
65535 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Abb. 6

15 D1 und D2, die relativen Adressen des 1. und 2. Operanden, nehmen in unserer Maschineninstruktion jeweils 12 Bitpositionen ein.

Stellen Sie bitte in Abb. 6 fest, wie die größte Dezimalzahl heißt, die man in 12 Bits dual darstellen kann.

4095

16 In 12 Bits können somit relative Adressen von 0 bis 4 095 dual dargestellt werden.

Die (WW) eines Operanden kann also nicht größer als (Z) werden.

relative Adresse

4095

17 Die Zahl "4095", die die maximale Größe einer (WW) angibt, wird uns im Laufe dieses Lehrganges noch öfter begegnen.

Prägen Sie sich bitte ein, daß eine relative Adresse nicht größer als (Z) werden kann.

relativen Adresse

4095

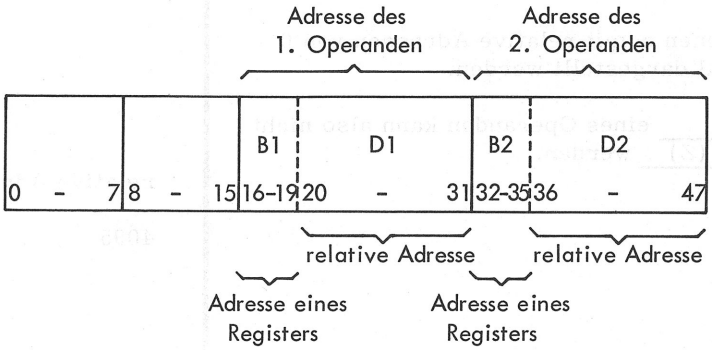

Adresse des
1. Operanden
Adresse des
2. Operanden
B1
D1
B2
D2
0 - 7
8 - 15
16-19
20 - 31
32-35
36 - 47
relative Adresse
relative Adresse
Adresse eines
Registers
Adresse eines
Registers

Abb. 7

Wir wenden uns nun der zweiten Komponente bei der Bildung der Hauptspeicheradresse zu.

Betrachten Sie bitte Abb. 7. -

In den Bitpositionen 16 - 19 und 32 - 35 wird jeweils die Adresse eines Registers angegeben. Wie die relativen Adressen sind auch die Register-Adressen dual dargestellt.

18 Ein Allgemeines Register, das für die Adressierung benutzt wird, nennt man auch B a s i s r e g i s t e r .

Wird für die Adreßbildung das _____ 5 verwendet, so steht in den Bitpositionen 16 - 19 und 32 - 35 der Maschineninstruktion die Dualzahl _(Z)_ .

Basisregister/Register

0101

19 Den Inhalt eines Registers, das als Basisregister benutzt wird, nennt man die B a s i s a d r e s s e .

B1 und B2 geben das jeweilige _____ an, das die _____ enthält.

Basisregister

Basisadresse

20 Die Basisadresse - der Name sagt es schon - stellt die Basis, das Fundament für die Bildung der Hauptspeicher - _____ dar.

Adresse

21 Die 16 Allgemeinen Register in der Zentraleinheit haben eine Größe von jeweils _(Z)_ Bytes.

Wird ein Register als Basisregister benutzt, so werden nur drei der vier Bytes benötigt.

4

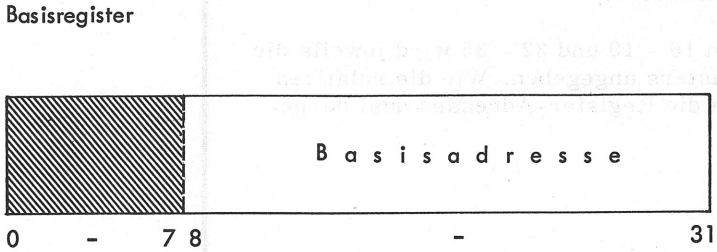

Basisregister
B a s i s a d r e s s e
0 - 7 8 - 31

Abb. 8

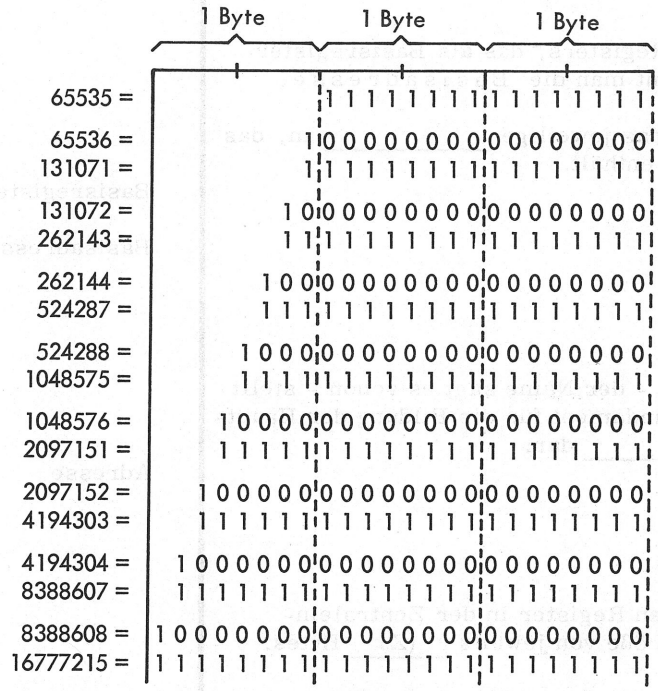

1 Byte
1 Byte
1 Byte
65535 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
65536 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
131071 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
131072 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
262143 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
262144 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
524287 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
524288 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1048575 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1048576 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2097151 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2097152 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4194303 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4194304 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8388607 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
8388608 = 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16777215 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Abb. 9

22 Betrachten Sie das Basisregister in Abb. 8. -

Das linke Byte des Basisregisters, die Bitpositionen 0 - 7, bleiben bei der Adreßbildung unberücksichtigt.

Die Basisadresse wird in den Bits __(Z)__ bis __(Z)__ des _____ gespeichert.

Für die Basisadresse stehen also __(Z)__ Bytes im Basisregister zur Verfügung.

8

31

Basisregisters

3

23 Aus der Tabelle in Abb. 9 ersehen Sie, daß man in 3 Bytes die Zahlen 0 bis __(Z)__ dual darstellen kann.

16 777 215

24 Wie wir in LE 6 gesehen haben, kann der Hauptspeicher des Modells 85 eine Größe von 4096 K haben. Das sind 4 194 304 Bytes.

16 777 216 Bytes (16 384 K) können wir jedoch mit Hilfe des Basisregisters, das die _____ enthält, adressieren.

Das ist also ausreichend, um auch die Adressierung des größten Hauptspeichers im System /360 sicherzustellen.

Basisadresse

25 Wir fassen zusammen:

Die beiden Komponenten, die die Hauptspeicheradressen in den Maschineninstruktionen darstellen, sind die Adresse eines _____ und die ___(WW)___.

Das Basisregister enthält die _____.

Die relative Adresse kann maximal __(Z)__ groß sein.

Basisregisters

relative Adresse

Basisadresse

4095

In den folgenden Lernelementen erfahren Sie nun, wie die Hauptspeicheradressen durch die Angaben in der Maschineninstruktion gebildet werden.

26 Die Summe aus Basisadresse und relativer Adresse ergibt die Hauptspeicheradresse, auch e f f e k t i v e A d r e s s e genannt.

Basisadresse + Relative Adresse = ___(WW)___.

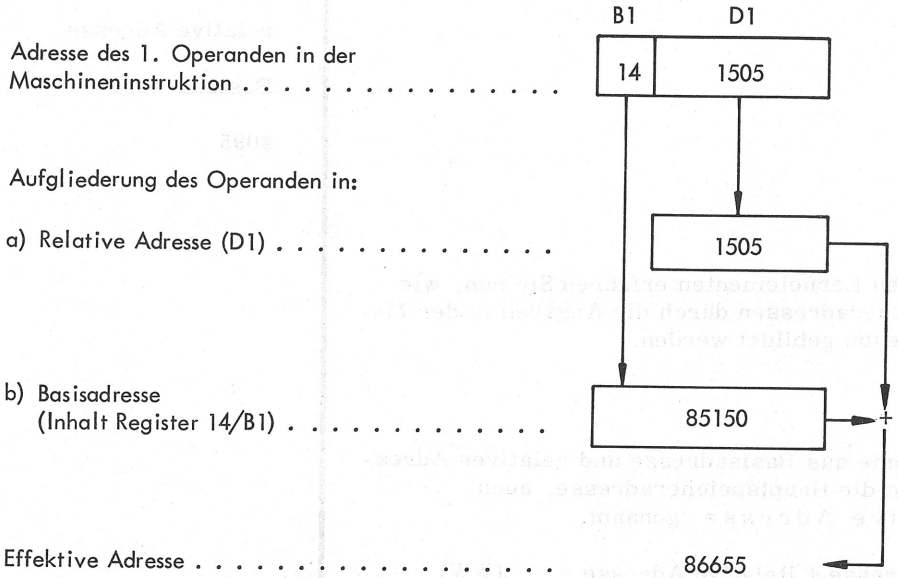
effektive Adresse

27 Jede Adresse im Hauptspeicher wird auf diese Weise durch die beiden Komponenten _____ und ___(WW)___ ausgedrückt.

Basisadresse

relative Adresse

BEISPIEL ZUR BILDUNG EINER EFFEKTIVEN ADRESSE



Anmerkung:

Aus Gründen der besseren Überschaubarkeit sind in dem obigen Beispiel Dezimalzahlen verwendet. In Wirklichkeit sind die Zahlen dual dargestellt. Ebenso erfolgt die Addition dual.

Abb. 10

28 Abb. 10 soll Ihnen die Bildung der effektiven Adressen im System /360 verdeutlichen.

In dem Beispiel der Abb. 10 enthält der Adreßteil der Maschineninstruktion für den 1. Operanden die Adresse des Registers __(Z)__ (B1) und die __(WW)__ 1505 (D1).

14

relative Adresse

29 Der Inhalt des Registers 14 ist die _____ 85 150 (vgl. Abb. 10).

Basisadresse

30 Die relative Adresse und die Basisadresse (Inhalt Register 14) werden addiert:

```
   1505
+ 85150
-------
  86655
```

Das Ergebnis dieser Addition ist die __(WW)__ __(Z)__.

effektive Adresse

86655

31 Ermitteln Sie bitte in den folgenden Fällen die effektiven Adressen:

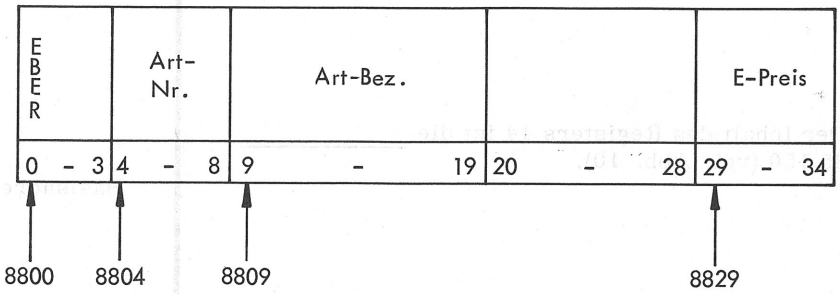
	Basisadresse	rel. Adr.	eff. Adr.
a)	10 400	150	
b)	10 400	300	
c)	10 400	475	

a) 10550

b) 10700

c) 10875

Eingabebereich:



Basisadresse: 8500

Abb. 11

32 Betrachten Sie bitte Abb. 11. -

Ein Eingabebereich soll in den Hauptspeicher ab der (effektiven) Adresse 8 800 gespeichert sein. - Die Basisadresse, die für alle Felder gleich ist, sei 8 500.

Wie muß die relative Adresse lauten, damit der Eingabebereich (EBER) die effektive Adresse 8 800 erhält ?

300 (8500 + 300 = 8800)

33 Ermitteln Sie für die folgenden Felder des Eingabebereichs in Abb. 11 die relativen Adressen:

	Basisadr.	rel. Adr.	eff. Adr.
Art-Nr.	8 500		8 804
Art-Bez.	8 500		8 809
E-Preis	8 500		8 829

304

309

329

34 Am Beispiel des Eingabebereichs in Abb. 11 sehen Sie:

Von Feld zu Feld ändert sich die effektive Adresse.

Entsprechend zur effektiven Adresse ändert sich auch die ___(WW)___.

Die Basisadresse jedoch ___(WW)___.

relative Adresse

ändert sich nicht / bleibt unverändert

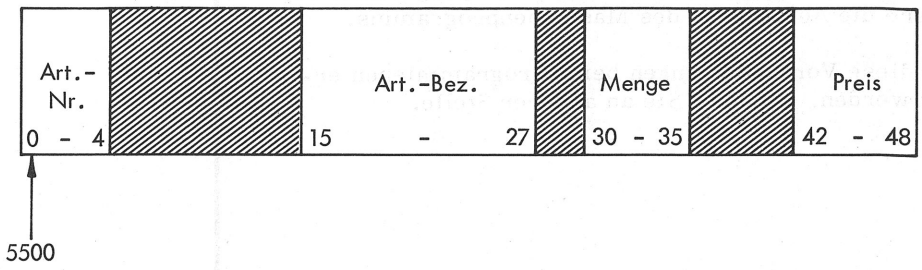
Sie wissen nun, wie eine Hauptspeicheradresse gebildet wird und in welcher Form die hierfür erforderlichen Angaben in einer Maschineninstruktion gespeichert werden.

Damit die Adressbildung, wie in diesem Kapitel beschrieben, erfolgen kann, müssen zwei wesentliche programmiertechnische Voraussetzungen erfüllt sein.

Die eine Voraussetzung betrifft die Umwandlung des symbolischen Programms in das Maschinenprogramm, die andere die Ausführung des Maschinenprogramms.

Wie diese Voraussetzungen beim Programmieren erfüllt werden, erfahren Sie an anderer Stelle.

Eingabebereich:



Basisadresse: 5000

Abb. 12

FRAGEN ZUM KAPITEL V

Benutzen Sie für die Beantwortung der folgenden Fragen ein Extrablatt. - Wenn Sie alle Fragen beantwortet haben, finden Sie die richtigen Antworten auf der nächsten Seite.

1. Wie nennt man ein Register, das für die Adreßbildung benutzt wird und wie heißt sein Inhalt ?

2. Die Adresse eines Feldes im Hauptspeicher wird in der Maschineninstruktion durch die Adresse eines _____ und eine ___(WW)___ dargestellt.

3. Wie groß kann die relative Adresse eines Operanden maximal werden ?

4. Ordnen Sie bitte die Begriffe effektive Adresse, relative Adresse, Basisadresse ein:

   _____ + _____ = _____

5. Abb. 12 zeigt einen Eingabebereich, der ab der Adresse 5500 im Hauptspeicher gespeichert ist.

   Ermitteln Sie die relative und die effektive Adresse der einzelnen Felder im Hauptspeicher:

	Basisadr.	rel. Adr.	eff. Adr.
Art.-Nr.	5 000		
Art.-Bez.	5 000		
Menge	5 000		
Preis	5 000		

ANTWORTEN ZUM KAPITEL V

Wenn Ihnen bei der Beantwortung der Fragen Fehler unterlaufen sind, sollten Sie die entsprechenden Lernelemente (Angaben in Klammern) nochmals wiederholen.

1. Ein Register, das zur Adreßbildung benutzt wird, nennt man Basisregister. Sein Inhalt ist die Basisadresse.

   (LE 18 - 24)

2. Die Adresse eines Feldes im Hauptspeicher wird in der Maschineninstruktion durch die Adresse eines Basisregisters und eine relative Adresse dargestellt.

   (LE 25)

3. Die relative Adresse kann maximal 4095 werden.

   (LE 15 - 17)

4. Basisadresse + relative Adresse = effektive Adresse

   (LE 26 - 30)

5.

	Basisadr.	rel. Adr.	eff. Adr.
Art.-Nr.	5 000	500	5 500
Art.-Bez.	5 000	515	5 515
Menge	5 000	530	5 530
Preis	5 000	542	5 542

(LE 32 - 34)

Schauen Sie sich nun Merkblatt 206 an. -



→ Band 3, Seite 275 (USING)

KAPITEL VI

INSTRUKTIONSFORMATE

Lernstoff:

Das SS-Format

Das SI-Format

Das RX-Format

Das RR-Format

In diesem Kapitel werden wir uns mit der Form der Maschineninstruktionen befassen.

Der Programmierer bedient sich bei seiner Arbeit der Assemblersprache. Die Maschine übersetzt dann mit Hilfe des Umwandlungsprogramms das Assemblerprogramm in ein Maschinenprogramm.

Trotzdem ist es für den Programmierer manchmal wichtig, die Maschinensprache zu kennen.

Er kann veranlassen, daß bei der Umwandlung des symbolischen in das echte Programm vom DVS eine Umwandlungsliste erstellt wird. Auf dieser Liste erscheinen u. a. die Instruktionen des Programms in ihrer echten Form. Außerdem kann er sich während eines Probelaufs mit dem Programm den momentanen Speicherinhalt in einer Liste herausdrucken lassen.

Beide Listen bilden Unterlagen zum Überprüfen des Programms auf seine Richtigkeit, also zum Austesten und auch zur Dokumentation.

Um diese Listen lesen zu können, muß der Programmierer Länge und Aufbau der echten Befehle kennen.

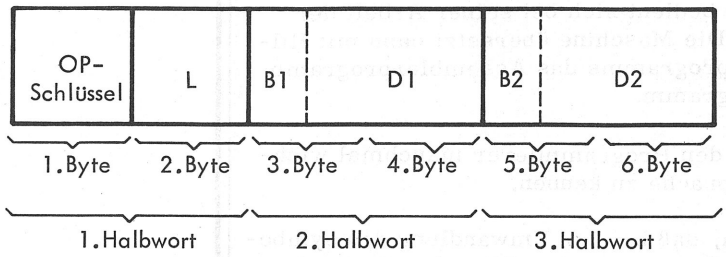

OP-
Schlüssel
L
B1
D1
B2
D2
1.Byte
2.Byte
3.Byte
4.Byte
5.Byte
6.Byte
1.Halbwort
2.Halbwort
3.Halbwort

Abb. 1

1 Dieses Kapitel soll Ihnen das nötige Wissen über A u f b a u und L ä n g e der Maschineninstruktionen des Systems /360 vermitteln. Wir können mit einem Fachausdruck auch sagen: wir wollen die I n s t r u k t i o n s f o r m a t e kennenlernen.

Mit dem Ausdruck "Instruktionsformat" meinen wir also _____ und _____ der Maschineninstruktionen.

Aufbau

Länge

2 Es gibt eine große Anzahl von Instruktionen, aber nur 5 verschiedene Instruktionsformate.

Daraus folgt, daß viele verschiedene Instruktionen das gleiche _____ haben.

Instruktionsformat

3 In diesem Lehrgang werden nicht alle Befehle behandelt, die es für das System /360 gibt.

Daher befassen wir uns auch nur mit 4 der insgesamt __(Z)__ Instruktionsformate.

5

4 Das Instruktionsformat, mit dem wir beginnen, ist Ihnen in diesem Lehrgang schon vorgestellt worden.

Betrachten Sie Abb. 1.

Das Format umfaßt __(Z)__ Bytes oder __(Z)__ Halbworte.

6

3

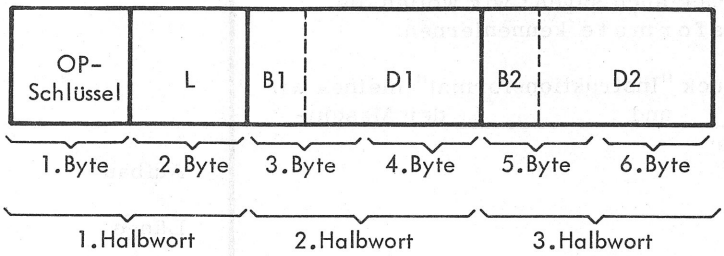

OP-
Schlüssel
L
B1
D1
B2
D2
1.Byte
2.Byte
3.Byte
4.Byte
5.Byte
6.Byte
1.Halbwort
2.Halbwort
3.Halbwort

Abb. 1

5 Das 1. Byte enthält die Angabe darüber, w a s zu tun ist. Es enthält den _____.

Die Angaben im 2. und 3. Halbwort sind für Sie nicht neu.

B bedeutet immer die Adresse eines _____.

D bedeutet immer eine ___(WW)_____.

Jedes der beiden Halbworte enthält also die Angaben für eine _____.

Operationsschlüssel

Basisregisters

relative Adresse

Hauptspeicheradresse/effektive Adresse

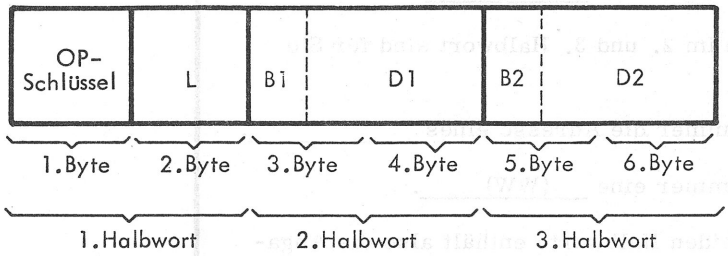

OP-
Schlüssel
L
B1
D1
B2
D2
1.Byte
2.Byte
3.Byte
4.Byte
5.Byte
6.Byte
1.Halbwort
2.Halbwort
3.Halbwort

Abb. 2

6 Wie Sie aus Abb. 2 ersehen, heißt dieses Format "SS-Format".

"SS" ist die Abkürzung für "Speicher-Speicher". Sowohl der erste als auch der zweite Operand sind Felder im _____.

Speicher/Hauptspeicher

7 Die Adressen der beiden Felder im Speicher stehen im __(Z)__ und __(Z)__ Halbwort des SS-Formats.

Außerdem muß dieses Format eine Angabe über die L ä n g e der Felder im Speicher enthalten.

Diese Angabe steht im __(Z)__ Byte. In Abb. 2 ist sie mit _____ abgekürzt.

2. (und) 3.

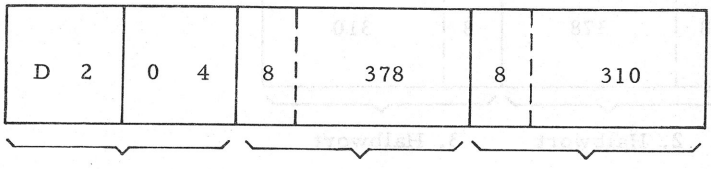
2.

L

ein Bit	ein Bit	ein Bit	ein Bit	Hexadezimal-Ziffer
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F
8	4	2	1	Stellenwert

Abb. 3

8 Das folgende Beispiel zeigt Ihnen eine Instruktion im SS-Format. (Die relativen Adressen sind der besseren Übersicht wegen dezimal angegeben.)



Die Instruktion ist aufgebaut aus den Angaben:

Operationsschlüssel: __(ZZ)__

Längenangabe: __(ZZ)__

Adresse des 1. Operanden: __(ZZ)__

Adresse des 2. Operanden: __(ZZ)__

D 2

0 4

8 (und) 378

8 (und) 310

9 Jeder echte Operationsschlüssel wird in der Fachliteratur mit 2 Hexadezimalziffern angegeben, z. B. "FA", "48", "5F".

Im obigen Beispiel heißt der Operationsschlüssel "D2".

Jedes der Zeichen "D" und "2" ist eine _____ -Ziffer und ist in __(Z)__ Bits verschlüsselt.

Welchen Inhalt haben also die ersten 8 Bits dieser Instruktion ? (Benutzen Sie Abb. 3.)

Hexadezimal

4

1101 0010

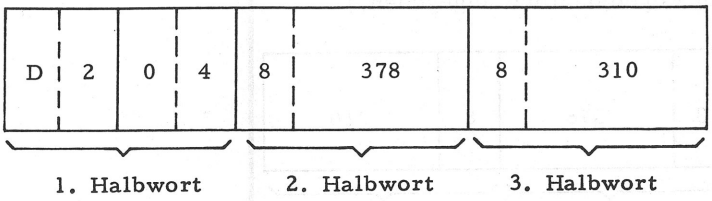


Abb. 4

10 Das 2. Byte der Instruktion enthält hexadezimal die Angabe zur Länge oder den L ä n g e n s c h l ü s s e l. In unserem Beispiel ist der Längenschlüssel "04" (vgl. Abb. 4).

Welchen Inhalt haben in diesem Fall die 8 Bits ?

0000 0100

11 Auch die restlichen Angaben in der Instruktion sind dual verschlüsselt.

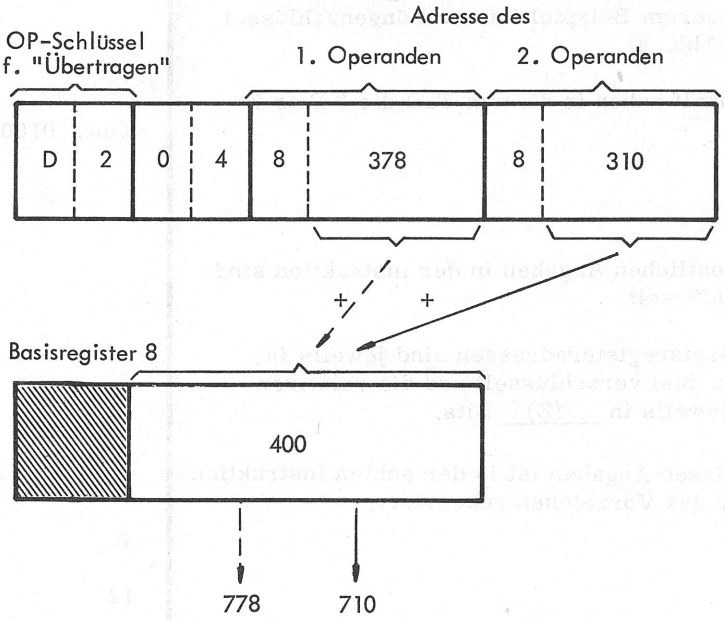
Die zwei Basisregisteradressen sind jeweils in __(Z)__ Bits dual verschlüsselt und die relativen Adressen jeweils in __(Z)__ Bits.

4

12

Bei allen diesen Angaben ist in der echten Instruktion kein Bit für das Vorzeichen reserviert.

ÜBERTRAGUNGSBEFEHL (SS-FORMAT)



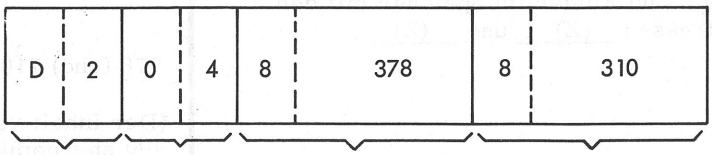
12 Das Beispiel in Abb. 5 stellt einen Übertragungsbefehl dar.

Die Operanden sind Felder im Speicher mit den effektiven Adressen __(Z)__ und __(Z)__.

778 (und) 710

(Der Inhalt von Basisregister 8 ist mit 400 angenommen.)

ÜBERTRAGUNGSBEFEHL (SS-FORMAT)



effektive Adresse : 778 (1. Operand) 710 (2. Operand)

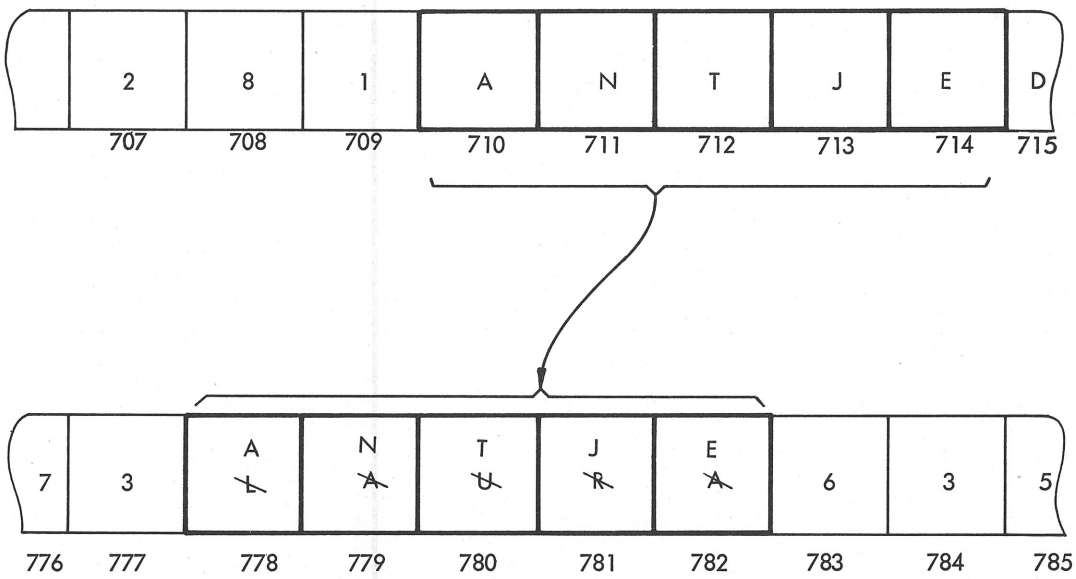


Abb. 6

13 Mit diesem Übertragungsbefehl werden Daten aus dem 2. Operanden übertragen.

Abb. 6 zeigt die beiden Operanden.

Der 2. Operand beginnt bei dem Byte mit der Adresse __(Z)__, der 1. Operand bei dem Byte mit der Adresse __(Z)__.

Die Operanden haben eine Länge von __(Z)__ Bytes.

Der 1. Operand hat nach der Operation den Inhalt _____.

710

778

5

ANTJE

14 Vergleichen Sie die Länge der Operanden mit dem Längenschlüssel in der Operation.

Der wirklichen Länge von 5 Bytes steht die Angabe __(Z)__ in der Instruktion gegenüber.

04

15 Es gilt die Regel:

Die Längenangabe in der echten Instruktion ist um 1 kleiner als die tatsächliche Byte-Anzahl der Operanden.

Durch diese Festlegung werden die 8 Bits in der Instruktion am besten ausgenutzt, wie Sie gleich sehen werden.

Wir nehmen einmal an, daß nur der Inhalt von 1 Byte übertragen werden soll. In diesem Fall enthält die Instruktion die Längenangabe __(Z)__.

0

16 "0" ist die kleinste Längenangabe, die in einer Instruktion stehen kann.

Wir werden uns jetzt überlegen, welches die größt mögliche Längenangabe ist.

Im SS-Format stehen __(Z)__ Bits für die Längenverschlüsselung zur Verfügung.

Der größte Wert ist dargestellt, wenn alle diese Bits die Binärziffer (0, 1) _____ enthalten.

8

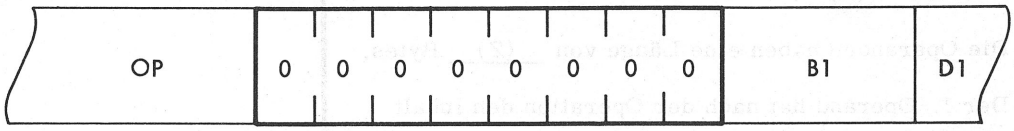
1

SS-FORMAT

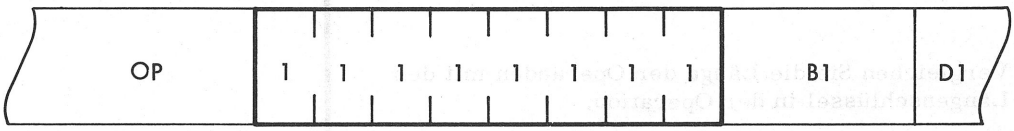
Feldlänge

(L)

binäre Darstellung des Wertes 0



binäre Darstellung des Wertes 255



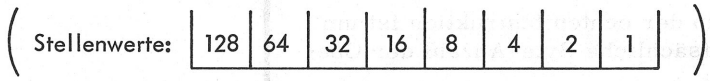


Abb. 7

17 Betrachten Sie Abb. 7.

Enthalten alle 8 Bits des Längenschlüssels die Binärziffer 1, so ist der Wert __(Z)__ dargestellt.

Diese maximale Längenangabe bezieht sich auf Operanden mit einer Länge von __(Z)__ Bytes.

255

256

18 Jetzt sind Ihnen der Aufbau und die Länge des SS-Formats bekannt.

Das 1. Byte enthält den _____, das 2. Byte enthält die Angabe der _____, das 3. und 4. Byte enthalten die _____ des 1. Operanden und das 5. und 6. Byte enthalten die __(WW)__.

Eine Instruktion im SS-Format hat immer eine Länge von __(Z)__ Bytes.

Operationsschlüssel

Feldlänge

Adresse/Hauptspeicheradresse

Adresse des 2. Operanden

6

19 Wir haben gesehen:

"SS" steht als Abkürzung für "Speicher-Speicher". Das SS-Format gilt also für Instruktionen, die 2 Felder im Speicher ansprechen.

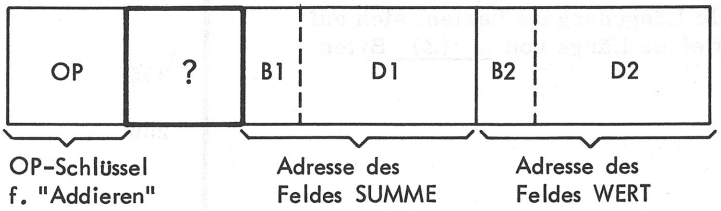
Entscheiden Sie:

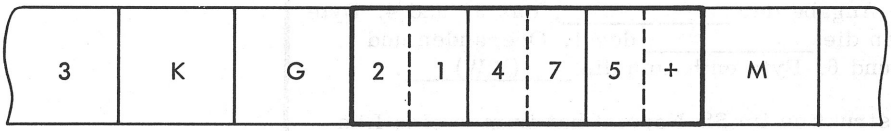
Ein Additionsbefehl, aufgrund dessen ein Wert im Speicher in ein Summenfeld im Speicher addiert werden soll, (hat, hat nicht) _____ das SS-Format.

hat

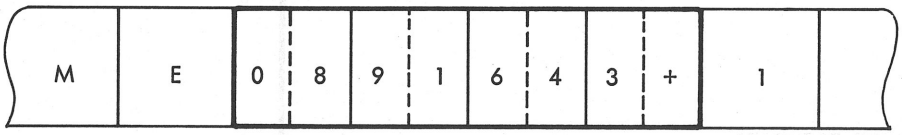
SS-FORMAT

FÜR ARITHMETISCHE BEFEHLE





WERT
(2. Operand)



SUMME
(1. Operand)

Abb. 8

20 Befehle für Rechenoperationen haben das SS-Format, wenn sie 2 Felder im Speicher ansprechen.

Solche Befehle haben also eine Länge von __(Z)__ Bytes.

Sie enthalten die genannten 4 Angaben:

den _____, die Angabe zur _____ der Datenfelder und die 2 _____.

6

Operationsschlüssel

Länge

Adressen/Hauptspeicheradressen

21 Befehle für Rechenoperationen, auch "arithmetische Befehle" genannt, haben allerdings eine kleine Besonderheit.

Die beiden Felder WERT und SUMME in Abb. 8 haben (gleiche, verschiedene) _____ Länge.

Damit die Additionsoperation richtig ausgeführt wird, müssen beide Operanden in ihrer eigenen Länge verarbeitet werden.

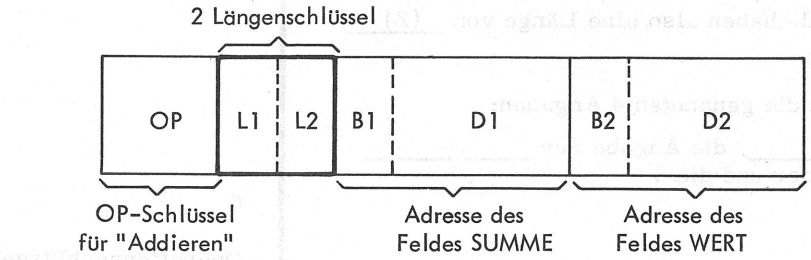
Das 2. Byte in der Instruktion muß daher (1, 2) _____ Längenangabe (n) enthalten.

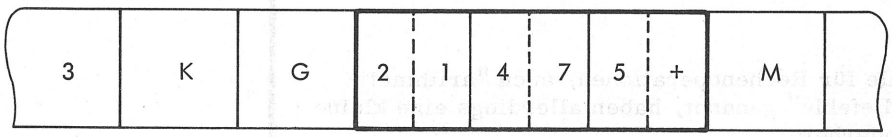
verschiedene

2

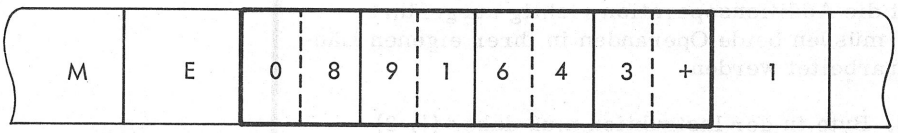
SS-FORMAT

FÜR ARITHMETISCHE BEFEHLE





WERT
(2. Operand)



SUMME
(1. Operand)

Abb. 9

22 In Abb. 9 enthält das 2. Byte der Instruktion die Abkürzungen L1 und L2.

"L1" ist der Längenschlüssel des Feldes _____ und "L2" der Längenschlüssel des Feldes _____ .

SUMME

WERT

23 Denken Sie an die Regel bezüglich der tatsächlichen Länge eines Feldes einerseits und der Längenangabe für dieses Feld in einer echten Instruktion andererseits.

Diese Regel gilt auch für die Angaben L1 und L2 in einem arithmetischen Befehl.

Für das Beispiel in Abb. 9 ist also L1 = __(Z)__ und L2 = __(Z)__.

3 (Das Feld SUMME ist 4 Bytes groß.)

2 (Das Feld WERT ist 3 Bytes groß.)

24 Für jede der beiden Angaben L1 und L2 stehen 4 Bits oder 1 Halbbyte zur Verfügung.

Sind alle 4 Binärziffern 0, so ist der Wert __(Z)__ dargestellt.

Sind alle 4 Binärziffern 1, so ist der Wert __(Z)__ dargestellt.

0

15 (= 1 + 2 + 4 + 8)

25 Für beide Operanden ist 0 die kleinste und 15 die größte Längenangabe.

Beide Operanden können also Längen zwischen __(Z)__ und __(Z)__ Bytes haben.

1 (und) 16

26 Bei diesen arithmetischen Befehlen enthalten die Operanden Dezimalzahlen in gepackter Darstellung.

Wieviel Ziffern hat eine Zahl, die in 16 Bytes gepackt gespeichert ist ?

31 (und 1 Vorzeichen)

SS-FORMAT

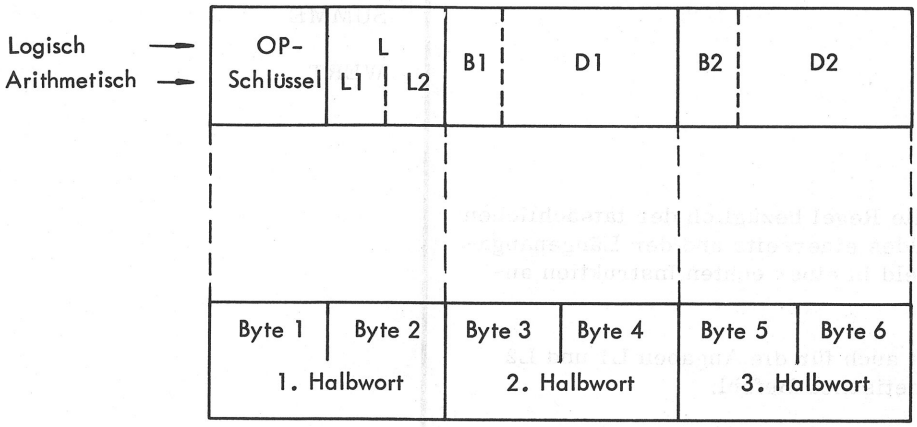


Abb. 10

27 Abb. 10 stellt die beiden Formen des SS-Formats einander gegenüber.

Sie heißen "arithmetisches" und "_____" SS-Format.

Das logische SS-Format hat __(Z)__ Längenschlüssel.

Das arithmetische SS-Format hat __(Z)__ Längenschlüssel.

logisches

1

2

28 Die Bezeichnung "logisch" gilt für alle Instruktionen im SS-Format, die nicht zu den arithmetischen Befehlen gehören.

In diesem Sinn hat eine Übertragungs-Instruktion die _____ Form des SS-Formats.

logische

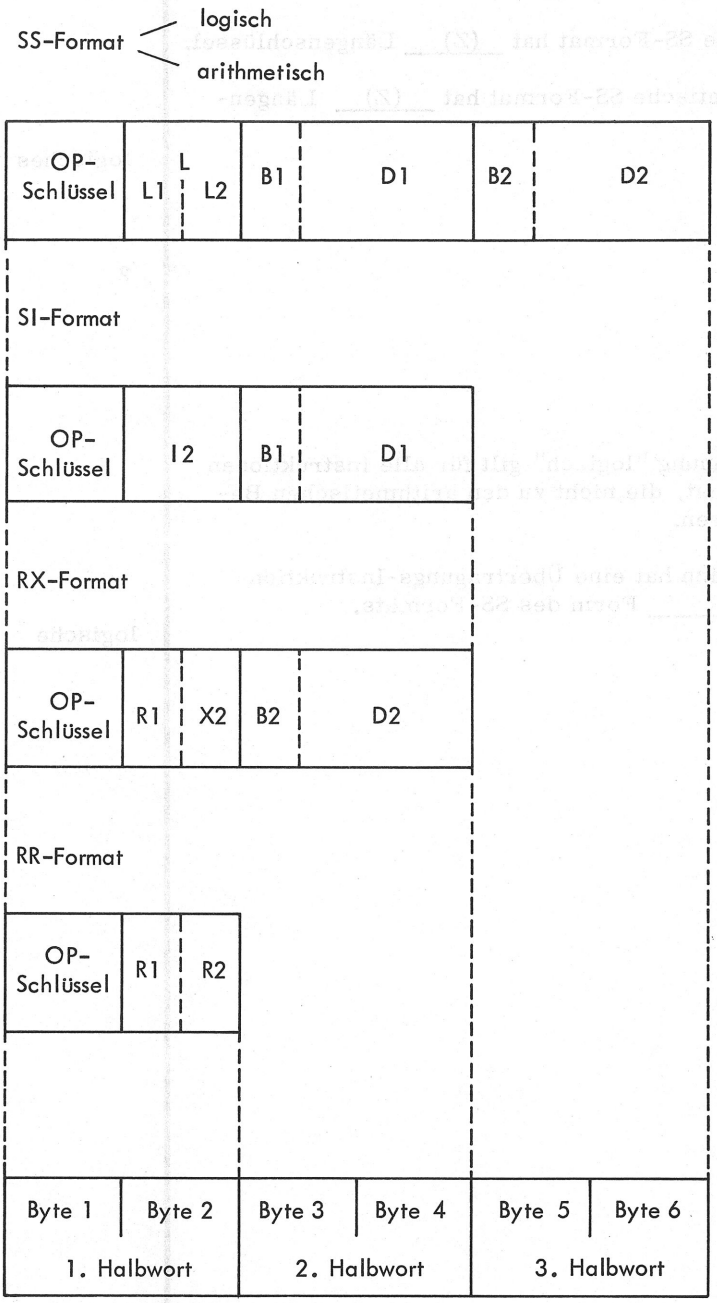

SS-Format
logisch
arithmetisch
OP-Schlüssel
L
L1
L2
B1
D1
B2
D2
SI-Format
OP-Schlüssel
I2
B1
D1
RX-Format
OP-Schlüssel
R1
X2
B2
D2
RR-Format
OP-Schlüssel
R1
R2
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
1. Halbwort
2. Halbwort
3. Halbwort

Abb. 11

29 Abb. 11 zeigt Ihnen alle 4 Formate, die in diesem Lehrgang behandelt werden.

Mit seinen 6 Bytes ist das SS-Format das längste.

Das SI- und das RX-Format haben beide eine Länge von __(Z)__ Bytes, und das RR-Format hat eine Länge von ___(ZW)___.

4

2 Bytes

30 Wir können auch sagen:

Die 4 Formate haben Längen von __(Z)__, __(Z)__ und __(Z)__ Halbworten.

Allen 4 Formaten ist eines gemeinsam:

Der _____ steht immer an der gleichen Stelle und zwar im ___(ZW)___.

3, 2 (und) 1

Operationsschlüssel

1. Byte

31 Bei der Durchführung einer Arbeit im DVS sind die Instruktionen im Hauptspeicher gespeichert. Ihre Länge beträgt immer ein Vielfaches von einem Halbwort oder 2 Bytes.

Damit hängt es zusammen, daß die Instruktionen im Speicher immer bei einer Adresse beginnen, die durch 2 teilbar ist.

Bei welchen der folgenden Adressen könnte also eine Instruktion beginnen:

3205, 168, 304, 1692, 2079 ?

168

304

1692

32 Eine Maschineninstruktion beginnt also im Speicher immer an einer H a l b w o r t g r e n z e .

Bei der Übersetzung der Assembler-Befehle in Maschinen-Befehle wird automatisch den Befehlen eine solche _____-grenze zugeteilt.

Halbwort

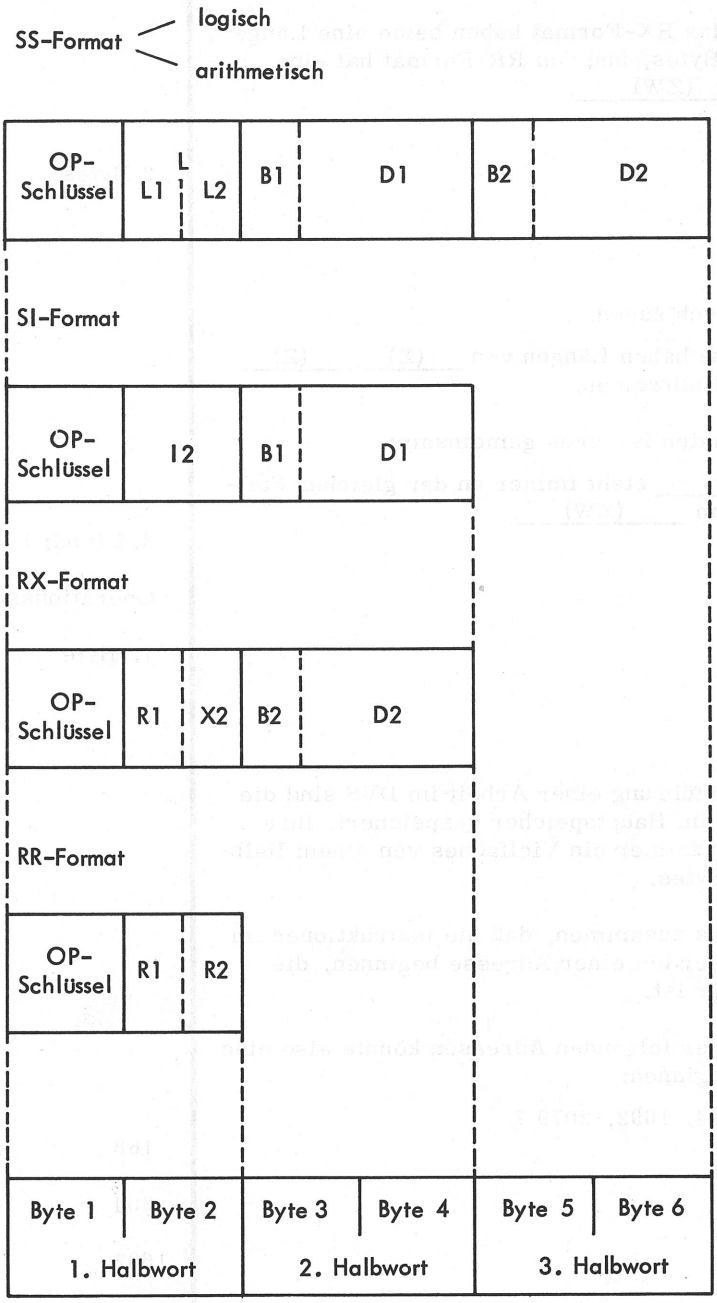

SS-Format
logisch
arithmetisch
OP-Schlüssel
L
L1
L2
B1
D1
B2
D2
SI-Format
OP-Schlüssel
I2
B1
D1
RX-Format
OP-Schlüssel
R1
X2
B2
D2
RR-Format
OP-Schlüssel
R1
R2
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
1. Halbwort
2. Halbwort
3. Halbwort

Abb. 11

33 In den folgenden Lernelementen werden wir uns mit den Formaten SI, RX und RR befassen.

Wir beginnen mit dem SI-Format. Aus Abb. 11 ersehen Sie Länge und Aufbau dieses Formats.

Jede Instruktion im SI-Format enthält im 1. Byte, den _____ und im 3. und 4. Byte die _____ des 1. Operanden.

Operationsschlüssel/OP-Schlüssel

Adresse

34 Im 2. Byte des SI-Formats steht eine neue Bezeichnung, nämlich _____.

Das "I" ist auch in dem Ausdruck "_____ -Format" enthalten.

I2

SI

35 Das "I" steht als Abkürzung für das englische Wort "immediate" (sprich: immidiet) = unmittelbar, direkt. Das 2. Byte enthält nämlich beim SI-Format einen D i r e k t operanden (Direktzeichen).

Durch den Index "2" ist ausgedrückt, daß der Direktoperand den (Z) Operanden einer SI-Instruktion darstellt.

Die Adresse im 3. und 4. Byte ist die Adresse des (ZW). Sie ist die Adresse eines Feldes im Hauptspeicher.

2.

1. Operanden

SI-FORMAT

BEISPIEL EINER INSTRUKTION

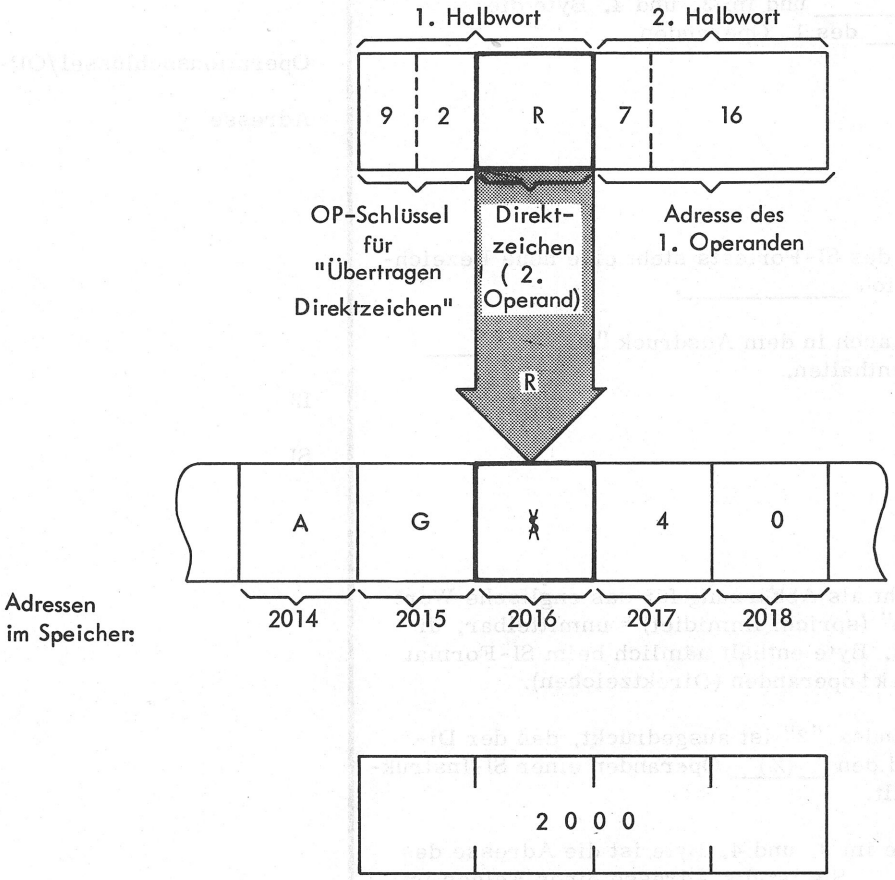


Abb. 12

36 Betrachten Sie bitte Abb. 12.

Sie finden hier ein Beispiel für eine SI-Instruktion. Es handelt sich um die Operation "___(WW)___".

In diesem Beispiel wird das Zeichen _____ übertragen.

Es ersetzt das Zeichen _____ im Byte mit der effektiven Adresse __(Z)__.

Übertragen Direktzeichen

R

S

2016

37 In der SI-Instruktion stehen die Adreßangaben für den 1. Operanden im __(Z)__ Halbwort.

Die effektive Adresse 2016 ergibt sich aus dem Inhalt __(Z)__ des Basisregisters __(Z)__ und aus der relativen Adresse __(Z)__.

2.

2000

7

16

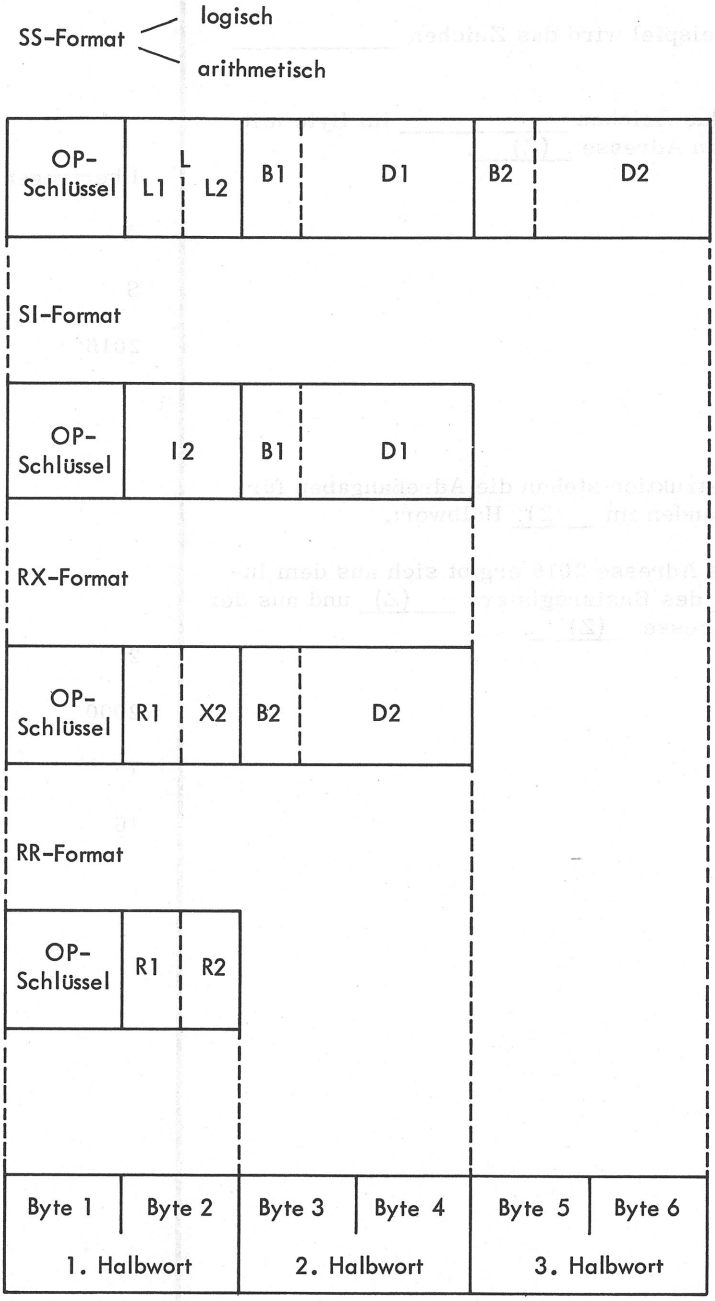

SS-Format
logisch
arithmetisch
OP-Schlüssel
L
L1
L2
B1
D1
B2
D2
SI-Format
OP-Schlüssel
I2
B1
D1
RX-Format
OP-Schlüssel
R1
X2
B2
D2
RR-Format
OP-Schlüssel
R1
R2
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
1. Halbwort
2. Halbwort
3. Halbwort

Abb. 11

38 Vergleichen Sie bitte in Abb. 11 den Aufbau des SS- und des SI-Formates.

3 Bytes haben in beiden Formaten den gleichen Inhalt.

Sie enthalten ____(WW)____.

den Operationsschlüssel und die Adresse des 1. Operanden

39 In den rechten 2 Bytes enthält das SS-Format die Adresse des 2. Operanden.

Das SI-Format enthält im ____(ZW)____ den ____(ZW)____ selbst.

2. Byte

2. Operanden

SI-FORMAT

BEISPIEL EINER INSTRUKTION

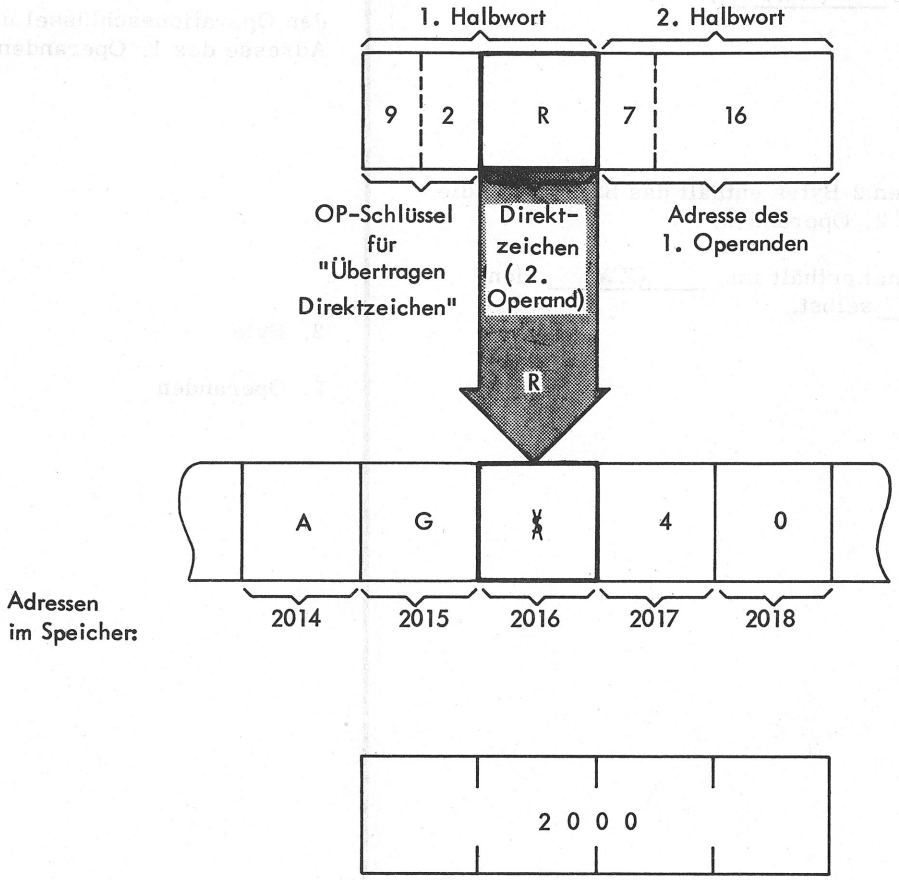


Abb. 12

40 Im 2. Byte einer SI-Instruktion steht also nicht die Adresse des 2. Operanden, sondern der 2. Operand selbst.

Er ist direkt im Befehl enthalten.

Darum heißt er auch "_____-Operand".

In Abb. 12 enthält er das Direktzeichen _____.

Direkt

R

41 Da der 2. Operand selbst im 2. Byte der SI-Instruktion steht, ist seine Länge konstant.

Er ist immer ____(ZW)____ groß.

Im Gegensatz zum SS-Format erübrigt sich also beim SI-Format die Angabe der _____.

1 Byte

Länge

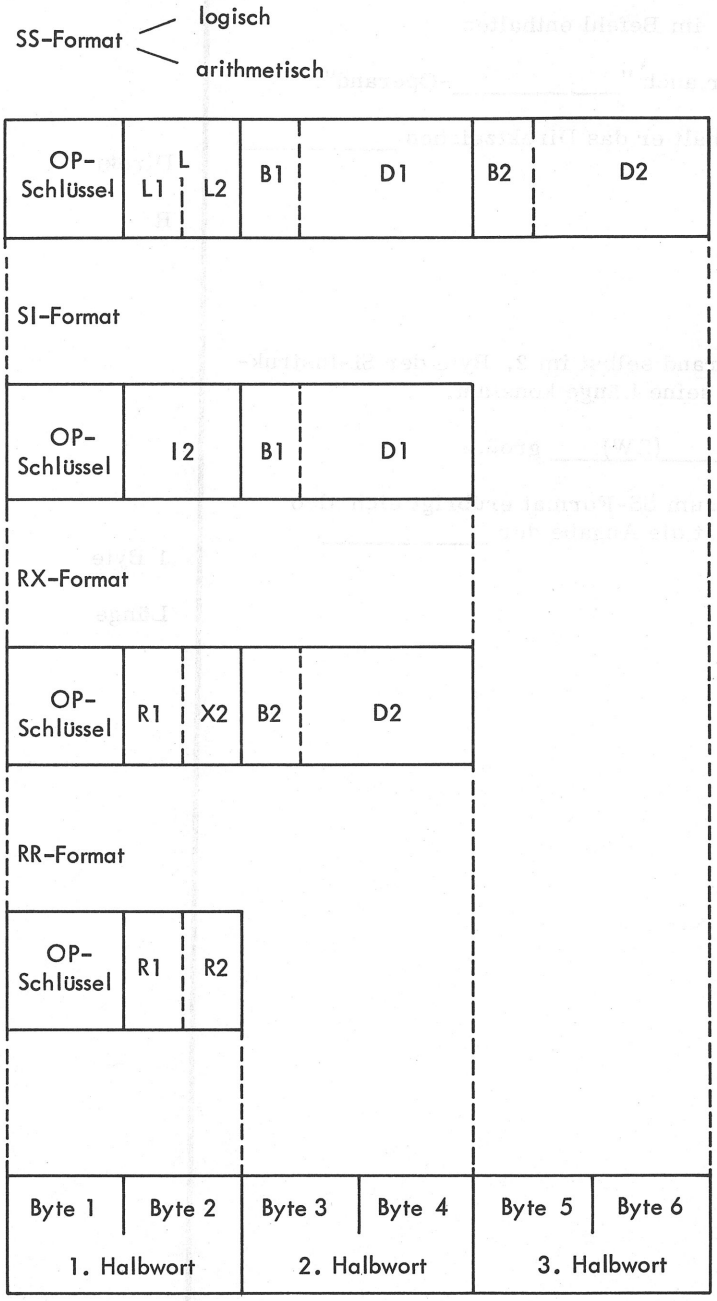

SS-Format
logisch
arithmetisch
OP-Schlüssel
L
L1
L2
B1
D1
B2
D2
SI-Format
OP-Schlüssel
I2
B1
D1
RX-Format
OP-Schlüssel
R1
X2
B2
D2
RR-Format
OP-Schlüssel
R1
R2
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
1. Halbwort
2. Halbwort
3. Halbwort

Abb. 11

42 Nennen Sie bitte zur Wiederholung die 3 Teile einer SI-Instruktion:

Das 1. Byte enthält den _____.

Das 2. Byte enthält den _____.

Das 3. und 4. Byte enthalten die ____(WW)_____.

Operationsschlüssel

Direktoperanden/2. Operanden

Adresse des 1. Operanden

43 Abb. 11. zeigt Ihnen noch ein weiteres Instruktionsformat mit einer Länge von 4 Bytes.

Es heißt _____.

Diesen Namen hat das Format nach dem Inhalt des ___(Z)___ Bytes.

RX-Format

2.

44 Alle Angaben außer "R1" und "X2" sind Ihnen in ihrer Bedeutung schon bekannt.

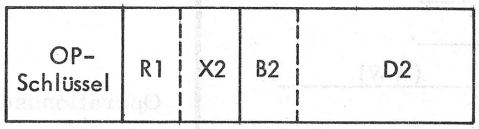
Das 1. Byte enthält wie immer den _____ und das 3. und 4. Byte enthalten Angaben zur ____(WW)____.

Operationsschlüssel

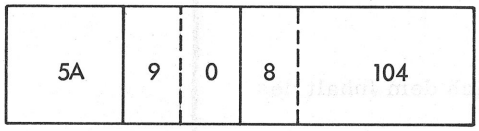
Adresse des 2. Operanden

RX-FORMAT

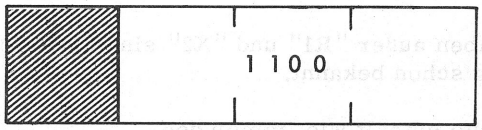
Allgemein:



Beispiel: "Addieren"



Register 8



Adresse des 2. Operanden:

Relative Adresse	104
Basis- Adresse	+ 1100
Effektive Adresse	1204

Abb. 13

45 Abb. 13 zeigt Ihnen ein Beispiel einer RX-Instruktion.

Das 1. Byte enthält den Operationsschlüssel für die Operation "_____". Der Inhalt des 2. Operanden soll a d d i e r t werden in ein Summenfeld.

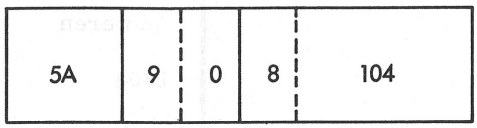
Der 2. Operand steht im Speicher und hat die (effektive) Adresse ___(Z)___.

Addieren

1204

RX-FORMAT

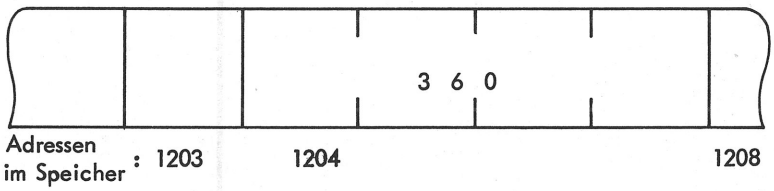
Beispiel: "Addieren"



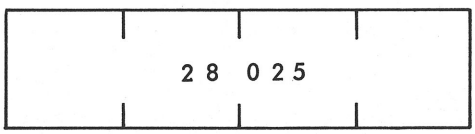
Adresse des 2. Operanden:

Relative Adresse	104
Basis- Adresse	+1100
Effektive Adresse	1204

Addition:

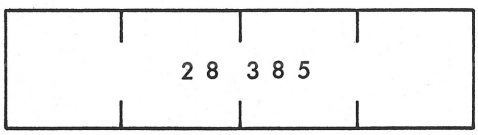






Register 9
(vorher)





Register 9
(nachher)

Abb. 14

46 In Abb. 14 sind beide Operanden der Additions-Instruktion dargestellt.

Der 2. Operand steht im Speicher. Er beginnt bei der Adresse __(Z)__ und enthält den Wert __(Z)__.

Der 1. Operand ist das _____ mit der Nummer __(Z)__.

Es enthält vor Ausführung des Additionsbefehls den Wert __(Z)__.

1204

360

Register

9

28 025

47 Der Additionsbefehl in Abb. 14 bewirkt, daß der Inhalt eines Feldes im Speicher in ein _____ addiert wird.

Das Register 9 enthält also nach der Ausführung des Befehls den Summenwert __(Z)__.

Register

28 385

48 Die beiden Operanden des Additionsbefehls in Abb. 14 stehen an ganz verschiedenen Stellen in der Zentraleinheit. Der 1. Operand ist ein Register, der 2. Operand steht im Hauptspeicher.

Beide haben jedoch die gleiche Länge von __(Z)__ Bytes oder einem _____.

4

Wort

49 In einem Register ist ein Zahlenwert immer (ungepackt, gepackt, dual) _____ dargestellt.

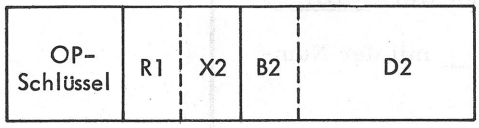
Wird der Inhalt eines Feldes im Speicher in ein Register addiert, so müssen die Daten auch in diesem Feld _____ verschlüsselt sein.

dual

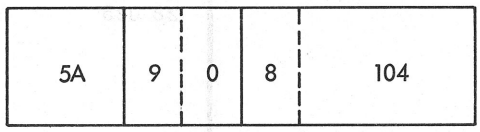
dual

RX-FORMAT

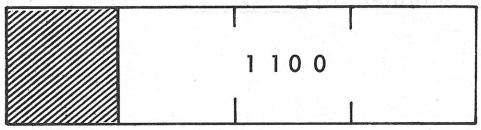
Allgemein:



Beispiel: "Addieren"



Register 8



Adresse des 2. Operanden:

Relative Adresse	104
Basis- Adresse	+ 1100
Effektive Adresse	1204

Abb. 13

50 Ist ein Register an einer Rechenoperation beteiligt, so handelt es sich immer um (duales, dezimales) _____ Rechnen.

duales

Wir fassen zusammen:

1. Der Additionsbefehl in Abb. 13 bezieht sich auf die Addition dualer Daten.

2. Die Operanden haben Wortgröße.

3. Der 2. Operand steht im Hauptspeicher, während der 1. Operand ein Register ist.

4. Das Ergebnis bildet sich im 1. Operanden, also im Register.

51 Die zusammengestellten Merkmale eines Additionsbefehls gelten für alle arithmetischen Befehle im RX-Format.

Es gibt noch andere Instruktionen im RX-Format. Allen Befehlen in diesem Format ist gemeinsam, daß der 2. Operand ein Feld im _____ ist.

Hauptspeicher

52 Betrachten Sie noch einmal das RX-Format im oberen Teil der Abb. 13.

Ein Halbbyte in diesem Format haben wir noch nicht besprochen.

Es enthält die Bezeichnung _____.

Die "2" deutet darauf hin, daß diese Angabe zur Adresse des __(Z)__ Operanden gehört.

X2

2.

53 Im Instruktionsbeispiel in Abb. 13 hat dieses 4. Halbbyte der Instruktion den Inhalt __(Z)__.

Diese Angabe 0 hat zur Bildung der effektiven Adresse 1204 des 2. Operanden (beigetragen, nicht beigetragen) _____.

0

nicht beigetragen

RX-FORMAT

ADRESSE DES 2. OPERANDEN

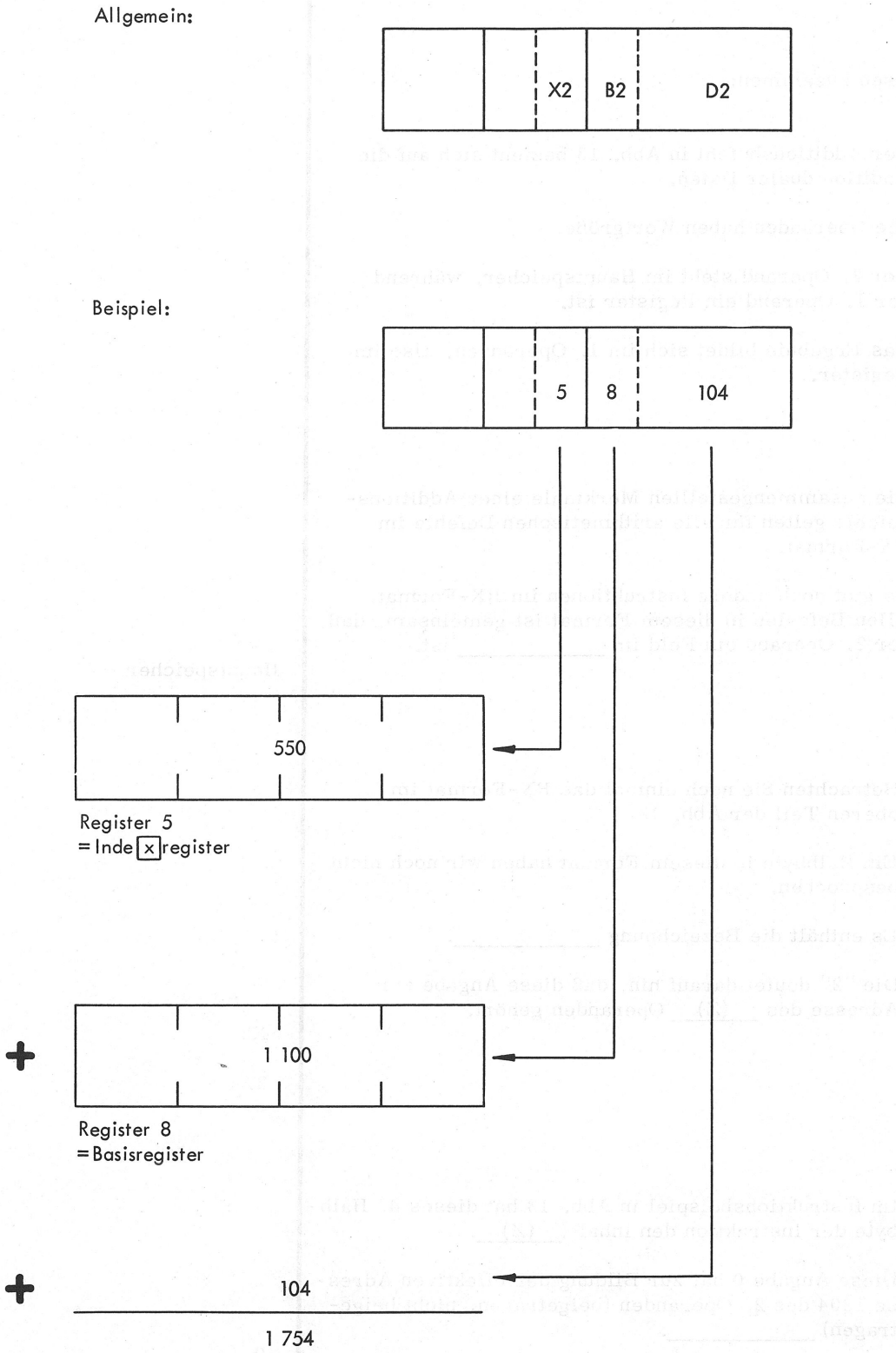


Abb. 15

Wir merken uns:

Hat dieses 4. Halbbyte beim RX-Format den Inhalt 0, so ist es für die Adreßbildung ohne Bedeutung.

Die 0 bedeutet "Register 0". Allgemein gilt:

Immer wenn Register 0 im Zusammenhang mit der Bildung einer Hauptspeicheradresse auftaucht, wird sein Inhalt ignoriert.

54 Betrachten Sie Abb. 15.

Der Halbbyte-Inhalt X 2 = __(Z)__ trägt zur Bildung der effektiven Adresse des 2. Operanden bei.

"5" ist die Nummer des __(WZ)__.

5

Registers 5

55 Abb. 15 zeigt, wie die effektive Adresse 1754 zustande kommt.

Ergänzen Sie bitte:

__(Z)__ + __(Z)__ + __(Z)__ = 1754

550 + 1100 + 104 = 1754

Das 4. Halbbyte kann also beim RX-Format zur zusätzlichen Erhöhung oder "Indizierung" der Adresse des 2. Operanden benutzt werden.

Der Buchstabe "X" bei diesem Format ist der Bezeichnung "Indexregister" für das zusätzliche Register entnommen.

RX-FORMAT

Sonderform mit Bedingungsmaske

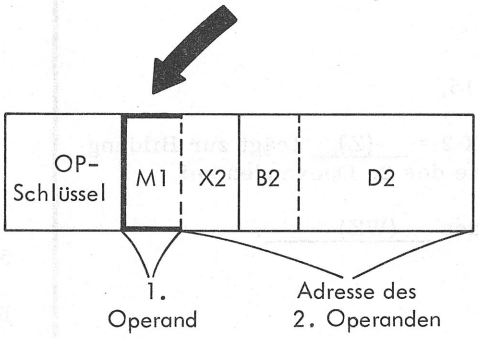


Abb. 16

56 Sie kennen jetzt das RX-Format in der normalen Form.

Es gibt aber noch eine Sonderform. Bei ihr hat das "R" in der Bezeichnung "RX" eigentlich keine Berechtigung mehr. Sie finden diese Sonderform in Abb. 16.

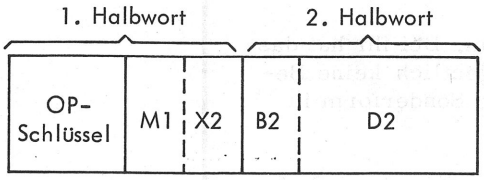
Hier ist der __(Z)__ Operand kein Register. Er ist angegeben mit _____.

1.

M 1

"RX-FORMAT"

Allgemein:



Beispiel: "Verzweigen bei Bedingung"

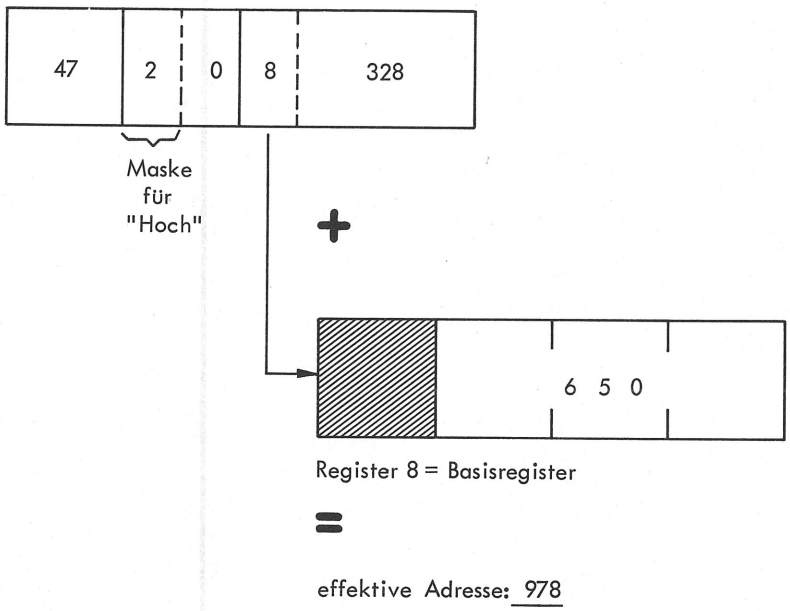


Abb. 17

57 Abb. 17 zeigt Ihnen ein Beispiel zur Sonderform des RX-Formats.

Die Operation heißt " ___(WW)___ ".

Das 3. Halbbyte enthält die _____ für die Bedingung "Hoch".

Verzweigen bei Bedingung

Maske

58 Das "M" bei dieser Sonderform des RX-Formats ist also die Abkürzung für M a s k e (Bedingungsmaske).

In unserem Beispiel hat das mit M1 bezeichnete Halbbyte den Inhalt 2. Es wird nur verzweigt, wenn die Bedingung "_____" erfüllt ist.

Hoch

59 Ist die Bedingung "Hoch" erfüllt, so muß das Programm zu dem Befehl verzweigen, der bei der Adresse ___(Z)___ steht.

Diese Adresse ist die Verzweigadresse.

978 (= 650 + 328)

RX-FORMAT

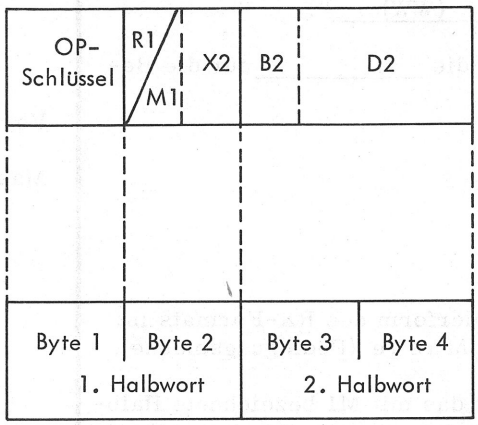


Abb. 18

60 Wir fassen zum RX-Format zusammen (vgl. Abb.18):

Das RX-Format umfaßt __(Z)__ Bytes oder __(Z)__ Halbworte.

Es enthält im 1. und 2. Halbbyte den _____,
im 3. Halbbyte eine _____ oder eine
_____ (1. Operand)
und im 4. bis 8. Halbbyte Angaben zu einer
_____ (2. Operand).

4

2

Operationsschlüssel

Registeradresse

Bedingungsmaske

Hauptspeicheradresse

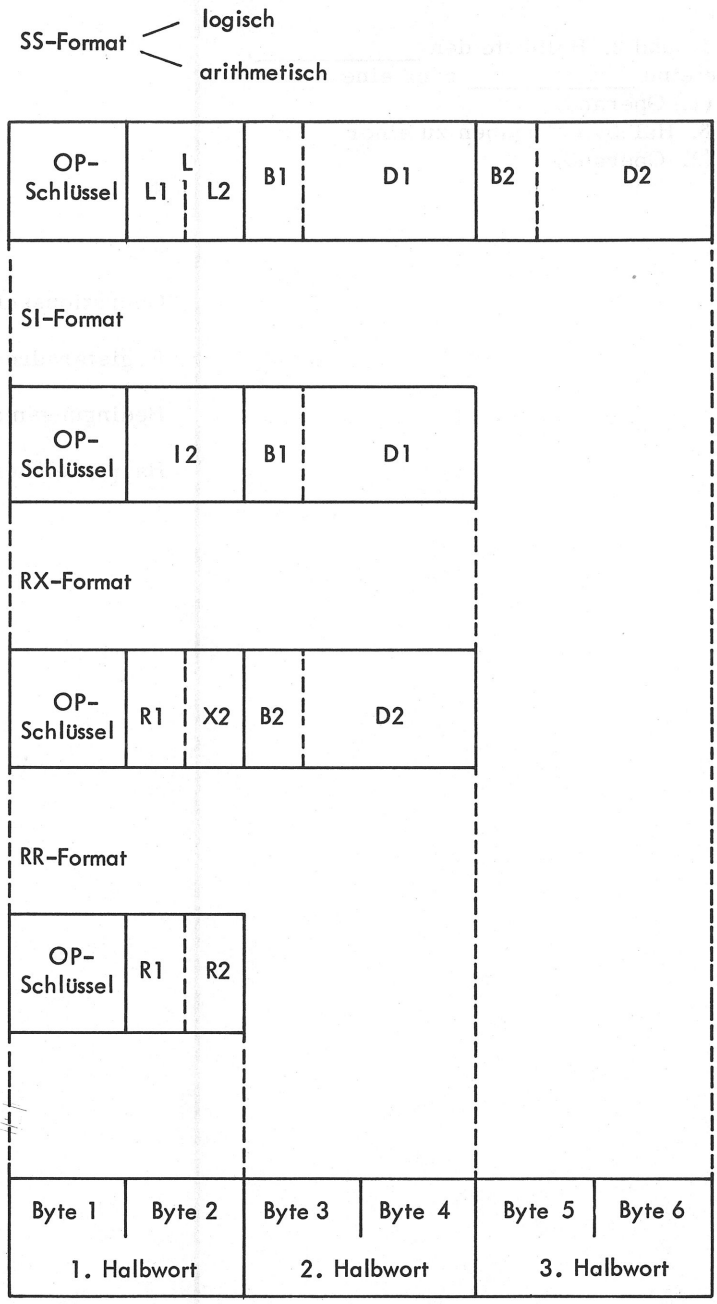

SS-Format
logisch
arithmetisch
OP-Schlüssel
L
L1
L2
B1
D1
B2
D2
SI-Format
OP-Schlüssel
I2
B1
D1
RX-Format
OP-Schlüssel
R1
X2
B2
D2
RR-Format
OP-Schlüssel
R1
R2
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
1. Halbwort
2. Halbwort
3. Halbwort

Abb. 11

61 Wir wenden uns jetzt dem 4. Format in Abb. 11 zu.

Es heißt _____ und ist __(Z)__ Bytes groß.

Der Name deutet die Art der Operanden an. Sowohl der 1. als auch der 2. Operand sind _____.

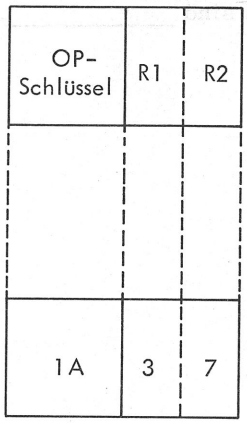
RR-Format

2

Register

RR-FORMAT

Allgemein:



Beispiel: "Addieren Register"

1A	3	7

Abb. 19

62 Abb. 19 zeigt als Beispiel zum RR-Format eine Additions-Instruktion.

Der Inhalt des 2. Operanden, also von Register (Z), soll in den 1. Operanden, also in (WZ) addiert werden.

Das Ergebnis bildet sich in Register 3, also im (Z) Operanden.

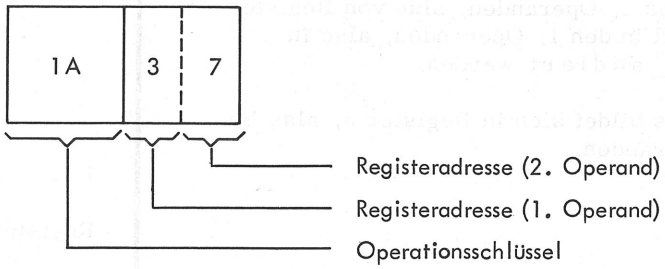
7

Register 3

1.

RR-FORMAT

Beispiel:
"Addieren Register"



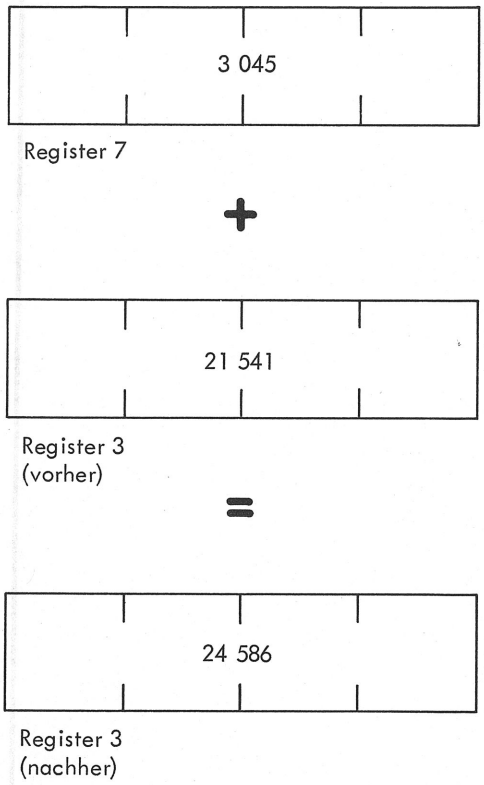


Abb. 20

63 In Abb. 20 sind die Werte in den beiden Operanden, Register 7 und Register 3, dezimal angegeben.

Da es sich um Register handelt, sind die Werte in Wirklichkeit _____ verschlüsselt.

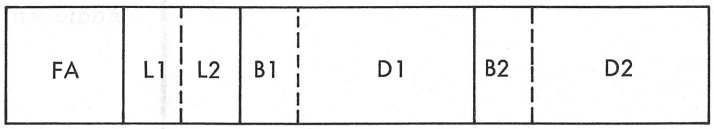
Die Operation " _____ " ist also eine duale Addition.

dual

Addieren Register

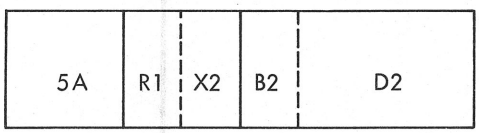
I . SS-Format

"Addieren Dezimal"



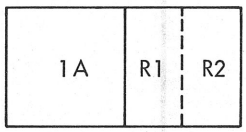
II . RX-Format

"Addieren"



III . RR-Format

"Addieren Register"



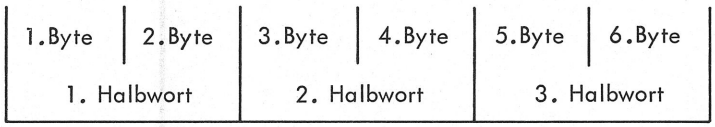


Abb. 21

64 Sie haben in diesem Kapitel 3 verschiedene Additions-Instruktionen kennengelernt. Sie dienten als Beispiele zu den Formaten SS, RX und RR.

Abb. 21 zeigt diese 3 Befehle.

Die Länge der Instruktionen hängt vom Ort der Operanden in der Zentraleinheit ab.

Notieren Sie sich auf einem Extrablatt, wie die folgende Tabelle auszufüllen ist.

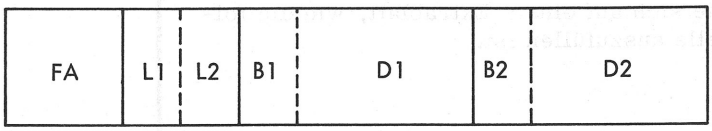
Beispiel	Instruktions-Format	Anzahl der Speicherfelder als Operanden	Anzahl der Register als Operanden	Länge der Befehle in Bytes
I				
II				
III				

Die Lösung finden Sie auf der folgenden Seite.

3 ADDITIONS-INSTRUKTIONEN

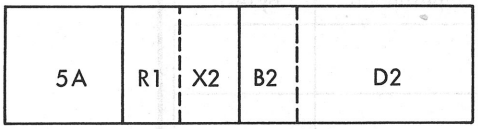
I . SS-Format

"Addieren Dezimal"



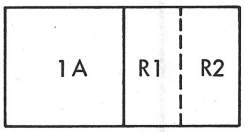
II . RX-Format

"Addieren"



III . RR-Format

"Addieren Register"



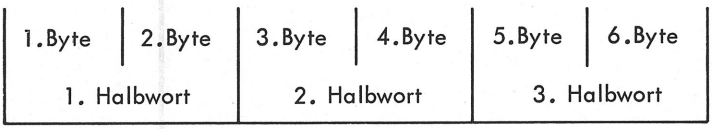


Abb. 21

Beispiel	Instruktions-Format	Anzahl der Speicherfelder als Operanden	Anzahl der Register als Operanden	Länge der Befehle in Bytes
I	SS	2	-	6
II	RX	1	1	4
III	RR	-	2	2

65 Die unterschiedliche Länge der Instruktionsformate ergibt sich aufgrund des unterschiedlichen Platzbedarfs für die Adressen der Operanden.

Die Adreßangabe für ein Register benötigt
___(ZW)___.

4 Bits / 1 Halbbyte

Die Adreßangabe für ein Hauptspeicherfeld benötigt
___(ZW)___.

2 Bytes
(bei Indizierung 5 Halbbytes)

Bei Instruktionen im SS-Format kommt noch der Platzbedarf für die Längenangaben hinzu.

66 Wir schließen die Behandlung des RR-Formates jetzt ab und wiederholen:

Alle Instruktionen, die sich auf 2 Register als Operanden beziehen, haben das RR-Format.

Bei einer Länge von 1 Halbwort enthalten sie

im 1. und 2. Halbbyte den _____,

Operationsschlüssel

im 3. Halbbyte die ___(WW)___,

Adresse eines Registers (1. Operand)

im 4. Halbbyte die ___(WW)___.

Adresse eines Registers (2. Operand)

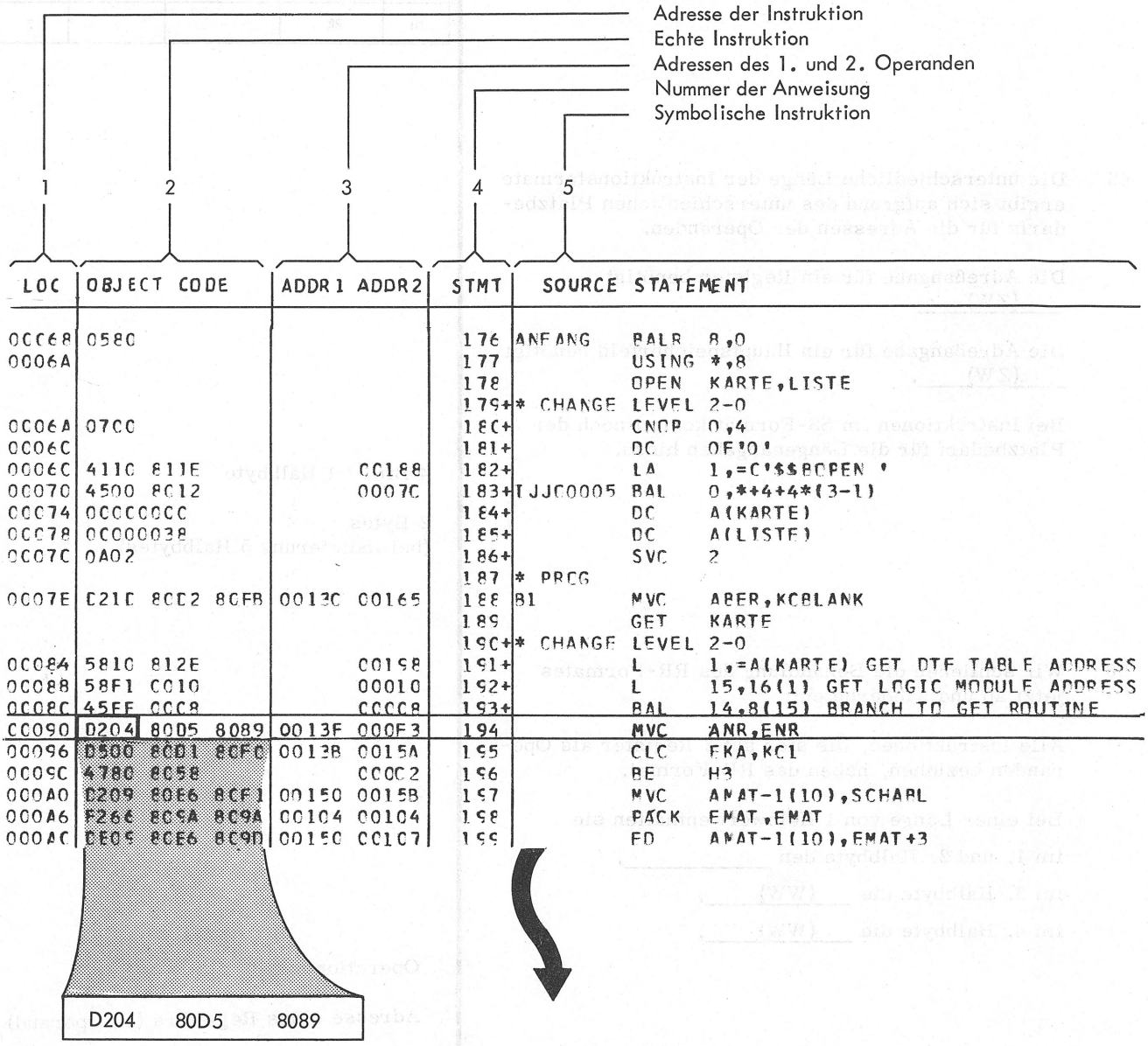

Adresse der Instruktion
Echte Instruktion
Adressen des 1. und 2. Operanden
Nummer der Anweisung
Symbolische Instruktion
1
2
3
4
5
LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT
00C68 058C 176 ANFANG BALR 8,0
0006A 177 USING *,8
178 OPEN KARTE,LISTE
179+* CHANGE LEVEL 2-0
00C6A 07CC 180+ CNOP 0,4
00C6C 181+ DC 0F'0'
00C6C 4110 811E 00188 182+ LA 1,=C'$$BOPEN '
00C70 4500 8012 0007C 183+IJJ00005 BAL 0,*+4+4*(3-1)
00C74 00CCCCCC 184+ DC A(KARTE)
00C78 00000038 185+ DC A(LISTE)
00C7C 0A02 186+ SVC 2
187 * PROG
00C7E D21D 80D2 80FB 0013C 00165 188 B1 MVC ABER,KCBLANK
189 GET KARTE
190+* CHANGE LEVEL 2-0
00C84 5810 812E 00198 191+ L 1,=A(KARTE) GET DTF TABLE ADDRESS
00C88 58F1 C010 00010 192+ L 15,16(1) GET LOGIC MODULE ADDRESS
00C8C 45EF CCC8 CCCC8 193+ BAL 14,8(15) BRANCH TO GET ROUTINE
00C90 D204 80D5 8089 0013F 000F3 194 MVC ANR,ENR
00C96 D500 80D1 80FC 00138 0015A 195 CLC EKA,KC1
00C9C 4780 8C58 CC0C2 196 BE H3
00CA0 D209 80E6 80F1 00150 0015B 197 MVC AMAT-1(10),SCHABL
00CA6 F266 80CA 80CA 00104 00104 198 PACK EMAT,EMAT
00CAC DE05 80E6 80CD 00150 00107 199 ED AMAT-1(10),EMAT+3
D204 80D5 8089

Abb. 22

Sie kennen jetzt die 4 Instruktionsformate, die Gegenstand dieses Lehrgangs sind. Sie wissen, wie 6, 4 oder 2 Bytes bei den verschiedenen Formaten aufgeteilt sind.

Abb. 22 zeigt nun einen Ausschnitt aus einer Umwandlungsliste, in der auch die echten Befehle vorkommen. Wie Sie wissen, wird diese Liste vom DVS während der Umwandlung des symbolischen Programms in das echte Programm erstellt. Sie dient der Fehlersuche beim Austesten und der Dokumentation.

Im folgenden erhalten Sie Hinweise zum Verständnis einer solchen Umwandlungsliste. Es kommt jedoch nicht darauf an, daß Sie Einzelheiten behalten.

67 Die Bedeutung der einzelnen Spalten in der Liste finden Sie rechts oben in Abb. 22 angegeben:

Die Liste enthält für jeden Befehl des Programms die symbolische Form (Spalte 5) und die echte Form (Spalte __(Z)__).

Die Adressen der Operanden sind außer in den echten Befehlen selbst noch ein zweitesmal in anderer Schreibweise angegeben (Spalte __(Z)__).

Alle Anweisungen des Programms sind durchnumeriert (Spalte __(Z)__).

Während der Ausführung des Programms werden die in Spalte 2 angegebenen Befehle im Hauptspeicher stehen. Die Umwandlungsliste zeigt, an welchen Adressen die echten Befehle dann gespeichert sein werden (Spalte __(Z)__).

2

3

4

1

68 Eine der Maschineninstruktionen ist in Abb. 22 unten besonders hervorgehoben. Es ist dies die echte Form zu der symbolischen Instruktion MVC ANR, ENR.

Diese Maschineninstruktion (Spalte 2) umfaßt __(Z)__ Zeichen.

Jedes dieser Zeichen gibt den Inhalt eines halben Bytes an. Der Befehl hat also eine Länge von __(ZW)__ und gehört deshalb zum _____ -Format.

12

6 Bytes/3 Halbworten

SS

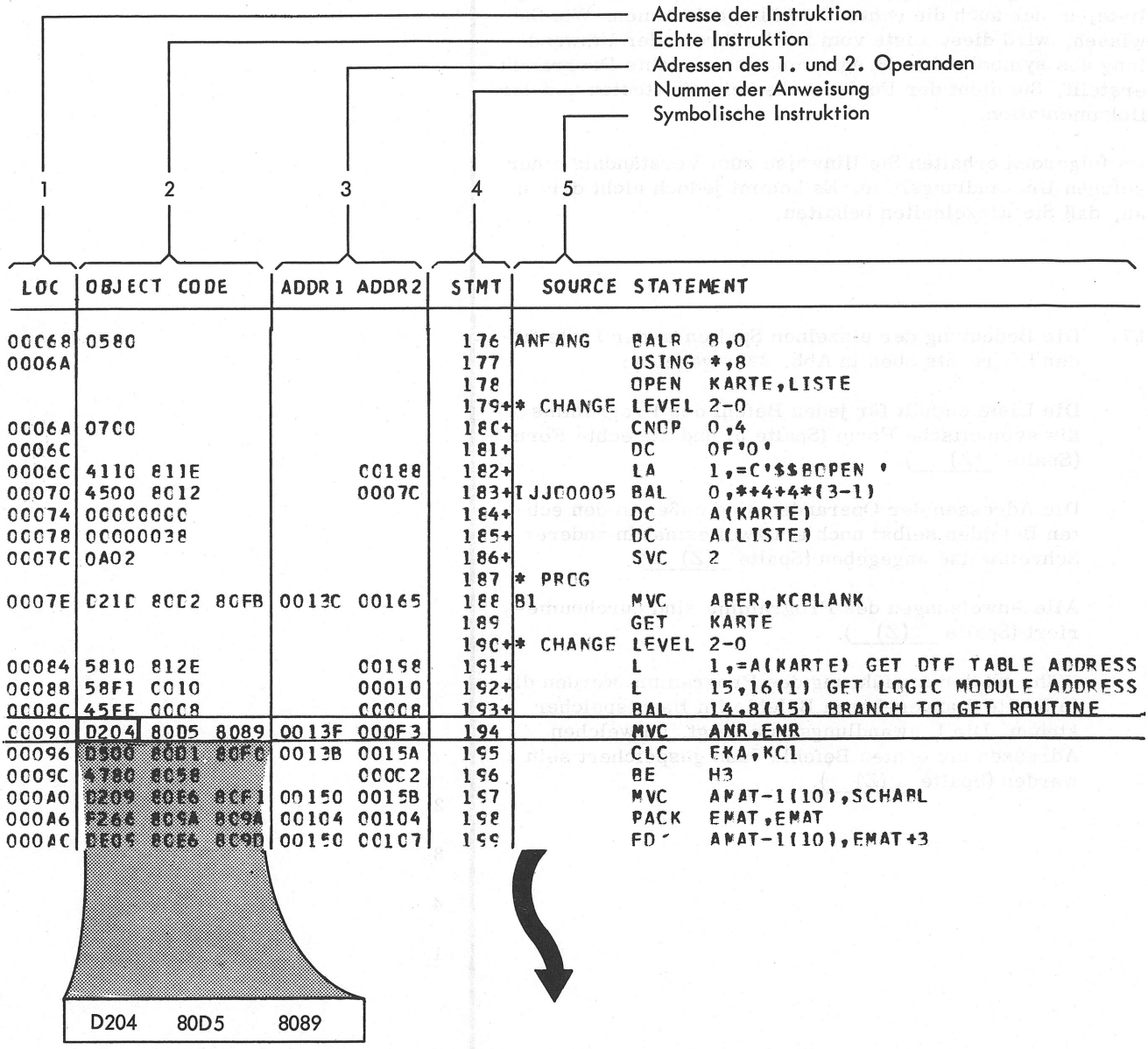

Adresse der Instruktion
Echte Instruktion
Adressen des 1. und 2. Operanden
Nummer der Anweisung
Symbolische Instruktion
1
2
3
4
5
LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT
00068 0580 176 ANFANG BALR 8,0
0006A 177 USING *,8
178 OPEN KARTE,LISTE
179+* CHANGE LEVEL 2-0
0006A 07C0 180+ CNOP 0,4
0006C 181+ DC 0F'0'
0006C 4110 811E 00188 182+ LA 1,=C'$$BOPEN '
00070 4500 8012 0007C 183+IJJ00005 BAL 0,*+4+4*(3-1)
00074 00000000 184+ DC A(KARTE)
00078 00000038 185+ DC A(LISTE)
0007C 0A02 186+ SVC 2
187 * PROG
0007E D21D 80D2 80FB 0013C 00165 188 B1 MVC ABER,KCBLANK
189 GET KARTE
190+* CHANGE LEVEL 2-0
00084 5810 812E 00198 191+ L 1,=A(KARTE) GET DTF TABLE ADDRESS
00088 58F1 C010 00010 192+ L 15,16(1) GET LOGIC MODULE ADDRESS
0008C 45EF 00C8 000C8 193+ BAL 14,8(15) BRANCH TO GET ROUTINE
00090 D204 80D5 8089 0013F 000F3 194 MVC ANR,ENR
00096 D500 80D1 80FC 0013B 0015A 195 CLC EKA,KC1
0009C 4780 80C8 000C2 196 BE H3
000A0 D209 80E6 80F1 00150 0015B 197 MVC AMAT-1(10),SCHABL
000A6 F266 80EA 80EA 00104 00104 198 PACK EMAT,EMAT
000AC DE09 80E6 80ED 00150 00107 199 ED AMAT-1(10),EMAT+3
D204 80D5 8089

Abb. 22

69 Ergänzen Sie bitte die folgende Tabelle für den in Abb. 22 hervorgehobenen echten Befehl im SS-Format:

	1. Byte	2. Byte	3. u. 4. Byte	5. u. 6. Byte
Inhalt	D2			
Bedeutung			Angaben zur Adresse des 1. Operanden	

	1. Byte	2. Byte	3. u. 4. Byte	5. u. 6. Byte
Inhalt	D2	04	80D5	8089
Bedeutung	Operations-schlüssel	Längen-angabe	Angaben zur Adresse des 1. Operanden	Angaben zur Adresse des 2. Operanden

70 Der Operationsschlüssel "D2" bewirkt die Übertragung von Zeichen im Hauptspeicher. Es handelt sich bei diesem Befehl um die logische Form des SS-Formates.

Der Befehl hat also (einen, zwei) _____ Längenschlüssel.

einen

71 Der Übertragungsbefehl in Abb. 22 enthält die Längenangabe "04". Mit diesen 2 hexadezimalen Ziffern ist die Dezimalzahl "4" dargestellt.

Die Operation spricht also Felder mit einer Länge von __(Z)__ Bytes an.

5

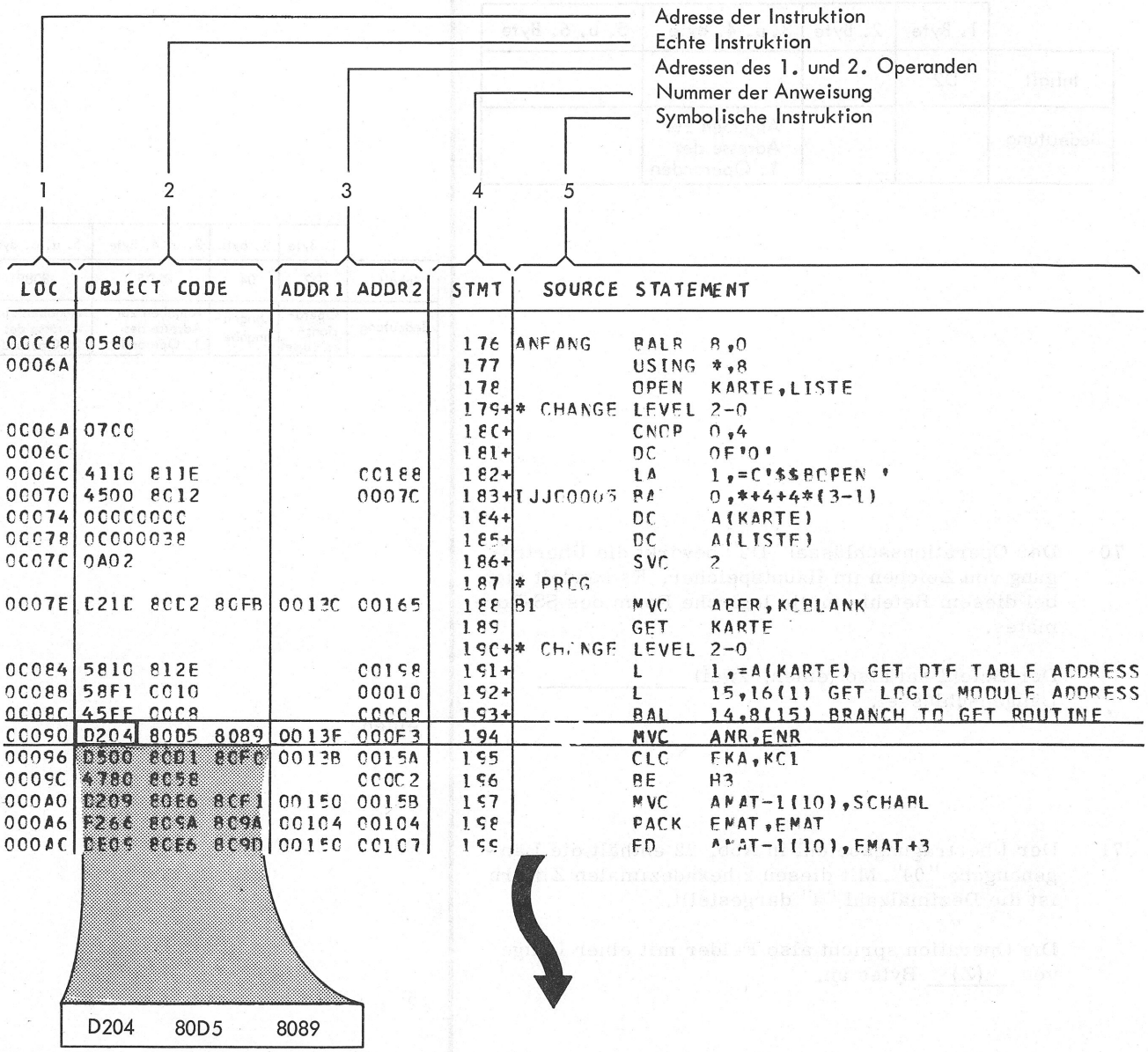

Adresse der Instruktion
Echte Instruktion
Adressen des 1. und 2. Operanden
Nummer der Anweisung
Symbolische Instruktion
1
2
3
4
5
LOC | OBJECT CODE | ADDR1 ADDR2 | STMT | SOURCE STATEMENT
00068 0580 176 ANFANG BALR 8,0
0006A 177 USING *,8
178 OPEN KARTE,LISTE
179+* CHANGE LEVEL 2-0
0006A 07C0 180+ CNOP 0,4
0006C 181+ DC 0F'0'
0006C 4110 811E 00188 182+ LA 1,=C'$$BOPEN '
00070 4500 8012 0007C 183+IJJ00003 BA 0,*+4+4*(3-1)
00074 00000000 184+ DC A(KARTE)
00078 00000038 185+ DC A(LISTE)
0007C 0A02 186+ SVC 2
187 * PROG
0007E D21D 80D2 80FB 0013C 00165 188 B1 MVC ABER,KOBLANK
189 GET KARTE
190+* CHANGE LEVEL 2-0
00084 5810 812E 00198 191+ L 1,=A(KARTE) GET DTF TABLE ADDRESS
00088 58F1 C010 00010 192+ L 15,16(1) GET LOGIC MODULE ADDRESS
0008C 45EF CCC8 CCCC8 193+ BAL 14,8(15) BRANCH TO GET ROUTINE
00090 D204 80D5 8089 0013F 000F3 194 MVC ANR,ENR
00096 D500 80D1 80FC 0013B 0015A 195 CLC EKA,KC1
0009C 4780 8058 00002 196 BE B3
000A0 D209 80E6 8CF1 00150 0015B 197 MVC AMAT-1(10),SCHARL
000A6 F26C 8C9A 8C9A 00104 00104 198 PACK EMAT,EMAT
000AC DE09 8CE6 8C9D 00150 00107 199 ED AMAT-1(10),EMAT+3
D204 80D5 8089

Abb. 22

72 Die Angaben zu den beiden Adressen sagen aus, wo die 5 Bytes großen Felder im Speicher stehen. Im Übertragungsbefehl in Abb. 22 enthalten das 3. und 4. Byte die Angaben

"80D5"

Daraus können Sie entnehmen:

Für die Adreßbildung ist das Basisregister __(Z)__ benutzt.

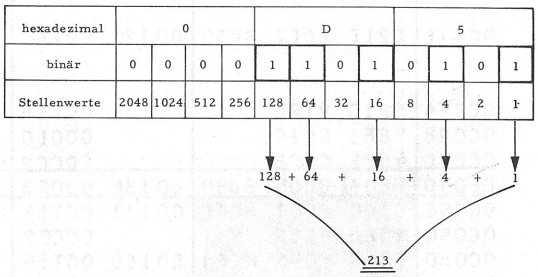
Die relative Adresse ist hexadezimal mit __(Z)__ angegeben.

8

0D5

73 Basisregister 8 enthält beim Programmierbeispiel in Abb. 22 den Wert "106"(hexadezimal "0006A", vgl. 2. Anweisung in der Liste in Abb. 22, linke Spalte).

Wir werden jetzt errechnen, welchen Wert die Hexadezimalziffern "0D5" als relative Adresse angeben:



Die relative Adresse beträgt also dezimal __(Z)__.

Bei einem Basisregisterinhalt von 106 beginnt dann der 1. Operand im Hauptspeicher bei der Adresse __(Z)__.

213

319

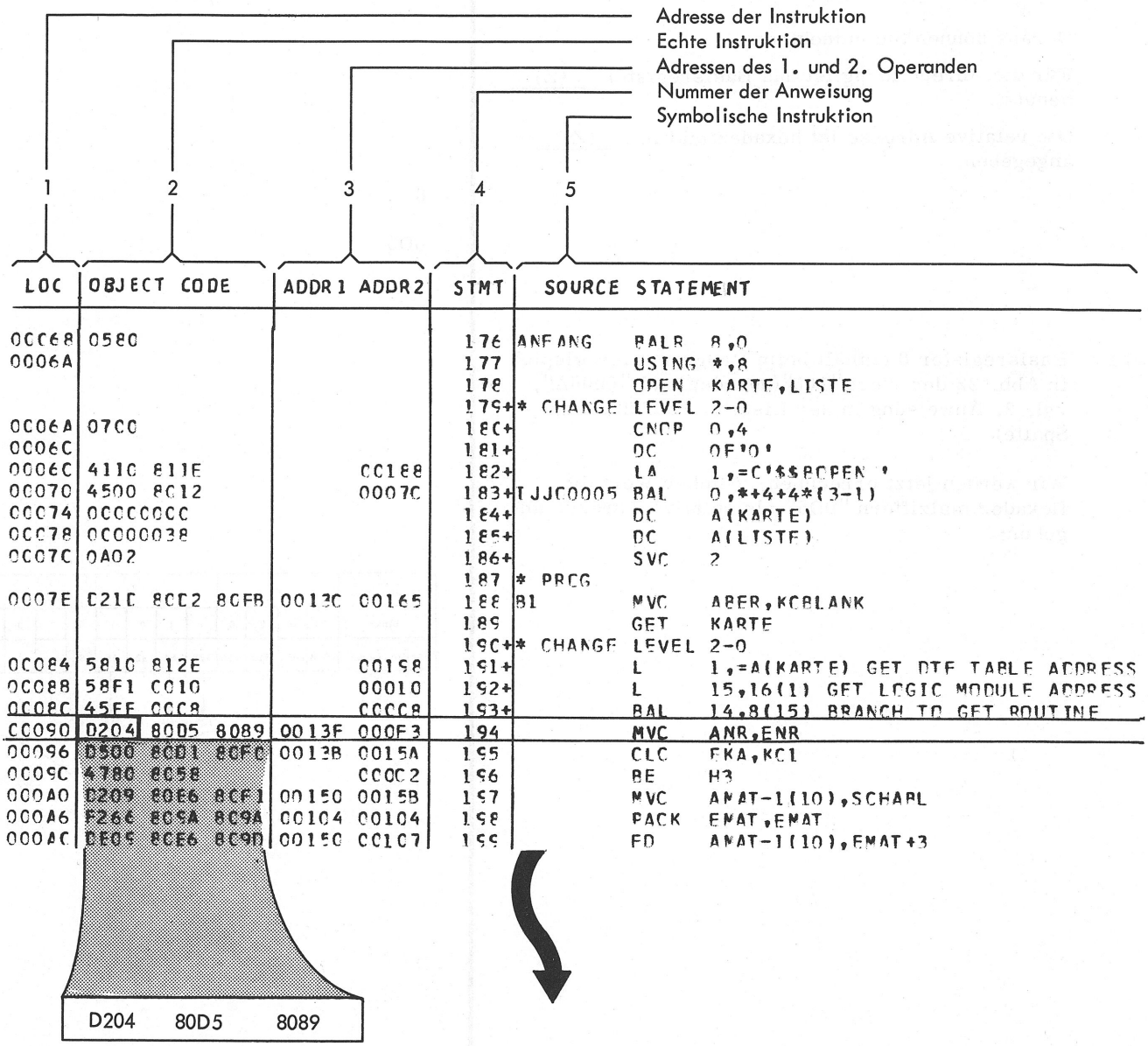

Adresse der Instruktion
Echte Instruktion
Adressen des 1. und 2. Operanden
Nummer der Anweisung
Symbolische Instruktion
1
2
3
4
5
LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT
00C68 0580 176 ANFANG BALR 8,0
0006A 177 USING *,8
178 OPEN KARTE,LISTE
179+* CHANGE LEVEL 2-0
0006A 07C0 180+ CNOP 0,4
0006C 181+ DC 0F'0'
0006C 4110 811E 00188 182+ LA 1,=C'$$BOPEN '
00070 4500 8012 0007C 183+IJJC0005 BAL 0,*+4+4*(3-1)
00074 00CCCCCC 184+ DC A(KARTE)
00078 00C00038 185+ DC A(LISTE)
0007C 0A02 186+ SVC 2
187 * PROG
0007E C21D 80D2 80FB 0013C 00165 188 B1 MVC ABER,KCBLANK
189 GET KARTE
190+* CHANGE LEVEL 2-0
00084 5810 812E 00198 191+ L 1,=A(KARTE) GET DTF TABLE ADDRESS
00088 58F1 C010 00010 192+ L 15,16(1) GET LOGIC MODULE ADDRESS
0008C 45EF CCC8 CCCC8 193+ BAL 14,8(15) BRANCH TO GET ROUTINE
00090 D204 80D5 8089 0013F 000F3 194 MVC ANR,ENR
00096 D500 80D1 80F0 0013B 0015A 195 CLC EKA,KC1
0009C 4780 8058 CC0C2 196 BE H3
000A0 C209 80E6 8CF1 00150 0015B 197 MVC AMAT-1(10),SCHABL
000A6 F26E 8C9A 8C9A 00104 00104 198 PACK EMAT,EMAT
000AC DE05 8CE6 8C9D 00150 CC1C7 199 ED AMAT-1(10),EMAT+3
D204 80D5 8089

Abb. 22

74 Für die Adresse des 2. Operanden wird auch das Basisregister 8 mit dem Inhalt 106 benutzt.

Die relative Adresse ergibt sich aus den Hexadezimalziffern "0 8 9".

Stellen Sie mit Hilfe der folgenden Tabelle den Wert dieser relativen Adresse fest:

hexadezimal	0				8				9			
binär	0	0	0	0	1	0	0	0	1	0	0	1
Stellenwerte	2048	1024	512	256	128	64	32	16	8	4	2	1

Die relative Adresse hat den Wert ___(Z)___.

137 (128 + 8 + 1)

75 Der 2. Operand des Übertragungsbefehls in Abb. 22 hat dezimal ausgedrückt die Basisadresse 106 und die relative Adresse 137.

Daraus ergibt sich die effektive Adresse ___(Z)___.

243

Wir haben festgestellt:

Die Hexadezimalziffern

" D204 80D5 8089 "

in der Umwandlungsliste in Abb. 22 stellen eine Übertragungs-Instruktion in echter Form dar.

Die Operanden sind Felder im Hauptspeicher mit Längen von 5 Bytes und den Adressen 319 und 243.

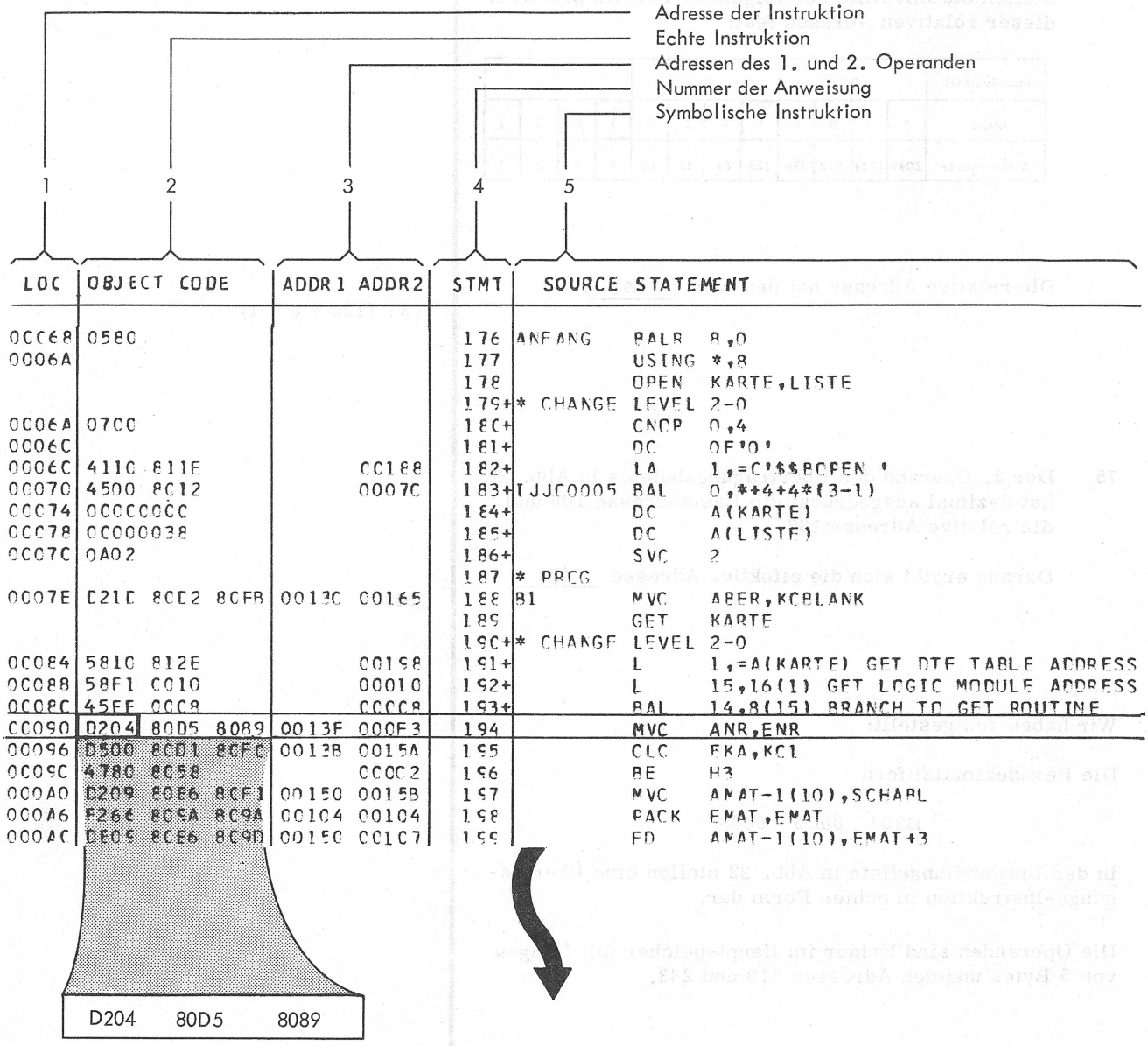

Adresse der Instruktion
Echte Instruktion
Adressen des 1. und 2. Operanden
Nummer der Anweisung
Symbolische Instruktion
1
2
3
4
5
LOC | OBJECT CODE | ADDR1 ADDR2 | STMT | SOURCE STATEMENT
000068 0580 176 ANFANG BALR 8,0
00006A 177 USING *,8
178 OPEN KARTE,LISTE
179+* CHANGE LEVEL 2-0
00006A 07C0 180+ CNOP 0,4
00006C 181+ DC 0F'0'
00006C 4110 811E 000188 182+ LA 1,=C'$$BOPEN '
000070 4500 8012 000070 183+IJJC0005 BAL 0,*+4+4*(3-1)
000074 00000000 184+ DC A(KARTE)
000078 00000038 185+ DC A(LISTE)
00007C 0A02 186+ SVC 2
187 * PROG
00007E C21D 80D2 80FB 0013C 00165 188 B1 MVC ABER,KOBLANK
189 GET KARTE
190+* CHANGE LEVEL 2-0
000084 5810 812E 000198 191+ L 1,=A(KARTE) GET DTF TABLE ADDRESS
000088 58F1 C010 000010 192+ L 15,16(1) GET LOGIC MODULE ADDRESS
00008C 45EF C0C8 0000C8 193+ BAL 14,8(15) BRANCH TO GET ROUTINE
000090 D204 80D5 8089 0013F 000E3 194 MVC ANR,ENR
000096 D500 80D1 80FC 0013B 0015A 195 CLC EKA,KC1
00009C 4780 8058 0000C2 196 BE H3
0000A0 D209 80E6 80E1 00150 0015B 197 MVC AMAT-1(10),SCHABL
0000A6 F26E 80EA 80EA 00104 00104 198 PACK EMAT,EMAT
0000AC DE05 80E6 80ED 00150 001C7 199 ED AMAT-1(10),EMAT+3
D204 80D5 8089

Abb. 22

76 Die Umrechnung der in Basisregister-Nummern und relative Adressen aufgegliederten Adressen in die dezimale Schweibweise erübrigt sich im allgemeinen für den Programmierer.

Betrachten Sie noch einmal Abb. 22.

Rechts neben dem Befehl, also in Spalte 3 der Liste, tauchen die Adressen der beiden Operanden noch einmal auf.

Die Angaben lauten hier
für die Adresse des 1. Operanden: (ZZ) ,
für die Adresse des 2. Operanden: (ZZ) .

0013F

000F3

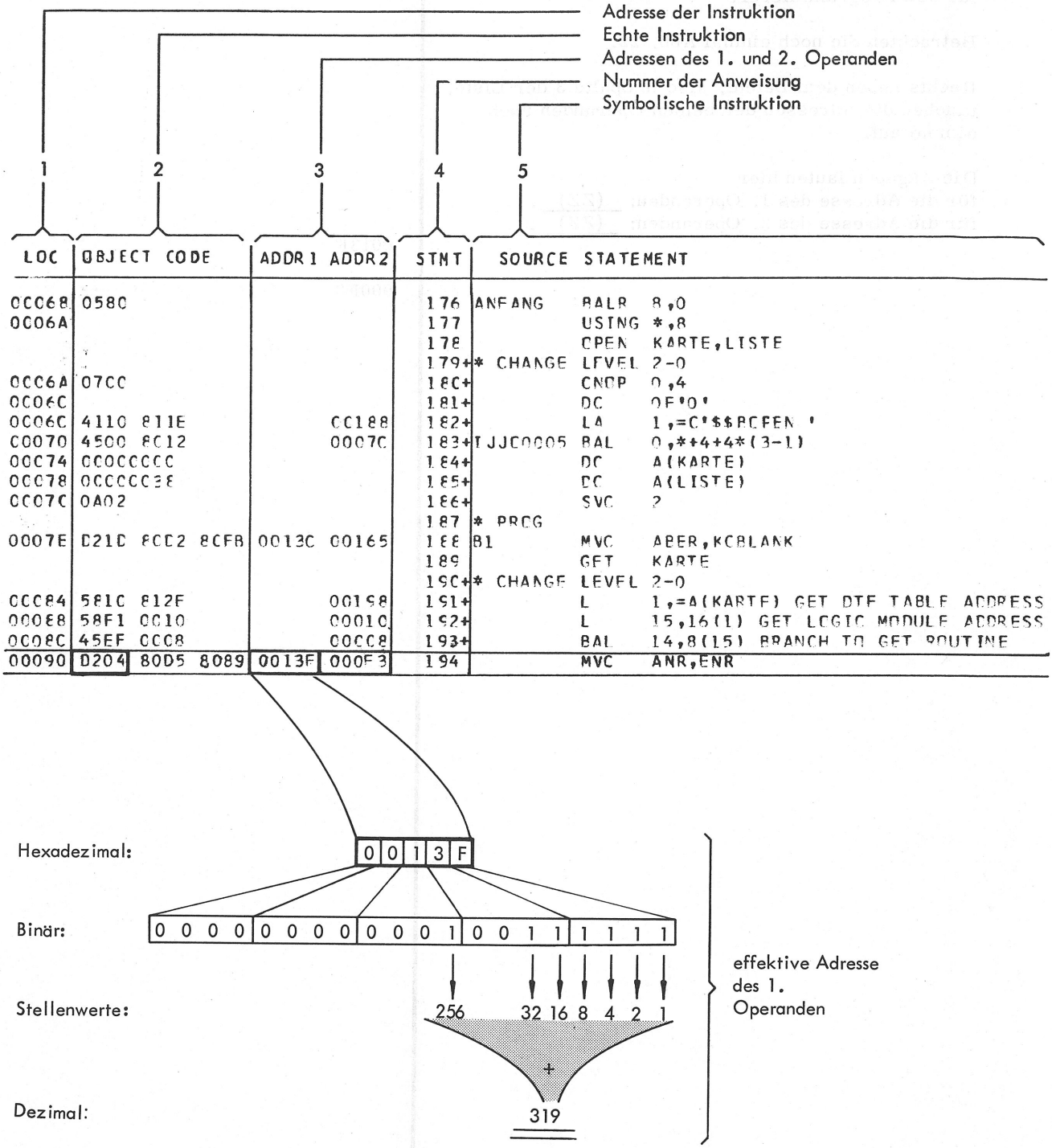

Adresse der Instruktion
Echte Instruktion
Adressen des 1. und 2. Operanden
Nummer der Anweisung
Symbolische Instruktion
1
2
3
4
5
LOC OBJECT CODE ADDR1 ADDR2 STMT SOURCE STATEMENT
00068 0580 176 ANFANG BALR 8,0
0006A 177 USING *,8
178 OPEN KARTE,LISTE
179+* CHANGE LEVEL 2-0
0006A 07CC 180+ CNOP 0,4
0006C 181+ DC 0F'0'
0006C 4110 811E CC188 182+ LA 1,=C'$$BCPEN '
00070 4500 8C12 0007C 183+IJJC0005 BAL 0,*+4+4*(3-1)
00074 CCCCCCCC 184+ DC A(KARTE)
00078 CCCCCC3E 185+ DC A(LISTE)
0007C 0A02 186+ SVC 2
187 * PROG
0007E D21D 8CD2 8CFB 0013C 00165 188 B1 MVC ABER,KCBLANK
189 GET KARTE
190+* CHANGE LEVEL 2-0
00084 581C 812F 00198 191+ L 1,=A(KARTE) GET DTF TABLE ADDRESS
00088 58F1 0010 00010 192+ L 15,16(1) GET LOGIC MODULE ADDRESS
0008C 45EF CCC8 00CC8 193+ BAL 14,8(15) BRANCH TO GET ROUTINE
00090 D204 80D5 8089 0013F 000F3 194 MVC ANR,ENR
Hexadezimal:
0 0 1 3 F
Binär:
0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1
Stellenwerte:
256 32 16 8 4 2 1
+
Dezimal:
319
effektive Adresse des 1. Operanden

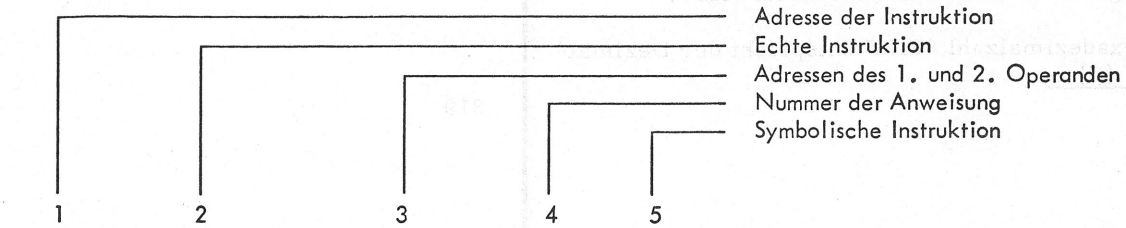
Abb. 23

"0013F" ist die effektive Adresse des 1. Operanden in hexadezimaler Schreibweise.

Zur Bestätigung zeigt Abb. 23 die Umrechnung dieser Angaben in die dezimale Schreibweise.

Die Hexadezimalzahl 0013F entspricht der Dezimalzahl ___(Z)___.

319



LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
000068	058C			176	ANFANG BALR 8,0
00006A				177	USING *,8
				178	OPEN KARTE,LISTE
				179+* CHANGE LEVEL 2-0
00006A	07CC			180+	CNOP 0,4
00006C				181+	DC OF'0'
00006C	4110 811F		011E8	182+	LA 1,=C'$$POPEN '
000070	4500 8012		00070	183+IJJ00005	BAL 0,*+4+4*(3-1)
000074	00000000			184+	DC A(KARTE)
000078	00000038			185+	DC A(LISTE)
00007C	0AC2			186+	SVC 2
				187	* PROG
00007E	D21E 8002 8CFB	0013C	00165	188	B1 MVC ABER,KOBLANK
				189	GET KARTE
				190+* CHANGE LEVEL 2-0
000084	5810 812E		01198	191+	L 1,=A(KARTE) GET DTF TABLE ADDRESS
000088	58F1 0010		00010	192+	L 15,16(1) GET LOGIC MODULE ADDRESS
00008C	45EF 0008		00008	193+	BAL 14,8(15) BRANCH TO GET ROUTINE
00090	D204 80D5 8089	0013F	000F3	194	MVC ANR,ENR
000096	D500 8001 8CFC	00138	011EA	195	CLC EKA,KC1
00009C	478C 8058		000C4	196	BE B3
0000A0	D209 8CE6 8001	001E0	011E8	197	MVC ANAT-1(10),SCHARL
0000A6	F266 8C9A 8C9A	001CA	001C4	198	PACK EMAT,EMAT
0000AC	DEC9 8CE6 8C9D	001EC	001C7	199	ED ANAT-1(10),EMAT+3
				222	* DEFINITIONEN DER BEREICHE, FELDER
000EC				223	EBER DS 0CL80 EINGABE-BEREICH
000EC7				224	DS CL7
000F3				225	ENR DS CL5
000F8				226	DS CL12
001C4				227	ELOHN DS 0CL7
001C4				228	EMAT DS CL7
001C8				229	DS CL48
001EB				230	EKA DS CL1
001EC				231	ABER DS 0CL30 AUSGABE-BEREICH
001EC				232	DS CL3
001F3				233	ANR DS CL5
00144				234	DS CL2

Abb. 24

78 Das DVS hat automatisch während der Umwandlungsphase die Angaben "80D5" in "0013F" umgerechnet.

Ohne Umrechnung in die dezimale Schreibweise kann man aus der Umwandlungsliste leicht entnehmen, welche Daten bei der Adresse "0013F" gespeichert werden. Die Adresse erscheint nämlich in gleicher Form in einer anderen Spalte der Liste (vgl. Abb. 24).

Es ist die Spalte __(Z)__.

1

79 Mit Hilfe dieser linken Spalte in Abb. 24 entnehmen wir der Umwandlungsliste:

Der 1. Operand der Übertragungsoperation mit der Adresse 0013F ist das Datenfeld mit dem symbolischen Namen "ANR".

Wie Sie sehen, reichen die Adreßangaben in Spalte 3 aus, um sich in der Umwandlungsliste zurechtzufinden.

Sie haben jetzt einen Einblick gewonnen, wie ein echter Befehl in einem der 4 besprochenen Formate aussieht und wie er in Listen erscheint, die das DVS ausgibt.

Im Band 4 werden wir zum Schluß alle Assemblerbefehle dieses Lehrgangs den 4 Formaten zuordnen.

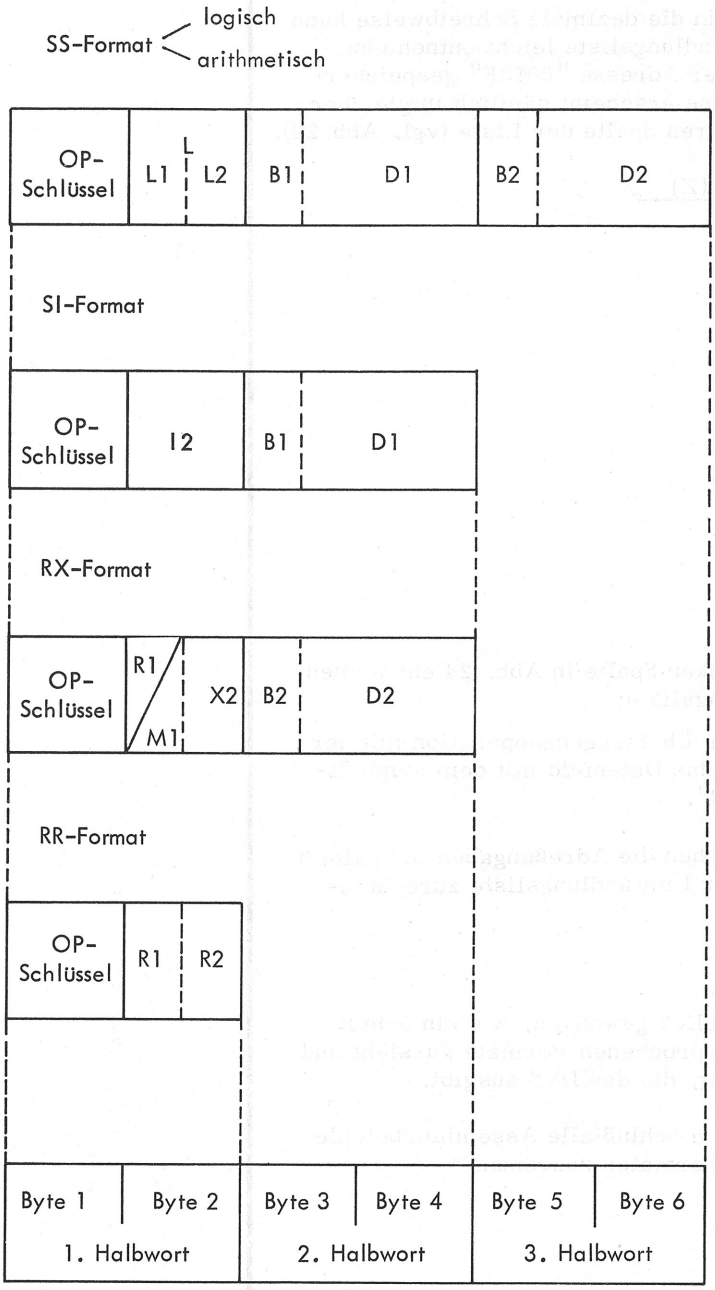

SS-Format
logisch
arithmetisch
OP-Schlüssel
L1
L
L2
B1
D1
B2
D2
SI-Format
OP-Schlüssel
I2
B1
D1
RX-Format
OP-Schlüssel
R1
M1
X2
B2
D2
RR-Format
OP-Schlüssel
R1
R2
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
1. Halbwort
2. Halbwort
3. Halbwort

Abb. 25

FRAGEN ZUM KAPITEL VI

Beantworten Sie bitte erst alle Fragen, ehe Sie Ihre Antworten überprüfen.

Benutzen Sie Abb. 25 als Hilfsmittel.

1. In welchem Byte steht der Operationsschlüssel bei allen 4 Formaten ?

2. Welches Format enthält Angaben zur Länge der Operanden ?

3. Warum enthalten die anderen Formate keine Längenangaben ?

4. Wieviel Bytes beansprucht die Adresse eines Hauptspeicherfeldes im echten Befehl

   a) im SS-Format

   b) im RX-Format ?

5. Wie lang ist die Adresse eines Registers im echten Befehl ?

6. In der Umwandlungsliste stehen die Angaben "1A59" für einen echten Befehl.

   a) Welches Format liegt vor ?

   b) Wie heißen die beiden Operanden, auf die sich dieses Beispiel bezieht ?

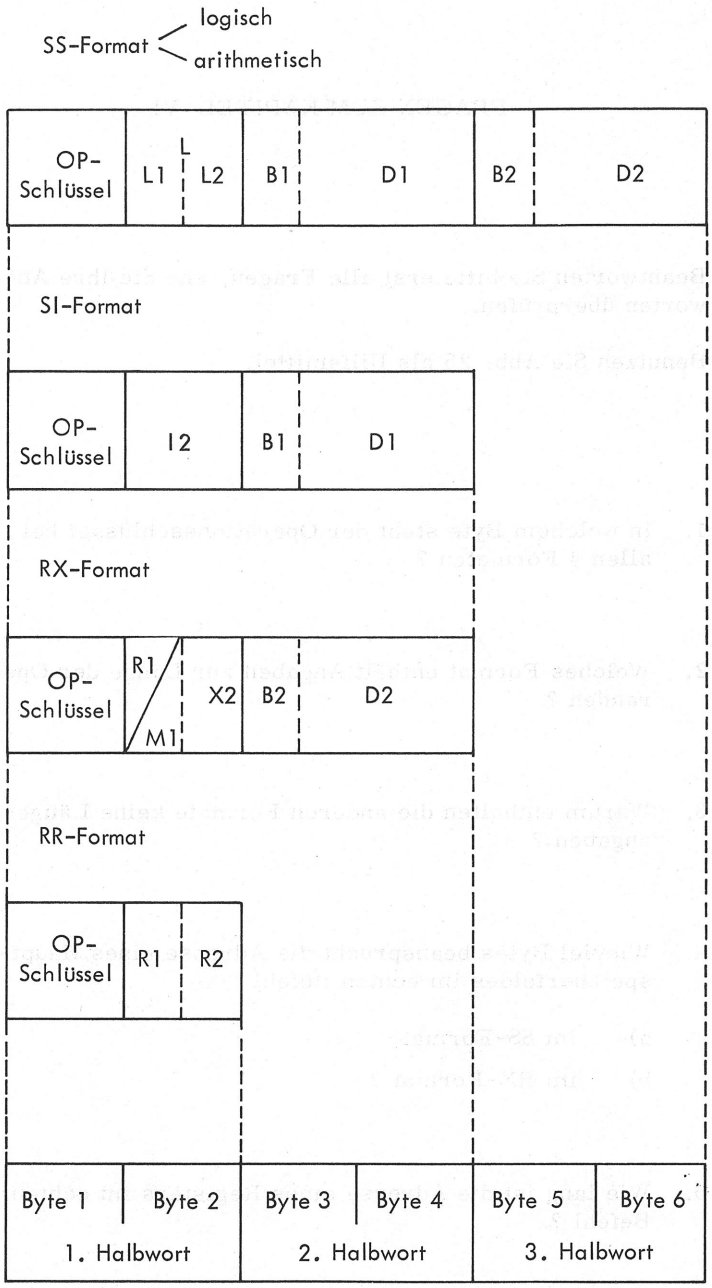

SS-Format
logisch
arithmetisch
OP-Schlüssel
L1
L
L2
B1
D1
B2
D2
SI-Format
OP-Schlüssel
I2
B1
D1
RX-Format
OP-Schlüssel
R1
M1
X2
B2
D2
RR-Format
OP-Schlüssel
R1
R2
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
Byte 6
1. Halbwort
2. Halbwort
3. Halbwort

Abb. 25

7. Es gibt Operanden, die in einer Instruktion nicht durch ihre Adresse vertreten sind, sondern selbst in der Instruktion stehen .

   a) Wie heißt ein solcher Operand ?

   b) Wie heißt das dazugehörige Format ?

8. Welche Instruktions-Längen treten auf bei den 4 Formaten ?

ANTWORTEN ZUM KAPITEL VI

Überprüfen Sie nun Ihre Antworten. Beseitigen Sie alle Unklarheiten, ehe Sie fortfahren. Wiederholen Sie gegebenenfalls die in Klammern angegebenen Lernelemente.

1. Im 1. Byte (LE 30)

2. Das SS-Format (LE 4 - 28)

3. Bei den anderen Formaten steht die Länge der Operanden fest. (LE 33 - 66)

4. a) 2 Bytes
   b) 5 Halbbytes (LE 64)

5. 1 Halbbyte (LE 64)

6. a) Das RR-Format
   b) Register 5 und Register 9 (LE 61 - 66)

7. a) Direktoperand
   b) SI-Format (LE 33 - 41)

8. 6, 4 und 2 Bytes, bzw. 3, 2 und 1 Halbwort (LE 29 - 30)

Schauen Sie sich nun das Merkblatt 207 (5 Seiten) an. -



Band 4. Seite 503 (Anhang zu den Instruktionen)

FRAGEN ZUM BAND 2

In den folgenden Fragen wird der Lernstoff des Bandes 2 wiederholt. Schreiben Sie Ihre Antworten bitte auf ein Extrablatt.

Die Lösungen finden Sie auf den anschließenden Seiten.

1. a) In wieviel Bits wird ein Zeichen im EBCD-Code verschlüsselt ?

   b) Mit wieviel Hexadezimalziffern kann eine solche Verschlüsselung angegeben werden ?

   c) Welche anderen Bezeichnungen kennen Sie

      α) für "linkes Halbbyte",

      β) für "rechtes Halbbyte" ?

2. Wie heißen die beiden Darstellungsformen für Dezimalzahlen im Speicher ?

3. Entnehmen Sie Ihren Arbeitsmitteln Übung 202. Es zeigt zwei 3-stellige Dezimalzahlen in Lochkarten.

   Führen Sie bitte die Subtraktion dieser beiden Zahlen durch und tragen Sie den Inhalt der eingezeichneten Speicherfelder und des Lochfeldes in der Ausgabe-Karte in das Übungsblatt ein.

4. a) Wandeln Sie die Dezimalzahlen 69 und 31 in Binärzahlen um.

   b) Addieren Sie diese Binärzahlen.

   c) Überprüfen Sie das Ergebnis, indem Sie es in eine Dezimalzahl umwandeln.

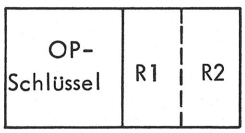
5. Für Dualdaten gibt es 3 Formate fester Länge. Wie heißen diese 3 Formate und wie lang sind sie ?

6. a) Wie viele Allgemeine Register gibt es und wie bezeichnet man ein Register, das zur Adreßbildung benutzt wird ?

   b) Durch welche beiden Komponenten wird eine Hauptspeicheradresse in einer echten SS-Instruktion dargestellt ?

   c) Wie errechnet sich eine effektive Adresse aus einer SS-Instruktion ?

7.

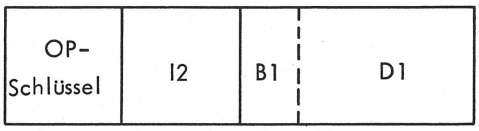


   a) Welches Instruktions-Format liegt vor ?

   b) Wieviel Bytes belegt eine Instruktion in diesem Format ?

   c) Welche Angaben enthält das zweite Byte bei diesem Format ?

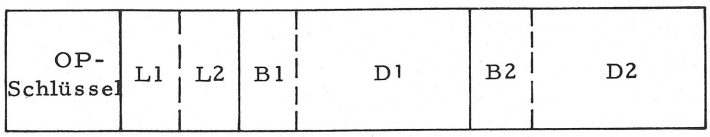
8.



Bei diesem Format ist der 1. Operand ___(WW)___ und der 2. Operand ___(WW)___.

Das Format heißt _____.

9.



a) Wie heißt die vollständige Bezeichnung für dieses Instruktions-Format ?

b) Wieviel Bytes belegt eine Instruktion in diesem Format ?

c) In welchem Halbbyte steht die Längenangabe für den 1. Operanden ?

d) Woraus setzt sich die Adresse des 1. Operanden zusammen ?

e) Wieviel Bytes umfaßt der 2. Operand, wenn L2 = 4 ist ?

ANTWORTEN ZUM BAND 2

Wenn Sie Fehler gemacht haben, sollten Sie zur Beseitigung der Unklarheiten zu den Fragen des betreffenden Kapitels (in Klammern) zurückschlagen. Kommt es auch dort zu Fehlern, so sollten Sie die betreffenden Lernelemente wiederholen.

1. a) in 8 Bits

   b) mit 2 Hexadezimalziffern

   c) α) Zonenteil

      β) Zifferntteil

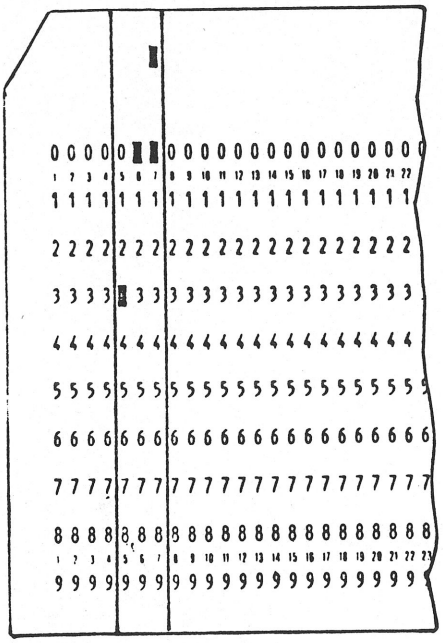
(Fragen zu Kapitel II, Seite 91)

2. ungepackt (gezont) und gepackt

(Fragen zu Kapitel III, Seite 131)

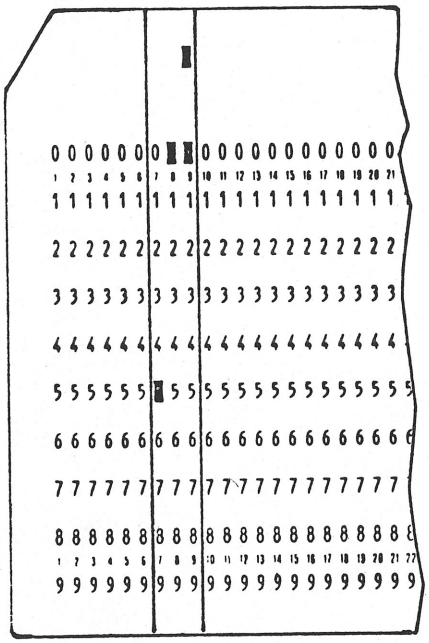
3. (s. folgendes Blatt!)

(Fragen zu Kapitel III, Seite 131)



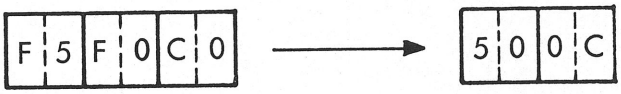
WERTA

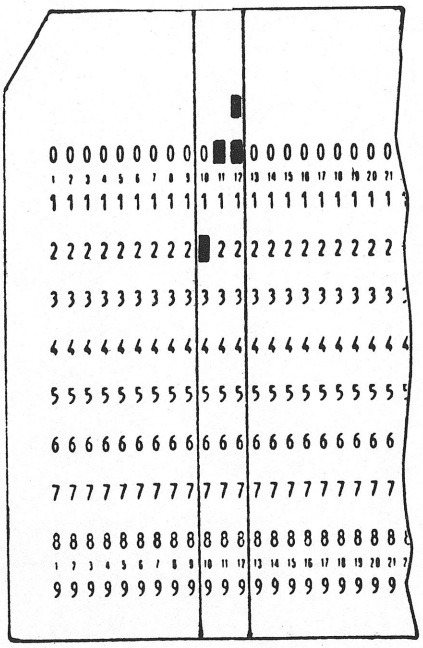




—

WERTB





=

ERGEBNIS





4. a) $69_{10} = 1000101_2$

$31_{10} = 11111_2$

---

b) 1100100_2

c) $1100100_2 = 100_{10}$

(Fragen zu Kapitel IV, Seite 177)

5. Halbwort - 2 Bytes
Wort - 4 Bytes
Doppelwort - 8 Bytes

(Fragen zu Kapitel IV, Seite 177)

6. a) Es gibt 16 Allgemeine Register. Ein Register, das zur Adreßbildung benutzt wird, nennt man Basisregister.

b) Die beiden Komponenten sind:
Adresse eines Basisregisters (B) und relative Adresse (D).

c) Eine effektive Adresse ist die Summe aus Basisadresse und relativer Adresse.

(Fragen zu Kapitel V, Seite 213)

7. a) das RR-Format
b) 2 Bytes
c) zwei Register-Adressen

(Fragen zu Kapitel VI, Seite 293)

8. Bei diesem Format ist der 1. Operand
ein Feld im Hauptspeicher und der 2. Operand
ein Direktoperand.

Das Format heißt SI-Format.

(Fragen zu Kapitel VI, Seite 293)

9. a) arithmetisches SS-Format
b) 6 Bytes
c) im 3. Halbbyte von links
d) aus dem Inhalt eines Basisregisters und einer relativen Adresse
e) 5 Bytes

(Fragen zu Kapitel VI, Seite 293)